Operations Orchestration

Software Version: 10.80
Windows and Linux

Use

Hewlett Packard
Enterprise

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notice

© September 2017 Hewlett Packard Enterprise Development LP

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

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

UNIX® is a registered trademark of The Open Group.

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

Documentation Updates

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

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

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

Contents

Use

Navigate through following topics to comprehend the usage of OO Central and OO Studio.

Command Line Interface

Running Central Commands from the Command Line

This document shows you how to use Operations Orchestration Shell (OOSH) utility to run OO commands from the command line, instead of from the Central 10.x web interface.

Running OO Central Commands from the Command Line

The OO Shell (OOSH) utility lets you run OO commands from the command line, instead of from the Central 10.x web interface. This enables you to create scripts to run commands, to automate your OO processes.

You can perform actions such as:

- Run flows

- Deploy content packs

- Manage files - import, upload, delete, and so on

- Display flow information

- Manage system configuration

```
Welcome to oosh. For assistance type "help" then hit ENTER.
oosh>connect --url http://16.59.62.205:8293/oo/
connected
oosh>finfo --uuid  b1e8dfa0-b975-41c7-84ac-249565aa8dda

FlowInfo
--------

    Flow ID (UUID): b1e8dfa0-b975-41c7-84ac-249565aa8dda
    Name          : Connectivity Test
    Content pack  : HPOO-oo-base (1.0-SNAPSHOT)
    Description   :
        Checks to see whether or not a connection exists, and if it exists check
s to see
        if it is adequately reliable.

        Inputs:
        host - The host to ping
        from.
        username - The username for host.
        password - The password for
        host.
        lossThreshold - What percentage of packet loss you consider to be
        acceptable for this connection.
        latencyThreshold - How long (in milliseconds)
        you consider to be an acceptable latency for this connection.
```

The OOSH utility is located in two places:

- In OO installations, under **<installation path>\central\bin** and **<installation path>\ras\bin**.

- In a standalone OOSH zip package, which is included in the OO release zip.

> **Note:**
>
> - Users invoking commands from the OOSH utility must have the relevant Central permissions (otherwise, they will face "Unauthorized" errors).

- If you used to work with **RSFlowInvoke** and **JRSFlowInvoke** with version 9.x, note that this is the alternative for OO 10.x.

- For OO version 10.80, the OOSH utility has been changed to access OO Central via the latest REST API (v2). This allows OOSH to also trigger CloudSlang flows.

Running Commands

You can run commands within the OOSH utility or from a normal prompt/shell.

1. Do one of the following:

   ○ Run the oosh executable file (oosh.bat on Windows). This will open the OOSH shell interface, where you can enter commands.

   ○ To run a command directly from the operating system's prompt/shell, change the directory to the location of the **oosh** executable file and enter: `oosh <command>`.

   If you want to run a sequence of commands automatically, it is recommended to use a script file (see).

2. Type the command in the following format:

   `<commandName> --<parameterName1> <parameterValue1>`

   `--<parameterName2>  <parameterValue2>`

For example, when running commands from a normal prompt/shell, type:

`c:\hpoo\central\bin> oosh.bat trigger --uuid ABC --user ABC --password ABC --url ABC`

Common OOSH Tasks

The sections below describe how to perform some common functions using OOSH. See for a complete list of commands that can be used with the OOSH utility and the options that can be used with each command.

Connect to a Central 10.x server

If you are running commands in the OOSH utility's shell interface, or from a script file, you can use the `connect` command to connect to the Central 10.x server.

If the connection is successful, consequent commands will use it (until you exit OOSH or the Central session times out).

All other commands support basic connection properties: `url`, `user`, `password`. This enables you to invoke them directly from the command line without using the `connect` command first.

> **Note:** It is possible to specify the user name (with the `user` property) without specifying the password (with the `password` property). You will be prompted for the password upon making the connection.
>
> Note also that in a multi-LDAP environment, you should enter the user name in the following format: **domain\username**. If you enter the user name without a domain, OOSH will try to

> authenticate the user with the default LDAP first and if this fails, will to authenticate within the OO internal domain.

The `connect` command has more options. It also allows connection via an HTTP proxy.

For a complete list of all the options you can use with the `connect` command, see "Connection to Central " on page 19 or type `help connect` in the command line.

The following steps are for running the `connect` command from the OOSH shell interface. The command can also be used in an OOSH script file (as the very first command).

1. Double-click the `oosh.bat` batch file (on Linux, run the `oosh` file), to start the OOSH utility.

2. In the command line, type (for example):

    `connect --url <Central server URL>`

3. Press the **Enter** key.

    If the connection is successful, the command line displays:

    `Connected`

Display information about a flow

1. In the command line of the OOSH utility, type:

    `finfo --uuid <the UUID of the flow>`

    > **Note:** The UUID of the flow is displayed in the Flow Library information pane.

2. Press the **Enter** key.

    The flow information is displayed in the command line.

Run a flow

Use the trigger command to run a flow via Central. For a complete list of all the options you can use with the `trigger` command, see "trigger" on page 21 or type `help trigger` in the command line.

1. In the command line of the OOSH utility, type (for example):

    `trigger --uuid <the UUID of the flow>`

2. Press the **Enter** key.

Deploy a content pack

Use the `deploy` command to deploy content packs in Central. For a complete list of all the options you can use with the `deploy` command, see "deploy" on page 20 or type `help deploy` in the command line.

1. In the command line of the OOSH utility, type (for example):

   ```
   deploy --url <URL of OO Central> --files <comma-separated list of files to
   deploy>
   ```

2. Press the **Enter** key.

Display help about the OOSH utility

To display a list and description of all available commands in the OOSH utility, type:

```
help
```

To display help about a particular command, type:

```
help <command>
```

For example:

```
help trigger
```

Configure how the X-Frame-Options response header is added to outgoing responses

This lets you configure whether or not a browser will be allowed to render a page in a <frame>, <iframe> or <object>. You can use this to avoid clickjacking attacks, by ensuring that your content is not embedded into other sites.

> **Note:** Some versions of Chrome ignore the **X-Frame-Options** response header.

Available values for the **xframe.options.mode** key are:

- **SAMEORIGIN** - only web pages containing iframes that come from the same URI as OO will be allowed. This is the default value. If you are using OO embedded in a browser, you may need to change this.
- **DENY**
- **NONE** - no **X-Frame-Options** response header will be used.
- **ALLOW-FROM** - you can specify the specific URI from which the frames will be allowed.

  If the value for **xframe.options.mode** is **ALLOW-FROM**, you must also add the **xframe.options.allow.from** key to specify the URI.

  > **Note:** Some versions of Chrome ignore the **ALLOW-FROM** value. Therefore, it is not recommended to use it.

1. In the command line of the OOSH utility, type (for example):

   ```
   set-sys-config --key xframe.options.mode --value DENY
   ```

   ```
   set-sys-config --key xframe.options.mode --value ALLOW-FROM --key
   xframe.options.allow.from --value http://www.example.com
   ```

2. Press the **Enter** key.

Enable flows to run interactively from other applications

This configuration item lets you run flows interactively from other applications, such as BSM.

1. In the command line of the OOSH utility, type (for example):

   ```
   set-sys-config --key headless.interactivity --value true
   ```

2. Press the **Enter** key.

Tell Central whether to trust all valid certificates from secured LDAP servers

When the **ldap.trustAllCertificates** configuration item is true, Central will trust any valid certificate from a secured LDAP server. There will be no need for you to add it to the **client.truststore**.

When this is not enabled, you will need to add the certificates from the LDAP servers used for LDAP authentication to the **client.truststore**. Central will not trust them otherwise.

1. In the command line of the OOSH utility, type (for example):

   ```
   set-sys-config --key ldaps.trustAllCertificates --value true
   ```

2. Press the **Enter** key.

Running OOSH Script Files

Create a script to run OO commands from the OOSH utility

1. To create a script, write a simple text file with the OOSH commands that you want to run. Write each command on a separate line, in the sequence that you want the commands to run.

2. To run the script, either set OOSH as its parser (Linux only, see the details below), or use the script command in OOSH. For example:

   ```
   script --file <path to script>
   ```

Add OOSH as the parser of an executable file

In Linux only, it is also possible to add OOSH as the parser of an executable file.

1. For example, create script-a:

   ```
   #!/<install-dir>/oosh
   ```

   ```
   <command-sequence>
   ```

2. Chmod the file:

   ```
   chmod 755 script-a
   ```

3. Run it from the command line:

   ```
   ./script-a
   ```

OOSH Reference Material

This section contains a list of all the OOSH commands organized into five categories:

Connection to Central

connect

Connects to an OO 10.x Central server.

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes
--user	OO Central user name In a multi-LDAP environment, enter the user name in the format `domain\username`.	If you enter the user name without a domain, OOSH will try to authenticate the user with the default LDAP first, and if this fails, will try to authenticate within the OO internal domain.	No
--password	OO Central password		No
--proxyhost	http proxy host to connect through		No
--proxyport	http proxy port to connect through	8080	No
--proxyuser	http proxy user to connect with		No

Argument	Description	Default Value/Behavior	Required
--proxypassword	http proxy password to connect with		No

disconnect

Disconnects from the OO 10.x Central server.

show connection

Shows the connection properties.

Content Management

deploy

Deploys a content pack (requires an active connection).

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	OO Central password		No
--files	Comma-separated list of content packs to deploy		Yes

lcp, list-content-packs

Lists all currently deployed content packs in the connected server

finfo, flow-info

Displays information about a specific flow.

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	OO Central password		No
--property	Displays a specific property (default: all). Available properties: `id, name, path, description, cpName, version`	all	No
--uuid	UUID of the flow. For example, `cd38b579-4220-4fa5-bafd-78d67f73eb6f`		Yes

Run Management

trigger

Runs a flow.

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No

Argument	Description	Default Value/Behavior	Required
--password	OO Central password		No
--uuid	UUID of the flow to run. For example, `cd38b579-4220-4fa5-bafd-78d67f73eb6f`		Yes

--inputs	Comma-separated key=value map of the flow inputs. For example, `key1=value1,key2=value2`
	
	You can include commas by repeating them. For example, `a,,b` results in `a,b`.
	
	Note: If the input value contains white spaces, and you are using a Windows command line, you need to use \" to escape the quotes. For example:
	`--inputs \"title=my title,message=my message\"`
	
	If you open the OOSH command line (by running oosh.bat without arguments), you can use regular quotes. For example:
	`--inputs "title=my title,message=my message"`
			No

--wait	Wait for the run to finish (synchronous mode)?	Do not wait (asynchronous mode)	No
--wait-limit	If running in synchronous mode, how long in seconds should oosh wait before giving up?	No limit	No

track

Shows information about a specific execution (tracking).

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	OO Central password		No
--id	Execution id		Yes
--property	Displays a specific property (default: all). Available properties: `executionId, executionName, status, resultStatusType, resultStatusName`	all	No

System Configuration

ssc, set-sys-config

Sets the system configuration.

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDAP environment, enter the user name in the format `domain\username`.		No

Argument	Description	Default Value/Behavior	Required
--password	OO Central password		No
--key	System configuration key See the keys in the table below.		Yes
--value	System configuration value	empty string	No

Some useful keys for the **ssc**, **set-sys-config** command:

Key	Description

xframe.options.mode	Lets you configure how the **X-Frame-Options** response header is added to outgoing responses.
	
	Available values:
	
	• **SAMEORIGIN** - only web pages containing iframes that come from the same URI as00 will be allowed. This is the default value. If you are using00 embedded in a browser, you may need to change this.
	
	• **DENY**
	
	• **NONE** - no **X-Frame-Options** response header will be used.
	
	• **ALLOW-FROM** - you can specify the specific URI from which the frames will be allowed, by using the **xframe.options.allow.from** key.
	
	> **Note:** Some versions of Chrome ignore the **ALLOW-FROM** value. Therefore, it is not recommended to use it.
	
	Examples:
	
	`set-sys-config --key xframe.options.mode --value DENY`
	
	`set-sys-config --key xframe.options.mode --value SAMEORIGIN --key xframe.options.allow.from --value http://www.example.com`

xframe.options.allow.from	If the value for **xframe.options.mode** is **ALLOW-FROM**, you must add the **xframe.options.allow.from** key to specify the URI. This value is considered only when the value for **xframe.options.mode** is **ALLOW-FROM**.
headless.interactivity	Lets you run flows interactively from other applications, such as BSM. For example, when the value is **true**, the user that invokes an00 flow from the BSM user infterface can run the flow interactively.
ldaps.trustAllCertificates.isEnabled	When the value is **true**, Central will trust any valid certificate from a secured LDAP server without you needing to add it to the **client.truststore**. When the value is **false**, you will need to add the certificates from the LDAP

Key	Description
	servers used for LDAP authentication to the **client.trustore**. Central will not trust them otherwise.
x509.principal.lookup.field	This controls which certificate field is used to extract the username. Possible values are: • `subjectDN` - represents the `Subject` field of the certificate, meaning that00 keeps its default behavior and attempts to extract the username from the **Subject** field. This is the default value. • `subjectAltNames.otherName.principalName` - represents the `User Principal Name` contained in the `Other Name` entry of the `Subject Alternative Names` certificate extension. For CAC Authentication, it might be required to use the value of the User Principal Name, so you would use this value.

lsc, list-sys-config

Lists the system configuration.

Argument	Description	Default Value/Behavior	Required
--url	URL of00 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	00 Central password		No
--key	Lets you request the value of a specific key only		No

license-file-import

Install an enterprise license from a file.

Use

Argument	Description	Default Value/Behavior	Required
--url	URL of OO10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	00 Central password		No
--file	The full path to the license file, with "dat" extension		Yes

license-report

Displays details of the current license. For example, the number of days left in a trial period.

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	OO Central password		No

Upgrade from OO 9.x

historical-data-start-date

Sets the start date for the import of historical data.

HPE Operations Orchestration (10.80) Page 26 of 790

Use

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	OO Central password		No
--date	The date from which to start the import, in format dd/MM/yyyy		Yes

historical-data-status

Gets the status of historical data import.

import-historical-data

Imports the historical data of deployed content with the data defined in the 9.x server.

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	OO Central password		No

import-permissions

Imports the permissions of deployed content with the permissions file.

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	OO Central password		No

import-schedules

Imports the schedules of deployed content with the schedules defined in the 9.x server.

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	OO Central password		No

permissions-status

Gets a list of all the permissions that were imported so far.

Argument	Description	Default Value/Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment,		No

Argument	Description	Default Value/ Behavior	Required
	enter the user name in the format `domain\username`.		
--password	OO Central password		No
--output	Full path where the output should be written		No

upload-permissions-file

Uploads the permissions file created during content upgrade.

Argument	Description	Default Value/ Behavior	Required
--url	URL of OO 10.x Central		Yes, unless already connected to Central with the `connect` command
--user	OO Central user name In a multi-LDP environment, enter the user name in the format `domain\username`.		No
--password	OO Central password		No
--file	The full path to the permissions file, which is in the **Output** folder of the Content Upgrade Utility		No

> **Note:** After an upgrade from OO 9.x to 10.x (earlier than 10.02), using the OO Shell Utility to migrate historical data will get the result "Resolved" instead of "RESOLVED", as required in OO 10.x. This causes duplicates in the results distribution on the Dashboard. If this occurs, upgrade to the latest version of OO 10.x.

Running Studio Commands from the Command Line

This document shows you how to use Operations Orchestration Shell Authoring (OOSHA) utility to run OO authoring commands from the command line, instead of from the Studio interface.

Introduction

The OO Shell Authoring (OOSHA) utility lets you run OO commands from the command line, instead of from the Studio 10.x interface. This enables you to automate some of your OO authoring processes.

> **Important:** In order to create and write to the oosha log file (**<oosha_installation_folder>/logs/oosha.log**) in the installation folder, you must run OOSHA with write permissions.

Installing OOSHA

There are two ways to install OOSHA:

- As a standalone package

- Within an OO installation (as of version 10.60)

Install a Standalone OOSHA

1. Download the **oosha.zip** file.

2. Unzip the **oosha.zip** file to the desired location.

3. To start OOSHA, navigate to the **bin** folder, and run the **oosha.bat** executable file for the Windows operating system or the **oosha.sh** executable file for the Linux operating system.

Install OOSHA within OO

1. OOSHA is installed and uninstalled with OO Studio.

2. To start OOSHA, navigate to **<oo_installation_folder>/studio/tools**, and run the **oosha.bat** executable file for the Windows operating system or the **oosha.sh** executable file for the Linux operating system.

Running Operations Orchestration Authoring Commands from the Command Line
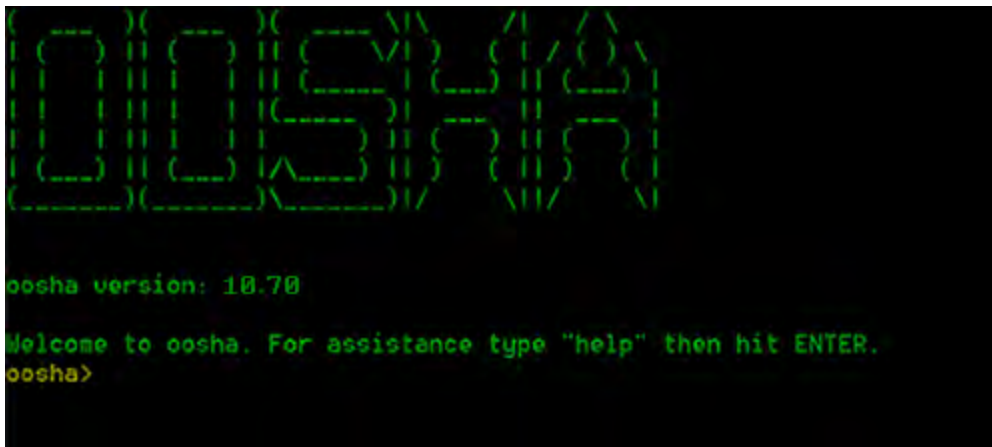
Run authoring commands

You can run commands within the OOSHA utility or from a normal prompt/shell.

1. Start the OOSHA shell: Navigate to **<oo_installation_folder>/studio/tools**, and run the **oosha.bat** executable file for the Windows operating system or the **oosha.sh** executable file for the Linux operating system.

> **Note:** : In a standalone OOSHA installation, **<oosha_instalation_folder>** is the location where the **oosha.zip** file was unzipped. In an installation of Operations Orchestration, **<oosha_instalation_folder>** is **<oo_installation_folder>/studio/tools**.

After you have run the executable file, the OOSHA shell appears.

2. Type the command in the following format:

   <commandName> --<parameterName1> <parameterValue1>

   --<parameterName2> <parameterValue2>

   Some of the available commands are:

   ○ `createCP` - enables you to package a content pack. For details, see "Creating Content Packs via OOSHA" on page 34

   ○ `help` - displays a list and description of all available commands

   ○ `exit, quit` - exits OOSHA

   ○ `cls, clear` - clears the console

   ○ `system properties` - displays the OOSHA system properties

   ○ `date` - displays the current date

   ○ `version` - displays the current version of OOSHA

3. You can run commands from a normal prompt/shell using the OOSHA utility by passing arguments to the oosha.bat or oosha.sh files respectively.

   Examples:

   ```
   oosha.bat createCP --projectFolder C:\Users\User1\.oo\Workspace\project1 --
   version 7
   ```

```
./oosha.sh createCP --projectFolder '/home/user1/.oo/Workspace/my project' -
-destinationFolder  /tmp/cps --description 'test content pack' --publisher
'the publisher' --version 1.9
```

Display Help in OOSHA

- To display a list and description of all available commands in the OOSH utility, type: `help`

- To display help about a particular command, type: `help` <command>

  For example: `help createCP`

- To display a list of arguments for a valid command that has at least one argument, type: <command>

For example: `createCP`

```
oosha> createCP
You should specify option (--projectFolder, --destinationFolder, --version, --publisher, --description, --includeSystemAcc
--incrementVersion) for this command
oosha>
```

Navigate the command history

Use the **Up** arrow key or **Down** arrow key on your keyboard to navigate inside the command history for OOSHA.

- The **Up** arrow key history navigates one item back in the history of commands.

- The **Down** arrow key navigates one item forward in the history of commands.

Troubleshoot OOSHA

The OOSHA application logs are stored inside the **<oosha_instalation_folder>/logs** folder.

By default, under the **logs** folder, there are two log files: **oosha.log** and **maven.log**.

Change the OOSHA log location (oosha.log)

1. Locate the **log4j.properties** file at **<oosha_instalation_folder>/conf/log4j.properties**.

2. Locate the following line inside the **log4j.properties** file:

   `log4j.appender.FILE=org.apache.log4j.RollingFileAppender`

3. Below that line, add the following line:

   `log4j.appender.FILE.File=<absolute path of the oosha.log file>`

   For example: `log4j.appender.FILE.File=C:/logs/oosha.log`

4. Save the **log4.properties** file and start the OOSHA shell.

Change the Maven log location (maven.log)

1. Locate the **oosha.properties** file at **<oosha_instalation_folder>/conf/oosha.properties**.

2. Edit the **oosha.properties** file by setting the **oosha.logs.location.maven** property with the value of the absolute path of the **maven.log** file

   For example:

   `oosha.logs.location.maven=C:/logs/maven.log`

3. Save the **oosha.properties** file and start the OOSHA shell.

Configure the OOSHA encryption

> **Note:** A content pack may contain either CloudSlang content or content that was developed from a Studio project. This section only applies to content that developed from a Studio project.

By default, OOSHA uses the same encryption options as a default Studio installation. To configure OOSHA to use the encryption of another Studio installation, set the value of the **oosha.encryptor.dir** property inside the **oosha.properties** file to point to the desired encryption folder.

1. Locate the **oosha.properties** file **<oosha_instalation_folder>/conf/oosha.properties**.

2. Edit the **oosha.properties** file by setting the **oosha.encryptor.dir** property with the value of the absolute path of the Studio installation security folder.

   For example:

   `oosha.encryptor.dir=`<mark>`<oo_installation_folder>`</mark>`/studio/var/security`

3. Save the **oosha.properties** file and start the OOSHA shell.

Creating Content Packs via OOSHA

Using the `createCP` command, you can create a content pack from a project developed in Studio or in CloudSlang.

While a content pack is being created via OOSHA, the project undergoes a series of structural and consistency validation checks.

> **Note:** For information on how to create a new project, see Create a new project in Studio and Create a new project from a template.

Create a content pack

The `createCP` command packages a project into a content pack. The content pack will contain all the objects in the project.

> **Note:** The invalid flows will also be included.

For a complete list of all the options you can use with the `createCP` command, see the Reference section or type `help createCP` in the command line.

1. Run the **oosha.bat** executable file for the Windows operating system or the **oosha.sh** executable file for the Linux operating system.

2. In the command line, type the following command, replacing the highlighted values with your own

values:

```
createCP --projectFolder <projectFolder> --destinationFolder
<CPDestinationFolder> --version <CPversion> --publisher <publisher> --
description <description> --includeSystemAccountPasswords
<shouldIncludeSAPasswords> --overrideDestination <shouldOverrideDestination> --
type <contentType>
```

3. Press the **Enter** key.

   The content pack is created in the location that was defined in the `--destinationFolder` argument.

   The name of the content pack is taken from the **pom.xml** file. If not defined there, it is taken from the **contentpack.properties** file. If not defined there, it is taken from the project folder.

Validate a content pack

When you run the `createCP` command from OOSHA, the structure of the project is validated in the same way as in Studio.

The `createCP` command will fail if the project includes corrupt XML or CloudSlang files, or files that do not respect the Operations Orchestration validation schema.

For example:

OOSHA validates that the project contains either XML or CloudSlang (.sl) files, but not both. If it contains both, you must specify the type of the content using the `--type` option.

Possible values:

- **afl** – the project was developed in Studio and contains XML content

- **cs** – the project contains CloudSlang content

If this option is specified, only items of the specified type (XML or CloudSlang) will be included in the resulting content pack.

> **Note:** In this version, OOSHA does not validate the following:
>
> - Start steps
>
> - Connected transitions
>
> - Lanes
>
> - Operation responses
>
> - Folder and file structure
>
> - No missing references to other flows, operations or configuration items inside this project, as well as other projects and content packs
>
> OOSHA does not exclude invalid items (flows, operations, configuration items) from the resulting content pack.

Include hard copies Inside the content pack

> **Note:** A content pack may contain either CloudSlang content or content that was developed from a Studio project. This section only applies to content that was developed from a Studio project.

In order for OOSHA to pack operation dependencies (operations' plugins and their dependencies) for hard copy operations, you must configure the **oosha.maven.home** property inside the **oosha.properties** file by following the steps below.

1. Locate the **oosha.properties** file **<oosha_instalation_folder>/conf/oosha.properties**.

2. Edit the **oosha.properties** file by setting the **oosha.maven.home** property with the value of the absolute path of the repository containing the operation dependencies.

   The value of this property is typically **<user_home>/.oo/data/maven**. This is the absolute path of the Studio maven repository folder that was used to create the project, **<studio_workspace>/.oo/data/maven**.

   If this property is not specified, the default maven home is **<user_home>/.m2/repository**.

3. Save the **oosha.properties** file and start the OOSHA shell.

Include CloudSlang operations inside the content pack

In order for OOSHA to correctly pack CloudSlang operations, it might need to include their Java dependencies transitively. All Java dependencies are packaged as Maven artifacts.

For collecting all necessary artifacts inside the content pack, you must configure some properties inside the **oosha.properties** file by following the steps below.

1. Locate the **oosha.properties** file **<oosha_instalation_folder>/conf/oosha.properties**.

2. Edit the **oosha.properties** file by setting the the following properties:

   ○ Set the **oosha.maven.home** property with the value of the absolute path of the local repository that OOSHA will use to get dependencies from. If a dependency cannot be found in this repository, OOSHA will try to download it from a remote one.

   If this property is not specified, the default maven home is `<user_home>/.m2/repository`

   ○ Set the **oosha.maven.remote.repository** property with the value of the URL of the remote repository that OOSHA will use for dependency download.

   If this property is not specified, the default remote repository is Maven Central.

   ○ Set the **oosha.maven.settings** property with the value of the absolute path of the Maven settings file to use for configuration.

   If no Maven settings file is provided, OOSHA will use the configurations provided using the two properties above. If a file is provided, it will override any configuration specified in the properties above.

3. Save the **oosha.properties** file and start the OOSHA shell.

   Note: OOSHA relies on the Maven Dependency plugin and the Maven Install plugin for collecting dependencies. If these do not exist in your local repository, OOSHA will try to download them from the remote one.

Configure System Accounts

Note: A content pack may contain either CloudSlang content or content that was developed from a Studio project. This section only applies to content that developed from a Studio project.

In some cases, you may want to a package a content pack from a project that contains system accounts without including the system account passwords.

For example, when you want to run the `createCP` command using a false value for the `--includeSystemAccountPasswords` argument or when the value for the `--includeSystemAccountPasswords` argument is not provided.

In such a case, you must first configure the **oosha.encryptor.dir** property inside the **oosha.properties** file by following the steps below.

1. Locate the **oosha.properties** file **<oosha_instalation_folder>/conf/oosha.properties**.

2. Edit the **oosha.properties** file by setting the **oosha.encryptor.dir** property with the value of the absolute path of the Studio installation security folder **<oo_installation_folder>/studio/var/security**.

3. Save the **oosha.properties** file and start the OOSHA shell.

Create a new project in Studio

The complete process of creating a new project using Studio is documented in the *OO Studio Authoring Guide.*

The project generated by Studio will have the following folder structure:
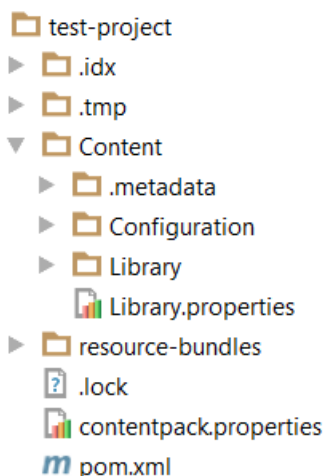
In the example above, the name of the project is **test-project**.

To place CloudSlang or XML files inside this project, you have to create a new subfolder under **Library** and place them inside. Then, you can build the content pack using OOSHA.

> **Note:** Do not place the files directly under the **Library** folder, as this will not work.

Create a new project from a predefined template

To create a new project without Studio:

1. Download **template-project.zip**.

2. Unzip **template-project.zip** to the desired location.

3. Find the directory template-project with the following structure:

```
☐ template-project
▼ ☐ Content
    ▶ ☐ Configuration
    ▶ ☐ Library
▶ ☐ resource-bundles
  ☐ contentpack.properties
  𝑚 pom.xml
```
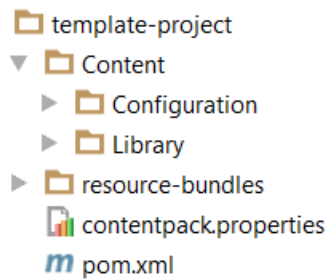
4. Rename the **template-project** directory to the name you want to provide for the new project.

    ○ For a CloudSlang project, the project naming convention is: **cs-[<vendor>]-<product|technology>**

    For example: cs-vmware-vcenter

    ○ For a XML project, the naming convention is: **[<vendor>]-<product|technology>**

    For example: vmware-vcenter

5. Locate the **pom.xml** file. Open it, and replace the value inside the **artifcactId** tag with the name of the project.

    For example: <artifactId>cs-vmware-vcenter </artifactId>)

6. Locate the **contentpack.properties** file. Open it, and provide values for the properties:

    ○ **content.pack.name** - It is recommended to specify a value for this property. The value should be the name of the project.

    ○ **content.pack.uuid** - This value is mandatory. You must provide a valid uuid value before running the **createCP** command.

    ○ **content.pack.version** - It is recommended to specify a value for this property. The value must be the same as the value from the version tag inside the **pom.xml** file. If you want to update the version of the project, you must update it inside both files.

    ○ **content.pack.publisher**

Reference Material

Available Arguments for the createCP Command

Argument	Description	Default Value/Behavior	Required
--projectFolder	The location of the project to be packaged. This must be the absolute		Yes

Argument	Description	Default Value/Behavior	Required
	path of the folder that represents a project in the Studio format.		
--destinationFolder	The path to a local folder where the project will be packaged into a content pack. The destination folder and all parent folders up to the destination folder will be created, if they do not exist on the file system.	• The default location is where the content pack was created the previous time. • The first time the content pack is created, the default path is: C:\Users\<user_name>\.oo\Content Packs	No
--version	Lets you specify the version of the content pack.	1.0.0 If the version is not specified, the default version number is taken from the project, with the "-SNAPSHOT" suffix removed. For example, if the project version is **1.7.0-SNAPSHOT**, the content pack version will be **1.7.0**.	No
--incrementVersion	The content pack version is taken from the project version with the "-SNAPSHOT" suffix removed, and the project version is increased by 0.0.1. For example, assuming that you have not specified a version number with --version: If the project version is **1.7.0-SNAPSHOT**, the content pack version will be **1.7.0** and the project	False The default is that the project version is not incremented.	No

Argument	Description	Default Value/Behavior	Required
	version will be moved up to **1.7.1-SNAPSHOT**.		
`--publisher`	Lets you specify the publisher of the content pack.	"" (empty)	No
`--description`	Lets you specify a description of the content pack.	If this is not specified, the description is taken from the **contentpack.properties** file.	No
`--includeSystemAccount Passwords`	Lets you specify whether the content pack should include system account passwords. See "Exporting a Content Pack" in the *Operations Orchestration Studio Authoring Guide*."Authoring Using Studio" on page 284. For security reasons, it is recommended to configure the system account passwords in Central.	False	No
`-- overrideDestination`	Lets you specify whether to override an existing file with the same name as the content pack in the destination folder. If the destination file that the content pack represents exists on the file system, and `overrideDestination` is set to false, the content pack will not be created. If the destination file that the content pack represents exists on the	False	No

Argument	Description	Default Value/Behavior	Required
	file system, and `overrideDestination` is set to true, the content pack will override the original destination file.		
`--type`	Specify the type of content from the project. Possible values: afl, cs `afl` means that the project contains XML files with flows and operations. `cs` means that the project contains CloudSlang files. Only files of the specified type will be included in the resulted content pack.	The type of the content from the project will be auto-detected. If the project contains both XML and CloudSlang files, OOSHA will not manage to detect the type, and will fail to build the content pack. The solution for this situation is to specify the type of the content. If the project is empty, the detected type will be `afl`.	No

Using Central

This document shows you how to use OO Central to promote, run, and schedule flows, configure the system, and extract and analyze data resulting from flow runs.

Welcome to Operations Orchestration Central User Guide

Operations Orchestration Central is a web-based application used for promoting flows, running and scheduling flows, administering the system, and extracting and analyzing data resulting from flow runs.

Major Tasks and Personas

The main personas who work with Central are the Operations Administrator (Op Admin) and the End User. However, some of the initial setup tasks are performed by the System Administrator (Sys Admin).

Operational Administrator (Op Admin)

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

The Op Admin uses Central to:

- Promote content packs - see "Promoting a Content Pack – Overview" on page 157

- Run and monitor flows - see "Running and Monitoring a Flow – Overview" on page 208

End User

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

The End User uses Central to:

- Run and monitor flows - see "Running and Monitoring a Flow – Overview" on page 208

System Administrator (Sys Admin)

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

The Sys Admin uses Central to set up the Central configurations:

- Configure the topology – see:

  "Setting Up Topology – Workers and RASes" on page 114

  "Setting Up Topology – Automatic RAS Upgrade" on page 127

- Configure security and authentication – see:

"Setting Up Security – Roles" on page 66

"Setting Up Security – Internal Users" on page 99

"Setting Up Security – LDAP Authentication" on page 76

"Setting Up Security – LWSSO" on page 110

"Setting up Security Settings" on page 60

Sample Scenario

	System Admin		Op Admin		Flow Author

Setup

	After the installation of Central, the **system administrator**, configures the LDAP authentication (see "Setting Up Security – LDAP Authentication" on page 76).
	The system administrator sets up the Central roles and maps these roles to the LDAP groups (see "Setting Up Security – Roles" on page 66).
	The system administrator assigns workers to the worker groups. He creates a new group, "Windows", for the external worker (see "Setting Up Topology – Workers and RASes" on page 114).
	The operations administrator deploys the base content in the Central servers (see "Deploying and Managing Content Packs" on page 160).

Promotion

	The flw author creates a new content pack in Studio and deploys it to the file system.
	The operations administrator receives an email from the flow author, saying that a new content pack is ready for deployment.
	The operations administrator deploys the new content pack to the Central Staging server (see "Deploying and Managing Content Packs" on page 160).
	The operations administrator maps the Central user accounts to the system accounts in the content pack (see "Setting Up Configuration Items for a Content Pack" on page 193).

	The operations administrator sets up the system properties in the content pack (see "Setting Up Configuration Items for a Content Pack" on page 193).
	The operations administrator verifies that the Windows alias was automatically assigned to the worker group and maps the other group aliases to actual worker groups (see "Setting Up Configuration Items for a Content Pack" on page 193).
	The operations administrator takes a look at the flow library and verifies that all the flows are there and that when she selects a flow, she can see the flow meta data (see "Managing the Flow Library" on page 180).
	The operations administrator sets the permissions for the flows in the content pack (see "Managing the Flow Library" on page 180).

Running and monitoring flows

	The operations administrator triggers one of the flows in the content pack (see "Running a Flow" on page 209).
	The operations administrator navigates to the **Runs** view to see the flow status (see "Tracking and Managing Flow Runs" on page 244).
	The operations administrator pauses, resumes, or cancels the flows, if required (see "Tracking and Managing Flow Runs" on page 244).
	The operations administrator drills down to the flow details and tests the flow, to see if there are any problems (see "Testing and Troubleshooting a Flow Run" on page 256).

Updating the content pack
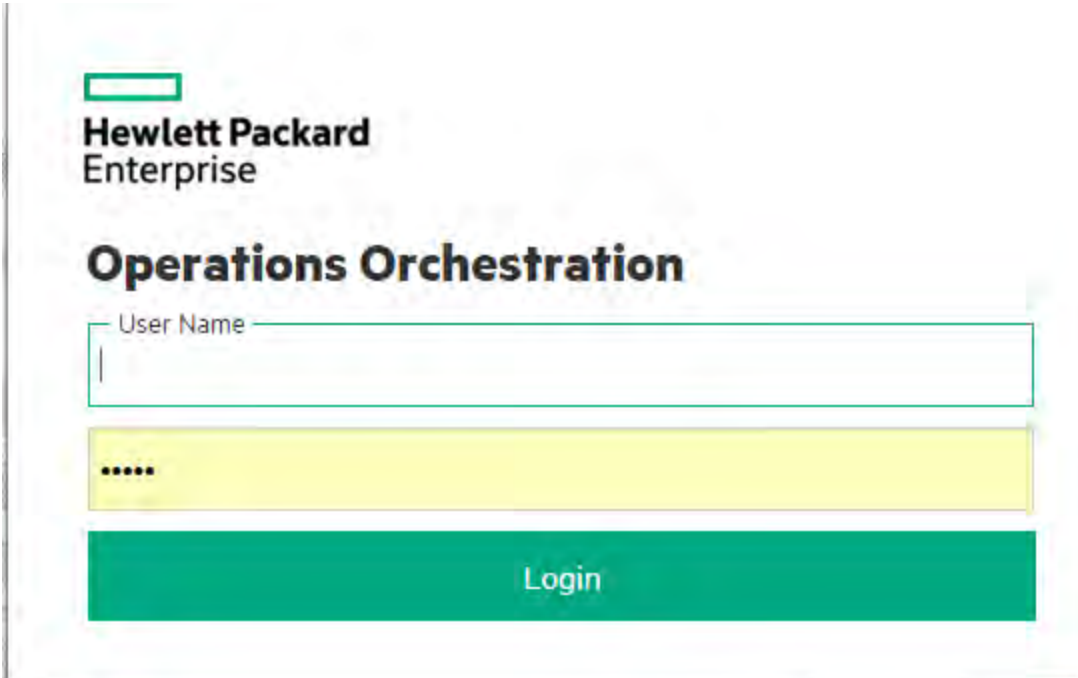
	The operations administrator identifies a problem with the content pack. It is too large, and she wants it to be split into two smaller content packs.
	The operations administrator sends an email to the flow author, asking him to split the content pack into two.
	The flow author splits the content pack into two smaller content packs and sends them to the operations administrator.

Final deployment

	The operations administrator deploys the new content packs to the Staging server and verifies that there are no other problems.
	The operations administrator deploys the content packs to the Production server.

Logging into OO

If authentication has been set up for OO, the Logon screen is shown when OO is started.

1. Enter your user name and password.

   Note: You cannot log in with a user name that includes a colon.

2. If multiple domains are available, you also need to select a domain. The domain that you select will remain the default choice for the browser that you're using.

   Note: Your current domain is selected by default.

   There is no choice of domain if:

   ○ Only internal users have been set up

   ○ There is only one domain and this has been set as the default

      Note: If there is a single domain, but it has not been set as the default, you will have a choice between this domain and the internal users.

      If there are multiple domains, you must select a domain name, unless you are logging in as an internal user or if you belong to the default LDAP.

3. If the option to capture logged-in user credentials has been enabled, the Login screen includes a

warning message, saying that OO will capture your credentials and may transfer them over the network.

For information about how to set up the capture of logged-in user credentials, see "Setting up Security Settings" on page 60.

4. Click **Login**.

If authentication has not been enabled, the Logon screen is not shown. For information about how to set up authentication for OO, see "Setting up the System Configuration" on page 59.

Navigating Operations Orchestration Central

Operations Orchestration (OO) Central is divided into four workspaces, so you can perform end-to-end tasks from the same place. A workspace is a unit that holds all the screens that belong to the same end-to-end task group.

- **Dashboard** – used to display the system's ROI, and analyzed flow aggregation. For users with an Op Admin ![icon] role.

- **Run Management** – used for running flows, monitoring runs, scheduling runs, and troubleshooting runs. Can be used by users with an End User ![icon] or Op Admin ![icon] role.

- **Content Management** – used for promotion tasks, such as deploying new content, setting permissions on flows, setting up configuration items, and rolling back to earlier versions of content packs. For users with an Op Admin ![icon] role.

- **System Configuration** – used to configure topology, and set up users, roles, LDAP authentication, IDM authentication, LWSSO, security banners, monitor and control the size of the database, and personalize the Central UI appearance. For users with a System Administrator ![icon] role.

Dashboard

Click the **Dashboard** button to display the Dashboard workspace.

This workspace provides statistical information about the system (popular flows, result distribution, execution time, and so on) and financial information about ROI (return on investment).
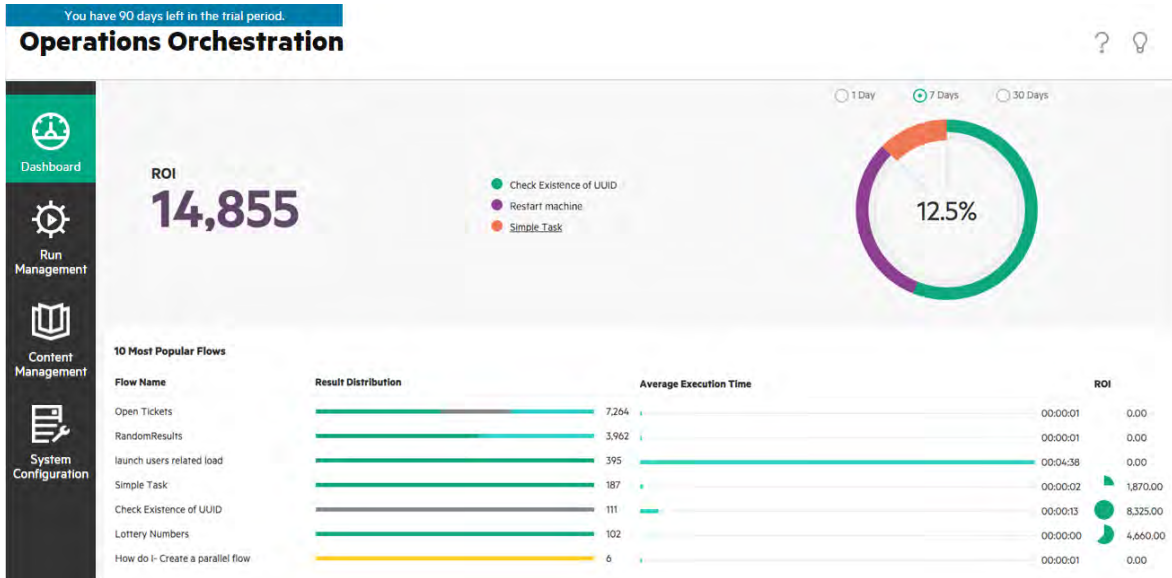
Run Management

Click the **Run Management** button to display the Run Management workspace.

This workspace includes the following modules:

- **Run Explorer** – In the Run Explorer module, the Op Admin or End User can monitor their running flows and the flows that have finished running. You can track flow runs, monitor their progress, and perform actions on flow runs, such as pausing, resuming, and canceling them. If a flow fails and you want to troubleshoot, you can drill down into the run to display detailed information. If you have many flows running at the same time, you can use filters to locate the flow that you need.

- **Flow Launcher** – In the Flow Launcher module, the Op Admin or End User can browse for a flow, view the flow information, name the run, enter inputs, and run the flow.
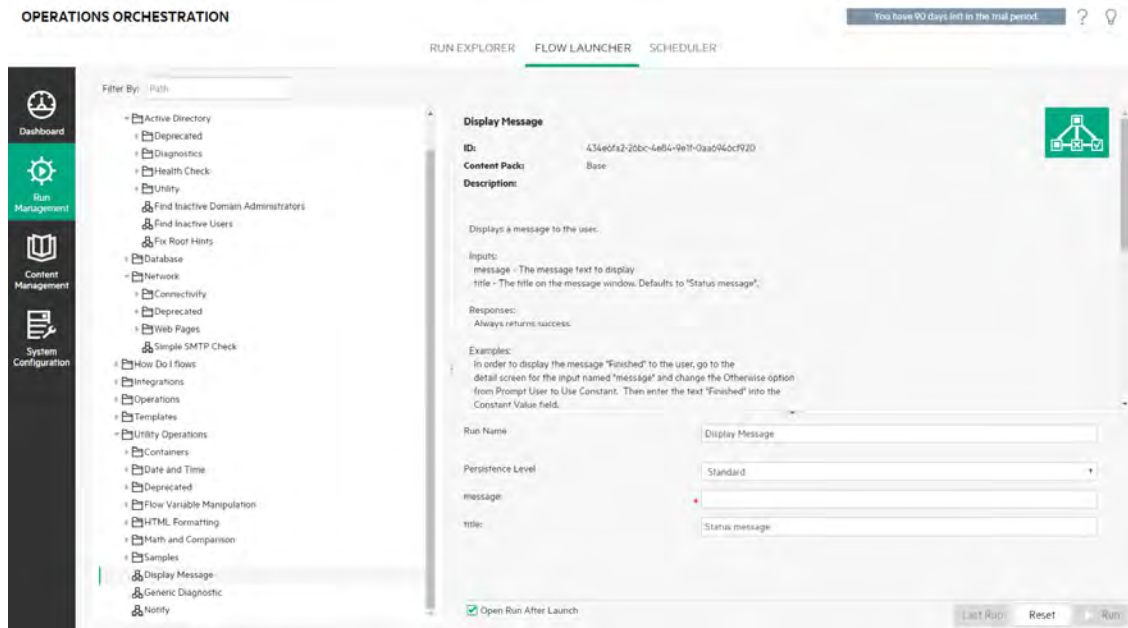
- **Scheduler** – In the Scheduler module, the Op Admin can view, create, and modify flow run schedules.
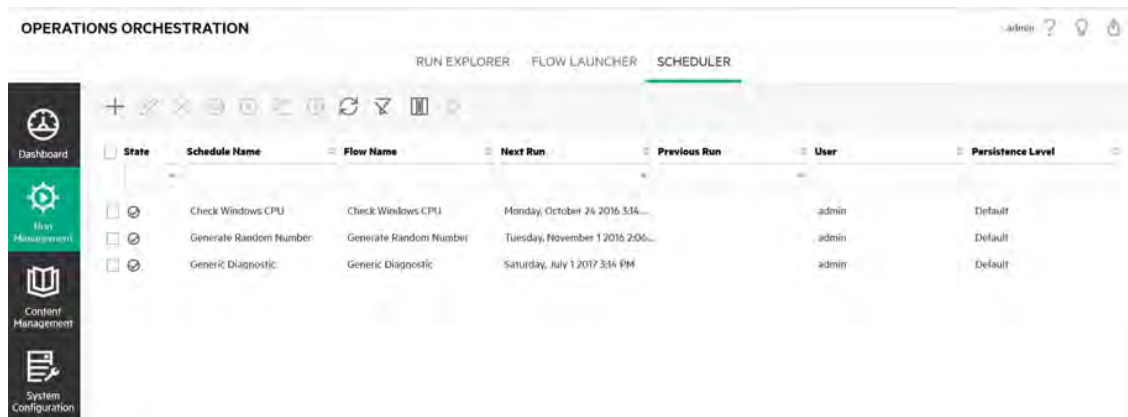
Content Management

Click the **Content Management** button to display the Content Management workspace.

This workspace includes the following modules:

- **Flow Library** – The Flow Library displays the flows that have been deployed. From here, the Op Admin can run or schedule a flow, view a flow's meta data, set the content permissions, or run a flow for testing.

- **Content Packs** - In this module, the Op Admin can view the deployed content packs, filter them, delete them, deploy or roll back content packs, view the details of content packs, compare statistics about deployed content packs, and view changes that were made to them.

- **Configuration Items** – In this module, the Op Admin can configure the contents of a content pack, in order to adjust it to the environment: mapping group aliases to actual worker groups, mapping system accounts in the content pack to Central user accounts, and mapping system properties.

System Configuration

Click the **System Configuration** button to display the System Configuration workspace.

This workspace contains the following modules:

- **Security** – Enables the System Administrator to set up roles for OO and to apply these roles to internal users or to users in an authentication system such as LDAP, IDM, or LWSSO. This is also where the System Administrator can enable authentication and auditing, and configure a security banner.
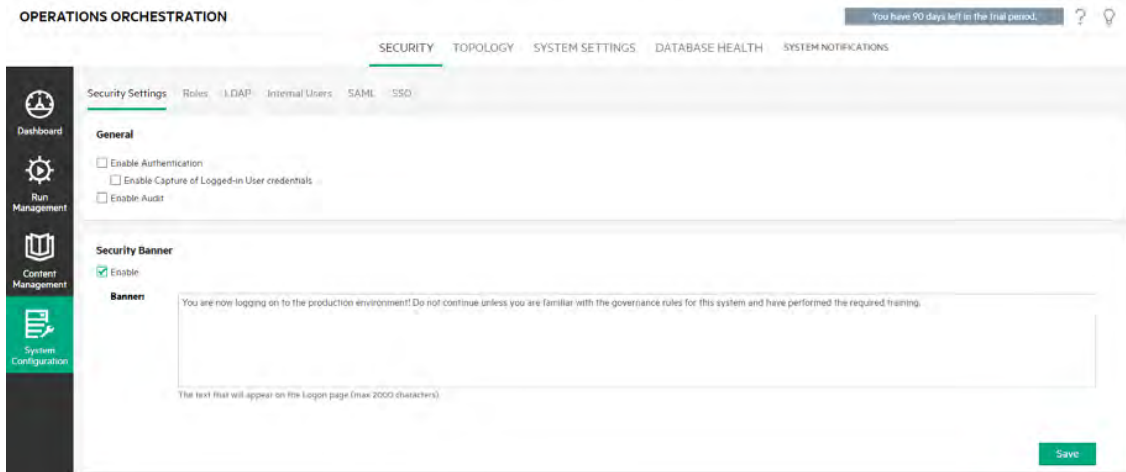
- **Topology** – Enables the System Administrator to configure workers, worker groups, and RASes, and to set up an automatic upgrade process for RASes.
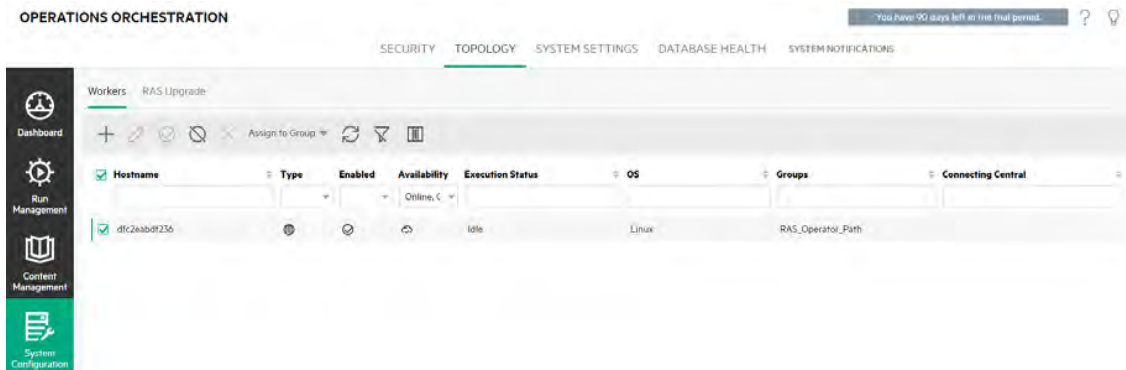
- **System Settings** – Enables the System Administrator to set defaults and customizations, and configure a forward proxy, reverse proxy, or load balancer. It also enables the System Administrator to install an Operations Orchestration license.

- **System Notifications** – Notifies users about system events through email messages.

Viewing the Workspaces

It is possible to set up the OO roles so that users who have been assigned particular roles will only be able to access the workspaces that are relevant to their role.

For example, you can set up roles so that:

- Users who belong to the **End User** role will see only the Run Management workspace.

- Users who belong to the **Promoter** role will see the Content Management workspace and the Run Management workspace.

- Users who belong to the **System Administrator** role will see the System Configuration workspace and the Run Management workspace.

For more information about roles, see "Setting Up Security – Roles" on page 66.

Adjusting the Display of Panes in the Workspace

Use the sliders between panes, to adjust the display:

- Drag the edge of a slider to adjust the height or width of a pane.

- Use the open/close button in a slider to toggle between hiding and displaying a pane.
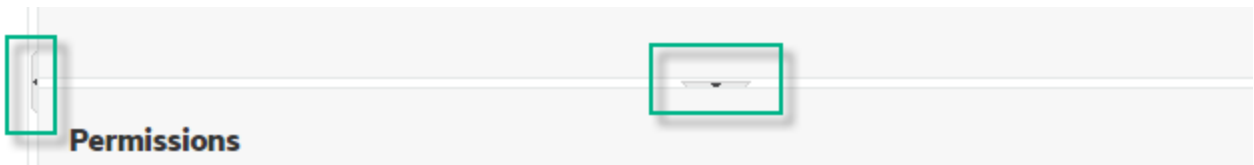
Working with the Dashboard

The Dashboard provides statistical information about the system (popular flows, result distribution, execution time, and so on) and financial information about return on investment (ROI).

The Dashboard shows information about all flows that the logged-in user is entitled to see.

The Dashboard enables the Op Admin to analyze the flow statistics, and compare the performance of the top ten flows.

> **Note:** The Dashboard does not include flows that were run using the Remote Debugger in Studio. This ensures that the statistics and ROI calculations in the Dashboard will not be affected by test runs.

The time range of the data in the Dashboard appears in the top right corner. This time range can be set to the last day, the last seven days, or the last thirty days.

> **Note:** It is also possible to perform a wider range of Dashboard tasks using APIs:
>
> - Set the date range with a wider range of possible date ranges.
>
> - Display a larger number of flows
>
> - Sort and filter data
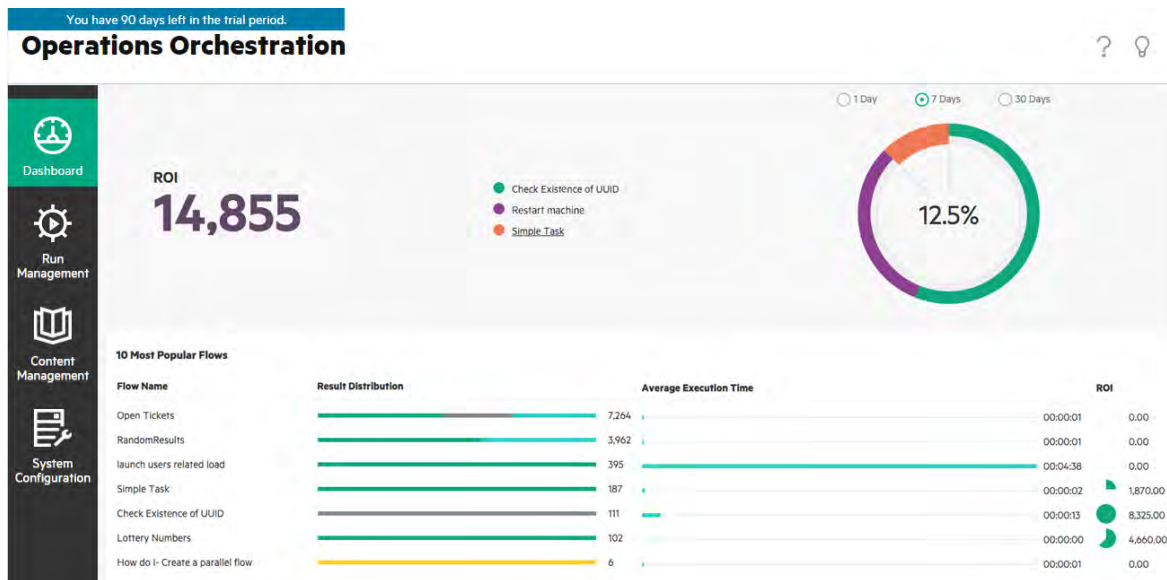
For information about how to interpret the Dashboard, see "Reference Material" on the next page.

> **Note:** The **Dashboard** tab is only visible if you have been assigned a role with the **View Dashboard** permission. For information about setting up roles, see "Setting Up Security – Roles" on page 66.

If you do not have ROI values assigned in your flows, the total ROI value will be 0.

> **Note:** You can personalize the Dashboard by adding text to the ROI title, in order to help users understand the meaning of the ROI values. For example, you can tell them if ROI is calculated in dollars or in work days. For more information, see "Setting Up the System Settings - General Settings" on page 134.

To see a movie on how to set an ROI value for your content, click the **Play** button under the movie thumbnail in the Dashboard.

What do you want to do?

Customize the Dashboard time range

In the dashboard, you can display information about all the flow runs that you are entitled to see, over the past day (the last 24 hours starting from now), the past week (the last 7 days starting from today), or the past 30 days (starting from today).
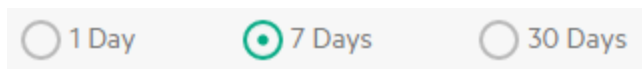
If you do not adjust the date range, the default is seven days.

> **Note:** It is also possible to set the date range using APIs, with a wider range of possible date ranges.

1. Click the **Dashboard** button to display the Dashboard workspace.

2. In the top right corner, select the option with the number of days that you want to include in the date range:

   ○ **1** - to display information about runs from the past day

   ○ **7** - to display information about runs from the past 7 days

   ○ **30** - to display information about runs from the past 30 days

Extract the ROI Information by API

You can extract the relevant ROI information by API, as a basis for periodical reports for the business manager.

For more information, see "Get Statistics" in the *OO API Guide.*

Reference Material

System ROI

GUI item	Description
Time range	All the data in the Dashboard is relevant to the specified time range in the top right corner. To adjust the date range, select one of the options : • **1 Day** - to display information about runs from the past day • **7 Days** - to display information about runs from the past 7 days • **30 Days** - to display information about runs from the past 30 days
Total ROI	This value is the sum of the ROI values of all performed transitions in the defined time range, which the logged on user is entitled to see.
Main Contributors pie chart	Displays the flows that were the major contributors to the ROI, with their relative contributions expressed in a pie chart.

10 Most Popular Flows

10 Most Popular Flows

Flow Name	Result Distribution		Average Execution Time	ROI	
ROI - Basic		46	00:00:05		460.00
Generate Random Number		6	00:00:02		0.00
Magic Vars - Basic		6	00:00:08		0.00
ROI - Parallel		4	00:00:15		1,780.00
ROI - Diagnosed		3	00:00:04		3,000.00

GUI item	Description
Results Distribution bar graph	For each of the top ten flows, displays the number of finished runs. Finished runs means those with a status of success, error, failed to complete, diagnosed, or no action taken. This does not include flows that are paused, running, canceled, pending cancel, or pending pause. The bar graph displays the relative distribution of runs according to status. The different statuses are expressed by the following colors: • **Green**: Completed - Resolved • **Red**: Completed - Error • **Gray**: Completed – No Action Taken • **Blue**: Completed - Diagnosed • **Yellow**: Failed to Complete The colors are the same as the colors of the status icons in the Run Explorer.

	Tip To identify the status indicated by a color, roll the cursor over a colored bar to display a tool-tip.
Average Execution Time bar	Displays the average duration of each flow, based on the finished runs for that flow over the time that was set as the date range. The bar displays the duration time of the flow, relative to the other flows in the top ten.
ROI column	Displays the ROI value of each flow. The coins represent the relative ROI score of the flow compared to the other flows in the top ten.

Setting up the System Configuration

Setting up the system configuration is usually done by the system administrator, after Operations Orchestration is installed.

Setting up Security Settings

The administrator can enable or disable the following security settings for Central:

- Authentication

- Capturing of logged-in user credentials

- Auditing

- A customized security banner
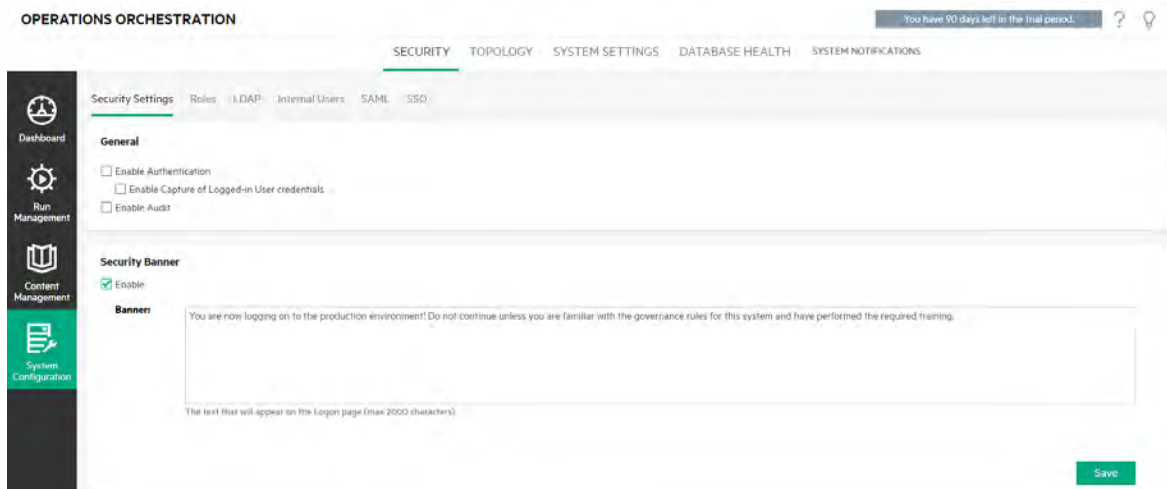
> **Note:** The **Security > Security Settings** tab is only visible if you have been assigned a role with the **Manage Security Configuration** or **View Security Configuration** permission. You will only be able to edit the settings if you have been assigned a role with the **Manage Security Configuration** permission.

Enabling Authentication

The **Enable Authentication** check box is available under the **Security > Security Settings** tab. You can use this to enable or disable the authentication functionality in Central.

> **Note:** You must enable authentication immediately after installation. If authentication is not enabled, all users will be able to access all tasks.

Only users with administrator credentials can enable authentication in Central. This is to prevent a user from becoming locked out of the system.

The **Enable Authentication** check box is only available if there are existing internal or LDAP users with permission to disable the authentication in the future.

Enabling Auditing

OO gives you the option to audit events, so that you can track security breaches. Auditing lets you track actions that took place on Central, such as logins, triggering flows, creating schedules, editing configurations, and so on.

The **Enable Audit** check box is available under the **Security > Security Settings** tab. You can use this to enable or disable the auditing functionality in Central.

Once auditing has been enabled, a user who has been granted **View Audit** permission can retrieve an audit trail.

Notes:

- By default, the **Enable Audit** check box is not selected.

- Audit events are persisted in the database.

- Currently, it is only possible to retrieve an audit trail via APIs.

Enabling the Capture of Logged-in User Credentials

You can enable OO to capture the credentials of the logged-in user. These credentials will then be used in flows that have inputs assigned from logged-in user credentials.

- When this option is **not enabled** (this is the default), the flow will prompt for a password at the relevant step.

- When this option is **enabled**:

  ○ If a user starts or resumes a flow that includes logged-in credentials as an input parameter, the credentials of the logged-in user will be captured in the database and in memory (in a protected manner) and used during the flow run. This enables you to make the flow run automatically.

    Note: The logged-in credentials that are captured are available to Central only in the context of this session. When the user logs out, the credentials are not saved in the database.

    The logged-in user name will contain the domain unless the user belongs to the default LDAP or the user is an internal user and a default LDAP is configured in the system.

  ○ When a user logs in to Central, a message appears, warning that the user's credentials might be captured and used during flow runs, if this is required by the runs. When debugging a Central remotely, the Central login page in Studio displays this warning.

  When to use this?

  This feature is designed for backward compatibility with OO 9.x and is not recommended for new flows. The best practice for new flows is to pass credentials as inputs.

Limitations

There are limitations to using this option:

- Capturing of logged-in user credentials in not supported on scheduled runs

- Capturing of logged-in user credentials is supported only for users logging to OO via the default LDAP or an internal user repository

Note: It is also possible to enable this option by using OOSH to set the value of the **record.loggedin.user.credentials** system property to **true**, as follows:

```
ssc --key capture.loggedin.user.credentials --value true
```

For more information, see the *OO Shell (OOSH) User Guide*.

Warning! If you enable the capture of the logged-in user credentials, this causes these credentials to pass through the network. Therefore, we highly recommend to use a secured network in order to protect the user credentials.

Note: When this setting is changed, if auditing is enabled, an audit log is created.

Setting Up a Security Banner

You can configure a security banner to appear before the Logon screen. This banner may be useful if you want to warn users about security rules and issues.

When a security banner is enabled, it appears as a pop-up dialog box before the Logon screen, and the user needs to click an **OK** button to continue.

What do you want to do?

Enable authentication

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Click the **Security** tab and then click the **Security Settings** tab beneath it.

3. Select the **Enable Authentication** check box to display the Enable Authentication dialog box.

4. In the Enable Authentication dialog box, enter your administrator user name and password, and click **Yes**.

   Note: If a choice of LDAP domains has been set up, you will also need to select the domain for the authentication.

Disable authentication

1. After the authentication functionality has been enabled, clear the **Enable Authentication** check box.

2. In the confirmation dialog box, click **Yes**.

Enable auditing

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Click the **Security** tab and then click the **Security Settings** tab beneath it.

3. Select the **Enable Audit** check box.

4. In the confirmation dialog box, click **Yes**.

Disable auditing

1. After the auditing functionality has been enabled, clear the **Enable Audit** check box.

2. In the confirmation dialog box, click **Yes**.

Enable the capture of logged-in user credentials

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Click the **Security** tab and then click the **Security Settings** tab beneath it.

3. Select the **Enable capture of logged-in user credentials** check box.

4. In the confirmation dialog box, click **Yes**.

Disable the capture of logged-in user credentials

1. If the capture of logged-in user credentials functionality has been enabled, clear the **Enable capture of logged-in user credentials** check box.

2. In the confirmation dialog box, click **Yes**.

Add a security banner

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Click the **Security** tab and then click the **Security Settings** tab beneath it.

3. In the **Security Banner** section, select the **Enable** check box.

4. In the **Banner** text box, enter the text that you want to appear on the banner. The maximum length

of this text is 2000 characters.

5. Click **Save** to save the banner.

Edit a security banner

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Click the **Security** tab and then click the **Security Settings** tab beneath it.

3. Make the required changes in the **Banner** text box.

4. Click **Save**.

Remove a security banner

If you clear the **Enable** check box, but do not delete the text from the **Banner** text box, you can save this text for later use.

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Click the **Security** tab and then click the **Security Settings** tab beneath it.

3. Clear the **Enable** check box in the **Security Banner** section.

4. Click **Save**.

Reference Material

Security > Enable Authentication

GUI item	Description
User Name	Enter your administrator user name.

	Note: If a choice of LDAP domains has been set up, you will also need to select the domain for the authentication.
Password	Enter your administrator password.
OK	Click to save the credentials and enable the authentication.

Security > Security Settings

GUI item	Description
General section	
Enable Authentication check box	Select the check box to enable the user authentication functionality in OO. If this check box is cleared, there is no authentication and all users can access all tasks.
Enable Capture of Logged-in User Credentials	Select the check box to enable OO to capture the credentials of the logged-in user.
Enable Audit check box	Select the check box to enable the auditing functionality in OO. Users who have been granted **View Audit** permission will be able to retrieve an audit trail.
Security Banner section	
Enable check box	Select the check box to enable the security banner, so that it is displayed in the Logon screen.
Banner text box	Enter the text for the security banner. The maximum length of this text is 2000 characters.
Save	Click to save the security banner.

Setting Up Security – Roles

Access to OO Central tasks is managed using Role-Based Access Control (RBAC). RBAC lets the administrator control which actionsOO users are allowed to perform, based on their role.

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

- A **role** is a collection of permissions. Roles can be assigned to users.

For example, you could create a role called FLOW MANAGER with permission to run flows, view schedules, schedule flows, manage runs, and create reports. An END USER role could have permission to run flows, but not to schedule them. If a role is set up with no schedule permissions, users with this role do not even see the Scheduler module.

The large number of available permissions enables a refined definition of which parts of the web UI each role sees. The administrator can formulate a dedicated UI experience for each group of users . This enables end users to use Central and only be exposed to the information they need, limiting their ability to harm the system.

> **Note:** The **Role** tab is only visible if you have been assigned a role with the **Manage Security Configuration** or **View Security Configuration** permission. You will only be able to edit the roles if you have a role with the **Manage Security Configuration** permission.

Once the roles are set up, they can be assigned to internal users or mapped to LDAP or SAML users:

- The administrator can map a role to one or more groups, so that LDAP users who are assigned to these groups will be assigned the permissions in the role.

- The administrator can assign a role to an internal user, while setting up the user configuration. For more information, see "Setting Up Security – Internal Users" on page 99.

By default, Central includes the roles ADMINISTRATOR, END_USER, EVERYBODY, PROMOTER, and SYSTEM_ADMIN, corresponding to these roles in Studio.

> **Note:** Role-Based Access Control is not the same as content permissions. If you want to restrict access to data (flows and folders), you will need to configure the permission of that content. This is done as part of the promotion of a content pack. For information about content permissions, see "Managing the Flow Library" on page 180.

What do you want to do?

Create a role

> **Note:** You must have the **Manage Security Configuration** permission in order to create and edit roles.

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **Security** > **Roles**.

3. Click the **Add** ➕ button.

Role Details   ? ✕

Name   * ADMINISTRATOR

Description   Administration Role

Permissions   Group Mapping

Run Management

☑ View Schedules   ☑ Manage Schedules   ☑ Manage Others' Runs
☑ Remote Debugging

Content

☑ View Content Packs   ☑ View Configuration Items   ☑ Manage Content Settings
☑ Manage Content Packs ☑ Manage Configuration Items

System

☑ View System Settings   ☑ View Topology   ☑ View Security Configuration
☑ Manage System Settings ☑ Manage Topology   ☑ Manage Security
                                                      Configuration

☑ View Audit   ☑ Manage Data Cleanup

Dashboard

☑ View Dashboard

                                    Cancel   **Save**

---

4. In the **Name** box, enter a unique name for the role. The role name cannot have more than 255 characters.

5. In the **Description** box, enter a description of the role. The description cannot have more than 255 characters.

6. Select the check boxes next to the permissions that you want to assign to this role. The permissions are grouped according to the four Central workspaces. If a role is not assigned any permissions in a workspace, users with this role will not be able to enter that workspace.

Run Management

○ **View Schedules** – Permission to view schedules

○ **Manage Schedules** – Permission to create and modify schedules

○ **Manage Others' Runs** – Permission to modify runs that were triggered by other users

○ **Remote Debugging** – Permission to trigger the Studio Debugger on a remote Central. This permission also gives view and run permissions on all the Flow Library.

Content

○ **View Content Packs** – Permission to view content packs

○ **Manage Content Packs** – Permission to modify content packs, deploy content, and roll back a deployment.

○ **View Configuration Items** – Permission to view configuration items (for example, group aliases, system accounts, system properties)

○ **Manage Configuration Items** – Permission to modify the values of configuration items (for example, group aliases, system accounts, system properties)

○ **Manage Content Settings** - Permission to modify content permissions (for flows, system accounts, and folders), flow persistence settings, and flow timeout settings.

System

○ **View System Settings** – Permission to view the Monitoring and System Information reports and to view the system configuration values available through OOSH

> **Note:** These reports contain information for HP Support:
> - **/oo/reports/sysinfo** shows the system configuration
> - **/oo/monitoring** is a performance monitoring dashboard

○ **Manage System Settings** – Permission to configure the log level using REST API, to manage the system configuration values available through OOSH, and to manage the Audit configuration

○ **View Topology** – Permission to view workers and worker groups

- **Manage Topology** – Permission to enable/disable workers and configure worker groups

- **View Security Configuration** – Permission to view the security configuration. This includes viewing internal users, roles, LDAP authentication, LWSSO, SAML, and Security banner configurations.

- **Manage Security Configuration** – Permission to modify the security configuration. This includes configuring internal users, roles, LDAP authentication, LWSSO, SAML, and the Security banner.

- **View Audit** – Permission to access the audit trail.

- **Manage Data Cleanup** – Permission to use the data cleanup (purging) APIs.

Dashboard

- **View Dashboard** - Permission to view the Dashboard

7. To map this role to an LDAP group, click the **Group Mapping** tab.

   For more information, see Map a role to a group.

8. Click **Save** to save the role.

9. To enable the authentication functionality, select the **Settings** tab (under **Security**) and select the **Enable Authentication** check box to display the Enable Authentication dialog box. For more information, see "Setting up Security Settings" on page 60.

   > **Note:** The **Enable Authentication** check box is only available if there are existing internal or LDAP users with permission to disable the authentication in the future.
   >
   > If the **Enable Authentication** check box is not selected, there is no authentication and all users will be able to access all tasks.

Edit a role

You must have **Manage Security Configuration** permission in order to edit a role.

1. Select **Security** > **Roles**.

2. Select the role that you want to edit and click the **Edit** ✏️ button.

3. Make the required changes in the Role Details dialog box.

4. Click **Save**.

Delete a role
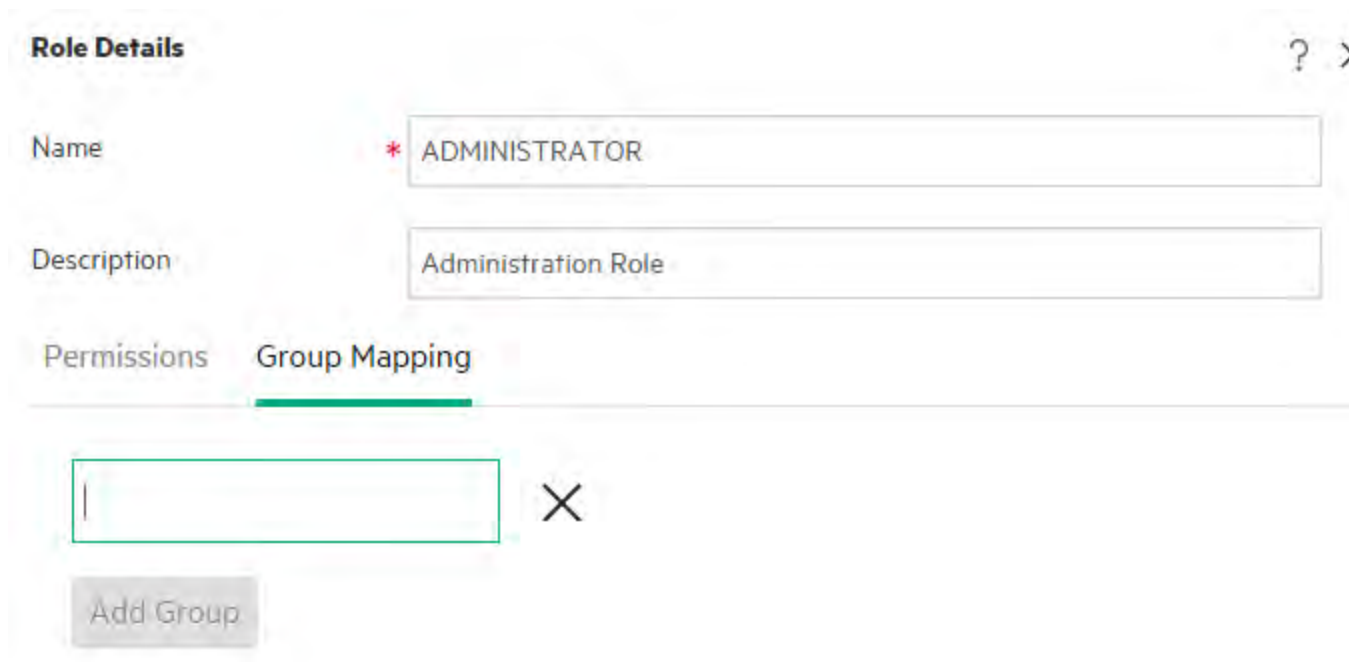
1. Select **Security** > **Roles**.

2. Select the role that you want to delete and click the **Delete** ✕ button.

3. Click **OK** in the confirmation dialog box.

Map a role to a group

Group mapping is case-sensitive. If you have two groups with similar names (for example, QG_Basic and qg_basic), which you assign to two different roles, a user logged in with one of these roles will not have access to the permissions of the other.

1. Select **Security** > **Roles**.

2. Select the role that you want to map to a group, and click the **Edit** 🖉 button.

3. In the Role Details dialog box, click the **Group Mapping** tab.

Role Details   ? ＞

Name	* ADMINISTRATOR
Description	Administration Role

Permissions   **Group Mapping**

[   ]   ✕

Add Group

4. Click the **Add Group** button to add a new mapping.

5. In the text box, type the name of the group to which you want to map this role.

6. If required, click the **Add Group** button again to map the role to further groups.

7. Click **Save**.

Delete the mapping of a role to a group

1. Select **Security** > **Roles**.

2. Select the role from which you want to remove the mapping to a group, and click the **Edit** ✎ button.

3. In the Role Details dialog box, click the **Group Mapping** tab.

4. Click the **Delete** ✕ button next to the mapping that you want to delete.

   > **Note:** There is no confirmation dialog box when you delete a mapping. If you change your mind, click **Cancel** to close the Role Details dialog box without saving.

5. Click **Save**.

Assign a role to be the default role

When a role is defined as the default role, it will be assigned to all users that have not been assigned a role. This means that all LDAP users will have the permissions from this default role, in addition to users that are explicitly associated with the role.

> **Note:** If you are setting a default role, it is recommended to set this as the one with the least privileges.
>
> Think carefully about whether you really want to set a default role. When you add permissions to that role, this will affect all LDAP users.

1. Select **Security** > **Roles**.

2. Select the role that you want to specify as the default role.

3. Click the **Set Default Role** ⬭ button in the **Roles** toolbar.

> **Note:** You can click the **Set Default Role** ⊘ button again to remove the default role functionality from this role.

Reference Material

Security > Roles

GUI item	Description
Add button +	Opens the Role Details dialog box, so that you can add a new role.
Edit button ✎	Opens the User Configuration dialog box, so that you can edit the selected role.
Delete button ✕	Deletes the selected role.
Set Default Role button ⊘	Makes the selected role the default role. If the selected role was set as the default role, makes it no longer the default role.
Role Name	Displays the role name.
Description	Displays a description of the role.
Group Mapping	Displays the groups that have been mapped to the role.
Filter Boxes	In the filter boxes at the top of the columns, enter text to filter the displayed roles.

Role Details dialog box > Permissions tab

GUI item	Description
Name	Enter a name for the role, of up to 255 characters.
Description	Enter a description of the role, of up to 255 characters.
Run Management	
View Schedules	Permission to view schedules.
Manage Schedules	Permission to create and modify schedules.
Manage Others' Runs	Permission to modify runs that were triggered by other users.
Remote Debugging	Permission to trigger the Studio Debugger on a remote Central. This

	permission also gives view and run permissions on all the Flow Library.
Content	
View Content Packs	Permission to view content packs.
Manage Content Packs	Permission to modify content packs, deploy content, and roll back a deployment.
View Configuration Items	Permission to view configuration items (for example, group aliases, system accounts, system properties).
Manage Configuration Items	Permission to modify the values of configuration items (for example, group aliases, system accounts, system properties).
Manage Content Settings	Permission to modify content permissions (for flows, system accounts, and folders), flow persistence settings, and flow timeout settings.
System	
View System Settings	Permission to view the Monitoring and System Information reports, which contain information for HP Support: • **/oo/reports/sysinfo** shows the system configuration • **/oo/monitoring** is a performance monitoring dashboard Permission to view the system configuration values available through OOSH.
Manage System Settings	Permission to configure the log level using REST API. Permission to manage the system configuration values available through OOSH. Permission to manage the Audit configuration.
View Topology	Permission to view workers and worker groups
Manage Topology	Permission to enable/disable workers and configure worker groups
View Security Configuration	Permission to view the security configuration. This includes viewing internal users, roles, LDAP authentication, LWSSO, SAML, and the Security banner.
Manage Security Configuration	Permission to modify the security configuration. This includes configuring internal users, roles, LDAP authentication, LWSSO, SAML, and the Security banner.
View Audit	Permission to access the audit trail.
Manage Data Cleanup	Permission to use the data cleanup (purging) APIs.

Dashboard	
View Dashboard	Permission to view the Dashboard.

Role Details dialog box > Group Mapping tab

GUI item	Description
Add Group button	Displays a text box so that you can create a mapping from this role to a group.
Text box	In the text box, type the name of the group to which you want to map this role.
Delete ✕ **button**	Deletes the mapping in the adjacent text box.

Setting Up Security – LDAP Authentication

If you are using an LDAP authentication service, you can add the LDAP configuration to Central. Users will be able to log in to the system by providing their credentials from their organization. This change is dynamic—there is no need to restart Central for the change to be applied.

The LDAP groups that users belong to can be mapped to OO roles, so that the administrator can control which actions users are allowed to perform in the system. For more information about roles, see "Setting Up Security – Roles" on page 66

If your organization works with multiple LDAP servers, it is possible to configure Central to work with all of them. This includes LDAPs with different schemes and from different vendors. For example, you might have an Active Directory (Microsoft LDAP) implementation for one part of the organization and a Sun One (Oracle LDAP) implementation for another part.

If you set up multiple LDAP configurations on different domains, when users log in to OO, they will need to select from a drop-down list with the active domains. There will be no drop-down list of domains if one of the LDAP configurations is set as the default or if only internal users are configured.

Login Conventions

Users will be able to log in using the following conventions:

- domain\username
- username@domain

The domain name is required unless a user is logging in as an internal user or if they belong to the default LDAP.

It is also possible to log in using the **samAccountName**, **mail**, and **principalName** attributes.

> **Note:** For non-Active Directory configurations, it is also possible to log in with any unique attribute.

Default LDAP

If you flag an LDAP configuration as default, users who belong to this LDAP will be able to log in without having to select a domain.

In earlier versions of OO 10.x, users did not provide a domain when authenticating. The behavior was to go first to the LDAP and after that, to the internal users (if the user was not found). The "default LDAP" feature preserves this behavior for upgraders. When a default LDAP is set, an authentication attempt without a domain will go first to the default LDAP and then to the internal users. See Set the default LDAP authentication configuration.

> **Note:** If you have upgraded from an earlier version of OO 10.x, the LDAP that was configured in the previous version is set as the default.

> **Note:** If you do not have an LDAP service, you can set up internal users, who will log in using their OO user name and password. For more information about this option, see "Setting Up Security – Internal Users" on page 99.

It is also possible to run Central without the user authentication feature. All Central users will be identified as "anonymous" and the system will not require authentication. All Central users will have full administrative access to do anything in the system—triggering, content management, and system configuration.

> **Note:** The **LDAP** tab is only visible if you have been assigned a role with the **Manage Security Configuration** or **View Security Configuration** permission. You will only be able to edit the LDAP authentication if you have a role with the **Manage Security Configuration** permission.

> **Note:** For a comparison between configuring LDAP in OO 10.x and in previous versions, see the OO *Release Notes*.

What do you want to do?

Add an LDAP authentication configuration (not Active Directory)

This procedure is relevant for all LDAP authentication configurations, apart from Active Directory. If you want to create an Active Directory configuration, see Add an Active Directory LDAP authentication configuration.

1. Click the **System Configuration** button to display the System Configuration workspace.
2. Select **Security** > **LDAP**.
3. From the **Add** menu, select the type of LDAP configuration that you want to add.
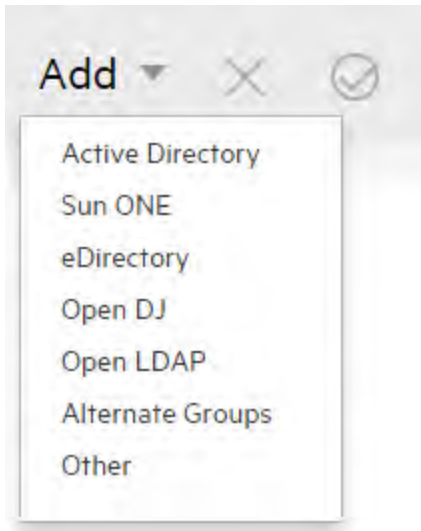
The **LDAP** tab displays the fields relevant to the selected LDAP configuration.
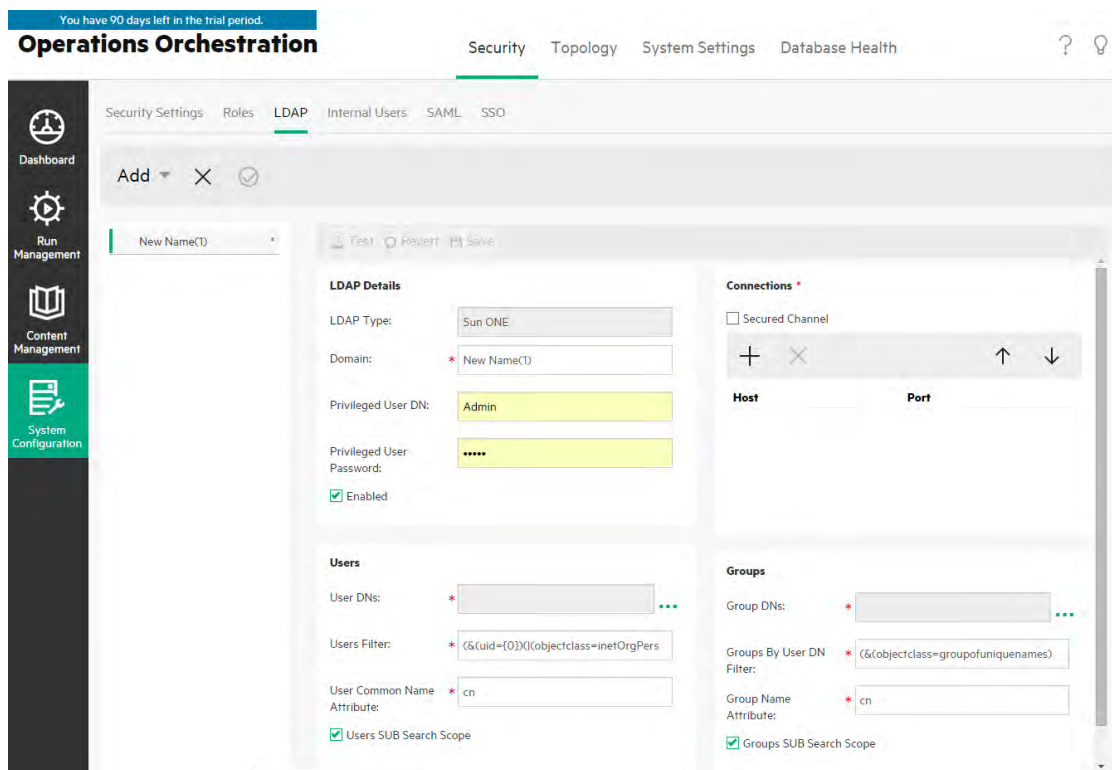
4. In the **Domain** box, enter a name for the domain.

   For example, if you have a single LDAP configuration, you might enter your company name. If you are setting up multiple LDAP configurations, you would give each one a domain that will identify it.

When users log in to OO, they will be able to select from a drop-down list of the active domains. They will be required to select a domain name unless they are logging in as an internal user or if they belong to the default LDAP.

The domain name appears in the LDAP navigation pane to the left.

5. In the **Privileged User DN** box, enter the distinguished name of a user with search privileges.

    It is recommended to provide a privileged user, because some OO functionality uses searching, for example, the scheduler. If there is no privileged user, this functionality may not work properly, depending on how the LDAP is configured.

    For example, `uid=john,ou=people,dc=devlab,dc=ad`

    > **Note:** Make sure to use the exact DN for the user with these privileges.

6. In the **Privileged User Password** box, enter the password of the user with search privileges.

7. Select the **Enabled** check box to enable the authentication of this LDAP.

8. In the **Connections** section, select **Secured Channel** to support secured transport layer connection.

9. In the **Hosts** box, enter the IP address or host name of the LDAP server.

10. In the **Ports** box, enter the port number of the LDAP server. This value must be between 0 and 64435.

11. If you have multiple LDAP servers, click the **Add** ╋ button to add a new line, and add the additional host and port.

12. Use the **Up** and **Down** ↑ ↓ arrows to move a connection up and down in the list. A connection that is higher in the list has a higher priority in the case of failover.

13. Click the browse button ⋯ next to the **User DNs** box, to open the User DNs dialog box, in order to define the distinguished names for users.

    ◦ If you know the user DN, click the **Add** ╋ button to add an empty row, and enter the user DN information manually. If you have multiple LDAP hosts, use ";" as the delimiter.

       For example, `ou=people,dc=devlab,dc=ad;ou=people,dc=devlab,dc=com`

    ◦ If you don't know the exact syntax, click **Add Base DN** to fetch a base DN from the LDAP.

       From the list of available root DNs, select the one that is closes to your needs and then refine

it.

User DNs   ? ✕

+   Add Base DN ▼   ✕   ↑   ↓

DC=   DC=qa,DC=ad,DC=com

CN=Configuration,DC=qa,DC=ad,DC=com

CN=Schema,CN=Configuration,DC=qa,DC=ad,DC=co

DC=DomainDnsZones,DC=qa,DC=ad,DC=com

DC=ForestDnsZones,DC=qa,DC=ad,DC=com

Cancel   OK

14. Use the **Up** and **Down** arrows ↑ ↓ to adjust the order of the user DN rows. The first item in the list is used first.

15. Click **OK** to close the User DNs dialog box and load the information into the **User DNs** box.

16. In the **Users Filter** box, enter the search filter parameters in order to filter users. This filter will be applied on each user DN that was entered in the **Users DNs** box.

    It is recommended to modify the default values, even if they work, because this filter is very general. It is highly recommended that the filter contain the `objectclass` attribute.

    > **Note:** In this filter, {0} is a placeholder for the user ID. When a user logs in, {0} is automatically replaced with the supplied user name.

17. In the **User Common Name Attribute** box, enter the attribute that defines how the user name will

be displayed on the screen.

> **Note:** In most LDAP implementations, `cn` is the default.

18. Select the **Users SUB Search Scope** check box if you want the users filter to run recursively.

> **Note:** If recursive searching is not required, then it is not recommended to select this check box.

19. Click the browse button [...] next to the **Group DNs** box, to open the Group DNs dialog box, in order to define the distinguished names for groups.

    This step is very similar to defining the distinguished names for users.

20. In the **Groups By User DN Filter** box, enter the search filter parameters in order to filter the groups. This filters the groups of which the given user is a member. The filter is applied on each Group DN.

    It is recommended to modify the default values even if they work, because this filter is very general. It is highly recommended that the filter contain the `objectclass` attribute.

21. In the **Group Name Attribute box**, enter the attribute that defines how the name of the group will be displayed in the UI.

> **Note:** In most LDAP implementations, `cn` is the default.

22. Select the **Groups SUB Search Scope** check box if you want the groups filter to run recursively.

> **Note:** If recursive searching is not required, then it is not recommended to select this check box.

23. Make sure to test the configuration before saving. For more information, see Testing the Configuration.

24. Click **Save** to save the configuration.

> **Note:** Make sure to save your changes before enabling authentication. If you don't save, the authentication will not include your changes!

    If the LDAP configuration is not saved, an asterisk appears next to the domain name in the LDAP navigation pane to the left. If you leave the page before saving, a message appears, warning that there are unsaved changes.

25. To enable the authentication functionality, select the **Settings** tab (under **Security**) and select the **Enable Authentication** check box to display the Enable Authentication dialog box. For more

information, see "Setting up Security Settings" on page 60.

> **Note:** The **Enable Authentication** check box is only available if there are existing internal or LDAP users with permission to disable the authentication in the future.
>
> If the **Enable Authentication** check box is not selected, there is no authentication and all users will be able to access all tasks.

Add an Active Directory LDAP authentication configuration

This procedure is relevant for Active Directory configurations only. If you want to create a different kind of LDAP configuration, see Add an LDAP authentication configuration (not Active Directory).

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **Security** > **LDAP**.

3. From the **Add** menu, select **Active Directory**.

   The **LDAP** tab displays the Active Directory view.

4. In the **Domain (NetBIOS name)** box, enter the AD short domain (NetBIOS name) that you want to authenticate against, for example, EMEA, US, or ASIA. When users log in to OO, they will be able to select from a drop-down list of the active domains.

   The domain name appears in the LDAP navigation pane to the left.

5. In the **Privileged User Name** box, enter the user name of a user with search privileges.

   It is recommended to provide a privileged user, because some Operations Orchestration functionality uses searching, for example, the scheduler. If there is no privileged user, this functionality may not work properly.

6. In the **Privileged User Password** box, enter the password of the user with search privileges.

7. Select the **Enabled** check box to enable the authentication of this LDAP.

8. In the **Connections** section, select **Secured Channel** to support secured transport layer connection.

9. In the **Hosts** box, enter the IP address or host name of the LDAP server.

10. In the **Ports** box, enter the port number of the LDAP server. This value must be between 0 and 64435.

11. If you have multiple LDAP servers, click the **Add** + button to add a new line, and add the additional host and port.

12. Use the **Up** and **Down** ↑ ↓ arrows to move a connection up and down in the list. A connection that is higher in the list has a higher priority in the case of failover.

13. Click the browse button ⋯ next to the **User DNs** box, to open the User DNs dialog box, in order to define the distinguished names for users.

    ○ If you know the user DN, click the **Add** + button to add an empty row, and enter the user DN information manually. If you have multiple LDAP hosts, use ";" as the delimiter.

      For example, `ou=people,dc=devlab,dc=ad;ou=people,dc=devlab,dc=com`

    ○ If you don't know the exact syntax, click **Add Base DN** to fetch a base DN from the LDAP.

      From the list of available root DNs, select the one that is closes to your needs and then refine

it.

User DNs   ? ✕

+   Add Base DN | ▼   ✕   ↑   ↓

DC=

> DC=qa,DC=ad,DC=com
>
> CN=Configuration,DC=qa,DC=ad,DC=com
>
> CN=Schema,CN=Configuration,DC=qa,DC=ad,DC=co
>
> DC=DomainDnsZones,DC=qa,DC=ad,DC=com
>
> DC=ForestDnsZones,DC=qa,DC=ad,DC=com

Cancel   OK

14. Use the **Up** and **Down** arrows ↑ ↓ to adjust the order of the user DN rows. The first item in the list is used first.

15. Click **OK** to close the User DNs dialog box and load the information into the **User DNs** box.

16. In the **Users Filter** box, enter the search filter parameters in order to filter users. This filter will be applied on each user DN that was entered in the **Users DNs** box.

    It is recommended to modify the default values, even if they work, because this filter is very general. It is highly recommended that the filter contain the `objectclass` attribute.

    > **Note:** In this filter, {0} is a placeholder for the `sAMAccountName` user identifier attribute. When a user logs in, {0} is automatically replaced with the supplied user name.

17. In the **User Common Name Attribute** box, enter the attribute that defines how the user name will

be displayed on the screen.

> **Note:** For example, `displayName`.

18. In the **Group Membership Attribute** box, enter the attribute that shows which groups the user belongs to.

    For example, `memberOf`

19. Select the **Users SUB Search Scope** check box if you want the users filter to run recursively.

    > **Note:** If recursive searching is not required, then it is not recommended to select this check box.

20. In the **Group Name Attribute box**, enter the attribute that defines how the name of the group will be displayed in the UI.

    > **Note:** In most LDAP implementations, `cn` is the default.

21. Make sure to test the configuration before saving. For more information, see Testing the Configuration.

22. Click **Save** to save the configuration.

    > **Note:** Make sure to save your changes before enabling authentication. If you don't save, the authentication will not include your changes!

    If the LDAP configuration is not saved, an asterisk appears next to the domain name in the LDAP navigation pane to the left. If you leave the page before saving, a message appears, warning that there are unsaved changes.

23. To enable the authentication functionality, select the **Settings** tab (under **Security**) and select the **Enable Authentication** check box to display the Enable Authentication dialog box. For more information, see "Setting up Security Settings" on page 60.

    > **Note:** The **Enable Authentication** check box is only available if there are existing internal or LDAP users with permission to disable the authentication in the future.

    > If the **Enable Authentication** check box is not selected, there is no authentication and all users will be able to access all tasks.

24. In the Enable Authentication dialog box, enter your administrator user name and password, and click **OK**.

Edit an LDAP authentication configuration

1. Select **Security** > **LDAP**.

2. Select the LDAP authentication configuration that you need to edit.

3. Enter the required changes.

4. Make sure to test the configuration before saving. For more information, see Testing the Configuration.

5. Click **Save**.

Configure a list of attributes whose values will be used as groups

It is possible to set up an LDAP authentication configuration with a list of attribute names that will be used as groups, as an alternative to group filters.

This enables organizations to base the functional groups of users on LDAP attributes, rather than LDAP groups.

1. Select **Security** > **LDAP**.

2. From the **Add** menu, select **Alternate Groups**.
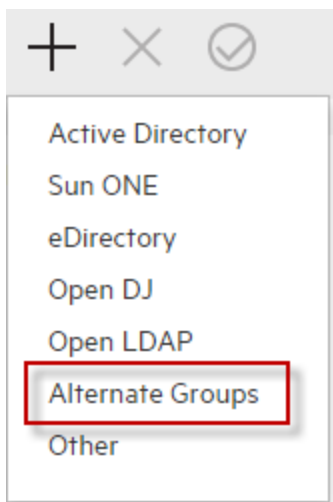
The **LDAP** tab displays the Alternate Groups view. This view is similar to the Non-Active Directory view, but it has an extra text box, **Alternate Group Attributes**, and does not include the **Groups** section.

3. In the **Alternate Group Attributes** box, enter the attribute names that will be used as groups. Use a semicolon (;) as the separator.

4. Enter the rest of the configuration details as described in Add an LDAP authentication configuration (not Active Directory), apart from the **Group** fields.

Set the default LDAP authentication configuration

If you set a default LDAP configuration, users who belong to this LDAP will not be required to select a domain, upon login.

Any authentication attempt without a domain will go first to the default LDAP and then to the internal users.

Note: The **Default** ⊘ button is only available if the selected LDAP configuration has been saved.

1. Select one of the LDAP authentication configurations from the navigation pane to the left.

2. Click the **Default** ⊘ button in the toolbar.

   The **Default** ⊘ icon appears next to the configuration name.
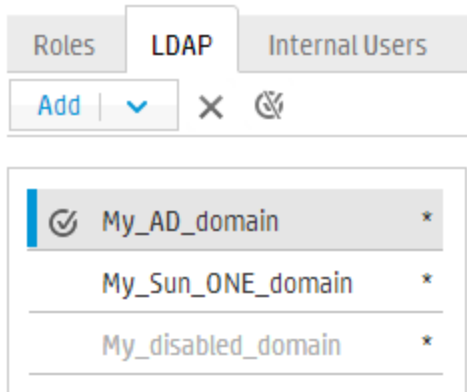
3. To stop this configuration from being the default, select the configuration name and then click the **Remove Default LDAP** button.

Test an LDAP authentication configuration

1. After entering the LDAP details, click the **Test** button.

2. In the Test LDAP Configuration dialog box, enter the user name and password, and click **Test**.

   A pop-up message displays the following information:

   ○ Whether the authentication is successful

   ○ Whether the configuration is valid

   ○ Groups to which the user belongs, if any were found

Test LDAP Configuration ⑦ ✕

User Name: * QU_Adv1

User Password: ••••••

✅ Authentication

Authentication succeeded

⚠ Configuration

The current LDAP requires privileged user credentials configured.
In the current configuration schedules might fail to run.
Verify you supply correct privileged user credentials.

✅ Groups

1. QG_Adv

Close   **Test**

3. Click **Close** to close the dialog box when the testing is complete.

Delete an LDAP authentication configuration

1. Select an LDAP configuration.

2. Click the **Delete** ✕ button in the toolbar.

3. Click **Yes** in the confirmation dialog box.

Disable an LDAP authentication configuration

You might want to temporarily disable an LDAP configuration, while saving the details for later.

1. Select an LDAP configuration.

2. Clear the **Enabled** check box in the **LDAP Details** section.

   The selected LDAP configuration name appears in gray text in the LDAP Navigation pane.
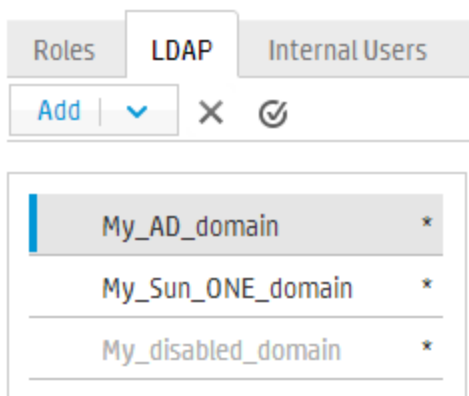
Tell Central whether to trust all valid certificates from secured LDAP servers

When this configuration item is enabled, Central will trust any valid certificate from a secured LDAP server. There will be no need for you to add it to the **client.truststore**.

When this is not enabled, you will need to add the certificates from the LDAP servers used for LDAP authentication to the **client.truststore**. Central will not trust them otherwise.

1. In the command line of the OOSH utility, type (for example):

   ```
   set-sys-config --key ldaps.trustAllCertificates --value true
   ```

2. Press the **Enter** key.

> **Note:** By default, the configuration item named **ldap.trustAllCertificates** is set to **false**. This means that OO will not trust all certificates, and you must configure the LDAP certificate in the Central truststore accordingly.

Reference Material

LDAP Navigation pane

GUI item	Description
Add button	Click to display the list of LDAP types that you can choose from.
Delete button	Click to delete the selected LDAP configuration.
Default button	Click to make the selected LDAP configuration the default. This means that an authentication attempt without a domain will go first to this LDAP and then to the internal users. If you do not set a default, an authentication attempt without a domain will go directly to the internal users. **Note:** The **Default** button is only available if the selected LDAP configuration has been saved.
Navigation pane	Displays the names of the LDAP configuration domain names. Click a domain name to edit the configuration.
Disabled	A domain name appears in gray text if the **Enabled** check box has been cleared for that LDAP configuration.

Security > LDAP>Non-Active Directory configuration

Mandatory fields are indicated with a red asterisk. *

GUI item	Description
Test button 🧪	Click to test the configuration. A pop-up message displays the following information: • Whether the authentication is successful • Whether the configuration is valid • Groups to which the user belongs, if any were found
Revert button ↻	Click to revert the configurations to the last saved configurations on the server.
Save button 💾	Click to save the selected LDAP configuration.
LDAP details section	
LDAP Type	Displays the LDAP type that you selected from the **Add** menu.
Domain	Enter a name for the domain. This is name will be displayed when users log in to OO. For example, if you have a single LDAP configuration, you might enter your company name. If you are setting up multiple LDAP configurations, you would give each one a domain that will identify it.

Privileged User DN	Enter the distinguished name of a user with search privileges.
	For example, `uid=john,ou=people,dc=devlab,dc=ad`
	It is recommended to provide a privileged user, because some OO functionality uses searching, for example, the scheduler. If there is no privileged user, this functionality may not work properly.
	Note: Make sure to use the exact DN for the user with these privileges.
Privileged User Password	Enter the password of the user with search privileges.
Enabled	Click to enable the authentication of this LDAP
Connections section	
Secured Channel	Select the **Secured Channel** check box to support secured transport layer connection.
Add button	Click to add a new connection.
Delete button	Click to delete the selected connection.
↑ ↓	Use the arrows to move a connection up and down in the list. A connection that is higher in the list has a higher priority in the case of failover
Host	Enter the IP address or host name of the LDAP server.
	For example, `16.55.222.71`
Port	Enter the port number of the LDAP server. This value must be between 0 and 64435.
	For example, `389`
Users section	
Users DNs	Define the distinguished names to use for the users. Click the browse button [...] to open the User DNs dialog box. See User DNs/Group DNs dialog box.
Users Filter	Enter the search filter parameters in order to filter users. This filter will be applied on each user DN that was entered in the **User DNs** box.
	It is recommended to modify the default values, even if they work, because this filter is very general. It is highly recommended that the filter contain the `objectclass` attribute.
	Note: In this filter, {0} is a placeholder for the user ID. When a user

	logs in, {0} is automatically replaced with the supplied user name.
User Common Name Attribute	Enter the attribute that defines how the user name will be displayed on the screen. **Note:** In most LDAP implementations, cn is the default.
Users SUB Search Scope check box	Select this check box if you want the users filter to run recursively. **Note:** If recursive searching is not required, then it is not recommended to select this check box.
Groups section	
Groups DNs	Define the distinguished names to use for the groups. Click the browse button ⋯ to open the Group DNs dialog box. See User DNs/Group DNs dialog box.
Groups By User DN Filter	Enter the search filter parameters in order to filter the groups. This filters the groups of which the given user is a member. The filter is applied on each Group DN. It is recommended to modify the default values even if they work, because this filter is very general. It is highly recommended that the filter contain the objectclass attribute.
Group Name Attribute	Enter the attribute that defines how the name of the group will be displayed in the UI **Note:** In most LDAP implementations, cn is the default.
Groups SUB Search Scope check box	Select this check box if you want the groups filter to run recursively. **Note:** If recursive searching is not required, then it is not recommended to select this check box.

Security > LDAP> Active Directory configuration

Mandatory fields are indicated with a red asterisk. *

Use

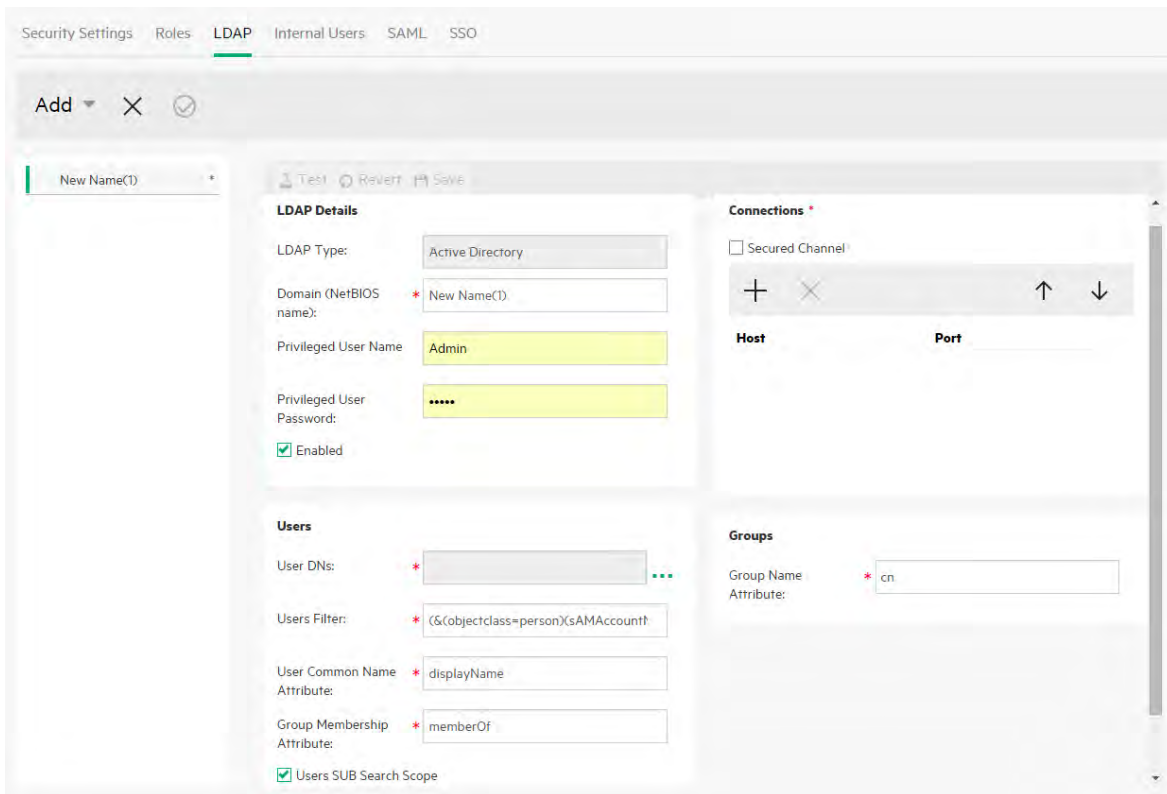| GUI item	Description
Test button 🧪	Click to test the configuration. A pop-up message displays the following information: • Whether the authentication is successful • Whether the configuration is valid • Groups to which the user belongs, if any were found
Revert button 🔄	Click to revert the configurations to the last saved configurations on the server.
Save button 💾	Click to save the selected LDAP configuration.
LDAP details section	
LDAP Type	Displays the LDAP type that you selected from the **Add** menu.
Domain (NetBIOS name)	Enter the name of the domain (NetBIOS name) that you want to authenticate against. The domain name appears in the navigation pane to the left.
Privileged User Name	Enter the user name of a user with search privileges.

	It is recommended to provide a privileged user, because some OO functionality uses searching, for example, the scheduler. If there is no privileged user, this functionality may not work properly.
Privileged User Password	Enter the password of the user with search privileges.
Enabled	Click to enable the authentication of this LDAP
Connections section	
Secured Channel	Select the **Secured Channel** check box to support secured transport layer connection.
Add button	Click to add a new connection.
Delete button	Click to delete the selected connection.
↑ ↓	Use the arrows to move a connection up and down in the list. A connection that is higher in the list has a higher priority in the case of failover
Host	Enter the IP address or host name of the LDAP server. For example, `16.55.222.71`
Port	Enter the port number of the LDAP server. This value must be between 0 and 64435. For example, `389`
Users section	
Users DNs	Define the distinguished names to use for the users. Click the browse button ⋯ to open the User DNs dialog box. See User DNs/Group DNs dialog box.
Users Filter	Enter the search filter parameters in order to filter users. This filter will be applied on each user DN that was entered in the **Users DNs** box. It is recommended to modify the default values, even if they work, because this filter is very general. It is highly recommended that the filter contain the `objectclass` attribute. **Note:** In this filter, `{0}` is a placeholder for the `sAMAccountName` user identifier attribute. When a user logs in, `{0}` is automatically replaced with the supplied user name.
User Common Name Attribute	Enter the attribute that defines how the user name will be displayed on the screen.
Group Membership	Enter the attribute that shows which groups the user belongs to.

Attribute	For example, `memberOf`
Users SUB Search Scope check box	Select this check box if you want the users filter to run recursively. **Note:** If recursive searching is not required, then it is not recommended to select this check box.
Groups section	
Group Name Attribute	Enter the attribute that contains the name of the group to be displayed in the UI. **Note:** In most LDAP implementations, `cn` is the default.

Security > LDAP > Alternate Groups configuration

The **Alternate Groups** view is similar to the other non-Active Directory views. However, it does not contain a **Groups** section, and it contains the **Alternate Group Attributes** box.

This enables you to set up an LDAP authentication configuration with a list of attribute names to be used as groups, instead of group filters.

GUI item	Description
Alternate Group Attributes	Enter the attribute names that will be used as groups. Use a semicolon (;) as the separator.

User DNs/Group DNs dialog box
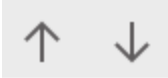
Click the browse button [...] next to the **User DNs** box, to open the User DNs dialog box.

Click the browse button [...] next to the **Group DNs** box, to open the Group DNs dialog box.

The fields in the two dialog boxes are the same.

User DNs ? ✕

\+   Add Base DN   ▼   ✕   ↑   ↓

DC=
    DC=qa,DC=ad,DC=com

    CN=Configuration,DC=qa,DC=ad,DC=com

    CN=Schema,CN=Configuration,DC=qa,DC=ad,DC=co

    DC=DomainDnsZones,DC=qa,DC=ad,DC=com

    DC=ForestDnsZones,DC=qa,DC=ad,DC=com

Cancel   OK
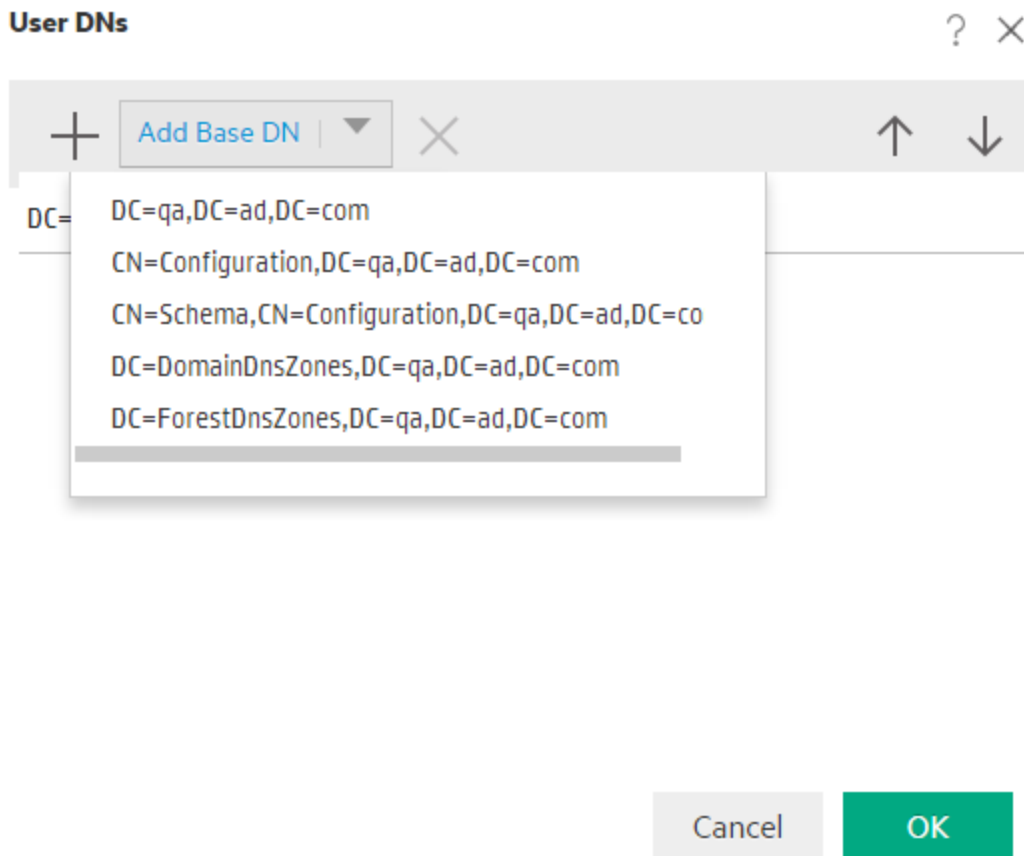
GUI item	Description
Add Base DN	Click to select from the list of available root DNs that are fetched from the LDAP. Select the one that is closes to your needs and then refine it.
Add +	Click to add a new, empty row, in order to manually enter the DN information. If you have multiple LDAP hosts, use ";" as the delimiter. For example, `ou=people,dc=devlab,dc=ad;ou=people,dc=devlab,dc=com`

Delete ✕	Click to delete the selected DN row.
↑　↓	Use the arrows to adjust the order of the DN rows. The first item in the list is used first.

Troubleshooting

- There is a limit to how many groups a user can have. Users who are members of more than 1,015 groups may fail logon authentication.

  For more information, see http://support.microsoft.com/kb/328889
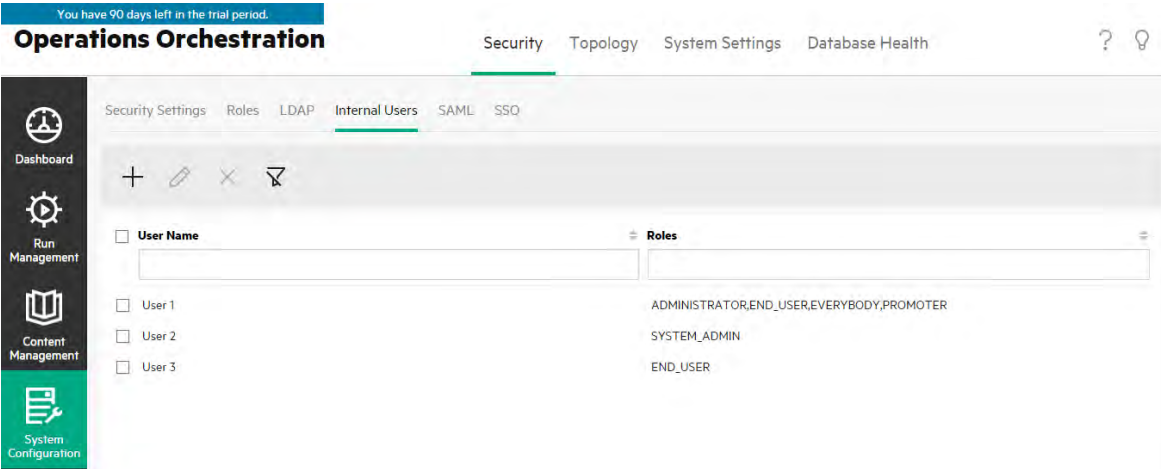
Setting Up Security – Internal Users

As an alternative to using an external authentication system, such as LDAP, it is possible to set up internal users in Central. Internal users will log in using their OO user name and password, and they will not require any external authentication.

By assigning roles to users, the administrator controls which actions users are allowed to perform in the system. For more information about roles, see "Setting Up Security – Roles" on page 66.

> **Note:** The **Internal Users** tab is only visible if you have been assigned a role with the **Manage Security Configuration** or **View Security Configuration** permission. You will only be able to edit the users if you have been assigned a role with the **Manage Security Configuration** permission.

If a user has been assigned the same user name for both LDAP user and internal user, Central will relate to the LDAP user.

What do you want to do?

Add a user

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **Security** > **Internal Users**.

3. Click the **Add** ✛ button.

4. Enter the details for the internal user:

Internal User Details   ? ✕

| User Name | * | User 1 |

| Password | •••••••••• |

| Confirm Password | •••••••••• |

Roles

☑ ADMINISTRATOR   ☑ PROMOTER
☑ END_USER   ☐ SYSTEM_ADMIN
☑ EVERYBODY

Cancel   **Save**

    a. In the **User Name** box, enter a unique user name. The user name cannot have more than 255 characters.

    b. Enter the password in both the **Password** and the **Confirm Password** boxes. The password cannot have more than 255 or less than 6 characters.

    c. Select the check box next to each role that you want to assign to this user. You can assign multiple roles to a user.

> **Note:** If you assign multiple roles to a user, the user will be able to access all the permissions in those assigned roles. A role with fewer permissions will not restrict this user's access to the permissions in their other roles.

5. Click **Save** to save the new user.

6. To enable the authentication functionality, select the **Settings** tab (under **Security**) and select the **Enable Authentication** check box to display the Enable Authentication dialog box. For more information, see "Setting up Security Settings" on page 60.

> **Note:** The **Enable Authentication** check box is only available if there are existing internal or LDAP users with permission to disable the authentication in the future.
>
> If the **Enable Authentication** check box is not selected, there is no authentication and all users will be able to access all tasks.

Edit a user account

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **Security** > **Internal Users**.

3. Select the check box next to the user that you want to edit and click the **Edit** ✎ button.

> **Note:** The **Edit** button is only enabled if a single user is selected.

4. Make the required changes in the User Configuration dialog box.

5. Click **Save**.

Delete a user

1. Select **Security** > **Internal Users**.

2. Select one or more users that you want to delete and click the **Delete** ✕ button.

> **Note:** The **Delete** button is only enabled if at least one user is selected.

3. Click **OK** in the confirmation dialog box.

> **Note:** A user cannot delete their own account.

Reference Material

Security > Internal Users
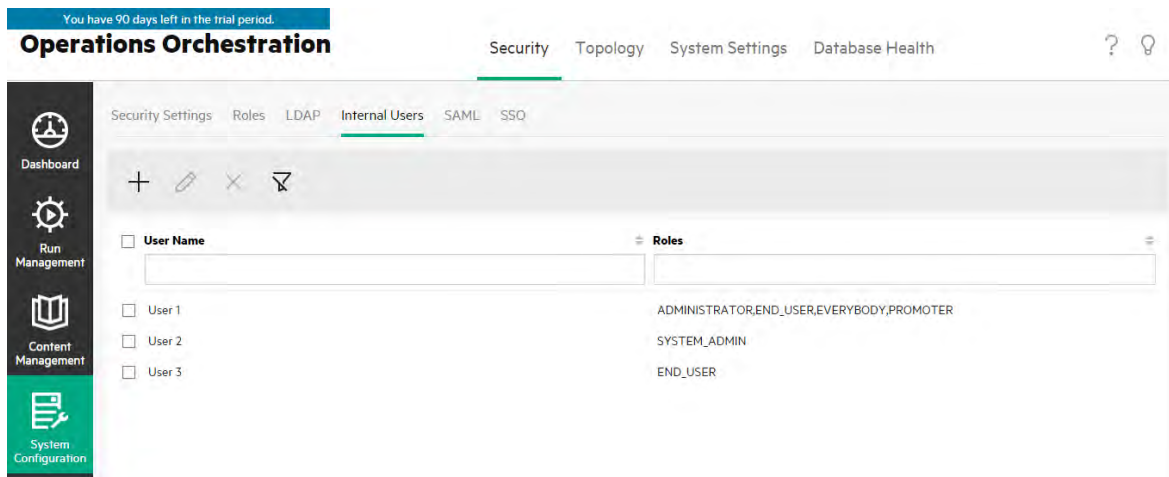
GUI item	Description
Add button ＋	Opens the Internal Users Configuration dialog box, so that you can add a new user.
Edit button ✎	Opens the Internal Users Configuration dialog box, so that you can edit the selected user.
Delete button ✕	Deletes the selected user.
User Name	Displays the user name.
Roles	Displays the roles that have been assigned to the user.
Filter boxes	In the filter boxes at the top of the columns, enter text to filter the displayed users.

Internal User Details dialog box

GUI item	Description
User Name	Enter a unique user name of up to 255 characters.
Password	Enter a password of 6 to 255 characters.
Confirm Password	Enter the password a second time for verification.
Roles	Select the check box next to each role that you want to assign to this user. You can assign multiple roles to a user.
Save	Click to save the user.

Setting Up Security – SAML Authentication

Administrators can configure Central to work with SAML 2.0 authentication. SAML (Security Assertion Markup Language) is an XML-based open standard data format for exchanging authentication and authorization data between an identity provider and a service provider. The process of authenticating and authorizing users with the identity provider involves the use of a SAML artifact and SAML assertions.

A **SAML assertion** is an XML document that includes the attributes that define the user and the user's group. The identity provider sends these attributes to the service provider, and based on these attributes, the service provider provides access rights to the user.

A **SAML artifact** is a sequence of bytes that encodes two pieces of information:

- The artifact ID for the SAML assertion, which is retrievable from the identity provider

- A SOAP endpoint, which can be used to resolve this handle.

The SAML artifact string should be passed as a query parameter over the request URL in Base64 encoding format.

For example:

```
http(s)://<FQDN>:<PORT>/saml/SSO/alias/defaultAlias?SAMLart=ARTIFACTEncodedString
```

Note: The **SAML** tab is only visible if you have been assigned a role with the **Manage Security Configuration** or **View Security Configuration** permission. You will only be able to edit the SAML authentication if you have a role with the **Manage Security Configuration** permission.

Tip: It is recommended to keep the admin user, so that you will be able to log in directly, if there is an error in your SAML configuration.

Note: You must use an **internal/ldap** user when using the OO Shell Utility (OOSH), the Studio remote debugger, when creating a flow schedule in Central, and when using RESTful APIs or backwards compatible SOAP and RESTful APIs from OO 9.x.

Known Issues and Troubleshooting

- If you get locked out, with no access via the identity provider, log into OO directly. Make sure to keep the admin user, for this. To log in directly, go to **http(s)/<host>:<port>/oo/login/direct**.

- The Central Scheduler does not work when SAML authentication is used.

Note: If you are using LDAP, you must set a default LDAP configuration. See "Set the default

LDAP authentication configuration" in "Setting Up Security – LDAP Authentication" on page 76.

What do you want to do?

Prerequisites

Before setting up SAML authentication, complete the following steps:

1. Configure OO for SSL.

2. If you have changed the **key.store** default password, you need to set it as a system configuration item.

   For instructions, see Change the default values of the SAML system configuration. For reference information, see SAML system configuration.

   > **Note:** This step is not needed if you use the default password.

3. If you have changed the server certificate (private key) default password and the alias default value of this certificate in the **key.store**, you need to set them as system configuration items.

   For instructions, see Change the default values of the SAML system configuration. For reference information, see SAML system configuration.

   > **Note:** This step is not needed if you use the default values.

4. Import the public keys of the identity provider's certificate into the Central **key.store**.

   For example:

   ```
   keytool -importcert -alias <any_alias> -keystore key.store -file <certificate_
   name.cer> –storepass <changeit>
   ```

5. Restart the server.

6. Open the Central user interface.

   > **Important!** Make sure to use the fully qualified domain name in the URL.

7. Create the administrator user and administrator role, and any other roles that are needed.

8. Map the roles to the required groups. For more information about creating roles and mapping them to groups, see "Setting Up Security – Roles" on page 66.

Change the default values of the SAML system configuration

If you have changed the **key.store** default password, you need to set it as a system configuration item.

If you have changed the server certificate (private key) default password and the alias default value of this certificate in the **key.store**, you need to set them as system configuration items.

> **Note:** These steps are not needed if you use the default values.

1. Connect to the OO Shell utility (OOSH).

   For more information about theOperations Orchestration Shell utility, see the Operations Orchestration *Shell User Guide*

2. If you want to change the password of the **key.store** or the Operations Orchestration server certificate (that was imported into the **key.store**), you need to encrypt the password with the encryption tool, located at **<install dir>/central/bin/encrypt-password**:

   ```
   encrypt-password --encrypt --password <your password>
   ```

   ```
   C:\Program Files\Hewlett-Packard\HP Operations Orchestration_saml_oracle\central\bin>encrypt-password.bat --encrypt --password mo
   {ENCRYPTED}a/MYfi/Op2Rh3IRO5aDYmQ==
   ```

3. Run `set-sys-config --key <key> --value <value>`

   For example:

   ```
   set-sys-config --key key.store.private.key.alias.name --value newValue
   ```

   > **Note:** You can see a list of all system configurations by running the '`lsc`' command.

See SAML system configuration in the **Reference** section.

Add a SAML authentication configuration

Before starting this procedure, read the Prerequisites section, and complete any necessary steps.

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **Security** > **SAML**.

3. In the **Service Provider** section, enter the entity ID of the service provider (OO). The default is **ooentityid**.

   In the **Service Provider** section, enter the entity ID of the service provider (OO). The default is **ooentityid**.

4. In the **Identity Provider** section, enter the details of the identity provider:

   ○ **IDP metadata URL**: Enter the SAML metadata for the identity provider with which you will interact.

   ○ **User name attribute**: Enter the SAML assertion attribute name that defines the user.

- **Group names attribute**: Enter the SAML assertion attribute name that defines the user groups.

- **Group names delimiter**: Enter the delimiter used for the groups in the SAML assertion group names attribute.

  > **Note:** Make sure that the attributes are correct; otherwise, access will be denied.

5. If OO is installed behind a proxy, define the forward (web) proxy host and port number.

6. Click **Enable** to enable SAML.

7. Click **Save**.

8. Click the **Download Metadata** button to save the metadata file.

   > **Note:** This button is enabled after you enable and save the SAML configuration.

9. Provide the OO public key to the identity provider.

10. Provide the OO public key to the identity provider.

11. Provide the metadata file to the identity provider.

12. To enable the authentication functionality, select the **Settings** tab (under **Security**) and select the **Enable Authentication** check box to display the Enable Authentication dialog box. For more information, see "Setting up Security Settings" on page 60.

    > **Note:** The **Enable Authentication** check box is only available if there are existing internal or LDAP users with permission to disable the authentication in the future.
    >
    > If the **Enable Authentication** check box is not selected, there is no authentication and all users will be able to access all tasks.

Add SAML authentication configuration to a load balancer

When setting up a load balancer, you need to resend the service provider metadata to the identity provider.

Before starting this procedure, read the Prerequisites section, and complete any necessary steps.

1. Complete all the steps in the Add a SAML authentication configuration section.

2. Set the external URL of the load balancer (under the **Topology** tab) and save. For more information, see "Setting Up Topology – Workers and RASes" on page 114.

3. Click the **Download Metadata** button to get the new metadata with the external URL details.

4. Provide the metadata file to the identity provider.

Change the SAML log level

You can change the SAML log level in the **log4j.properties** file.

1. Open the **log4j.properties** file (under **/<oo-installation>/central/conf/log4j.properties**).

2. Locate the line that refers to the SAML logging level. For example:

   `saml.log.level=INFO`

3. Change the logging level to `DEBUG` or `ERROR/WARNING` as required.

For more information, see "Adjusting the Logging Levels" in the *OO Admistration Guide*.

For more information, see "Adjusting the Logging Levels" in the *OO Admistration Guide*.

Reference Material

Security > SAML

GUI item	Description
Enable check box	Select to enable the SAML authentication
Entity ID	Enter the identity ID of the service provider (OO). The default is **ooentityid**.
IDP metadata URL	Enter the SAML metadata for the identity provider with which you will interact.

User name attribute	Enter the SAML assertion attribute name that defines the user.
Group names attribute	Enter the SAML assertion attribute name that defines the user groups.
Group names delimiter	Enter the delimiter used for the groups in the SAML assertion group names attribute.
Host	If OO is installed behind a proxy, define the forward (web) proxy host.
Port	If OO is installed behind a proxy, define the forward (web) proxy port number.
Download metadata	Click to save the metadata file, which then needs to be sent to the identity provider. This button is enabled after you enable and save the SAML configuration.

SAML system configuration

If you update the following configuration items using the OO Shell utility (OOSH), you will need to restart the server (security context). For more information about the OO Shell utility, see the OO *Shell User Guide*.

Configuration item	Action
`key.store.password`	To set the password used to access to the **key.store**. The default value is 'changeit'.
`key.store.private.key.alias.name`	To set the alias used to for the server certificate (private key) in the **key.store**. The default value is 'tomcat'.
`key.store.private.key.alias.password`	To set the password used to access the server certificate (private key) from the **key.store**. The default value is 'changeit'.

Setting Up Security – LWSSO

You can set up single sign-on (SSO) for OO through Lightweight SSO (LWSSO). LWSSO is an solution that enables single sign-on using one authentication across various applications.

LWSSO shares a cookie between products that are accessed from a web browser. As a result, if a user logs onto another product web client that has LWSSO enabled, such as the SM web client or the BSM web client, this user will be able to enter the OO Central application directly, bypassing the OO Central logon screen.

Prerequisites for this configuration:

- The LWSSO feature must be enabled on all the products between which you want to use single sign-on.

- The user's logon credentials in the other product must match those of an OOuser account.

  For example, if BSM is integrated with OO, the user that is logged into BSM must also exist inOO (either an LDAP user or an internal user).

> **Note:** Even if OO is set up with multiple LDAP configurations, it is only possible to authenticate the user using LWSSO with the default LDAP. LWSSO will first try to authenticate the user with the default LDAP, and if this fails, will try to authenticate within the OO internal domain.

After you save an LWSSO configuration, the changes take effect immediately, and there is no need to restart the server.

All nodes in the cluster configuration will refresh their configuration from the database periodically at a 10 minute interval. In the case of a single server, the node will update its configuration from the database at the same 10 minute interval.

When connecting another application that employs LWSSO to OO, you must specify the connection URL of OO using the following format:

`<protocol>://<FQDN>:<portNumber>`

For example, `http://lab.lab:8080`

> **Note:** The LWSSO enabling procedure may be different for other products. See the corresponding documentation for each product.
>
> For example, for some applications, the connection URL needs to be in the format `<protocol>://<FQDN>:<portNumber>/oo`.

> **Note:** The **SSO** tab is only visible if you have been assigned a role with the **Manage Security**

> **Configuration** or **View Security Configuration** permission. You will only be able to edit the LWSSO authentication if you have a role with the **Manage Security Configuration** permission.

When Central and a RAS are installed on the same machine, and the LWSSO settings are enabled, you must set the management URL property using the fully qualified domain name, in the **ras/conf/ras-wrapper.conf** file. For more information, see "Configuring LWSSO Settings" in the *OO Administration Guide*.

> **Note:** If you are using LDAP, you must set a default LDAP configuration. See "Set the default LDAP authentication configuration" in "Setting Up Security – LDAP Authentication" on page 76.

What do you want to do?

Set up the LWSSO authentication configuration

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **Security** > **SSO**.

3. In the **initString** box, enter the password you want to use to connect products. This value must be the same as those used in the LWSSO configuration of the other products. This password must contain at least 12 characters.

   The initstring is used to encrypt the LWSSO cookie and has no link to the user's password. However, when two applications are integrated using LWSSO, the initstring in both applications must match. Otherwise, the cookie encrypted by one application cannot be decrypted by the other application.

4. In the **Domain** box, enter the domain of the OO server.

5. If there are multiple domains used by applications that employ LWSSO, enter them in the **Protected Domains** box, using commas as separators.

   The OO server domain must appear in the **Protected Domains** list. When there are several domains in the **Protected Domains** box, all applications that are integrated through LWSSO must define the same list of protected domains.

6. Select the **Enable** check box to enable the LWSSO configuration. If you do this, after you save the settings, the icon next to the **LW SSO Settings** title will change to **Enabled** ⊘ .

   > **Note:** The **Enable** check box is there to let you disable an LWSSO configuration while keeping the general authentication feature enabled. For example, you might have authentication enabled for internal users and LDAP users, but you may or may not want the LWSSO configuration to be enabled.
   >
   > Note that the new state of LWSSO settings is not enabled immediately when you select the **Enable** check box, but when you save the new configuration.

7. Click **Save**.

8. To enable the authentication functionality, select the **Settings** tab (under **Security**) and select the **Enable Authentication** check box to display the Enable Authentication dialog box. For more information, see "Setting up Security Settings" on page 60.

   > **Note:** The **Enable Authentication** check box is only available if there are existing internal or LDAP users with permission to disable the authentication in the future.
   >
   > If the **Enable Authentication** check box is not selected, there is no authentication and all users will be able to access all tasks.

Edit an existing LWSSO authentication configuration

1. Select **Security** > **SSO**.

2. Enter the required changes.

3. Click **Save**.

Reference Material

Security > SSO

GUI item	Description
⊘	The icons next to the **LWSSO Settings** title indicate whether the settings are enabled or disabled.
Enable	Select the check box to enable LWSSO.
initString	The initstring is used to encrypt the LW SSO cookie and has no link to the user's password. However, when two applications are integrated using LWSSO, the initstring in both applications must match. Otherwise, the cookie encrypted by one application cannot be decrypted by the other application.
Domain	Enter the domain of the OO server.
Protected Domains	If there are multiple domains used by applications that employ LWSSO, enter them in the **Protected Domains** box, using commas as separators. The OO server domain must appear in the **Protected Domains** list. When there are several domains in the **Protected Domains** box, all applications that are integrated through LWSSO must define the same list of protected domains.
Save	Click to save the LWSSO authentication configuration.

Setting Up Topology – Workers and RASes

Network topology is the arrangement of the various elements (links, nodes, and so on) in a network. Workers and RASes are part of the topology and can be configured under the **Topology** tab.

The **Workers** tab displays information about the following items:

- **Central worker** - Central workers are responsible for executing flows. A Central worker obtains tasks (flow execution messages) to process from Central.

  A Central worker can be assigned to a worker group. This enables workers to withstand a high action execution load and increases availability of workers in a data center. See Assigning Workers to Groups.

- **Standard RAS** - a RAS is a remote action server, containing a worker and a remote protocol for connecting with Central. A standard RAS will initiate the connection to Central.

  After a standard RAS is installed, it will appear in the grid in the **Workers** tab.

- **Reverse RAS** - a reverse RAS will wait for Central to initiate the connection. To define a reverse RAS, you must:

  a. Install the reverse RAS during the installation procedure. This creates the **ras-connectivity.properties** file containing the RAS configuration details. See "Configure Central to register the Reverse RAS" in the *Installation, Upgrade, and Configuration Guide*.

  b. Then, register the reverse RAS in the **Workers** tab. See Registering a Reverse RAS.

  Note: One of the Centrals is selected to establish connection to the RAS. If this Central is shut down, all RASes that this Central was responsible to establish connection to, will be reassigned to other Central nodes in the cluster equally.

  While attempting to connect to a reverse RAS:

  - If the RAS does not answer after 30 seconds, a timeout will occur.

  - If there are several consecutive failed connection attempts (because the wrong shared secret was entered), this will result in a delay. If this occurs, check that you are using the correct shared secret.

Note: The **Topology** tab is only visible if you have been assigned a role with the **Manage Topology** or **View Topology** permission. You will only be able to edit the topology if you have a role with the **Manage Topology** permission.
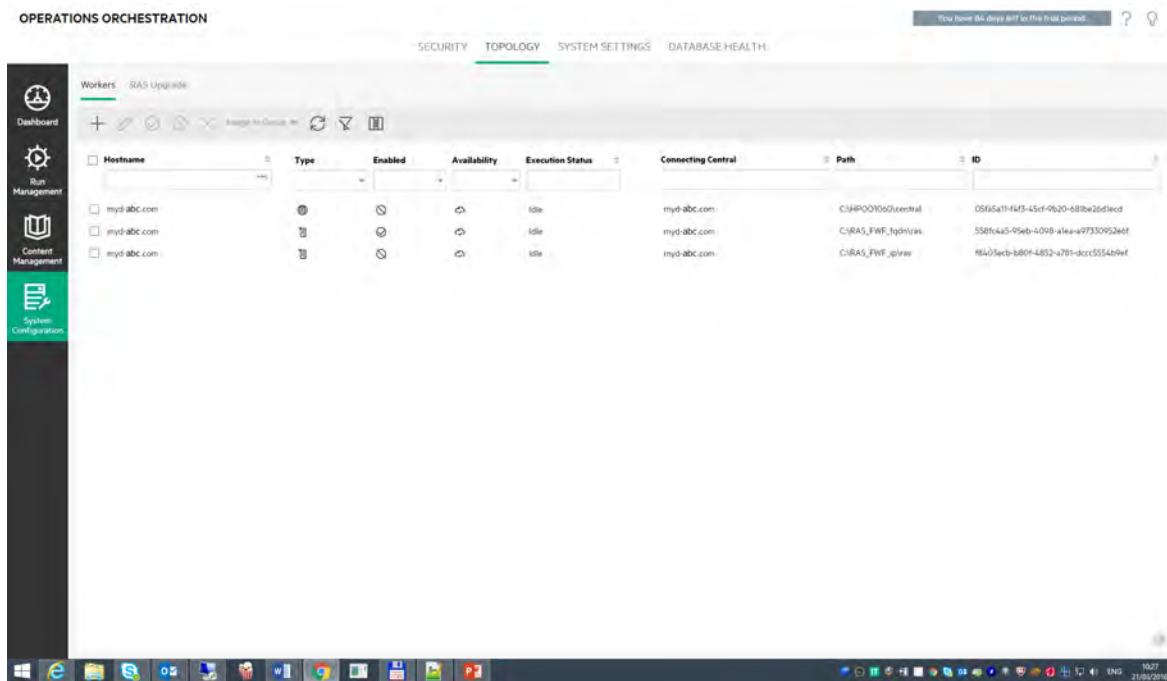
The **Topology** > **Workers** tab displays information about each worker/RAS: their status (**Running** or **Stopped**), installation path, connection details, operating system, ID, and so on.

> **Note:** The status can take a minute or two to update. If you stop a RAS service, it takes a few minutes until this is reflected in the **Topology** > **Workers** tab. There is no need to restart Central.

Execution Status

The **Topology** > **Workers** tab displays information about the resources that are being consumed by each worker/RAS, under **Execution Status**. This information may help with troubleshooting the system, and may contribute to decisions about scaling and resource allocation.

The information about resource consumption is automatically refreshed periodically. Take into account that the displayed information is only relevant to the last few minutes, since the previous refresh, and is not cumulative. In order to properly understand what is happening in your environment, it is recommended to observe the statuses over a period of time.

A worker/RAS can have one of the following execution statuses:

- **Idle**: This worker has almost no tasks to process.

- **Low Utilization**: This worker is working at low utilization.

  If your workers are with idle/low utilization over a long period of time, you might consider removing them.

- **Working**: This worker is working with a regular load.

- **Pending on Central**: Most of the time, this worker is waiting to send execution results to Central. Central is busy with handling other tasks.

  If this state is displayed over a long period of time, you might consider adding more Centrals, or monitoring your database for space, connectivity, and so on, to make sure it is not a bottle neck.

- **Loaded**: At the current time, this worker works at maximum ability and is not able to get more tasks from Central. If all your workers are loaded most of time, consider adding more Centrals or RASes.

- **N/A**: No information is provided for this worker.

Registering a Reverse RAS

While installing a RAS, you chose whether the RAS will initiate the connection to Central (Standard RAS) or wait for Central to initiate the connection (Reverse RAS)

If a RAS was configured to accept connection from Central (Reverse RAS), you must configure Central to register the RAS by providing all required information: host/IP, port, and so on. Until you do this, the RAS will be idle, waiting for Central to initiate the connection.

> **Note:** If you configured the RAS to initiate connection with Central (Standard RAS), then there is no need to register the RAS.

> **Note:** You cannot connect a reverse RAS to more than one Central.

Assigning Central Workers to Groups

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

> **Note:** A worker may belong to more than one group simultaneously.

Under the **Topology** > **Workers** tab, you can assign workers to worker groups, remove workers from groups, enable, disable, or delete workers.

> **Note:** Some execution stages always run on the default group, which is **RAS_Operator_Path**. Therefore, make sure that at least one worker is assigned to this group.

It is recommended to assign most (or even all) workers to **RAS_Operator_Path**, in order to optimize the execution time. You should only exclude a RAS from this group if you want to minimize its usage, and if you have only few operations that use it.
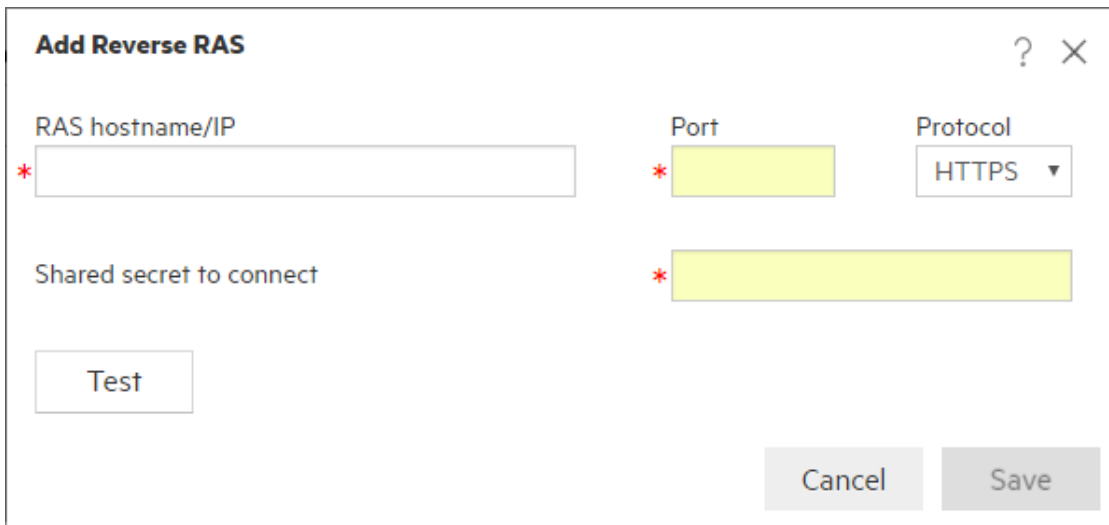
For more information about workers and worker groups, see "Worker Groups and Group Aliases" in the *OO Concepts Guide*.

What do you want to do?

Register a new reverse RAS

For each reverse RAS that you installed (those that were configured to accept connection from Central), you must configure Central to register the RAS. Until you do this, the RAS will be idle, waiting for Central to initiate the connection.

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **Topology** > **Workers**.

3. Click ➕ **Add Reverse RAS**.

4. In the dialog box, enter the RAS connection details: host name or IP, port, and protocol.

5. Enter the shared secret, which was entered when the reverse RAS was installed.

6. Click **Test**.

    If you encounter an error, adjust the details and try again.

7. Click **Save**.
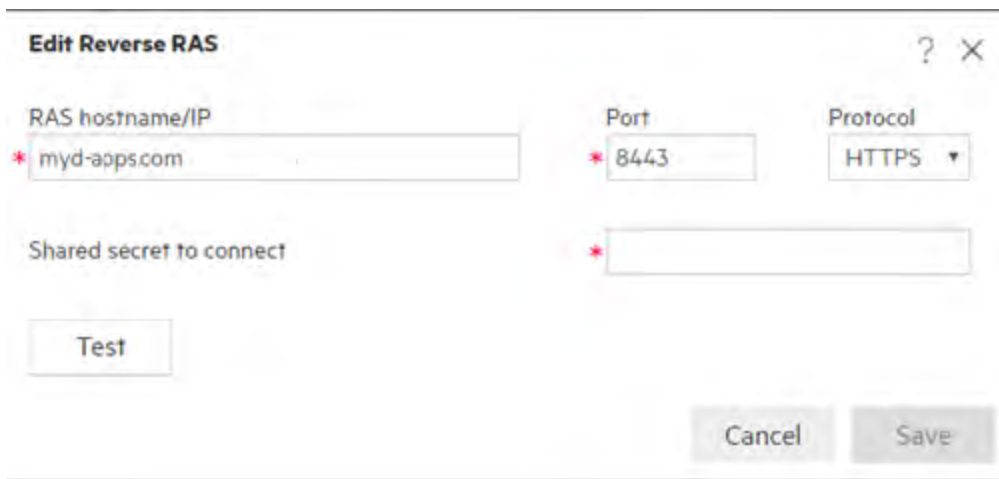
Central initializes the connection and the connection remains active. If the connection becomes disconnected, Central will try to reconnect at specified intervals.

The new reverse RAS appears in the table below.

Move a reverse RAS to a different port or IP address

1. Select **Topology** > **Workers**.

2. Select the reverse RAS you want to move.

3. Click [pencil icon] .

4. In the dialog box, change the reverse RAS connection details: host name or IP, port, and protocol.
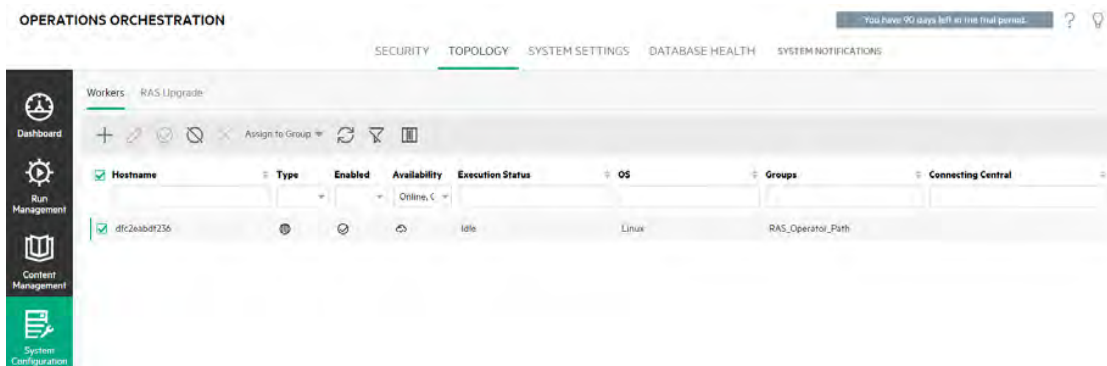
5. Click **Test**.

   If you encounter an error, adjust the details and try again.

6. Click **Save**.

The new reverse RAS appears in the table below.

Assign a Central worker to an existing worker group

1. Select **Topology** > **Workers**.

2. Select the check box next to the worker name.

3. Click the **Assign to Group** button.

4. Select the check boxes next to the worker groups that you want to assign this worker to.
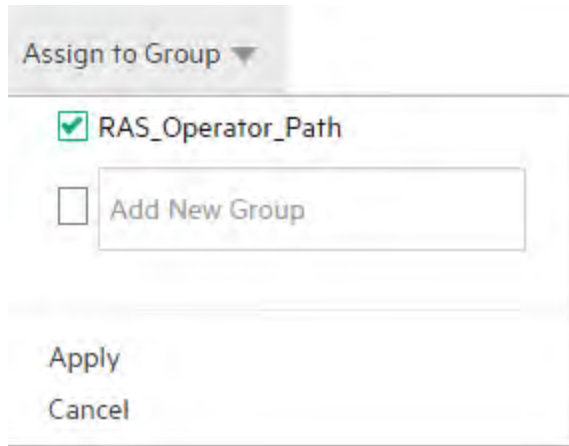
5. Click **Apply**.

Create a new worker group and assign a Central worker to it

1. Select **Topology** > **Workers**.

2. Select the check box next to the worker name.

3. Click the **Assign to Group** button.

4. Click the check box next to the empty box and enter the name of the new worker group that you want to create.
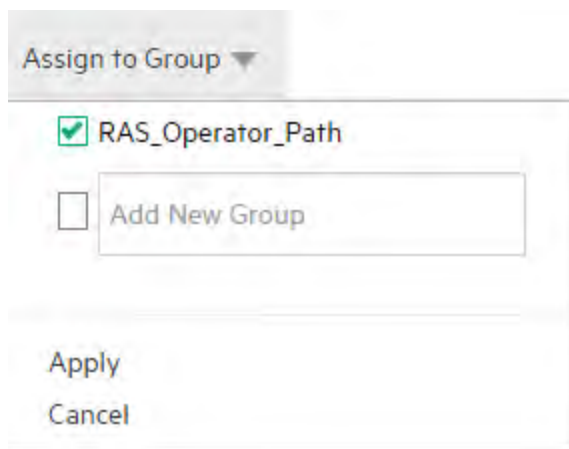
5. Click **Apply**.

Remove a Central worker from the a worker group

1. Select **Topology** > **Workers**.

2. Select the check box next to the worker name.

3. Click the **Assign to Group** button.

4. Clear the check boxes next to the worker groups that you want to remove this worker from.

5. Click **Apply**.

Disable a worker/RAS

1. Select **Content Management** > **Topology** > **Workers**.

2. Select the check box next to the worker name.

3. In the Workers toolbar, click **Disable** .

   > **Note:** The **Disable** button is only available if at least one enabled worker is selected.

Enable a worker/RAS

1. Select **Topology** > **Workers**.

2. Select the check box next to the name of the disabled worker.

3. In the Workers toolbar, click **Enable** .

   > **Note:** The **Enable** button is only available if at least one disabled worker is selected.

Delete a RAS

You can delete a RAS only if it is stopped.

It is not possible to delete an internal worker, which is within Central.

> **Note:** If you have stopped a RAS service (or even uninstalled the RAS), it takes a few minutes until this is reflected in the **Topology** > **Workers** tab. You may need to wait for a few minutes before that status is updated to **Stopped**, before you can delete the RAS. There is no need to restart Central.

1. Select **Topology** > **Workers**.

2. Select the check box next to the worker name.

3. In the Workers toolbar, click **Delete** .

   > **Note:** The **Delete** button is only available if at least one worker is selected and if the selected worker is not currently running.

Verify that a component (RAS or cluster node) was successfully installed

You can use the **Topology**/**Workers** tab to verify that a component was successfully installed.

If a worker is successfully installed, it will be visible in the grid and shown to be online.

Reference Material

Topology > Workers
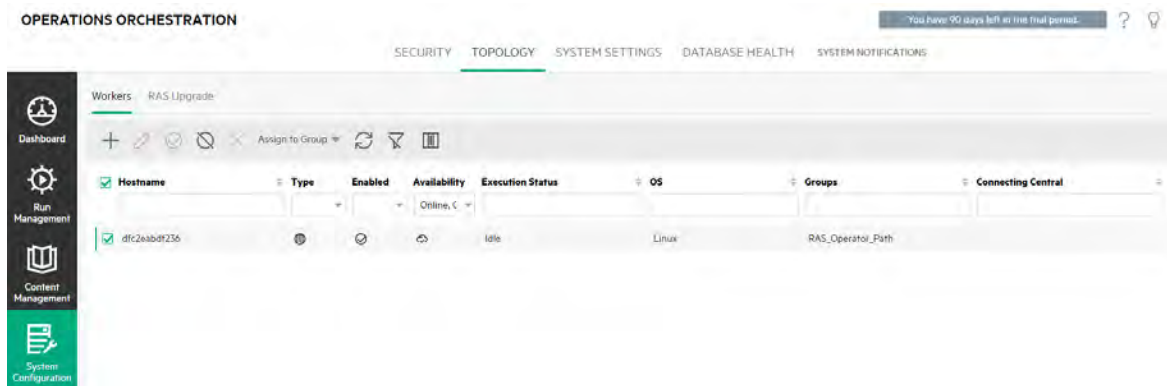
GUI item	Description
Add Reverse RAS button ➕	Click to create a new reverse RAS connection. You must do this for each reverse RAS that you installed (those that were configured to accept connection from Central).
Edit Reverse RAS button ✏️	Click to modify the connection details of the selected reverse RAS.
Enable button ⊘	Click to enable the selected worker/RAS. The Enable button is only available if at least one disabled worker/RAS is selected.
Disable button ⊘	Click to disable the selected worker/RAS. The Disable button is only available if at least one enabled worker/RAS is selected.
Delete button ✕	Click to delete the selected worker/RAS. The Delete button is only available if at least one worker/RAS is selected and if the selected worker/RAS is not currently running.
Refresh button ⟳	Click to refresh the display of workers/RASes.
Clear Filters button ▽	Click to remove the filters and display all the workers/RASes.

| Select Columns button	Click to display the column picker, to choose which columns to display.
Assign to Group button	Click to display the **Assign to Group** pop-up menu.
Filter boxes	In the filter boxes at the top of the columns, enter text to filter the displayed workers/RASes.
Host Name	The name of the host where this worker/RAS is located.
Enabled	Whether the worker/RAS is enabled or disabled.
Availability	Whether the worker/RAS is online or offline.
Execution Status	Displays information about the resource consumption of the worker/RAS. The information about resource consumption is automatically refreshed periodically. Take into account that the displayed information is only relevant to the last few minutes, since the previous refresh, and is not cumulative. In order to properly understand what is happening in your environment, it is recommended to observe the statuses over a period of time. • **Idle**: This worker/RAS has almost no tasks to process. • **Low Utilization**: This worker/RAS is working at low utilization. If your workers/RASes are with idle/low utilization for a long while, consider removing them. • **Working**: This worker/RAS is working with a regular load. • **Pending on Central**: Most of the time, this worker/RAS is waiting to send execution results to Central. Central is busy with handling other tasks. Consider adding more Centrals, or monitor your database for space, connectivity, and so on, to make sure it is not the bottle neck. • **Loaded**: At the current time, this worker/RAS is not able to process all Central tasks. If a worker/RAS is in a loaded state for a long time, consider scaling out or scaling up by adding more worker threads.
Path	The path to the host where this worker is located. The path indicates if the worker is in Central or in a RAS.
OS	The operating system of the machine where this worker/RAS is located.
JDK	The JDK version that is installed on the worker/RAS machine.
.NET	The .NET version that is installed on the worker/RAS machine.
ID	The UUID (unique identifier) of the worker/RAS.
Groups	The worker groups to which the selected worker/RAS is assigned.

Type	Whether the item is a Central worker, a standard RAS, or a reverse RAS.	
	Central worker	Central workers are responsible for executing flows. An external worker connects to Central to obtain tasks (flow execution messages) to process. It can be assigned to a worker group. See Assigning Workers to Groups.
	Standard RAS	A RAS is a remote action server, containing a worker and a remote protocol for connecting with Central. A standard RAS will initiate the connection to Central.
	Reverse RAS	A reverse RAS will wait for Central to initiate the connection. See Registering a Reverse RAS.
Connecting Central	**For a reverse RAS only:** The host name of the Central that connects to the reverse RAS.	

Connect Reverse RAS dialog box

GUI item	Description
RAS host name/IP	Enter the name or IP of the RAS (mandatory).
Port	Enter the port of the RAS (mandatory).
Protocol	Enter whether the protocol is HTTP or HTTPS.
Share secret to connect	Enter the shared secret that was entered in the installation wizard when this RAS was installed (mandatory).
Test	It is recommended to test the connection before submitting.

Topology > Workers > Assign to Group pop-up

GUI item	Description
Assign to Group	Click the button to display the **Assign to Group** pop-up menu.
Add New Group	Select the check box and enter the name of a new group, to which the selected worker will be assigned.
<Worker groups>	Select the check boxes next to group names to assign the selected worker to these groups. The display in the dialog box will vary, depending on the groups that have been set up.
Apply	Click to apply the changes to the worker.
Cancel	Click to close the Assign to Group pop-up without saving the changes.

Setting Up Topology – Configuration

If you are using a load balancer, reverse proxy, or DNS load balancer, you need to tell Operations Orchestration where the relevant external URL is located. For example, **https://my.server.com:443/oo**.

If the URL of the load balancer host is not configured, the Run Tree and the Run Log will refer to the IP/host from the HTTP request.

> **Note:** It is also possible to configure the load balancer host URL via REST API for configuration items.

Compression should not be set on the load balancer configuration. This can affect the connection between Central and workers and can lead to problems in new worker registration or existing worker functionality.

What do you want to do?

Configure a external URL for a load balancer, reverse proxy, or DNS load balancer

1. Click the **System Configuration Workspace** button to display the System Configuration workspace.

2. Select **Topology** > **Configuration**.

3. In the **URL** box, enter the URL of the load balancer, reverse proxy, or DNS load balancer.

4. Click **Save**.

Reference Material

Topology > Configuration
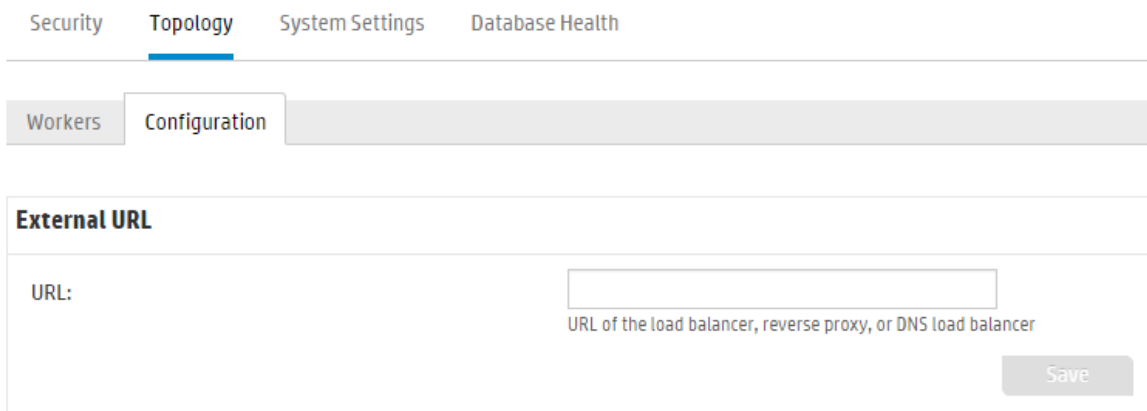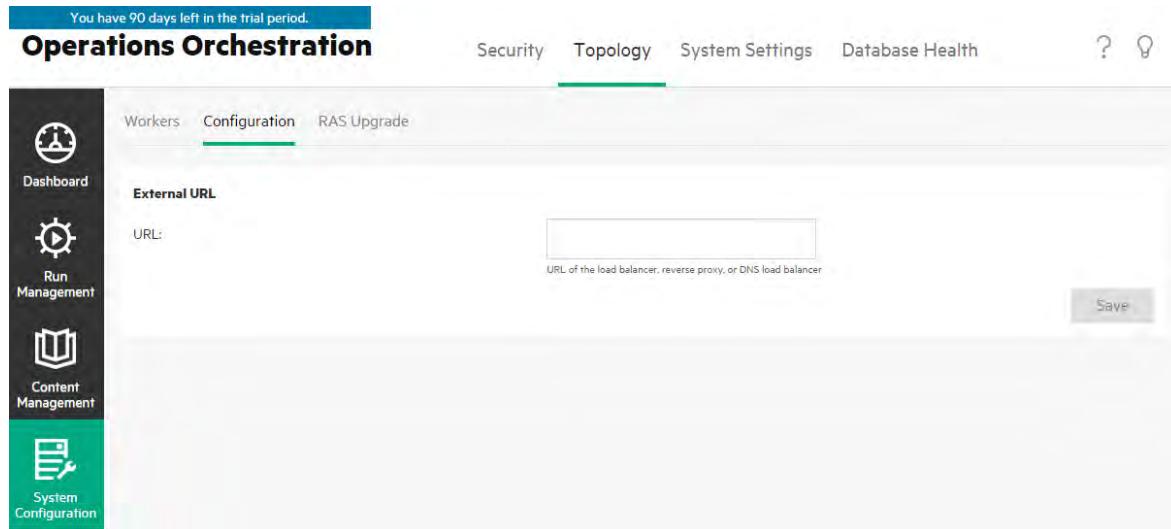
GUI item	Description
URL	Enter the URL of the load balancer, reverse proxy, or DNS load balancer.
Save	Click to save the configuration.

Setting Up Topology – Automatic RAS Upgrade

Your Centrals and RASes must be the same version. Central will only assign jobs to RASes that are of the same version.

Since version 10.6x, OO supports the automatic upgrade of RASes from Central, without manual intervention and without needing to access the RAS machines physically.

Automatic RAS upgrade is useful for customers with a large number of RASes in their environment, or with RASes in different time zones, making manual upgrading difficult.

Note: For FIPS140-2 compliant RASes you cannot use the automatic RAS upgrade procedure, so the RASes must be upgraded manually. This is because for FIPS140-2 compliance, the OpenJDK needs to be replaced by the Oracle JDK. In addition, the JDK needs to be configured for FIPS compliance which causes the upgrade package's signature to be voided.

As the upgrade package must have a valid signature in order for the automatic RAS upgrade procedure to be successful, this procedure cannot be used.

Note: Automatic RAS upgrade is only supported when both Central and the RASes are already at version 10.60 or later. For earlier versions, you will still need to upgrade the RASes manually. For example, you can't use it to upgrade RASes from 10.5x to 10.60.

Note: The **RAS Upgrade** tab is only visible if you have been assigned a role with the **Manage Topology** or **View Topology** permission.

There are four steps that you need to complete in order to upgrade your RASes using Automatic RAS Upgrade:

1. Upload the **RAS-Upgrade.zip** file to Central. This is a zip file containing the files that are required to upgrade the RASes to the relevant version. You can download the **RAS-Upgrade.zip** file from the same location where you download the new OO installation. Then, you upload the file to the Central database.

2. Prepare the RASes for upgrade. In this step, the RAS automatically runs a set of preparations, to ensure that the upgrade will be successful. This step runs pre-checks to verify that there are no problems in each RAS environment, and prepares those environments for upgrade. The RASes then download the upgrade package from Central.

   a. To start the preparation process, you need to manually select the required RASes from the grid and click the **Prepare for Upgrade** button.

During the preparation process, each RAS displays its status in the RAS grid. At the end, it displays whether the process was successful.

If a RAS encounters a problem, the details are displayed on the RAS grid. You need to fix the problem before upgrading that RAS. Examples of problems might be:

- Insufficient disk space. The RAS upgrade file may require up to 2 GB.

- Missing OS permissions (administrator permissions).

- Another OO component (Central or Studio) is installed alongside the RAS in the same OO installation directory. This kind of setup is not supported by this feature, and will require a manual upgrade.

- Problems where you need to check in the logs for more information.

  b. Once you have fixed the problems, you should run the preparation process again. This process can be repeated multiple times.

During the first two steps, you can use the RASes as usual. There is no need to schedule downtime.

The **RAS-Upgrade.zip** file is signed and verified before uploading and also after downloading. If packages are lost, the download will fail, so if this occurs, the download should be restarted.

When RAS transfers the **RAS-Upgrade.zip** file, this is done by streaming very small packets. This means that it can be sent to multiple RASes simultaneously, while taking up only minimal resources. This also means that if the download fails, on retry, it will start from where it failed, so the packets that were transferred successfully don't need to be transferred again.

3. Upgrade Central. The next step is to upgrade your environment, by upgrading Central to the new version. This is done manually. For more information, see the *OO Installation, Upgrade, and Configuration Guide*.

4. Upgrade the RASes. After Central is upgraded, you need to upgrade your RASes to the same version.

> **Important!** The versions of Central and the RASes *must be aligned*. RASes with older versions will not be able to run flows.

In the RAS grid, the **Version** column displays the version of each RAS. Check this column to see if your RASes are aligned with the Central version.

> **Note:** A RAS can only be upgraded if the preparation was successful, and its **Upgrade State** shows as **Ready for upgrade**.

Use

To start the upgrade process, you need to manually select the required RASes and click the

Upgrade Now ⊕ button.

During this process, each RAS runs the upgrade batch file, which is located in the **Upgrade-RAS.zip** file.

When the process ends, the **Upgrade State** column in the RAS grid shows whether the upgrade was successful. Each RAS that was successfully upgraded restarts and reports to Central that it is up in a new version.

The first two steps should be performed before the Central upgrade. This minimizes downtime and enables you to fix any issues that may be encountered, before the upgrade.

> **Note:** If Central is rolled back to the previous version, you will have to manually roll back the upgraded RASes.

> **Note:** This feature only works if you are upgrading from 10.6x to a later version of 10.6x. It will not work for RASes from earlier versions.

For more information about RASes, see the *Operations Orchestration Concepts Guide*.

What do you want to do?

Step 1: Upload the upgrade file to Central

This preparatory step should be completed before Central is upgraded.

1. Get a local copy of the **RAS-Upgrade.zip** file for the new version. You can do one of the following:

   ○ Go to the new OO version page in the SSO portal, and download the **RAS-Upgrade.zip** file and save it to your local file system. (This is supported for versions later than 10.60 only).

   ○ Go to the new OO version page in the SSO portal, and download the complete Central upgrade/installation package for the new version, and save it to your local file system.

2. Click the **System Configuration** button.

3. Select **Topology** > **RAS Upgrade**.

4. Click the **Upload** button.

5. In the Upload File dialog box, click **Browse** to navigate to the **RAS-Upgrade.zip** file, in the location where you saved it.
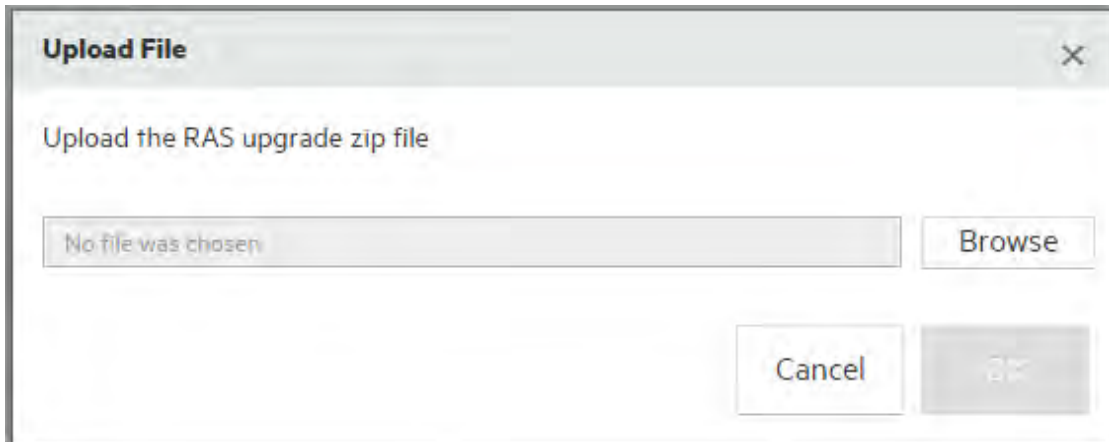
> **Note:** Make sure to upload the **RAS-Upgrade.zip** file, and not the full upgrade package.

6. Click **OK** to upload the upgrade file to Central.

   Central performs validations (for example, verifies the digital signature of the upgrade file), and receives the upgrade file.

Step 2: Prepare the RASes for upgrade

This step prepares the environment for upgrade by running pre-checks on the RASes, checking for disk space, permissions, and so on, and downloading the upgrader package to the RASes. The preparation is done in the background, and doesn't interrupt the routine work of your RASes.
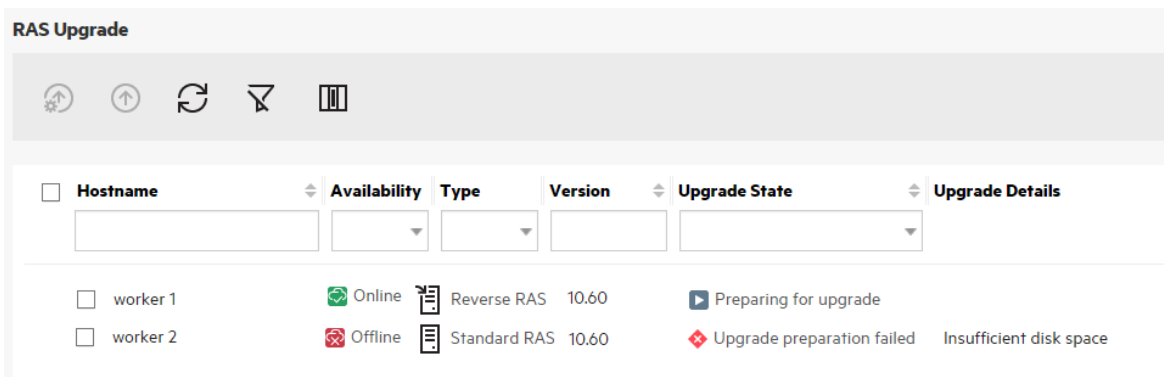
1. In the **Topology** > **RAS Upgrade** tab, select one or more RASes and click the **Prepare for Upgrade** button.

2. Check the status of the RASes, as displayed in the **Upgrade State** column.

   If there are any issues encountered during the download of the upgrader package to any of the RASes, the details are displayed next to that RAS.

3. If required, fix the problems and click the **Prepare for Upgrade** button again.

   When the checks and download are completed successfully for a RAS, the **Upgrade State** for that RAS is updated to **Ready for Upgrade**.

> **Note:** You can prepare RASes for upgrade either before or after you upgrade Central.
>
> You can only prepare RASes for upgrade if you have uploaded the **RAS-Upgrade.zip** file to Central.

Step 3: Upgrade Central

Upgrade Central to the relevant version, by running the upgrade script.

For more information, see the *OO Installation, Upgrade, and Configuration Guide*

Step 4: Upgrade the RASes

This step should be done after Central has been upgraded and after the RASes have been prepared for upgrade.

> **Note:** A RAS can only be upgraded if the preparation was successful, and its **Upgrade State** shows as **Ready for upgrade**.

1. Select **Topology** > **RAS Upgrade**.

2. Select the RASes that you want to upgrade (from those that passed the preparation), and click the **Upgrade Now** button. This triggers the auto upgrade in the RASes.

   > **Note:** When you click the **Upgrade Now** button, the RASes perform the prerequisite checks again. This is done because time may have passed since when you clicked the **Prepare for Upgrade** button, and the situation may have changed.

3. Follow the **Upgrade State** to monitor when your RASes have the status of **Upgrade succeeded**.

Reference Material

Topology > RAS Upgrade

GUI item	Description
Upload	Opens the Upload File dialog box, so that you can select the relevant upload zip file to upload to Central.
Clear	Click this button to delete the RAS upgrade files from the Central database.
Prepare for Upgrade	Click this button to get each selected RAS to perform the prerequisite checks and download the upgrader package from Central.
Upgrade Now	Click this button to trigger the auto upgrade in the selected RASes.
Clear Filters	Click this button to clear the filters and display all the RASes.
Refresh	The RAS table is automatically refreshed every three seconds. However, you can also click this button to refresh the display.
Select Columns	Click this button to adjust the display of columns.

Host Name	Displays the name of the host where the worker for this RAS is located.
Availability	Displays whether the RAS is online or offline.
Type	Displays whether the RAS is a standard RAS or reverse RAS. For more information about RAS types, see "Setting Up Topology – Workers and RASes" on page 114.
Version	Displays the version of OO that the RAS is currently running.
ID	Displays the UUID (unique identifier) of the RAS.
Path	Displays the path where the worker for this RAS is located.
Upgrade State	Displays the status of the upgrade process for the RAS. • Preparing for upgrade • Upgrade preparation failed • Ready for upgrade • Upgrading • Upgrade failed • Upgrade succeeded
Upgrade Details	If there is a problem with the upgrade, this column displays the details of the problem. For example, if there was insufficient space for the upgrade file to be downloaded to the RAS.

Setting Up the System Settings - General Settings

In the general system settings (**System Configuration** > **System Settings** > **General**), you can set defaults and customizations, and configure a forward proxy, reverse proxy, or load balancer.

Customize elements of the OO user interface

You can add extra text to the header title in the top banner in Central, in addition to the default text of "Operations Orchestration".

By adding text, you can make Central look different in different environments, so that users will be able to identify which environment they are in. For example:

You can add a description of the ROI counter, in the Dashboard, in order to help users understand the meaning of the ROI values. For example, by entering the words "Total ROI in dollars" or "Total ROI in work days" you can show users that ROI is calculated in dollars or in work days.

> **Note:** The text that you enter as the ROI description overwrites the default title of "Total ROI". Note also that this text does not get translated if you change the locale of the browser.

Set default timeout settings for runs

You can configure the default timeout settings for the system, so that if a flow run exceeds the specified duration, it will be canceled and will show the status **Canceled-Timeout Expired**.

> **Note:** The system timeout setting is applied to all flows in the system. However, if a flow is given a custom timeout setting in the Flow Library, or set to have none, this will overwrite the default. For more information, see "Managing the Flow Library" on page 180.

Configure the external URL

If you are using a load balancer, reverse proxy, or DNS load balancer, you need to tell OO where the relevant external URL is located.

For example, **https://my.server.com:443/oo**.

If the URL of the load balancer host is not configured, the Run Tree and the Run Log will refer to the IP/host from the HTTP request.

Compression should not be set on the load balancer configuration. This can affect the connection between Central and workers and can lead to problems in new worker registration or existing worker functionality.

Configure a forward proxy

If you are using a forward proxy, you need to tell OO its IP address, port number, username and password.

What do you want to do?

Add extra text to the header title

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **System Settings**.

3. In the **Extra text in title** box, enter the extra text to be displayed in the header title after "Operations Orchestration".

   The maximum number of characters that can be entered is 30 for Latin letters and 19 for double-byte letters.

4. Click **Save**.

Add a description of the ROI counter

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **System Settings** > **General**.

3. In the **Override ROI description** box, enter a description of the value that is used to calculate ROI. For example, "Total ROI in US dollars".

4. Click **Save**.

Enable/disable the rerun functionality

1. Select the **System Configuration > System Settings** tab.

2. In the Run Settings area, select the **Enable the rerun option for failed or canceled runs** check box.

3. Reselect the check box to disable the rerun functionality. The rerun icon will no longer appear next to steps that can be rerun.

Configure the default timeout settings

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **System Settings** > **General**.

3. In the **Timeout (minutes)** box, enter a number larger than 0, to set the number of minutes for the default timeout setting. A flow run will be canceled if it exceeds this duration.

   | **Note:** This value must be an integer.

4. Click **Save**.

Configure a external URL for a load balancer, reverse proxy, or DNS load balancer

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **System Settings** > **General**.

3. In the **URL** box, enter the URL of the load balancer, reverse proxy, or DNS load balancer.

4. Click **Save**.

Configure a forward proxy

1. Click the **System Configuration** button to display the System Configuration workspace.

2. Select **System Settings** > **General**.

3. Enter the IP address, host number, username, and password for the forward proxy.

4. Click **Save**.

Reference Material

System Settings> General tab

GUI item	Description

Use

General Settings	
Extra text in the title	Enter the extra text to be displayed in the header title after "Operations Orchestration". The maximum number of characters that can be entered is 30 for Latin letters and 19 for double-byte letters.
Override ROI description	Enter a description of the value that is used to calculate ROI. For example, "Total ROI in US dollars".
Run Settings	
Timeout (minutes)	Enter a number larger than 0, to set the number of minutes for the default timeout setting. A flow run will be canceled if it exceeds this duration. If nothing is entered, the system timeout is disabled. **Note:** This value must be an integer.
External URL	
URL	Enter the URL of the load balancer, reverse proxy, or DNS load balancer.
Forward Proxy	
IP	Enter the IP address and port number for the forward proxy.
Username	Enter the user name for the forward proxy.
Password	Enter the password for the forward proxy.

Setting Up the System Settings - Licensing

📖 When you install or upgrade to OO, this installs the Trial version license. You will need to install an Enterprise Edition license within 90 days.

OO offers the following options for licensing:

- **OO Enterprise Edition Trial**: A temporary license that runs for 90 days, and includes all out-of-the-box content packs. After the trial license is expired, you will need to purchase the OO Enterprise Edition license to use the content packs. This the default license that comes with the OO installer.

- **OO Enterprise Edition**: A full license, unlimited in time, which includes all out-of-the-box content packs. This license unlocks the OO Community Edition and OO Enterprise Edition Trial licenses.

- **OO Community Edition**: A partial license, which includes the Base content pack and the CE content packs, and enables 500 runs per month.

You can view the license information by clicking the **Info** 💡 button in the top right corner, to display the **About** popup window. This window displays the current status of the license: if the license is limited in time or number of runs, it shows how much is left.

About HPE Operations Orchestration ✕

OPERATIONS ORCHESTRATION

Version:

10.70 - Trial Edition

Build Number:

1368

Licensing:

You have 84 days left in the trial period.

(c) Copyright 2015 Hewlett Packard Enterprise Development Company, L.P.

Open source and third-party software license agreements for this product are available here

Hewlett Packard Enterprise

Close
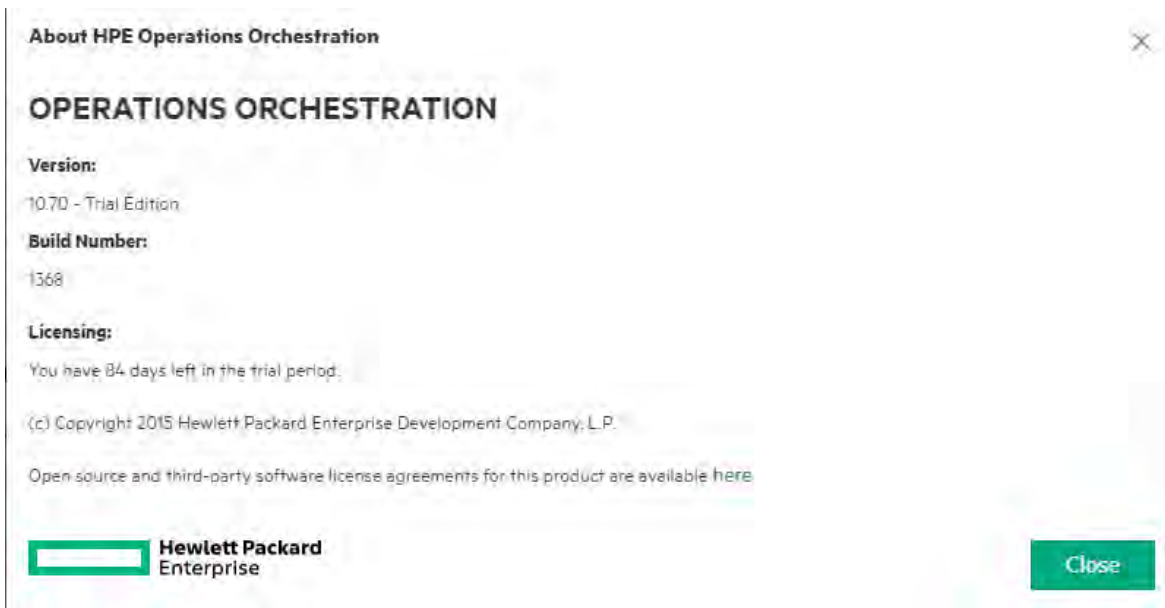
If you try to run flows after the license has expired, an error message appears, suggesting that you install an Enterprise Edition license.

You can also view details about the current license in the **License** tab under **System Configuration** > **System Settings**. From there, you can install a new license.
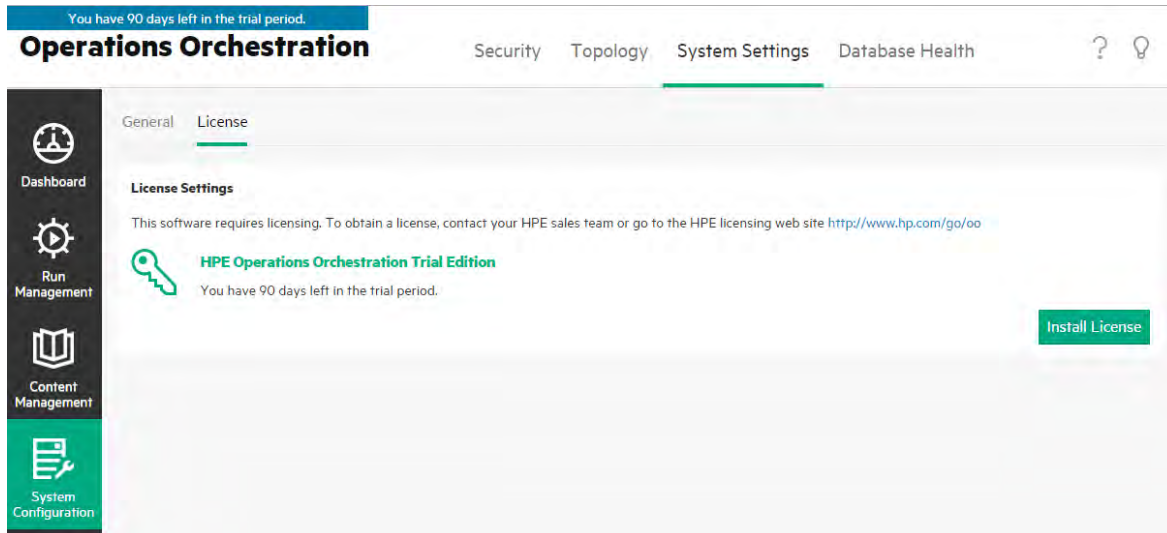
You can also see the remaining days/runs in the banner that appears at the top of all the Central windows.

You can click this banner from any location in Central, to open the **License** tab.

If you upgrade to a new version, the license is still valid.

Licensing a cluster

For a cluster, you need to request the license for one of the cluster IPs (and not on the load balancer IP) and install the license on that same cluster node.

This sets the license for the entire cluster even if that node is subsequently removed from the cluster.

What do you want to do?

Acquire a license

There are two ways to purchase a license:

- **Option 1**: Download the license from the OO web site at

  http://enterpriselicense.hpe.com/redirector/home

  Alternatively, you can go to
  https://h30580.www3.HPE.COM/poeticWeb/portalintegration/hppWelcome.htm

After logging in, enter the EON (Entitlement Order Number) in the Welcome page to see your license.

For more information about licensing details, contact your Sales team or Account Manager.

- **Option 2**: Issue a license using the OO License Management system:

  - For a standalone Central installation, issue the license with the IP address of the Central server.

  - If you have CSA and OO installed on the same machine, issue one license for that IP address.

  - If you have CSA and OO installed on separate machines, issue two licenses for the two IP addresses.

  - For a cluster, choose one of the nodes and request a license for the IP address of this node. Then, install the license on that node.

    > **Note:** Make sure to open the Central UI of the selected node (and not via the Load Balancer IP) when you install the license.

  - If you have CSA and an OO cluster, issue two licenses: one for CSA and one for a selected node.

    > **Note:** Make sure to open the Central UI of the selected node (and not via the Load Balancer IP) when you install the license for OO.

Install a license

1. Acquire the relevant license, as described above.

2. In Central, click the **System Configuration** button to display the System Configuration workspace.

   > **Important!** If you are working with a cluster or if you have CSA and OO on different machines, make sure that the open Central is at the IP address for which the license was issued.

3. Select **System Settings** > **License**.

4. Click the **Install License** button.

5. In the Install License dialog box, click **Browse** to navigate to the location of the license file, which was provided when you purchased the license.

Install License

Install License

| No file was chosen | Browse |

Cancel   OK

6. Click **OK**.

> **Note:** You can also install a new license via the OO Shell utility or via API.

Troubleshooting

In some cases, you may receive an error message saying that the license file is invalid.

If this error message appears, it could be because of the following:

- If you have **CSA and OO installed on separate machines**, you issued two licenses for the two IPs.

  ○ Make sure that you selected the license for the OO IP and not the CSA IP.

- For a **cluster**, you chose one of the nodes and issued a license for the IP address of this node.

  ○ Make sure that you are working from the Central at the IP address that was specified when you issued the license.

  ○ Make sure that you are not working from the Load Balancer IP.

- If you have **CSA and an OO cluster**, you issued two licenses: one for CSA and one for a selected node.

  ○ Make sure that you are working from the Central at the IP address that was specified when you issued the license.

  ○ Make sure that you are not working from the Load Balancer IP.

  ○ Make sure that you selected the license for the OO IP and not the CSA IP.

Reference Material

License tab
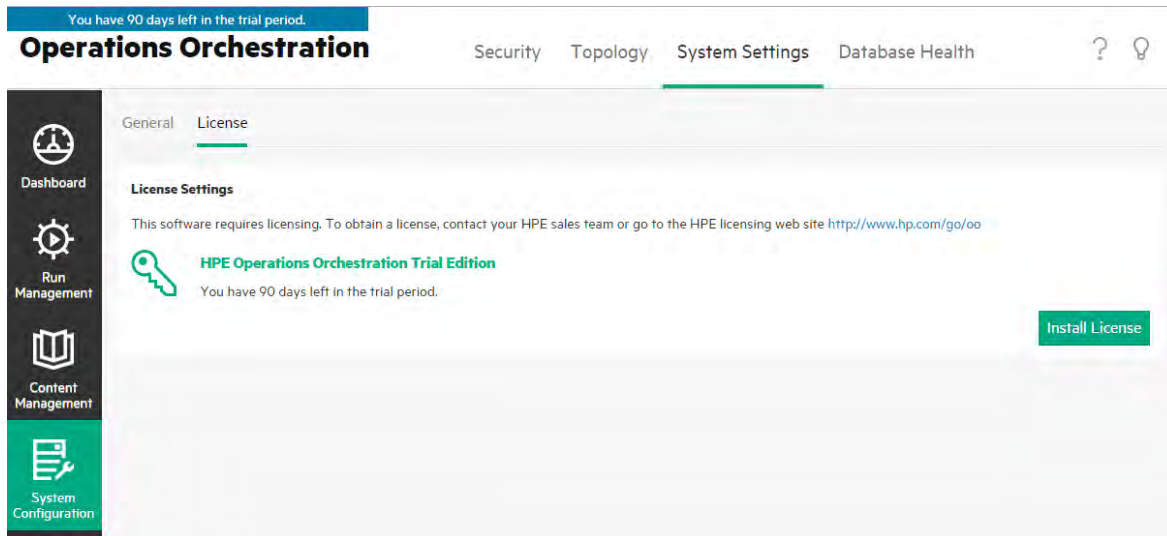
GUI item	Description
Install License	Click to display the Install License dialog box.

Install License dialog box

GUI item	Description
Browse	Click to navigate to the location of the license file, which was provided when you purchased the license.

Monitoring and Controlling Database Size

It is possible to monitor and control the size of your database, in order to enhance performance.

Under the **Database Health** tab in the System Configuration workspace, you can:

- Monitor the size of your database over different periods of time, and see how data is divided between execution, debugger and auditing usage.

- See information about how to schedule purging flows, in order to reduce the database size.

- Set the default level of information that is persisted in the Run Log. Reducing this level can help reduce the database size.
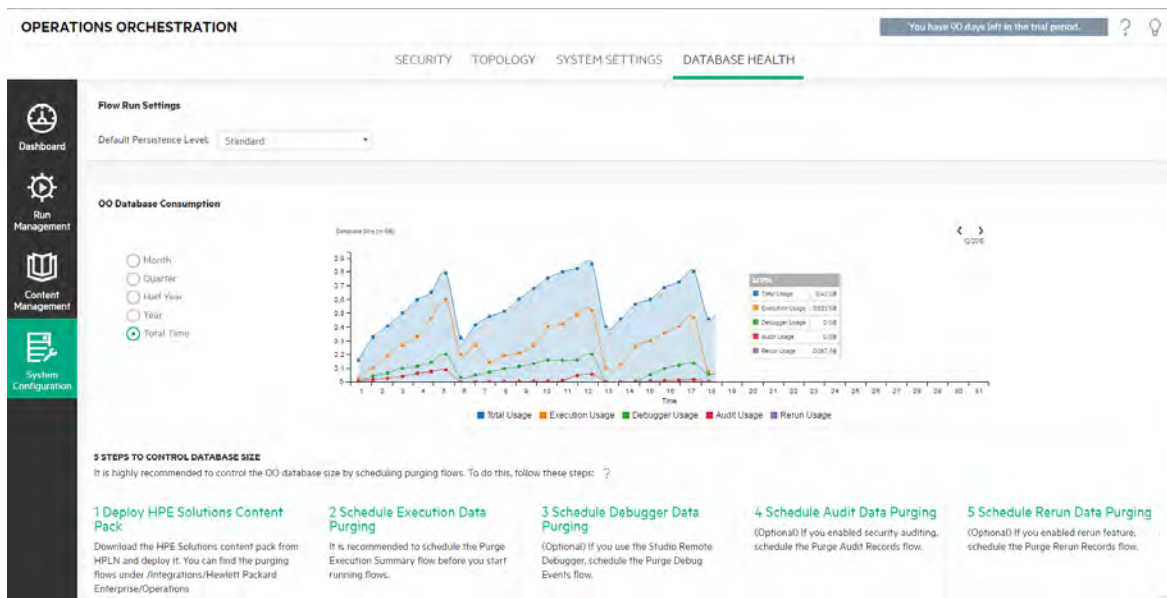
Monitor the Database Size

In the **Database Size** section, you can monitor the size of your database, comparing the current size to the size that it was in the past. You can adjust the graph to display the database size over the past month, quarter, half year, year, or total time. You can also see how the data is divided between execution, debugger and auditing usage.

> **Note:** The graph is updated approximately every 24 hours from the time Central was first up. For example, if you run a purging flow to reduce the database size, the updated size will be displayed on the following day.

The **OO Database Consumption** section displays information about how to schedule purging flows, in order to reduce the size of the database. It is highly recommended to follow the described steps to deploy the HPE **Solutions** content pack and run the purging flows.

Use

A Help button is available, which you can click to see more information about scheduling purging flows.

> **Important!** You should schedule the **Purge Execution Summary** flow before you start to run flows in Central. This will help reduce the size of the database and thus enhance the performance of OO.

> **Note:** The Step Persistence type selected for a step in a flow (Concise or Detailed) also affects the database size.
>
> See "**Step Inspector > Advanced tab**" in the *OO Studio Authoring Guide* for details on how to select the Step Persistence type and "Run Info > Step Details tab (Detailed Step Persistence)" on page 269 and "Run Info > Step Details tab (Concise Step Persistence)" on page 268 for details on how this effects the flow run.

Default Persistence Level for the Run Log

In OO 10.x, a very detailed run history is saved to the database. This makes it easy to troubleshoot, as all the information is available in the Run Log. However, you may want to reduce the level of information that is persisted. For example, if your database size has increased to the limit or if some of the persisted data is not relevant for you.

By selecting the default persistence level, you can control the level of detail that is saved to the Run Log. You can choose between:

- **Extended** - large input/output values are not truncated when persisted.
- **Standard** - when persisted, large input/output values are truncated when they exceed the truncation threshold (in English, 4,000 bytes) in UTF-8 encoding.

  > **Note:** The **Extended** level is similar to the behavior of previous 10.x versions. Therefore, when you upgrade from a previous version, the default persistence level is **Extended**, while in a new installation, the default is **Standard**.
  >
  > The top-level inputs and outputs of the launched flow are always logged with their full values (even at **Standard** level).

The default persistence level is applied to all flows that are run. You can override this default for individual flows in the flow library, or when triggering or scheduling a flow run.

> **Note:** All runs started from the Remote Debugger have the **Extended** persistence level.

How Values are Truncated at Standard Level

At **Standard** level, large input/output values are truncated when they exceed the truncation threshold in UTF-8 encoding. For English letters and symbols, this is 4000 bytes, but for other collations it may vary from 1000 to 4000 bytes (3000, in most cases).

This translates to:

- 4,000 ASCII characters

- 2,000 Latin-based characters (non-ASCII)

- 1,333 common CJK characters (Chinese, Japanese and Korean)

- 1,000 rare CJK characters and symbols

If the value combines different types of characters, the truncation threshold will be somewhere in between the above numbers. For example, Latin-based languages use a lot of ASCII characters, so the truncation point for such texts can be anywhere between 2,000 and 4,000 characters (typically over 3,000).

When the value is truncated, a "(truncated)" suffix is added to the stored value, so there are 3,985 bytes of actual content (instead of 4,000).

What do you want to do?

Set the persistence level for the Run Log

1. Click the **System Configuration** button to display the System Configuration workspace, and select the **Database Health** tab.

2. Select one of the options from the **Default persistence level** list:

   - **Extended** - large input/output values are not truncated when persisted

   - **Standard** - when persisted, large input/output values are truncated when they exceed the truncation threshold (in English, 4,000 bytes) in UTF-8 encoding.

This default level will be applied to all flows that are run. You can override this default for individual flows or folders in the flow library, or when triggering or scheduling a flow run.

Overwrite the persistence level for the Run Log

It is possible to overwrite the default persistence level that is set on the **Database Health** tab, in the following places:

- **Flow Library** - You can set the persistence level for individual flows in the Flow Library. For more information, see "Managing the Flow Library" on page 180.

- **Flow Launcher** - You can set the persistence level for a specific flow, when triggering a run. By

default, the persistence level is taken from the system configuration (first) and the Flow Library (second). For more information, see "Running a Flow" on page 209

- **Scheduler** - You can set the persistence level for the specific scheduler configuration. By default, the persistence level is taken from the system configuration (first) and the Flow Library (second). For more information, see "Scheduling Flow Runs" on page 225.

Monitor the size of your database

1. Click the **System Configuration** button to display the System Configuration workspace, and select the **Database Health** tab.

2. Under **OO Database Consumption**, select one of the options to display the database size (in MB) over the past month, quarter, half year, year, or total time.

3. If required, click the arrows to display areas of the graph that are hidden.

4. Click the Legend blocks under the graph to filter the display, in order to hide/show the execution, debugger, and auditing and rerun data.

■ Total Usage   ■ Execution Usage   ■ Debugger Usage   ■ Audit Usage   ■ Rerun Usage

Control the size of your database

It is strongly recommended to schedule the purging flows, in order to control the size of the OO database and enhance the performance of OO.

Important! Make sure to schedule the **Purge Execution Summary** flow *before you start* to run flows in Central!

Note: For each purge type (rerun, steplog, and so on), you can only run one purge flow at a time.

1. Download the **Solutions** content pack from ITOM Marketplace.

2. Deploy this content pack to Central.

   a. Open the **Content Packs** tab in Content Management and click the **Deploy New Content** button.

   b. In the Deploy New Content dialog box, click the **Add** ✛ button.

   c. Browse to and select the **Solutions** content pack, and then click **Open**.

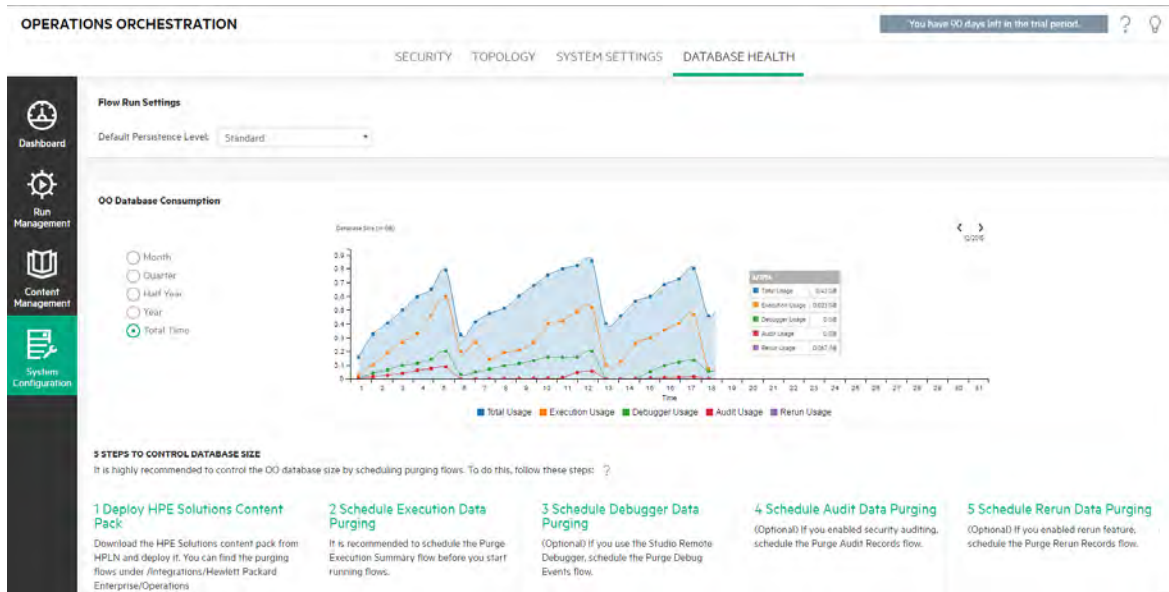   For more information, see "Deploying and Managing Content Packs" on page 160.

3. Open the **Flow Library** tab in Content Management and locate the purging flows, under **Integrations/Hewlett-Packard Enterprise/Operations Orchestration/10.x/Database**.

4. Open the **Scheduler** tab in **Run Management.**

5. Create schedules to run the following flows on a regular basis:

   ◦ **Purge Execution Summary** - recommended for all users

   ◦ **Purge Debug Events** - if you use the Studio Remote Debugger

   ◦ **Purge Audit Records** - if you have enabled security auditing

   ◦ **Purge Rerun Info** - if you have defined rerun points
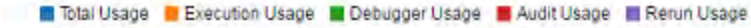
   For more information, see "Scheduling Flow Runs" on page 225.

Reference Material

Database Health tab

GUI item	Description
Run Log Persistence Level	Select the default level of detail that will be saved to the Run Log when a flow is run: • **Standard** - large input/output values are truncated at approximately 4,000 bytes when persisted • **Extended** - large input/output values are not truncated when persisted > **Note:** The **Extended** level is similar to the behavior of previous 10.x versions. > The top-level inputs and outputs of the launched flow are always

	logged with their full values (even at **Standard** level).
OO Database Consumption	See a graph showing the database size in MB, over a period of time, ending with the current date. Select the options to change the displayed time period. If required, click the arrows to display areas of the graph that are hidden.
Legend blocks	■ Total Usage ■ Execution Usage ■ Debugger Usage ■ Audit Usage ■ Rerun Usage Click the Legend blocks under the graph to filter the display, in order to hide/show the execution, debugger, and auditing data.
5 Steps to Control Database Size	View the steps describing how to deploy the **Solutions** content pack and run the purging flows.

Controlling Database Size by Purging

It is highly recommended to schedule the purging flows, in order to control the size of the Operations Orchestration database and enhance the performance of Operations Orchestration.

> **Important!** Make sure to schedule the **Purge Execution Summary** flow *before you start* to run flows in Central!

1. Download the HPE **Solutions** content pack from ITOM Marketplace.

2. Deploy this content pack to Central.

   a. Open the **Content Packs** tab in Content Management and click the **Deploy New Content** button.

   b. In the Deploy New Content dialog box, click the **Add** button.

   c. Browse to and select the **Solutions** content pack, and then click **Open**.

   For more information about deploying content packs, see "Deploying and Managing Content Packs" on page 160.

3. Open the **Flow Library** tab in Content Management and locate the purging flows, under **Integrations/Hewlett Packard Enterprise/Operations Orchestration/10.x/Database**.

4. Open the **Scheduler** tab in the Run Management workspace.

5. Create schedules to run the following flows on a regular basis:

   ○ **Purge Execution Summary** - recommended for all users

   ○ **Purge Debug Events** - if you use the Studio Remote Debugger

   ○ **Purge Audit Records** - if you have enabled security auditing

   ○ **Purge Rerun Info** - if you have defined rerun points

   For more information about scheduling, see "Scheduling Flow Runs" on page 225.

For more information about using the purging flows, see the Operations Orchestration 10 Tutorials section, available on ITOM Marketplace here.

Send Email Notifications

You can notify Operations Orchestration users about system events through email messages. Notifications can be sent only if the OO Entrprise Edition license is installed. By default, no notifications are sent.

Operations Orchestration supports the following types of events:

- Authentication was disabled

- Flow running for more than a specified threshold

You need to configure a mail server and the notifications that you want to receive using the **System Notifications** screen. To configure notifications:

1. Click **System Configuration > System Notifications**.

2. In the **Host** field, enter the host name of the email server host.

3. In the **Port** field, enter the port number of the email server port.

4. In the **Username** field, enter the name of the

5. In the **Password** field,

6. In the **From** field, enter the name of the person from whom the email message is sent.

7. In the **To** field, enter the names of Operations Orchestration users who must receive the notifications.

8. From the **Notifications** area, select the notifications for which emails must be sent.

Depending on the type of notification you selected, emails are sent to all users specified in the To field

if authentication is disabled or if a flow runs longer than a specified threshold.

Customize Email Notifications

You can customize the email notifications that you can send by creating your own email templates. Email templates are located in the central/conf/notifications folder. For the "*Authentication was disabled*" event, you have to customize the **authenticationDisabled.html** file and for the "*Flow running for more than a specified threshold*" event, you must customize the **runningTimeExceeded.html** file.

Create email templates

1. Open the **authenticationDisabled.html** or the **runningTimeExceeded.html** file depending on which email notification you want to customize.

2. Insert the following values:

Variable	Description

${url}	URL of the Central installation
${id}	ID of the execution
${name}	Name of the execution
${flow}	Path of the running flow
${count}	Number of hours for which the execution is running

3. You can also include images in your templates. These images must be located in a directory with the same file name as that of the template (that is, `authenticationDisabled`, `runningTimeExceeded`). Images are included in the template by reference.

   For example, if you have an image named `image01.png` in the `authenticationDisabled` directory, you can reference it in the **authenticationDisabled.html** template as
   `<img src="cid:image01"/>`

Configure the resend interval

Email notifications about flows running for more than a specified threshold are resent once in 24 hours by default. You can configure the resend interval by changing the value of the `runningTimeExceede.repeat.interval` configuration item.

Setting the Browser Language

Central language support is according to the browser language. If the language is not supported, Central is displayed in English.

Set up your browser for foreign language support.

> **Note:** Make sure that the added language is the first in the list.

Change the language preferences on Internet Explorer

1. Go to **Tools** > **Internet Options** > **General** (tab) > **Languages** > **Language** preference.

2. Select the required language. Make sure the primary language is the first in the list.

Change the language preferences on Firefox

1. Go to **Tools** > **Options**, **Content** (tab) > **Languages** > **Choose** (button).

2. Select the required language.

Change the language preferences on Google Chrome

1. Click the **Customize and control Google Chrome** button, and then select **Settings** > **Show advanced settings**.

2. In the **Languages** section, click **Language and input settings**.

3. Click **Add**, and select the required language.

   You can add languages to this from a predefined set, and Chrome will send all the languages you choose with the Accept-Language header in the order in which you arrange them.

Change the language preferences on Safari

In Windows, you can't change the language sent by Safari. The language sent with the Accept-Language header is derived from your system settings.

> **Note:** It is also possible to change the MS SQL collation language or the **central-wrapper.conf** language for content. For more information, see the *Operations Orchestration Installation, Upgrade, and Configuration Guide*.

Working with CloudSlang Content Flows

It is also possible to deploy and run content that was authored using CloudSlang 1.0.

CloudSlang

CloudSlang is a flow-based orchestration tool for managing deployed applications.

The CloudSlang project is composed of three main parts: the CloudSlang Orchestration Engine, the CloudSlang language and the ready-made CloudSlang content.

The engine is packaged as a lightweight Java .jar file and can be embedded into existing Java projects.

The language is a YAML-based DSL for writing workflows. The workflows can be run by an embedded instance of the engine or the stand-alone CloudSlang CLI. The main types of CloudSlang content are operations, flows, and decisions. An operation contains an action, which can be written in Python or Java. Operations perform the "work" part of the workflow. A flow contains steps, which stitch together the actions performed by operations, navigating and passing data from one to the other based on operation results and outputs. Flows perform the "flow" part of the workflow. A decision provides the capability to select different outcomes.

The supported file extensions are: .sl, .sl.yaml and .sl.yml.

> **Note:** For more information about CloudSlang, see:
>
> http://www.cloudslang.io/#/docs
>
> https://github.com/cloudslang/cloud-slang
>
> https://github.com/cloudslang/score

You can run CloudSlang flows in all the different OO topologies—single node, Central and RAS, and clustering.

Version OO <10.80 can run CloudSlang <1.0 and OO 10.80 can run CloudSlang 1.0. See the OO *Release Notes* for the supported version of CloudSlang.

Content Lifecycle when working with CloudSlang

CloudSlang content has the same life cycle as OO content:

- Deploy a CloudSlang content pack. See "Deploying and Managing Content Packs" on page 160.

- Manage Flows: View flows in the flow library and set flow permissions. See "Managing the Flow Library" on page 180

- View and edit System Properties. For CloudSlang, System Properties can also be configured as Sensitive.See "Setting Up Configuration Items for a Content Pack" on page 193.

- Run and monitor flows: Run flows, schedule runs, track and manage flow runs. See "Running and Monitoring Flows" on page 207.

- Troubleshooting Flows Runs: Drill down into the flow run. See "Testing and Troubleshooting a Flow Run" on page 256.

Features supported by CloudSlang and/or OO Flows

This section describes where CloudSlang content and OO Flows content behavior is different:

- **Content Permissions.** Permissions to run a CloudSlang flow is are computed when the flow is launched or resumed. If you don't have the correct content permissions for all dependencies, you will not be able to launch or resume the flow. See "Setting Permissions for Content" on page 181

- **Flow graphical representation.** A flow graphical representation is not available: If you open the Flow Graph tab for a flow that was written in CloudSlang, a **not supported** message is shown. See "Viewing a Flow Graph" on page 189.

- **Sensitive Data for System Properties.** In CloudSlang, OO flows, also CloudSlang flows can include Sensitive Inputs and Outputs, and they will be hidden in Central.

- **Decisions.** CloudSlang step type called **Decision**, works along with existing subflows and operations. When running a flow that includes a decision, the result of a decision looks like a Custom result.

- **Content Distribution.** All required resources are resolved and downloaded from Central to the RAS at the start of the operation execution using maven. For Python based content, the administrator must add manually, in the python library in each RAS, all the dependencies needed to run the flow and are not included in HPE OO out-of-the-box python lib. This is done by adding them to the HPE OO class path in ALL HPE OO nodes in your environment (Central or RAS):

  - For Central: **<Installation_path>\central\lib\pythonlib**

  - For RAS: **<Installation_path>\RAS\lib\pythonlib**

- **CloudSlang content IDs** use the format **<namepace>.<name>**.

Feature	Supported in CloudSlang	Supported in OO Flows
Content Packs. Deploy, rollback, delete, view details in the Content Management workspace. See "Deploying and Managing Content Packs" on page 160.	V	V
Distribution Capability.	V Java automatically V Python manually	V
Content Permission.	V	V

Feature	Supported in CloudSlang	Supported in OO Flows
See ."Managing the Flow Library" on page 180		
Run Flows. See "Running a Flow" on page 209	V	V
Schedule a flow. See "Scheduling Flow Runs" on page 225.	V	V
Monitor Flows. See "Adjusting the Display of Flow Runs" on page 249	V	V
Manage Flow Runs. See"Tracking and Managing Flow Runs" on page 244.	V	V
Decisions.	V	X
Sensitive Data in Input and Output Fields	V	V
Sensitive Data for System Properties	V	X
View the Flow Graph for a flow. See "Viewing a Flow Graph" on page 189.	X	V
Worker Group. See Worker Groups and Group Aliases in the *HPE OO Concepts Guide*.	X	V

Promoting Content Packs

The Op Admin is usually the one responsible for promoting content packs.

Promoting a Content Pack – Overview

What is promotion?

Promotion is the process of deploying content packs across several environments, in order to ensure that results are predictable and without (unknown) risks. This is in particular important for maintaining a stable production environment that contains content that has been tested and validated prior to deployment.

For example, deploying the content pack on the following Central environments: Development, QA, Staging, Production.

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

Is promotion the same as deployment?

Deployment is part of promotion. However, the promotion process also includes other tasks, such as:

- Configuring the content pack: configuring worker group aliases, mapping system accounts, and so on
- Testing and troubleshooting the flows in the content pack

What is a content pack?

A content pack is a file containing operations, flows, actions, and configuration items. When flow authors complete a project, they package it into a content pack for promotion on the Central server.

- CloudSlang operations use actions that are Java-based or python-based.
- Native OO operations use actions that are Java-based or .Net based.

For more information about content packs, see the *OO Concepts Guide*.

When do you promote a content pack?

You promote a content pack when:

- There is a need for new content in the Production environment.
- You have fixed a bug in a flow and need to upload the new version of the flow to the Production environment.

- You need to add new functionality to an existing flow in the Production environment.

- You are upgrading to a new version of an out-of-the-box content pack.

Who does the promotion?

The promotion process in each of the environments is slightly different and may be done by a different persona. However, in most cases, it is the Op Admin who is responsible for promotion.

The person doing the promotion must be assigned a role with **View Content Packs** and **Manage Content Packs** permissions.

For more information, see the *Best practices for OO content promotion* document, available on ITOM Marketplace.

Promotion steps
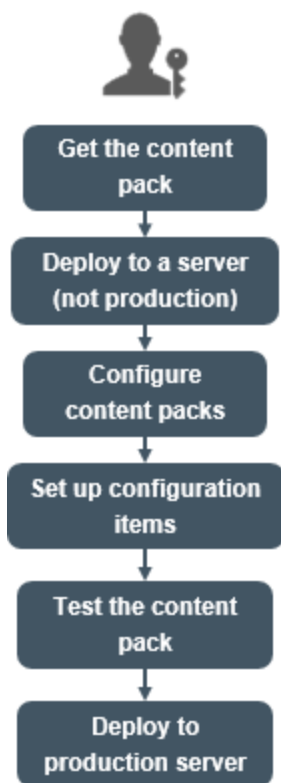
> **Note:** This is just a high level look at the Central workflow, and there are many options that are not described here. For more detailed information about any of the steps, use the links to learn about the options in-depth.

Step 1: Get the Content Pack From the Flow Author

1. The Flow Author creates a content pack in Studio. See "Exporting a Content Pack" in the *Studio Authoring Guide*.

2. The Flow Author saves the content pack to the Artifact Repository.

3. The Flow Author shares the content pack with the Op Admin/Promoter.

Step 2: Deploy to a non-Production Server

1. The Op Admin receives the content pack from the Flow Author.

2. The Op Admin deploys the content pack on the a non-Production server, for example, the Staging server. See "Deploying and Managing Content Packs" on the next page.

Step 3: Configure the Content in the Content Pack

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

1. The Op Admin sets the content permissions on the flows. See "Managing the Flow Library" on page 180.

2. The Op Admin sets the persistence level and run timeout for the flows. See "Managing the Flow Library" on page 180.

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

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

1. If the content pack includes system and system properties, the Op Admin assigns values to these in the content pack. See "Setting Up Configuration Items for a Content Pack" on page 193.

2. The Op Admin maps the aliases of the worker groups to actual worker groups. See "Setting Up Configuration Items for a Content Pack" on page 193.

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

1. The Op Admin runs each flow from **Content Management** > **Flow Library** and checks

whether it runs properly. See "Tracking and Managing Flow Runs" on page 244.

2. ![icon] If a flow run fails, the Op Admin can drill down into a flow to identify the problems. See "Testing and Troubleshooting a Flow Run" on page 256.

Step 6: Deploy to the Production Server

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

2. ![icon] Finally, the Op Admin deploys the content pack to the Production server.

Deploying and Managing Content Packs

![icon] In the **Content Packs** tab, you can deploy content packs to the Central environment. For example, Development, Testing, Production. Deployment is performed at least twice in the promotion process.

> **Note:** It is advised to deploy the base content pack and any other OO content packs that are used in Studio before deploying your own content packs.

Viewing Content Pack Details

After content packs have been deployed, they are displayed on the **Content Packs** tab.

> **Note:** After an upgrade, the deployment time is not displayed for content packs that were deployed to the previous version.

The **Content** tab on the right side displays more details about the selected content pack. This information includes the version, publisher, description, digital signature details, the name of the user who deployed it, the Source Control Management tag, SCM revision (for SVN), and the entities inside the content pack. The contents are arranged in a tree, under **Library** and **Configuration** folders, reflecting the way that the content pack appeared in Studio. You can drill down into the folders and see exactly which items exist in the content pack and where they are stored. This includes flows, operations, and configuration items.
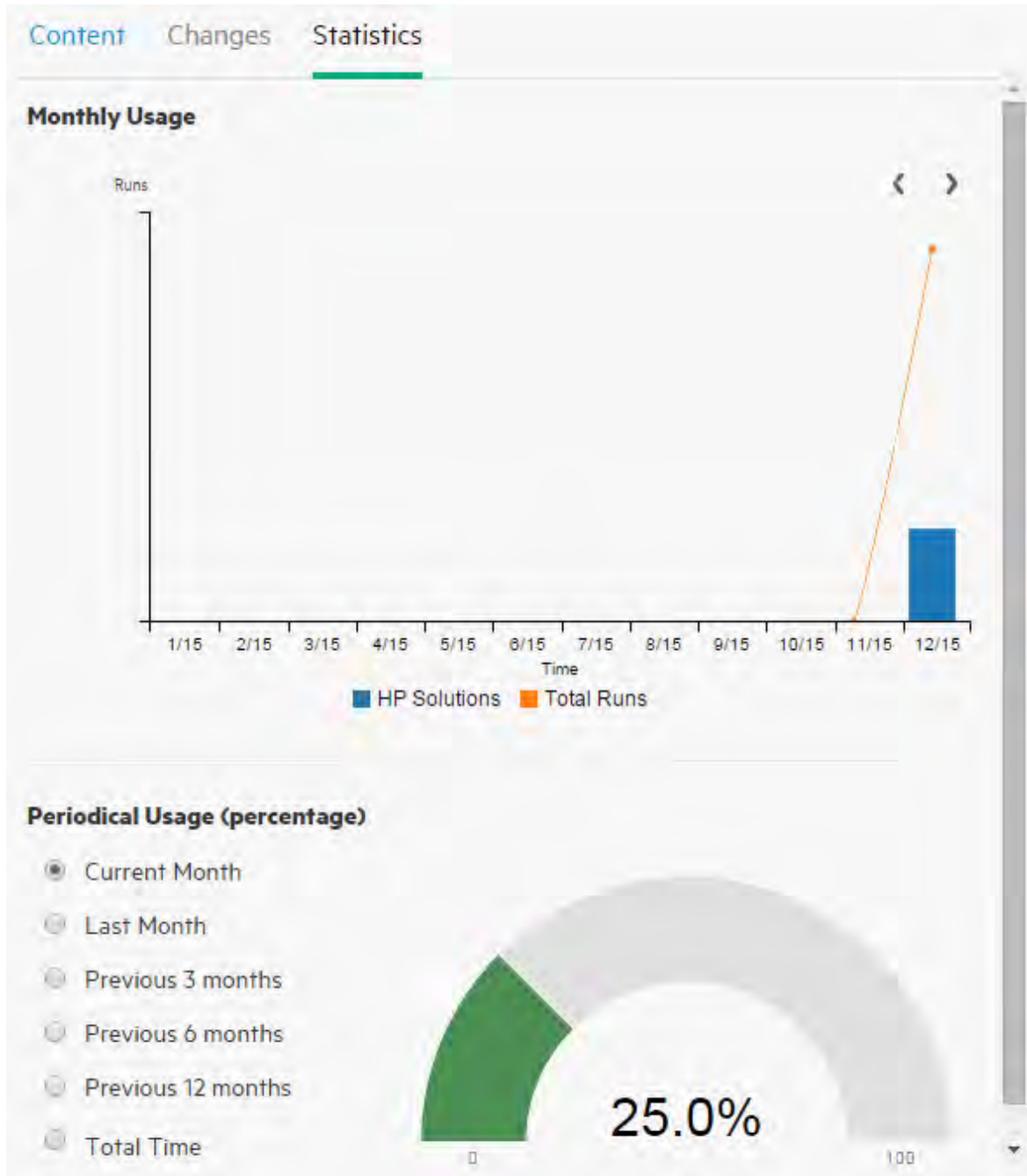
The **Changes** tab displays the changes that have been made to the selected content pack since the version that was previously deployed on Central. This information tells you which items have been added, deleted, updated, or moved.

The **Statistics** tab displays statistical information about the selected content pack:

- See how many flows were run were using content from this deployed content pack for each month since first use, and compare this with the total number of all runs over the same period.

- See how this content pack usage fits into the percentage of all runs during the current month, previous month, quarter, half year, year, or in total.

Content   Changes   **Statistics**

Monthly Usage

Runs   ‹ ›

1/15 2/15 3/15 4/15 5/15 6/15 7/15 8/15 9/15 10/15 11/15 12/15
Time
■ HP Solutions   ■ Total Runs

Periodical Usage (percentage)

⦿ Current Month

◯ Last Month

◯ Previous 3 months

◯ Previous 6 months

◯ Previous 12 months

◯ Total Time

25.0%

0   1.00

You can also click the Statistics �📊 button in the toolbar to display and compare statistical information about all deployed content packs.

Content Pack Versions

You can deploy a content pack to a server multiple times, as the content is updated by flow authors, and new versions of the content pack are created.

After a content pack has been deployed, it is possible to roll back to the last deployment version. This might be useful if you find that there are problems with the flows in a new deployment. For more information, see Roll back to a previous deployment.

> **Note:** When you roll back to the last deployment, you return the content to the state that it was in before your most recent deployment. If the most recent deployment involved multiple content packs, the rollback returns all of these content packs to the state they were in, previously.
>
> Note also that you can only roll back one version. You cannot roll back to earlier versions than the last one.

Deploying Multiple Content Packs

A single user can deploy multiple content packs at once, by using the **Add** ✛ button in the Deploy New Content dialog box.

However, it is not possible for two users to deploy content packs to the same Central at the same time. If you try to deploy content to Central while another user is already deploying content, you will see an error message, indicating that another deployment is currently in progress.

Split Deployment

After a content pack has been deployed, it is possible to split the original content pack into two or more parts in Studio and deploy each of these separately. Note that if one of these smaller content packs has the same name as the one that was originally deployed, it will overwrite the original one on the Central server. If it has a different name, it will be deployed alongside it.

## Content Pack Validation - Digital Signatures

A valid digital signature on a content pack verifies that the content pack was created by a known sender and that it was not altered in transit.

During the upload process, before deployment, Central checks that content packs with digital

signatures have not been altered or expired, and displays the details of the certificate.

> **Note:** OO verifies the following common signature types: .DSA, .RSA, .EC, .SF

If a signed content pack has been altered, that content pack cannot be deployed. If the signature has expired, the content pack can be deployed only if you select a check box agreeing to ignore this expiry. For more information, see Deploy a content pack.
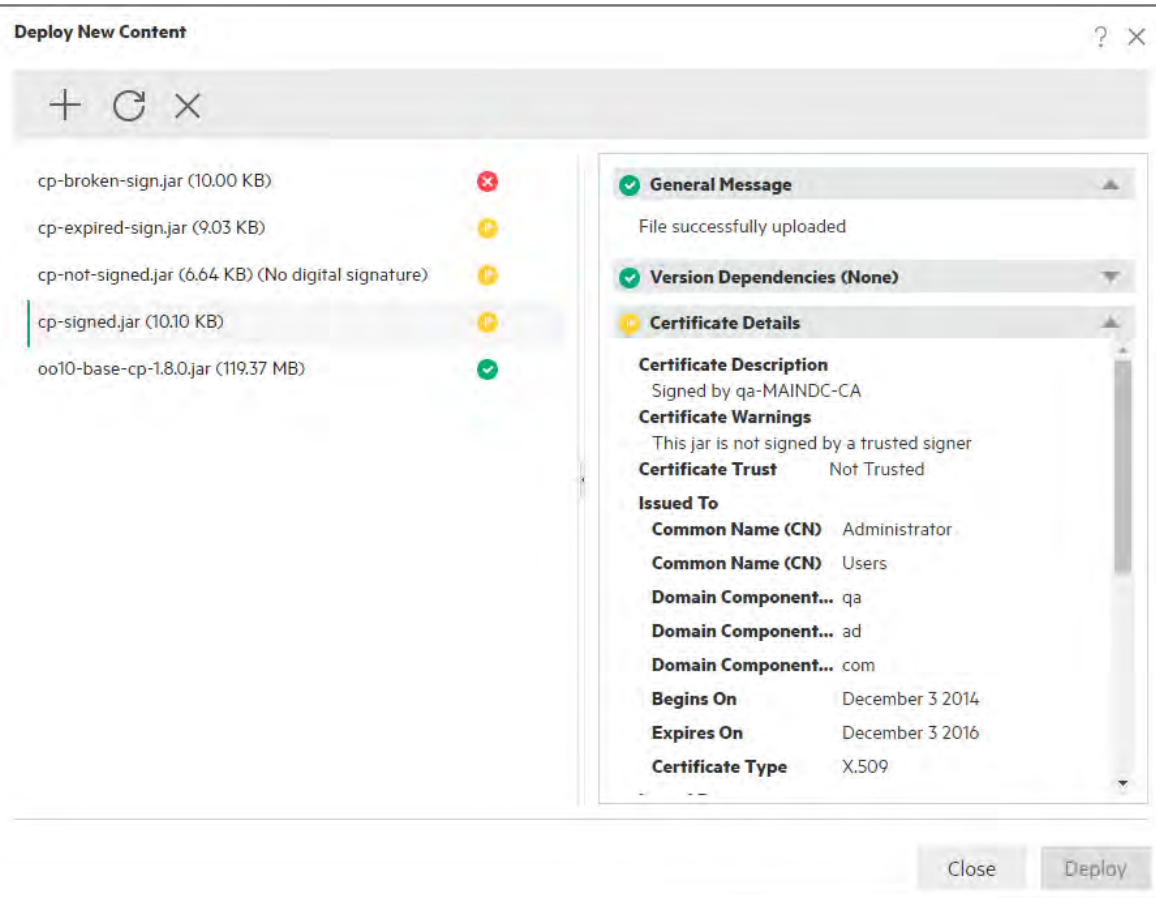
The scenarios for signatures are the following:

- The out-of-the-box content packs created by OO are signed with the private key.
- When you buy a content pack from a content pack vendor, the vendor uses an official certificate authority (CA) certificate.

  If the signing authority is not already in the Central TrustStore, you will need to import the CA certificate into it, in order to enable Central to verify the content pack's certificate.

- When OO authors create content packs in-house, they need a CA private key to be able to add a signature to a content pack. You can request this certificate from a company such as VeriSign, for

example. It is also possible to use a self-signed signature, but this is not recommended, for security reasons.

a. The promoter adds a digital signature to the content pack, using an external signing tool, such as Java JarSigner.

b. The administrator adds the trusted root certificate authority (CA) into the OO **client.truststore** file (under **central/var/security**). In a cluster, import the CA to the **client.truststore** file for all nodes.

For details about how to add a CA to the Central TrustStore, see "Importing a CA Root Certificate to the Central TrustStore" in the *OO Security and Hardening Guide*.

Security Note: It is recommended to verify the digital signature of a content pack manually using well known tools before importing it into Studio.

Content Pack Validation - Dependencies

During the upload process, before deployment, Central checks whether the content pack has any dependencies. If you are deploying a content pack that uses elements (for example, operations or properties) that are in a different content pack, the content pack that you are deploying will have a dependencies file that points to this other content pack.

However, what happens if any of these items were deleted from the other content pack, and the new version of this content pack has been deployed in your environment? If this occurs, the content pack that you are now trying to deploy will have missing dependencies, and this could cause the deployment to fail.

If there are missing dependencies, Central displays a warning icon in the Deploy New Content dialog box and displays the expected version of the content pack that contained these dependencies. This gives you an opportunity to add the expected version of this content pack to the deployment, so that the deployment of the content pack that depends on it will be successful.

Note, however, that you have the option to continue with the deployment without adding the dependencies.

For more information, see "Managing Content Packs and Dependencies in a Project" in the *OO Studio Authoring Guide*.

Content Pack Validation - Duplications

During the deployment process, Central performs a complete path validation for all elements that have a unique path on Central (such as flow, group alias, domain term, system property, system account, and selection list).

If there are any duplications of these elements (either within the deployment unit, or within previously deployed entities), an error message is displayed, listing the duplications.

Content Pack Validation - Structural Integrity

During the deployment process, Central validates the content packs to make sure that the structural integrity of the deployed content is intact.

Checking the structural integrity includes checking that there are no missing references, there are no duplicate UUIDs, the flows in the content packs are correctly formed, and there are no duplicate flow paths or duplicate system property names.

For example:

Deployment After Making Changes to a Content Pack

Each entity (flow, operation, and configuration item) in deployed content packs must have a unique UUID. If you try to deploy a content pack with a duplicated entity (with the same UUID as an entity in another deployed content pack), the deployment will fail.

If you move an entity from one content pack to another, this may result in a duplication, which will cause deployment to fail. For example, you have two content packs, A and B. You deploy content pack A to Central, and then you move entity X from A to B. When you try to deploy content pack B, the deployment will fail because entity X appears in both A and B.

In such a situation, the correct workflow is to deploy **both A and B** together after moving the entity, and not to deploy B on its own. The new version of A, without entity X, overwrites the previous version on Central, so there is no duplication.

When you want to promote content from one Central environment to another, you will only need to deploy all of the latest versions of the content packs that were deployed in the original Central environment.

Best Practices for Content Pack Deployment After Upgrade

After an upgrade, there may be content deployed in previous versions that does not meet the requirements for uniqueness. If this occurs, there is an indication in the **Content Packs** tab, showing that the content pack may not be valid (some of its contents might have moved to a different content pack without a new version of the first content pack being re-deployed).

If you are upgrading a large repository from OO 9.x, the recommended best practice is to:

1. Upgrade the repository to OO 10.x.

2. Deploy it to Central.

3. After the repository has been successfully deployed, split it into multiple content packs in Studio.

4. Deploy those content packs together.

5. If you modify, remove, or move entities in these content packs, deploy all affected content packs again.

What do you want to do?

Deploy a content pack

When you deploy a content pack, OO checks the content pack for a digital signature.

- If a content pack with a digital signature has been altered, this content pack cannot be deployed.

- If the signature has expired, you will need to select the **Ignore expiry of digital signature** check box in order to deploy the content pack.

- If there is no digital signature on the content pack, you can deploy it, but a warning is displayed, because there is no verification that the content pack was created by a known sender and that it was not altered in transit.

- If a content pack is signed, but the signature is not from a known and trusted authority, you can still deploy it.

1. Click the **Content Management** button to display the Content Management workspace.

2. Under the **Content Packs** tab, click the **Deploy New Content** ＋ button.

3. In the Deploy New Content dialog box, click the **Add** ＋ button.

4. Browse to and select one or more content packs, and then click **Open**.

   > **Tip:** You can also drag-and-drop a content pack from your file management system into the Deploy New Content dialog box.

5. During the upload process, OO verifies the selected content packs. Icons next to the content packs indicate if there are any issues with their certification.

   Select a content pack to display its certification details.
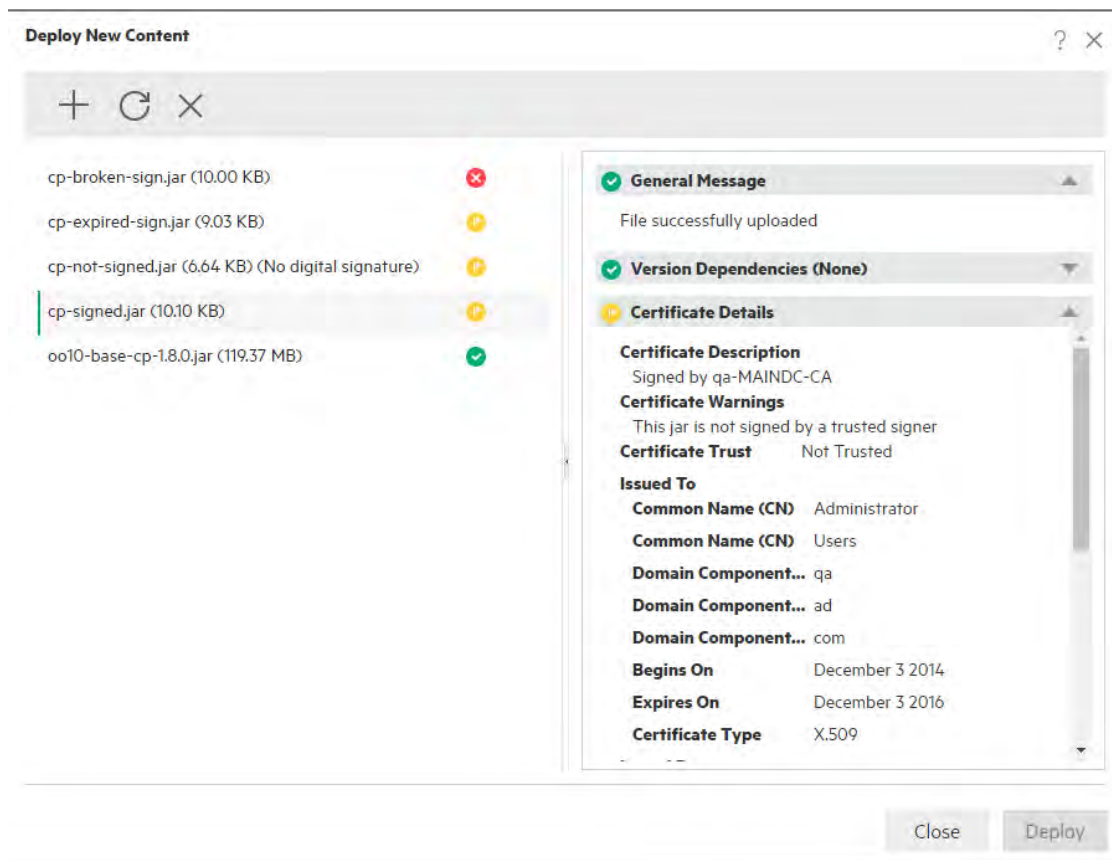
- If a content pack appears with a **Delete** icon ⊗, this content pack cannot be deployed. You need to click the **Delete**

$\times$ button in the toolbar to remove this content pack from the dialog box.

For example, a content pack whose digital signature has been altered cannot be deployed.

- If a content pack appears with a **Warning** icon , this content pack can be deployed, but there is information you need to check about it.

  For example, the signature has expired. In this situation, you need to select the **Ignore expiry of digital signature** check box in order to deploy the content pack.

- If a content pack appears with an **OK** icon , this content pack is okay to be deployed.

- If a content pack has no digital signature, and appear with a **None** icon you can deploy the content pack, but be aware that there is no verification of the content pack.

6. Repeat Steps 3 - 5 to add more content packs to the Deploy New Content dialog box, if desired.

7. Click **Deploy**.

   The deployment may take a few minutes. A progress bar displays the progress of the deployment, showing when each step is completed.

   > **Note:** It is not possible for two users to deploy a content pack to the same Central at the same time. If you try to deploy content to Central while another user is already deploying content, you will see an error message, indicating that another deployment is currently in progress.

   - If deployment was successful, an icon indicating success is displayed in the dialog box and you can click **Close**.

   - If deployment fails, messages are displayed, describing the causes of the failure.

     For example, a content pack will fail deployment if it lacks structural integrity (for example, missing references, duplicate UUIDs, duplicate flow paths, duplicate system property names). For more information, see Content Pack Validation - Structural Integrity.

     If the deployment failed, especially in the case of dependency failure, you can click the **Back** button to return to the **Deployment** page. There, you can verify the dependency data, clear the pane, upload all required content packs, and try again.

8. Click **Close** to close the dialog box.

Roll back to the last deployment

After a content pack has been deployed, it is possible to roll back to the last deployment. This might be useful if you find that there are problems with the flows in a new deployment.

If you roll back after a content pack has been deleted, this will undo the deletion.

1. Click the **Content Management**> **Content Packs** tab.

2. Select a content pack and click the **Roll Back Deployment** button.

3. Click **OK** in the confirmation dialog box.

Delete a deployed content pack

1. Click the **Content Management** > **Content Packs** tab.

2. Select a content pack and click the **Delete** button.

3. Click **Delete** in the confirmation dialog box.

Filter content packs

1. Click the **Content Management** > **Content Packs** tab.

2. In the filter box under **CP Name** or **Deployed By**, enter a content pack name, or part of a name, to filter the displayed content packs.

CP Name		Version		Deployed		Deployed By	

3. To remove the filter, click the **Clear Filters** button.

Display details of a selected content pack

1. Click the **Content Management** > **Content Packs** tab.

2. Select a deployed content pack from the list.

3. Click the **Content** tab in the Details pane, to the right.

4. View the displayed details about the version, publisher, signature, SCM tag, SCM revision (for SVN), description, and the user who deployed it.

5. See how the entities in the content pack are arranged in a tree, under **Library** and **Configuration** folders.

6. Expand and collapse the nodes to see the contents of different folders within the content pack.

7. Clear the check box next to the name of a content pack to remove its details from the Details pane.

Use

Display changes that were made to a selected content pack

1. Click the **Content Management** > **Content Packs** tab.

2. Select a deployed content pack from the list.

3. Click the **Changes** tab in the Details pane, to the right, to see which items have been added, deleted, updated, or moved in the content pack.

   For an item that has been moved, hover the mouse over the item to see where it has been moved from.

4. If desired, you can filter the changes so that only certain types of changes are displayed:

   ○ Under **Type**, select one or more entity type to display only the changes relating to this type of entity. Options are **Flow**, **Operation**, **Configuration Item**, or **All**.

   ○ Under **Path**, enter part or all of the path to display only changes to entities in a path containing this string.

   ○ Under **Change**, select one or more type of change to display only this type of change. Options are **Added**, **Deleted**, **Modified**, **Moved**, or **Content pack moved**.
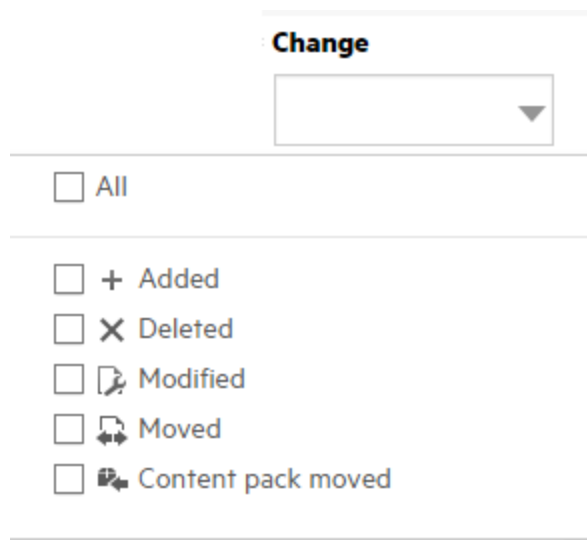
Adjust the display of panes

Use the open/close button in the slider on the edge of the panes to toggle between hiding and displaying the pane.

Drag the slider to adjust the size of a pane.

## View statistics about an individual content pack
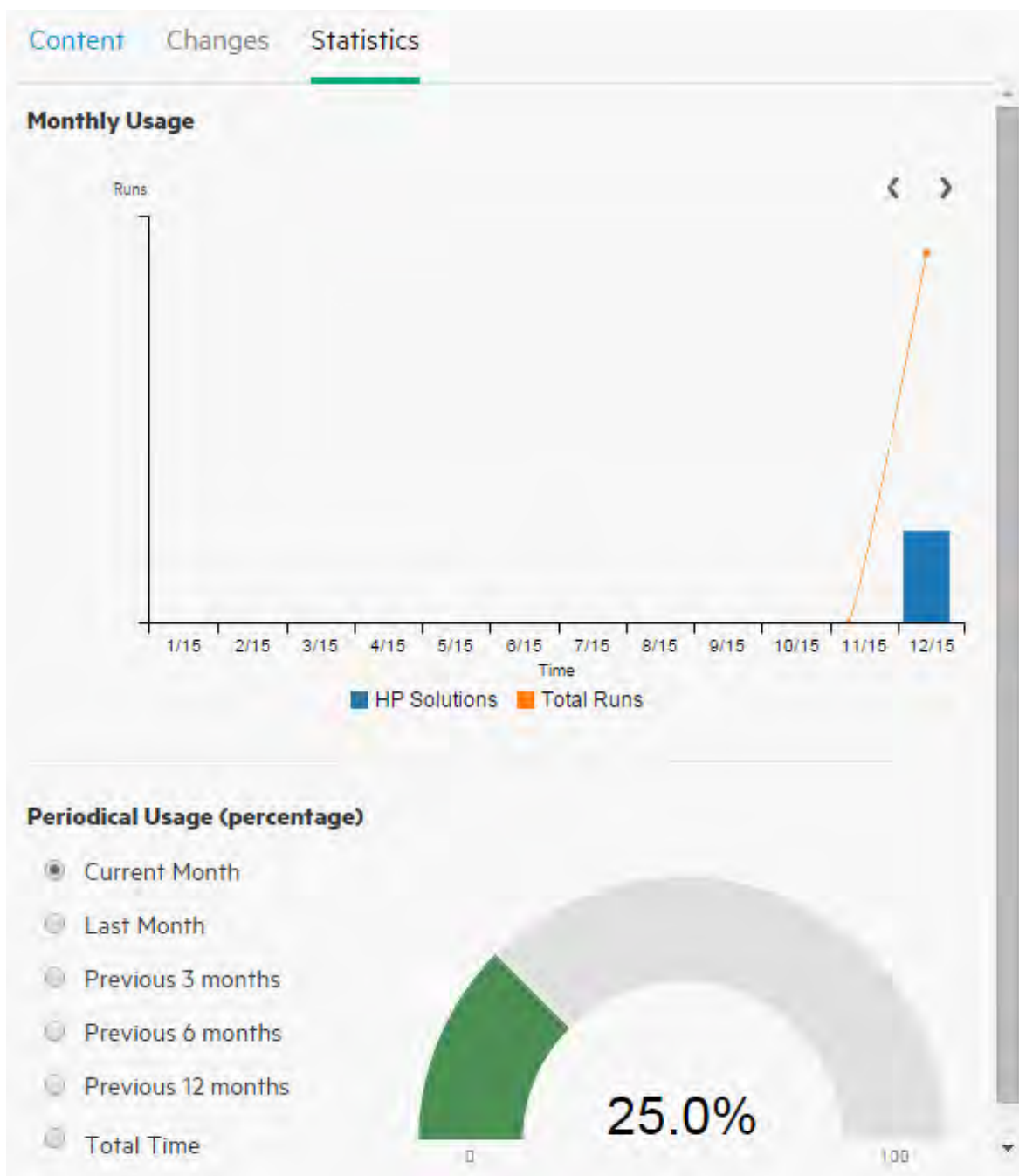
1. Click the **Content Management** > **Content Packs** tab.

2. Select a deployed content pack from the list.

3. Click the **Statistics** tab in the Details pane, to the right.

4. In the **Monthly Usage** section, see how many flows were run were using content from this

deployed content pack for each month since first use, compared with the total runs over the same period.

   a. Roll over a month in the bar chart to see a pop-up with a summary of the total number of runs for that month and the number of runs from the content pack.

   b. Click the arrows 〈 〉 to display more months, if not all are displayed onscreen.

   c. Click the Legend blocks under the bar chart to filter the display.

5. In the **Periodical Usage (percentage)** section, select a time period, to see how this content pack usage fits into the percentage of all runs during the current month, previous month, quarter, half year, year, or in total.

> **Note:** There may be a delay of 10-20 minutes between the time that a flow is run and when it is reflected in the statistics.

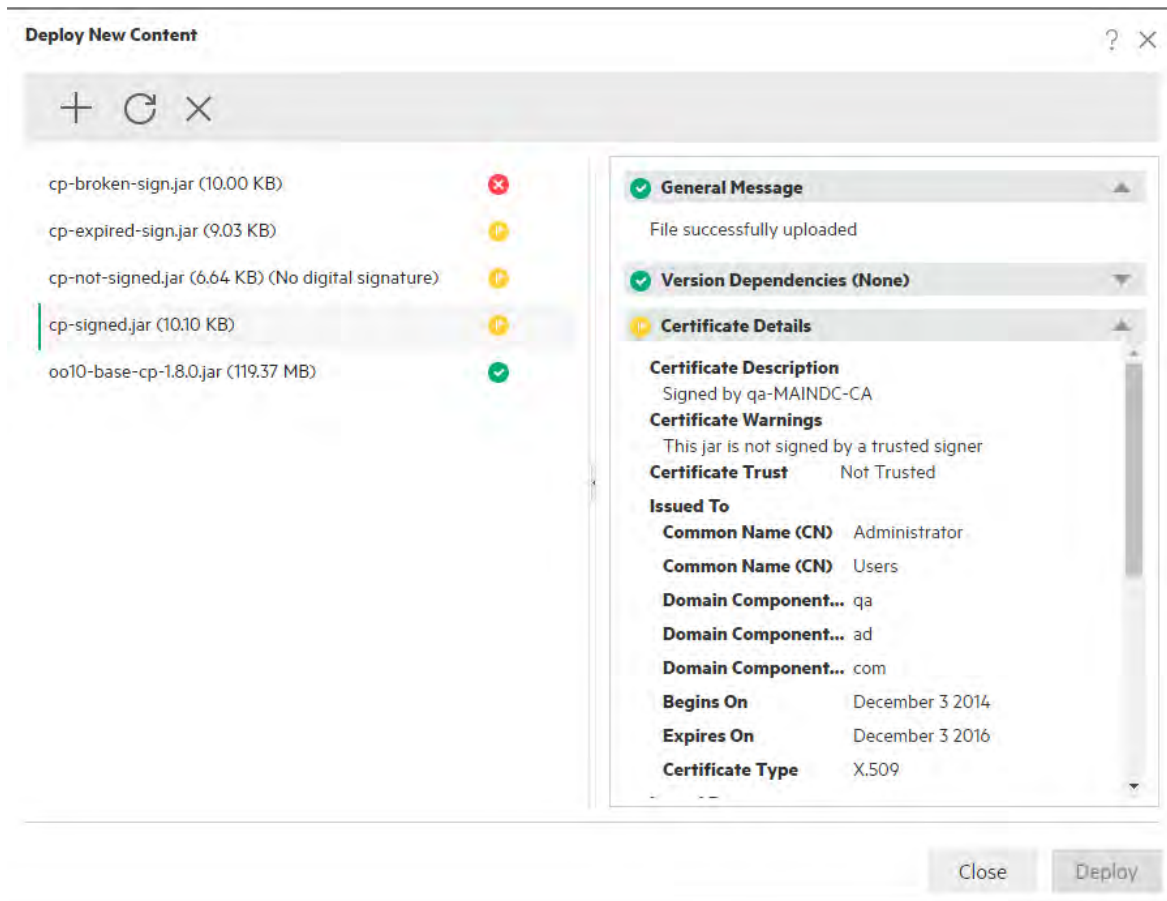## View statistics about all deployed content packs
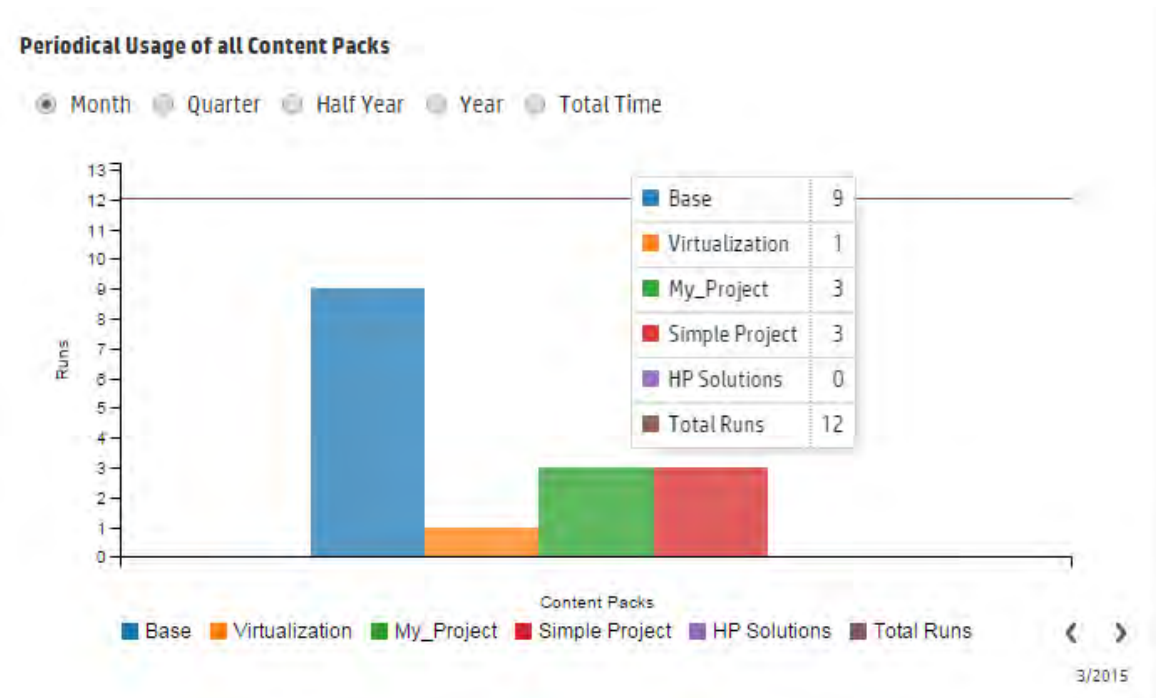
1. Click the **Content Management** > **Content Packs** tab.

2. Click the Statistics ⩘ button in the toolbar to display the Periodical Usage of all Content Packs dialog box.

3. Select a time period, to compare the number of runs of all deployed content packs over that period.

4. Click the Legend blocks under the bar chart to filter the display.

5. Roll over the bar chart to see a pop-up with a table showing the number of runs for each content pack during the selected period and the total number of runs.

6. Click the arrows ❮   ❯ to display more content packs if not all are displayed onscreen.

7. Click anywhere outside the dialog box to close it.

Reference Material

Deploy New Content dialog box

Content Packs tab

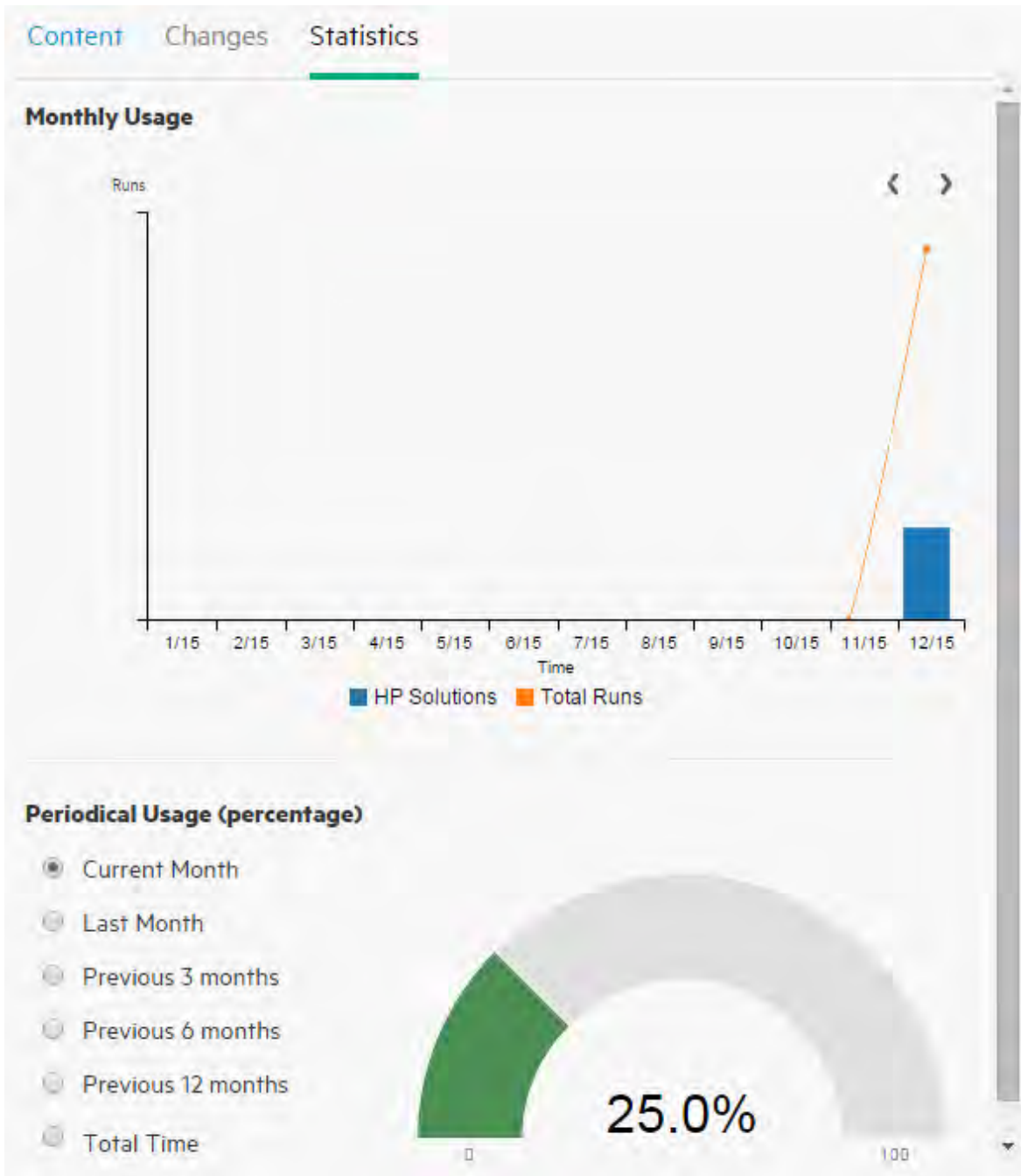| GUI item	Description
Deploy ➕ button	Click the **Deploy** button to open the Deploy New Content dialog box.
Roll Back ↩ button	Click the **Roll Back** button to roll back to the last deployment.
Delete ✕ button	Click to delete the selected content pack.
Clear ▽ Filters button	Click to remove the filters and display all the content packs.
Statistics 📊 button	Click to display and compare statistical information about all deployed content packs.
Select Columns button ▥	Click to display the column picker, to choose which columns to display.
Clear Filters button ▽	Click to remove the filters and display all the content packs.
Filters	Enter a content pack name or the name of the user who deployed it, or part of a name, to filter the displayed content packs.
Version column	Displays the version number of the content pack.
Deployed column	Displays the time and date when the content pack was deployed.

	Note: If the content pack was deployed during the current day, only the time of day is displayed.
Deployed By column	Displays the user who deployed the content pack.
Content tab	Click the **Content** tab to display the details of the selected content pack. This information includes the: • **Version** • **Publisher** • **Certification**- the digital signature of the content pack, verifying that the content pack was created by a known sender and that it was not altered in transit • **Description** • **Deployed By** - the user who deployed the content pack • **SCM Tag** - from the source control management system in Studio • **SCM Revision** - this is available only for a content pack that was created in a Studio that is connected to SVN Under **What's Inside**, the entities inside the content pack are arranged in a tree, under **Library** and **Configuration** folders, reflecting the way that the content pack appeared in Studio. You can drill down into the folders and see exactly which items exist in the content pack.
Changes tab	Click the **Changes** tab to display the changes that have been made to the selected content pack. This information tells you which items have been added, deleted, updated, or moved. For CloudSlang Content Packs, you can also view Descriptions. For an item that has been moved, hover the mouse over the item to see where it has been moved from.
Statistics tab	Click the **Statistics** tab to display statistical information about the selected content pack.
Open/Close button	Use the **Open/Close** button in the slider on the left edge of the Details pane to toggle between hiding and displaying the pane.

Content Packs tab > Changes Pane

GUI item	Description
Type	Select one or more entity type to display only the changes relating to this type of entity. Options are **Flow**, **Operation**, **Configuration Item**, or **All**.
Path	Enter part or all of the path to display only changes to entities in a path containing this string.
Change	Select one or more type of change to display only changes of this type. Options are **Added**, **Deleted**, **Modified**, **Moved**, or **Content pack moved**.

Content Packs tab > Statistics Pane

GUI item	Description
Monthly Usage bar chart	See how many flows were run were using content from this deployed content pack for each month since first use, compared with the total runs over the same period. Roll over a month in the bar chart to see a pop-up with a summary of the

	total number of runs for that month and the number of runs from the content pack.
Legend blocks	Click to filter the display.
Arrow buttons ❮ ❯	Click to display more months, if not all are displayed onscreen.
Periodical Usage (percentage)	Select a time period, to see how this content pack usage fits into the percentage of all runs during the current month, previous month, quarter, half year, year, or in total.

Periodical Usage of all Content Packs dialog box
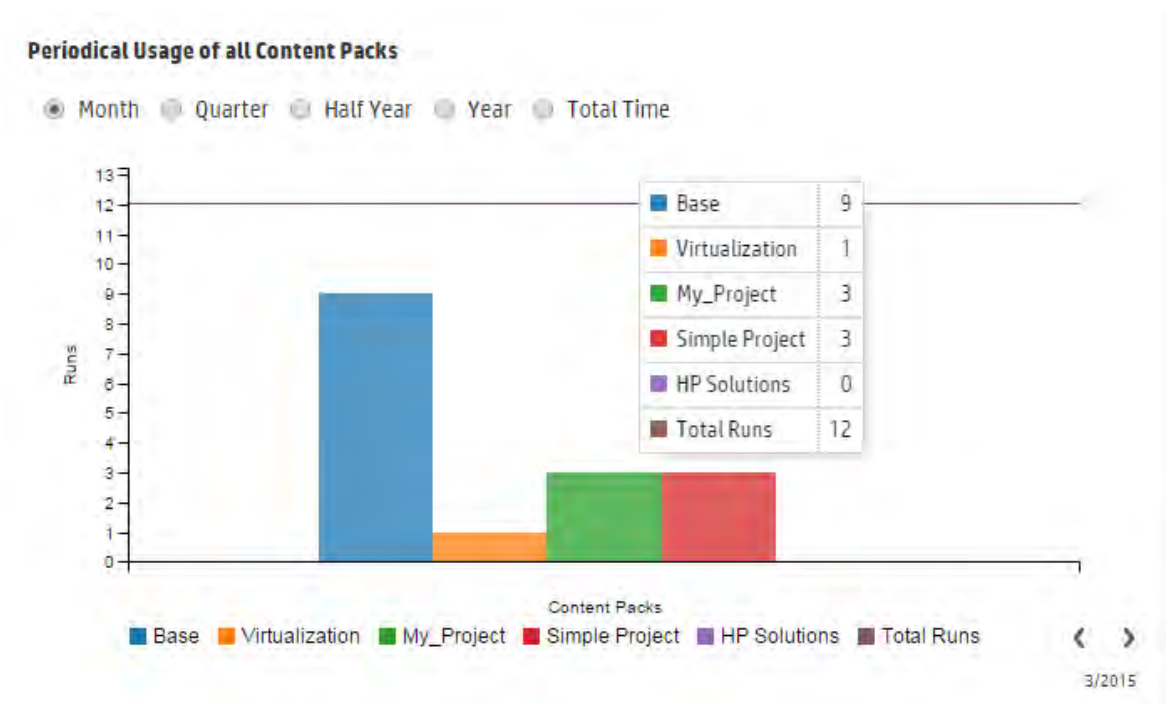
GUI item	Description
Bar chart	Compare the number of flows that were run from the different deployed content packs over the selected period. Roll over the bar chart to see a pop-up with a table showing the number of runs for each content pack during the selected period and the total number of runs.
Periodical Usage options	Select a time period, to compare the number of runs of all deployed content packs over that period.
Legend blocks	Click to filter the display.

Arrow buttons ‹ ›	>Click to display more content packs if not all are displayed onscreen.

Managing the Flow Library

After a content pack has been deployed, you can see the flows inside it in the Flow Library. From here, you can browse or filter flows, to find the one that you need. You can view the flow meta data, view reports about the last time that a flow was run, configure persistence level and run timeout, and set the content permissions.

> **Note:** You will only be able to edit the settings here if you have a role with the **Manage Content Settings** permission.

To access the Flow Library, click **Content Management** > **Flow Library** tab.

> **Note:** The icons of CloudSlang flows and Native OO flows are different.

When you select a flow in the flow library, information about that flow is displayed in the information pane to the right.

This information includes a description of the flow, if one exists, the path to the location where the flow is stored, the flow version, UUID, and ROI.

Configuring the Flow Run Settings for Flows

Flow Run Settings

Persistence Level: Taken from system default (currently Standard)   ▾

Timeout (in minutes):   Custom   ▾   25   Save

Taken from system default (currently, no timeout)
No timeout
Custom

Flow run settings let you control the persistence level and timeout for the selected flow.

- **Persistence Level** defines the level of detail that is saved to the Run Log for the flow. You can choose an extended or standard level, or use the system default. For more information, see Set up the Run Log persistence level for a flow.

- **Timeout** defines how long (in minutes) a flow can run before it is canceled and shows the status **Canceled-Timeout Expired**. You can set up a custom timeout setting for the flow, use the system

default, or set this flow to have no timeout. For more information, see Set up the timeout for a flow.

- The timeout setting applies to both triggered and scheduled flow runs.

- If you change the timeout value of a flow, this change is applied to new flow runs, but not to those that are already running.

- The timeout setting applied to a flow overrides the system default and is applied even if the system timeout is not enabled.

> **Note:** If a timeout setting was defined for an operation inside a flow, the timeout assigned for the flow is not affected. The flow timeout is triggered according to its value, regardless the values of the timeout values of the steps inside the flow.

Setting Permissions for Content

Content permissions let you restrict access to data (flows and folders) according to the roles that have been assigned to users. This should be done as part of the promotion of a content pack.

You can entitle a role to have either view permission or run permission, or both, for a flow or folder. For example, you might want to give users with the role **Promoter** permission to view and run the contents of a selected folder, and give users with the role **End User** permission to view the content only.

> **Note:** To modify the content permissions, you must be assigned a role with **Manage Content Settings** permission.

If you set the content permissions of a folder and apply the change to all children, these permissions will apply to all the flows within that folder. It is also possible to set the permissions of the entire library.

By default, only roles with permissions are displayed. However, you can select the **Show all roles** check box to display all the roles, including those that do not have permissions for the content.

> **Note:** Content permission is not the same as restricting the actions that OO users are allowed to perform in the system. This is done via Role-Based Access Control (RBAC). For more information about RBAC, see "Setting Up Security – Roles" on page 66

What do you want to do?

Display flow information

1. Click the **Content Management** button to display the Content Management workspace.

2. Click the **Flow Library** tab.

3. Expand the folders to look for the flow that you need.

4. (Optional) Enter part or all of the flow path in the **Filter By** text box to filter the flows.

   For example, if you enter "network", only flows that contain the word "network" or which are located in a path with a folder containing the word "network" are displayed.

   > **Note:** The search is not case-sensitive.

5. Select a flow.

6. View the flow information, displayed in the information pane to the right.

7. If required, use the scroll bar to scroll down and view the contents of the **Description** section.

8. If required, use the sliders at the edges of the information pane to adjust the width and height of the pane.

9. (Optional) Click the **Open Graph** button to display a graphical representation of the flow.

   For more information, see .

Display information about the last time a flow was run

You can view the details about what happened the last time a flow was run.

1. Click the **Content Management** button to display the Content Management workspace.

2. Click the **Flow Library** tab.

3. Select a flow in the flow library.

4. Click the **Last Run Report** button. The Drill Down window opens, showing what happened

the last time this flow was run.

The **Last Run Report** button is only available for flows that have been run at least once in the current session.

The information in the Drill Down window is the same as the information in the Run Explorer drill down view. For more details about this view, see "Testing and Troubleshooting a Flow Run" on page 256.

Set up content permission for a flow or folder

1. Select a flow or folder in the flow library.

2. In the **Permissions** section, click the **Edit** button

3. In the Edit Permissions dialog box, select the **View** and **Run** check boxes to assign permissions for the selected role.

- ◦ Select **View** to enable users with the selected role to view this flow or the contents of this folder in Central.

- ◦ Select **Run** to enable users with the selected role to run this flow or the contents of this folder in Central.

4. Click **Save**.

Set up the Run Log persistence level for a flow

By selecting the persistence level, you can control the level of detail that is saved to the Run Log for an individual flow.

1. Select a flow in the flow library.

2. From the **Persistence Level** list, select the persistence level for the flow. The options are:

   - ◦ **Standard** - Includes raw results, step results, and big inputs/outputs.

   - ◦ **Extended** - Includes the complete raw results, step results, subflow outputs (from step results), and big inputs/outputs.

   - ◦ **Taken from system default** - whatever was selected as the default persistence level. For information about setting the default persistence level, see "Monitoring and Controlling Database Size" on page 144.

     Note: It is possible to override a flow's persistence level, when running or scheduling a flow.

Set up the timeout for a flow

When you assign a timeout value to a flow, if the flow run exceeds the specified duration, it will be canceled and will show the status **Canceled-Timeout Expired**.

1. Select a flow in the flow library.

2. From the **Timeout (minutes)** list, select the timeout setting for the flow. The options are:

   - **Taken from system default** - whatever was selected as the default timeout setting for the system is applied for this flow. For information about setting the default timeout, see "Setting Up the System Settings - General Settings" on page 134.

   - **No timeout** - no timeout will be applied to this flow, even if a default timeout has been set for the system.

   - **Custom** - lets you specify a timeout for this flow.

3. If you selected **Custom**, enter a number larger than 0, to set the number of minutes for the timeout setting.

4. Click **Save**.

Reference Material

Flow Library pane

GUI item	Description

Filter By	To locate the flow that you need, enter part or all of the flow path in the **Filter By** text box to filter the flows.
Run button	Click to run the selected flow.
Schedule button	Click to open the Schedule Flow dialog box.
Last Run Report button	Displays the details about what happened the last time that the selected flow was run. The **Last Run Report** button is only available for flows that have been run at least once.
Open Graph Button	Click the **Open Graph** button to display a graphical representation of the flow. For more information, see "Viewing a Flow Graph" on page 189.
ID	Displays the ID of the selected flow. • Native OO flows IDs have a UUID format. The UUID is unique. • CloudSlang flows, IDs have the `<namespace>.<name>` format. **Note:** The recommended best practice is that the namespace is part of the path and that the name is the filename, but this is not required. The name and namespace are defined inside the yaml file. The id (<namespace>.<name>) must be unique in the system. The filters in the Flow Library use the flow path. So if you don't use the best practice, when you use the filter, using flow name/namespace will not work. For example, in the element `user.flows.hello_world`, the namespace is the folder path **user/flows** and the element name is **hello_world**.
Path	Displays the path to the location of the selected flow.
Content Pack	Displays the content pack that the selected flow came from.
Version	Displays the version number of the selected flow.
Run Link	You can embed parts of the Central UI into an external application. For more information, see "Embedding Central Views in External Web Pages" on page 275.
Description	Displays a description of the selected flow. If required, use the scrollbar to scroll down and view the contents of the **Description** section.

Input	Displays the input information for CloudSlang flows only.
	Note: This field may be encrypted as configured by the flow creator.
Output	Displays the output information for CloudSlang flows only.
	Note: This field may be encrypted as configured by the flow creator.
Persistence Level	From the **Persistence Level** list, select the persistence level for the flow. The options are:
	• **Standard** - Includes raw results, step results, and big inputs/outputs.
	• **Extended** - Includes the complete raw results, step results, subflow outputs (from step results), and big inputs/outputs.
	• **Taken from system default** - whatever was selected as the default persistence level. For information about setting the default persistence level, see "Monitoring and Controlling Database Size" on page 144.
	Note: It is possible to override a flow's persistence level, when running or scheduling a flow.
Timeout (minutes)	Enter the timeout setting, so that a flow run will be canceled if it exceeds this duration. The options are:
	• **Taken from system default** - whatever was selected as the default timeout setting for the system is applied for this flow. For information about setting the default timeout, see "Setting Up the System Settings - General Settings" on page 134.
	• **No timeout** - no timeout will be applied to this flow, even if a default timeout has been set for the system.
	• **Custom** - lets you specify a timeout for this flow. Enter a number larger than 0, to set the number of minutes for the timeout setting.
	Note: A custom timeout setting overrides the system default timeout setting and is applied even if the system timeout is not enabled.
Permissions	Displays the permissions that have been set for this flow or folder, for each role that has been defined in the system:
	• Displays **View** if users with the adjacent role have permission to view this flow (or the contents of this folder) in Central.
	• Displays **Run** if users with the adjacent role have permission to run this flow (or the contents of this folder) in Central.

	The display of roles in this section corresponds to the roles that have been defined in the system.
Edit	Click to open the Edit Permissions dialog box for the selected flow or folder.
Show all roles	By default, only roles with permissions are displayed. However, you can select the **Show all roles** check box to display all the roles, including those that do not have permissions for the content.

Edit Permissions dialog box
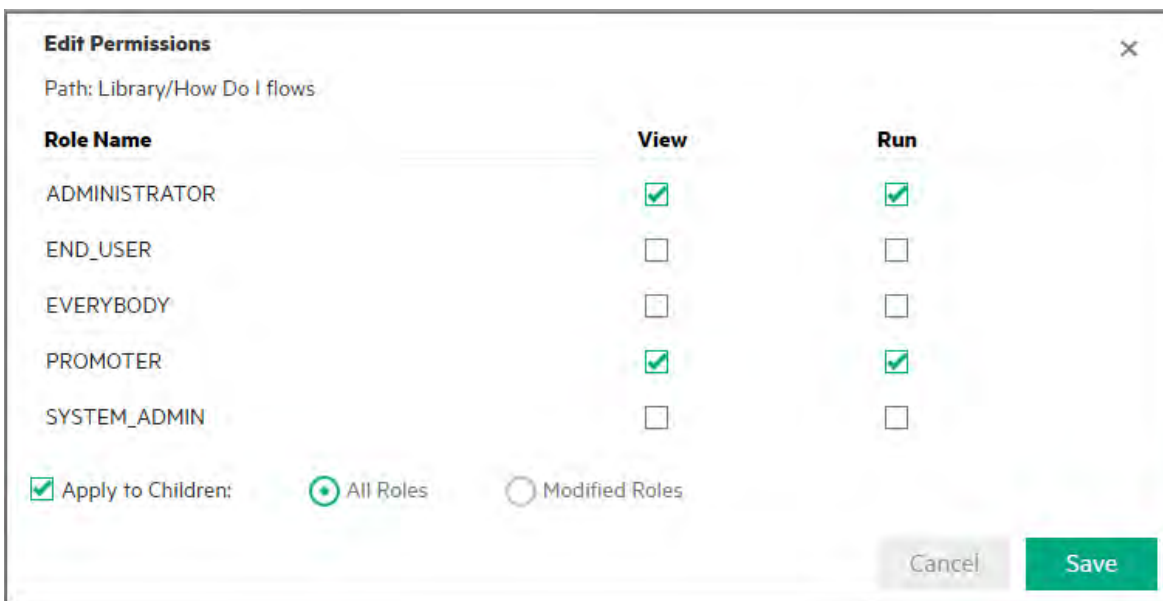
| GUI item	Description
View	Select **View** to enable users with the selected role to view this flow (or the contents of this folder) in Central.
Run	Select **Run** to enable users with the selected role to run this flow (or the contents of this folder) in Central.
Apply to children	If you have selected a folder, select the **Apply to Children** check box to apply the updated permissions to all the flows in the folder.
All Roles/Modified Roles	Select whether to apply the changes to all the roles in the folder or to modified roles only.
Save	Click to save the changes to the permissions.

Viewing a Flow Graph

From the Flow Library or Flow Launcher, you can click the **Open Graph** button to display a flow graph representing the flow. This is the same graph that existed in Studio, when the flow was created.

> **Note:** This feature is not supported when the flow was written in CloudSlang. A "not supported" message is shown when you click the **Open Graph** button.

Why View a Flow Graph?

- In the promotion phase, the Op Admin can view the flow graph in order to better understand what the flow does and which steps and outputs to expect.

- In the execution phase, the End User can use the flow graph to get a clear picture of what the flow does and to verify that this is the flow they need.

- In the troubleshooting phase, the End User and Op Admin can use the flow graph to understand exactly which series of steps caused a flow to fail.
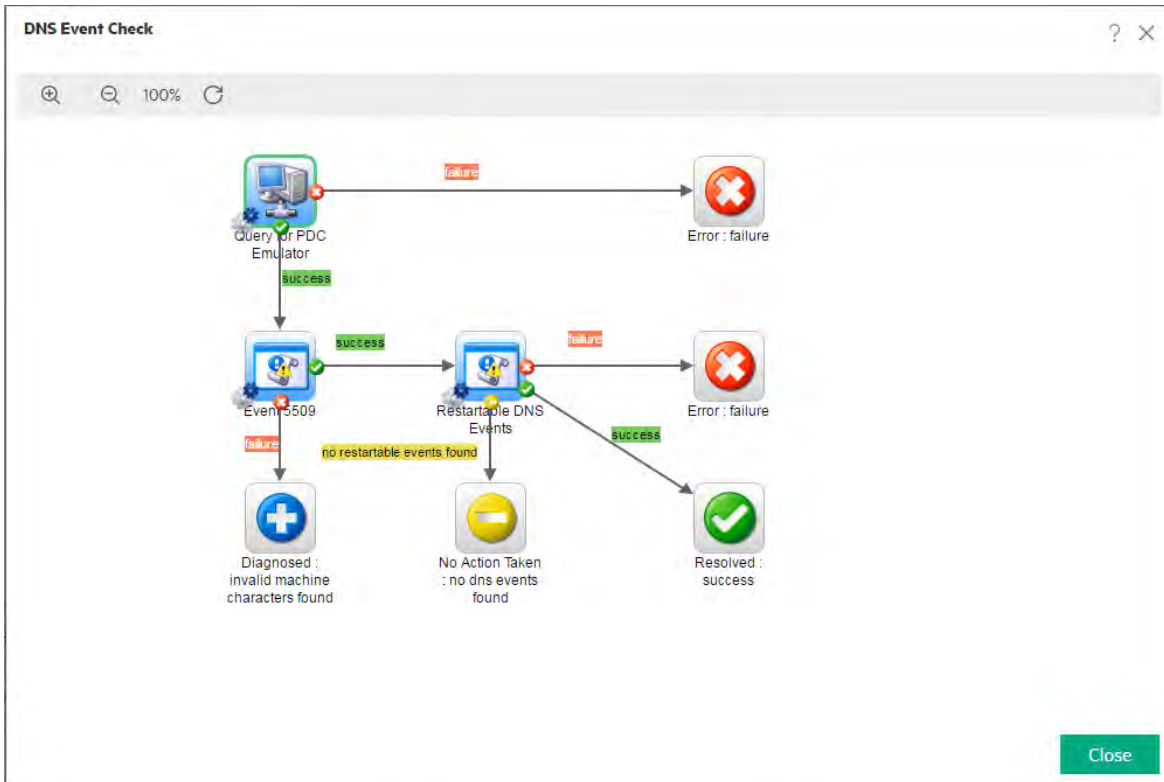
What do you want to do?

Display a flow graph

1. Open either the **Content Management** > **Flow Library** or the **Run Management** > **Flow Launcher**.

2. Select the flow that you want to view.

3. Click the **Open Graph** button to display a flow graph representing the flow.

Adjust the zoom of a flow graph

When you display a flow graph, the zoom level is automatically adjusted so that the entire flow fits into the window. However, you can increase or decrease this zoom level.

1. Display the flow graph for the selected flow.

2. Adjust the zoom levels:

- ○ Click the **Zoom In** ⊕ button to increase the zoom.

- ○ Click the **Zoom Out** ⊖ button to decrease the zoom.

- ○ Click the **Reset** ⟳ button to reset the view to the default size and position.
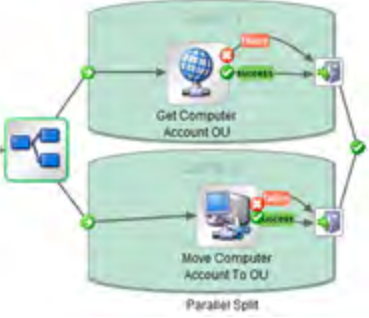
Pan across a flow graph

1. Display the flow graph for the selected flow.

2. Roll your mouse over the flow graph so that the cursor changes to a hand icon.

3. Press down on the cursor and drag across the graph to move the display to a specific location on the graph.

Reference Material

Flow graph elements

GUI item	Description
Operation Operation1	An operation has a gray background.
Responses Generate Report From	Operations have predefined response icons, for "error", "resolved", "diagnosed" and "no action".
Non-blocking Local Ping	A non-blocking step does not block the rest of the flow. While it is running, the flow run continues to carry out the steps that come after it. A non-blocking step appears with an orange lightning bolt.
Subflow Test Computer Account	A subflow is a flow within a flow. A subflow appears with a blue background.
Response steps	Response steps are end points for a flow.

Transitions	Transitions between steps have labels, color-coded using the same colors as the response steps.
Gated transition	A gated transition restricts access to the next step to users who have been assigned a particular role. Gated transitions appear with a red arrow.
Handoff transition	A handoff transition is one where a message appears, suggesting that the current user hand off the flow run to another user. Handoff transitions appear with a hollow arrow head.
Multi-instance step	A multi-instance step can run multiple times, with multiple targets. A multi-instance step resides within a multi-instance branch.
Parallel Split Step	A parallel split step is a set of step sequences that are carried out simultaneously. Each series of steps is represented visually in the flow diagram as a lane.

Flow graph toolbar
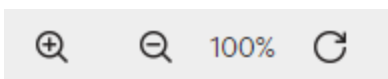
GUI item	Description
Zoom In ⊕	Click the **Zoom In** button to increase the zoom.
Zoom Out ⊖	Click the **Zoom Out** button to decrease the zoom.
Reset ↻ **button**	Click to reset the view to the default size and position.

Setting Up Configuration Items for a Content Pack

After a content pack has been deployed to a Central server, the Op Admin may need to configure the contents of this content pack, in order to adjust it to the environment. This includes setting up the:

- **System accounts** - In many cases, content packs come with system accounts configured. For example, if you are deploying a content pack for an Oracle database, it will include Oracle user accounts. You will need to create Central user names and map them to the system accounts in the content pack.

- **System properties** - In many cases, content packs come with system properties configured. You may want to override these properties in Central.

  Note: If a system account or system property was created via API, it is not possible to edit it in the Central UI. You can only edit it via API.

  In the Central UI, you can edit system accounts and system properties that were created in a content pack in Studio.

- **Group aliases** - If the content pack includes operations that have been assigned to group aliases, rather than actual worker groups, you will need to map these group aliases to actual worker groups. For more information about group aliases, see "Worker Groups and Group Aliases" in the *Concepts Guide*.

  Note: If a group name and alias name are the same, these are mapped automatically by default.

To access the configuration items, click the **Content Management** > **Configuration Items** tab.

Note: The **Configuration Items** tab is only visible if you have been assigned a role with the **Manage Configuration Items** or **View Configuration Items** permission. You will only be able to edit the items if you have a role with the **Manage Configuration Items** permission.

Assigning Permissions to a System Account

You can assign permissions for system accounts. This enables the administrator to control which users can view which system accounts and run (or resume) flows that use them. This feature is useful for customers with multiple organizations, who may wish to hide some of the system accounts from some users.

- Only users whose roles have **View and Run** permission on a system account will be able to view it in the Configuration Items tree.

- Only users whose roles have **View and Run** permission on a system account will be able to run a flow that uses this system account at the flow level. If a user without permissions tries to run such a flow, it will fail with a "Failed to Complete" status.

- Only users with **View and Run** permission on a system account will be able to remote debug a flow that uses it, from Studio.

You can assign permissions for multiple system accounts at once, by assigning permissions to folders.

If you do not specify the permissions, the system account inherits the permissions from its nearest parent.

When you move a system account, if it has permissions configured, these permission remain configured. If not, the system account inherits the permissions from its nearest parent in the new location.

Viewing Information About Configuration Items

The configuration items are displayed in a tree, with icons representing the different types of item.

Next to the Configuration Items tree, two columns display the values of the items:

- The **Deployed Value** column displays the original value of the configuration item from a deployed content pack.

- The **Central Value** column displays the new value, if the original value was overridden in Central.

When you select an item in the Configuration Items tree, information about that item is displayed in the information pane to the right. This information includes the ID of the item, the path to the location where the item is stored in the content pack, a description, information about the content pack, and so on.

For system accounts, permissions are also displayed. By default, only roles with permissions are displayed. However, you can select the **Show all roles** check box to display all the roles, including those that do not have permissions for the account.

Deployed Values and Central Values

If you override the value of a configuration item in Central, the **Deployed** value and the **Central** value are connected. If you move or rename a configuration item in Studio and deploy it again, the **Central** value is also moved or renamed.

Path Uniqueness for Configuration Items

Configuration items must be unique. If you deploy a configuration item in a folder containing a configuration item with the same name, it will either overwrite that item (if they have the same UUID) or fail deployment (if it has a different UUID).

Note: However, it is possible to have two configuration items with the same name if they are located in different folders.

There are limitations to the way that Central enforces path uniqueness:

- Unlike Studio, Central enforces path uniqueness for certain item types only – namely, flows and the specific configuration item types that are supported by the REST API **/v1/config-items**.

- Unlike in Studio, paths are case-sensitive in Central. This means that Central may not detect a conflict between "my item" and "My Item", for example.

System Accounts in Resumed Flows

When you resume a flow, OO fetches the system account configuration from the current system. So if a system account was changed between the time the flow was started and the time it was resumed, OO takes the new system account value.

What do you want to do?

Display configuration items in a content pack

1. Click the **Content Management** button to display the Content Management workspace.

2. Select **Configuration Items**.

3. In the Configuration Items tree, expand the folders to look for the item that you need.

4. (Optional) Enter part or all of the configuration item path in the **Filter By** text box to filter the items.

   For example, if you enter "network", only items that contain the word "network" or which are located in a path with a folder containing the word "network" are displayed.

   > **Note:** The search is not case-sensitive.

5. Select a configuration item or folder and view the information displayed in the pane to the right.

6. If required, use the scroll bar to scroll down and view the contents of the **Description** section.

7. If required, use the sliders at the edges of the information pane to adjust the width and height of the pane.

8. In the case of a system account, you can also view the roles that are assigned **View and Run** permission for the selected folder or account. By default, only roles with permissions are displayed. However, you can select the **Show all roles** check box to display all the roles, including those that do not have permissions for the account.

   > **Note:** To add **View and Run** permission for a system account or folder, click the **Edit** button.

Add configuration items to a content pack

It is not possible to add configuration items to a content pack in Central. These should be created in Studio and deployed to Central.

Edit a system account in a content pack

If a system account was created via API, it is not possible to edit it in the Central UI. You can only edit it via API.

In the Central UI, you can edit system accounts that were created in a Studio content pack.

1. Click the **Content Management** button to display the Content Management workspace.

2. Select **Configuration Items**.

3. Select the system account that you want to edit and click the **Edit** button.

   The Edit System Account dialog box opens. This dialog box will vary depending on the permissions of the currently logged in user. For example, if the logged in user is not entitled to assign permissions, the **Permissions** section is not shown in the dialog box.

Edit System Account

Path: Configuration/System Accounts/System_Account_Name.xml

Deployed User Name: My_username

User Name: New_username

Password:

Confirm Password:

Permissions:

Role Name	View and Run
ADMINISTRATOR	☑
END_USER	☐
EVERYBODY	☐
PROMOTER	☑
SYSTEM_ADMIN	☐

Cancel   Save

The **Deployed User Name** box displays the system account user name as it appears in the content pack, as defined in Studio.

4. In the **User Name** box, enter the Central value for the system account user name. This user name will be mapped to the system account from the content pack, overriding the deployed user name.

> **Note:** The user name does not have to be identical to the deployed user name.

5. In the **Password** box, enter a password for the user.

> **Note:** If the **Include system account passwords** check box was selected when this content pack was created in Studio, the password is automatically loaded into the **Password**

> box.

6. Enter the password a second time in the **Confirm Password** box.

7. In the **Permissions** section, select the check box under **View and Run** for all the roles for which you want to assign permission for this system account.

   Only users with **View and Run** permission on a system account will be able to view it in the Configuration Items tree or to run or resume flows and operations that use this system account.

8. Click **Save**.

Edit permissions for a system account folder

You can assign permissions to a system account folder, which assigns those permissions to all the system accounts in that folder.

1. Click the **Content Management** button to display the Content Management workspace.

2. Select **Configuration Items**.

3. Select a folder in the Configuration Items tree and click the **Edit** button.

4. In the **Permissions** section, select the check box under **View and Run** for all the roles for which you want to assign permission for the system accounts in this folder.

   Only users with **View and Run** permission on these system accounts will be able to view them in the Configuration Items tree or to run flows and operations that use these system accounts.

Edit System Account   ❓ ✕

Path: Configuration/System Accounts

Permissions:

Role Name	View and Run
ADMINISTRATOR	☑
END_USER	☐
EVERYBODY	☐
PROMOTER	☑
SYSTEM_ADMIN	☐

☑ Apply to Children: ● All Roles ○ Modified Roles   [Cancel] [Save]

5. Select the **Apply to Children** check box to apply these changes to all the system accounts in the folder.

   ○ Select **All Roles** to apply the changes to all roles, including those that have not been modified.

   ○ Select **Modified Roles** to apply the changes only to roles that have been modified.

6. Click **Save**.
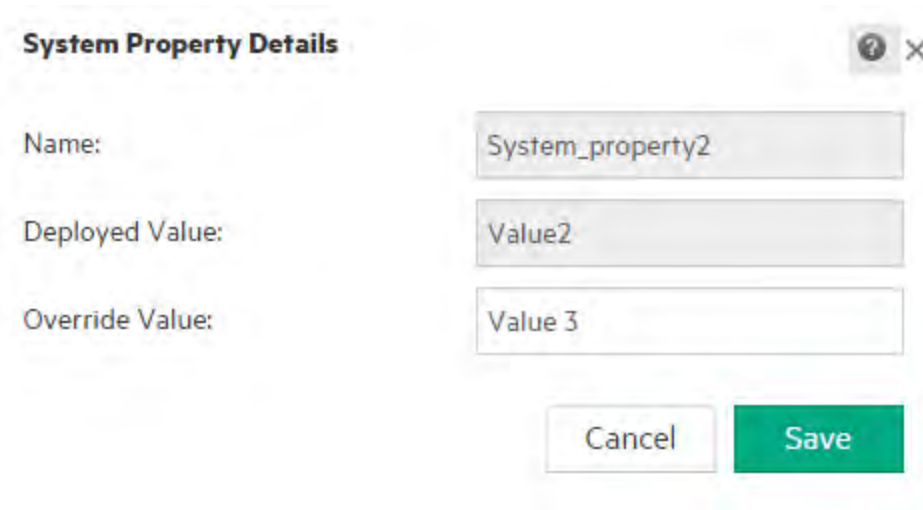
Edit a system property in a content pack

If a system property was created via API, it is not possible to edit it in the Central UI. You can only edit it via API.

In the Central UI, you can edit system properties that were created in a Studio content pack.

1. Click the **Content Management** button to display the Content Management workspace.

2. Select **Configuration Items**.

3. Select the system property that you want to edit and click the **Edit** ✏ button.

   The System Property Details dialog box opens.
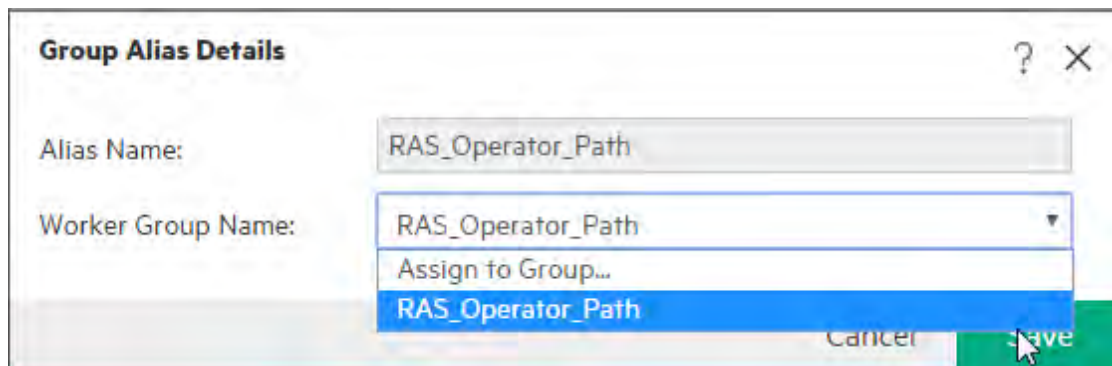
4. In the **Override Value** box, enter a new value for the system property.

5. Click **Save**.

Assign a worker group to a worker group alias

1. Click the **Content Management** button to display the Content Management workspace.

2. Select **Configuration Items**.

3. Select the worker group alias that you want to edit and click the **Edit** button.

   The Group Alias Details dialog box opens.

   The **Alias Name** box displays the name of the alias that was used in the operations.

4. From the **Worker Group Name** list, select the worker group to which you want to map this group alias.

5. Click **Save**.

> **Note:** If you want to map the group alias to a group that does not exist yet, go to **System Configuration** > **Topology** > **Workers** and create a new group, and then return to this task to do the mapping. For more information about creating worker groups, see "Setting Up Topology – Workers and RASes" on page 114.

Revert configuration items to their original value

If someone has changed the value of a configuration item that was in the content pack at the time of deployment, you can revert it to the original value.

1. Select the item and click the **Revert** button.

2. Click **Yes** in the confirmation dialog box.

Delete configuration items in a content pack

It is possible to delete configuration items that have a **Central** value but not a **Deployed** value. For example, if configuration items were deployed prior to OO 10.20, and then OO was upgraded, these will have a **Central** value and no **Deployed** value.

It is not possible to delete a configuration item that is part of a content pack that is currently deployed to Central.

1. Click the **Content Management** button to display the Content Management workspace.

2. Select **Configuration Items**.

3. Select the configuration item that you want to delete and click the **Delete** button.

4. Click **Yes** in the confirmation dialog box.

Reference Material

Configuration Items Tree

GUI item	Description
Name	Displays the name of the configuration item as it appears in the content pack.
Deployed Value	Displays the original value as deployed to Central.
Central Value	Displays the new value, which overrides the original value.
Edit button	Depending on which configuration item is selected, opens the relevant dialog box, so that you can edit the configuration item.
Revert button	Reverts the selected configuration item to its original deployed value, which was overridden in Central.
Delete button	Deletes the selected configuration item. It is possible to delete configuration items that have a **Central** value but not a **Deployed** value. For example, configuration items that were deployed prior to version 10.02, and OO was upgraded. It is not possible to delete a configuration item that is part of a content pack that is deployed to Central.

Edit System Account dialog box

The Edit System Account dialog box will vary depending on the permissions of the currently logged in user. For example, if the logged in user is not entitled to assign permissions, the **Permissions** section is not shown in the dialog box.

GUI item	Description
Path	Displays the path where the system account is located.
Deployed User Name	Displays the user name of the system account, as defined in the content pack.
User Name	Enter the Central value for the system account user name. This user name

	will be mapped to the system account from the content pack, overriding the deployed user name.
Password	Enter the password for the user name. **Note:** If the **Include system account passwords** check box was selected when this content pack was created in Studio, the password is automatically loaded into the **Password** box. You can change this value.
Confirm Password	Enter the password a second time, for confirmation.
View and Run	Select the check box under **View and Run** for all the roles for which you want to assign permission for this system account. Only users with **View and Run** permission on a system account will be able to view it in the Configuration Items tree or to run flows and operations that use this system account.

Edit System Account dialog box - Folder

Edit System Account   ? ✕

Path: Configuration/System Accounts

Permissions:

Role Name	View and Run
ADMINISTRATOR	☑
END_USER	☐
EVERYBODY	☐
PROMOTER	☑
SYSTEM_ADMIN	☐

☑ Apply to Children: ⦿ All Roles ◯ Modified Roles   Cancel   Save

GUI item	Description

View and Run	Select the check box under **View and Run** for all the roles for which you want to assign permission for the system accounts in this folder. Only users with **View and Run** permission on these system accounts will be able to view them in the Configuration Items tree or to run flows and operations that use these system accounts.
Apply to Children: All Roles	Select the **Apply to Children** check box and the **All Roles** option to apply these changes to all the system accounts in the folder, for all roles, including those that have not been modified.
Apply to Children: Modified Roles	Select the **Apply to Children** check box and the **Modified Roles** option to apply these changes to all the system accounts in the folder, but only for roles that have been modified.

System Property Details dialog box

GUI item	Description
Name	Displays the name of the system property, as defined in the content pack.
Deployed Value	If the system property had an original value in the deployed content pack, displays that value.
Override Value	Enter a value for the system property. If the system property had an original value in the deployed content pack, this value will override it.

Group Alias Details dialog box
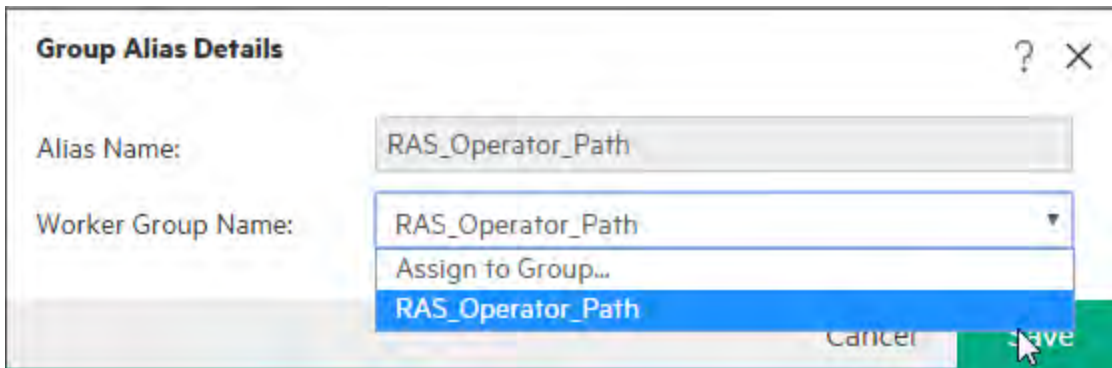
GUI item	Description
Alias Name	Displays the name of the group alias as it appears in the content pack.
Worker Group Name	Select the worker group to which you want to map this group alias.

Testing and Troubleshooting a Content Pack

In this step, the Op Admin tests and troubleshoots the flows in the Staging server.

What do you want to do?

Test a content pack on the Staging server

1. Click the **Content Management** button to display the Content Management workspace.

2. Click the **Flow Library** tab.

3. Select the flow that you want to test, and click the **Run** button.

4. Monitor the progress of the run, and check whether it runs as expected.

5. To further test the flow, run it again, with the same or different parameters.

Running and Monitoring Flows

The Op Admin and the End User are responsible for running and monitoring flows.

Running and Monitoring a Flow – Overview

Once the content packs have been deployed, the Op Admin or End User can run flows and monitor the flow runs.
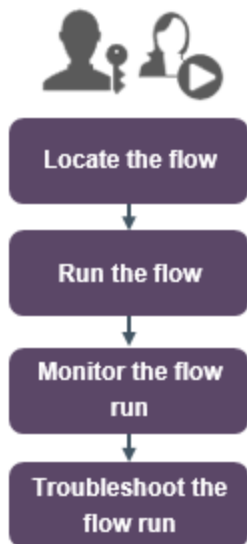
Note that this is just a high level look at the workflow, and there are many options that are not described here. For more detailed information about any of the steps, use the links to learn about the options in-depth.

Step 1: Find the flow that you want to run

The Op Admin/End User locates the flow either from the Flow Library or the Flow Launcher. Go to the **Content Management > Flow Library** module or to the **Run Management > Flow Launcher** module. See "Running a Flow" on the next page.

Step 2: Run the flow

The Op Admin/End User runs the flow. See "Running a Flow" on the next page.

Alternatively, the Op Admin/End User schedules the flow to run later. See "Scheduling Flow Runs" on page 225.

Step 3: Monitor the flow run

The Op Admin/End User tracks the flow run in the Run Explorer. See "Tracking and Managing Flow Runs" on page 244.
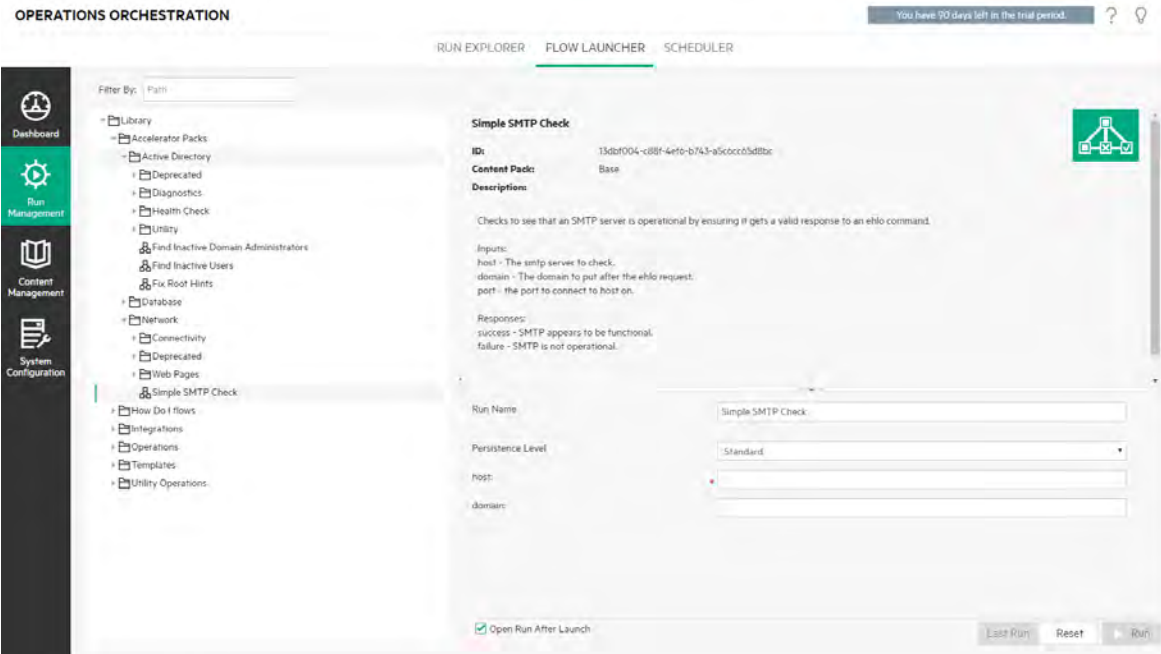
If required, the Op Admin/End User performs actions on flow runs, such as pausing, resuming, and canceling flow runs. See "Tracking and Managing Flow Runs" on page 244.

Step 4: Troubleshoot the flow run

If required, the Op Admin investigates any problems with the run. See "Testing and Troubleshooting a Flow Run" on page 256.

Running a Flow

In the **Flow Launcher** tab in the Run Management workspace, you can select a flow, view its details, prepare it for running, and start the run.

- The flows are displayed in a tree, where you can browse to locate the one you need. You can enter text in the **Filter By** box to filter the display of flows, and you can expand and collapse the tree nodes.

- In the **Flow Details** pane, on the upper right, you can view information about the selected flow, such as ID, content pack, description, and inputs.

- In the **Inputs** pane, on the lower right, you can give the run a name and enter any required inputs.

  > **Note:** The flow level input appears in Central only if it can be modified by a user. For example:
  >
  > ○ The input was defined as one that prompts the user
  >
  > ○ The value of **Assign From** is the same as the flow input name
  >
  > For example, the input will not appear if the flow author has not assigned a value to **Assign From** and has set **Otherwise** to **Use Constant**.
  >
  > For more information see "Input Inspector > Inputs tab" in the *Operations Orchestration Studio Authoring Guide*.

When you have entered the inputs and the flow is ready, you can click the **Run** button to start the flow.

> **Note:** It is also possible to run a flow from the Flow Library (in **Content Management**).

If you have run this flow previously, you can click the **Last Run** button to view the progress of your last run in the Drill Down dialog box.

For information about how to pause, resume, or cancel a flow run, see "Tracking and Managing Flow Runs" on page 244.

What do you want to do?

Run a flow

1. Click the **Run Management** button.

2. Click the **Flow Launcher** tab.

3. In the Flows tree on the left, browse to find the flow that you want to run.
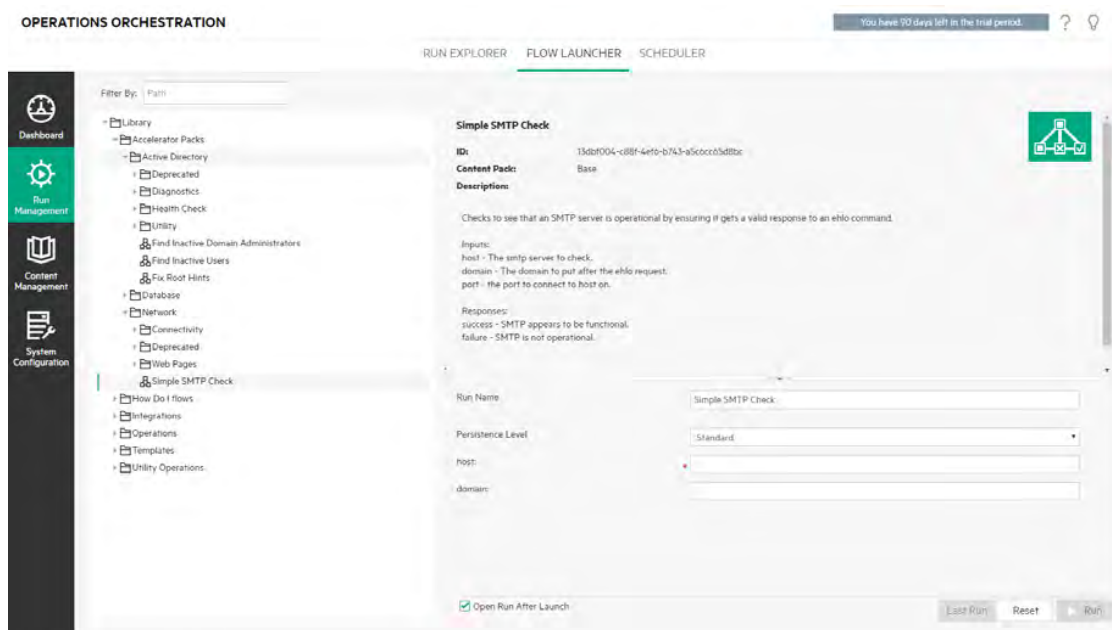
4. (Optional) To locate the flow that you need, enter part or all of the flow path in the **Filter By** text box.

   For example, if you enter "network", only flows that contain the word "network" or which are located in a path with a folder containing the word "network" are displayed.

   > **Note:** The search is not case-sensitive.

5. Select the flow that you want to run. Details about the flow are displayed in the **Flow Details** pane on the right.

6. In the **Run Name** box, enter a name for the run.

   By default, the run name is the flow name, but you can change this. You may want to give it a name that identifies this particular run. For example, you may want to run the same flow multiple times, each with different input. Naming the runs makes it easier to identify them based on the inputs.

7. From the **Persistence Level** list, select the persistence level, in order to control the level of detail that is saved to the Run Log. The options are:

   ○ **Standard** - Includes raw results, step results, and big inputs/outputs.

   ○ **Extended** - Includes the complete raw results, step results, subflow outputs (from step results), and big inputs/outputs.

   > **Note:** The flow's default persistence level is selected by default.

If you do not select a persistence level, the default is used. For information about how to set the default persistence level, see "Monitoring and Controlling Database Size" on page 144.

8. Enter any inputs that are required.

Run Name	Simple SMTP Check
Persistence Level	Standard ▼
host:	*
domain:	

> **Note:** You can click the **Reset** button to apply the default values.

9. (Optional) Click the **Open Graph** button to display a graphical representation of the flow.

10. (Optional) Select the **Open Run After Launch** check box to automatically track this flow after it is started.

11. Click **Run**.

     ○ If the **Open Run After Launch** check box was selected, the progress of the run is displayed in the Drill Down dialog box.

Click **Show Full Tracking** to navigate to the Run Explorer Drill Down. For more information about tracking the progress of a flow, see "Testing and Troubleshooting a Flow Run" on page 256.

○ If the **Open Run After Launch** check box was not selected, the Drill Down dialog box is not opened. A message displays whether the flow was launched successfully.

○ If you have run this flow previously, you can click the **Last Run** button to view the progress of your last run in the Drill Down dialog box.

Run a flow from the Flow Library

1. Click the **Content Management** button to display the Content Management workspace.

2. Click the **Flow Library** tab.

3. Browse for a flow in the flow library.

4. (Optional) To locate the flow that you need, enter part or all of the path in the **Filter By** text box.

   For example, if you enter "network", only flows that contain the word "network" or which are located in a path with a folder containing the word "network" are displayed.

   | **Note:** The search is not case-sensitive.

5. (Optional) Click the **Open Graph** button to display a graphical representation of the flow.

   Select the flow that you want to run, and click the **Run** button in the toolbar.

6. In the Run Flow dialog box, enter a name for this flow run.

   By default, the run name is the flow name, but you can change this. You may want to give it a name that identifies this particular run. For example, you may want to run the same flow multiple times, each with different input. Naming the runs makes it easier to identify them based on the inputs.

7. Enter the necessary inputs for the run in the Run Flow dialog box.

Run Flow   ? ✕

Flow: * Library/Accelerator Packs/Network/Simple SM   ...

Run Name: Simple SMTP Check

Persistence Level: Standard   ▾

host: *

domain:

                                                        Cancel   Run

8. Click **Run**.

   The progress of the run is displayed in the Drill Down dialog box.

Run an interactive flow

1. Run the flow.

2. When the run reaches an interactive point (such as **Inputs Required**, **Handoff**, **Display Message**, and so on), its status is changed to **Paused - Action Required**.

3. The flow can be resumed from the Run Explorer Drill Down or from the Drill Down dialog box.

   ○ Select the run in the Run Explorer and click the **Drill down** ❯ button at the end of the row to display the Run Information window, and click the **Expand** « button to expand the prompt message.

   ○ If the **Open Run After Launch** check box was selected, the prompt message appears in the Drill Down dialog box.

     Note: You can click the **Contract** » button on the left side of the Inputs Required dialog box to hide it again.

4. Read the message and follow its instructions to resume the run—enter the required inputs, handoff, and so on, and click **Resume**.

For information about the different statuses, see "Tracking and Managing Flow Runs" on page 244.

Run a flow with a gated transition

A gated transition restricts access to the next step to users who have been assigned a particular role.

> **Note:** Gated transition flows can only be resumed by users who have **Manage Others' Runs** permissions.

1. Run the flow.

2. When the flow run reaches the gated transition, if you have not been assigned a role with the required permission, the flow run is paused. The run status appears as **Paused – Action Required**.

3. If the **Open Run After Launch** check box was selected, the prompt message appears in the Drill Down dialog box.

    If not, select the run in the Run Explorer and click the **Drill down** button at the end of the row to display the Run Information window, and click the **Expand** button to expand the prompt message.

4. In the prompt message, click the **Hand off to another user (email this run)** link to create an email message to send to the user who needs to complete the flow run.

5. An email message opens with a message for the person to whom you are handing off the flow run and a link to the flow run. Enter the email address of this person and send the message.

Run a flow with a handoff transition

A handoff transition is one where a message appears, suggesting that the current user hand off the flow run to another user. Unlike with a gated transition, the first user has the option to resume the flow run without handing it off to another user.

1. Run the flow.

2. When the flow run reaches the handoff transition, the flow run is paused. The run status appears as **Paused – Action Required**.

3. If the **Open Run After Launch** check box was selected, the prompt message appears in the Drill Down dialog box.

If not, click the **Drill down** ❯ button at the end of the row to display the Run Information window,

and click the **Expand** « button to expand the prompt message.

4. You have two options:

   ○ To hand off the flow run to another user, click **Hand off to another user (email this run)**. An email message opens with a message for the person to whom you are handing off the flow run and a link to the flow run. Enter the email address of this person and send the message.

   ○ Click **Resume** to continue the flow without handing it off.

Run a flow with a Paused - No Workers interaction

A flow has the **Paused – No workers** status when there is a worker group that has not yet been mapped to any workers. Before you can resume the run, you will need to map the worker group to actual workers. For more information see "Setting Up Configuration Items for a Content Pack" on page 193.

1. Run the flow.

2. When the flow run reaches the point where it relates to a worker group with no workers, the flow run is paused. The run status appears as **Paused – No Workers**.

3. If the **Open Run After Launch** check box was selected, the prompt message appears in the Drill Down dialog box.

   If not, click the **Drill down** ❯ button at the end of the row to display the Run Information window,

   and click the **Expand** « button to expand the prompt message.

4. Map the worker group to the relevant workers. For details, see "Setting Up Configuration Items for a Content Pack" on page 193.

5. Return to the pause message and click **Resume**.

Reference Material

Flow Launcher tab

GUI item	Description
Filter By	To locate the flow that you need, enter part or all of the flow path in the **Filter By** text box to filter the flows.
Flows tree	Displays the flows that you have permission to view. Expand and collapse the nodes to see the flows in different folders.
Flow Details pane	Displays information about the selected flow.
Open Graph Button	Click the **Open Graph** button to display a graphical representation of the flow. For more information, see "Viewing a Flow Graph" on page 189.
ID	Displays the UUID of the selected flow.
Content Pack	Displays the content pack that the selected flow came from.
Description	Displays a description of the selected flow. If required, use the scrollbar to scroll down and view the contents of the **Description** section. For OO Content flows, you can add input and output information under **Description**.

Input	Displays the input information for CloudSlang flows only.
	Note: This field may be encrypted as configured by the flow creator.
Output	Displays the output information for CloudSlang flows only.
	Note: This field may be encrypted as configured by the flow creator.
Inputs	Displays an explanation of the inputs required in the selected flow.
Run Name	In the **Run Name** box, enter a name for the run.
	The name does not have to be the same name as the original flow. You may want to give it a name that identifies this particular run. For example, you may want to run the same flow multiple times, each with different input. Naming the runs makes it easier to identify them based on the inputs.
Persistence Level	Select the persistence level, in order to control the level of detail that is saved to the Run Log. The options are:
	• **Standard** - Includes raw results, step results, and big inputs/outputs.
	• **Extended** - Includes the complete raw results, step results, subflow outputs (from step results), and big inputs/outputs.
	Note: The flow's default persistence level is selected by default.
	If you do not select a persistence level, the default is used. For information about how to set the default persistence level, see "Monitoring and Controlling Database Size" on page 144.
<Input fields>	Enter inputs if these are required. The number of input fields and the labels on them will vary, depending on the inputs that are required in the selected flow.
Open Run After Launch check box	Select to automatically track the flow after it is started. If this check box is selected, the flow's progress is displayed in the Run Explorer Drill Down dialog box.
Last Run button	If you have run this flow previously, you can click the **Last Run** button to view the progress of your last run in the Drill Down dialog box.
Reset button	Click the **Reset** button to clear all the inputs from the **Flow Input** pane.
Run button	Click the **Run** button to run the selected flow.

Run Flow dialog box

The Run Flow dialog box is displayed when the **Run** button is clicked, in the Flow Library.

Run Flow   ? ✕

Flow:	* Library/Accelerator Packs/Network/Simple SM ...
Run Name:	Simple SMTP Check
Persistence Level:	Standard ▾
host:	*
domain:	

Cancel   Run

GUI item	Description
Flow	Displays the path to the flow.
Run Name	In the **Run Name** box, enter a name for the run. The name does not have to be the same name as the original flow. You may want to give it a name that identifies this particular run. For example, you may want to run the same flow multiple times, each with different input. Naming the runs makes it easier to identify them based on the inputs.
Persistence Level	Select the persistence level, in order to control the level of detail that is saved to the Run Log. The options are: • **Standard** - Includes raw results, step results, and big inputs/outputs. • **Extended** - Includes the complete raw results, step results, subflow outputs (from step results), and big inputs/outputs. For more information, see "Monitoring and Controlling Database Size" on page 144.
<Input fields>	Enter inputs if these are required. The number of input fields and the labels on them will vary, depending on the inputs that are required in the selected flow.

Run	Click to run the flow.

Drill Down dialog box

The Drill Down dialog box displays the progress of the run, when you run a flow from the Flow Launcher or from the Flow Library, when the **Open Run After Launch** check box is selected.

GUI item	Description
Show Full Tracking	Click **Show Full Tracking** to navigate to the Run Explorer Drill Down.
Other UI items	The UI items in the Drill Down dialog box are the same as those in the Run Explorer Drill Down. For more information about the Run Explorer Drill Down, see "Testing and Troubleshooting a Flow Run" on page 256.

Rerunning a Flow from a Specific Step

Central provides the ability to rerun a flow from a specific step. This is useful when recovering complex flows, where you do not want to rerun the entire flow from the beginning if it failed or was canceled.

The administrator enables/disables the rerun functionality using the **Enable the rerun option for failed/canceled runs** check box in **System Configuration > System Settings**.

You can rerun the flow from a specific step by clicking on the rerun icon ▷ . For example:

2	⚙	UUID Generator ▷
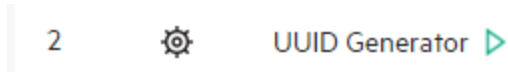
In order to rerun a flow from a specific step, you must first mark the step as a rerun point in Studio. See "Enable Rerunning a Flow from a Specific Step" in the *Operations Orchestration Studio Authoring Guide* .

> **Note:**
>
> - You cannot rerun steps inside parallel or multi-instance lanes. You will not see the rerun icon for these types of steps.
>
> - You cannot rerun steps in flows executed via a remote debugger. You will not see the rerun icon for these flows.
>
> - This functionality is available in Operations Orchestration version 10.80 and above. If you use the new Rerun functionality in a flow, you cannot deploy the flow on earlier Central versions.
>
> - Defining rerun points means that more data is persisted in the database.
>
> - Marking a large number of steps as rerun points increases OO resource consumption and may impact the overall flow execution performance. If you want to define many rerun points, it is recommended that you increase the maximum Java Heap Size. See "Increasing the JVM Heap Size" in the *Operations Orchestration Tuning Guide* .

What do you want to do?

Enable the rerun option

1. Select the **System Configuration > System Settings** tab.

2. In the Run Settings area, select the **Enable the rerun option for failed or canceled runs** check box.

Run Settings

Timeout (minutes):   No Timeout

Flows will be automatically canceled after the inserted number of minutes.

☐ Enable the Rerun option for failed or canceled runs ⓘ
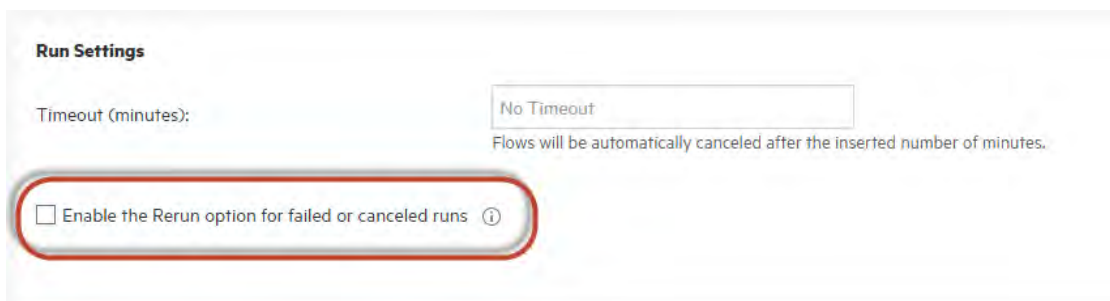
3. Reselect the check box to disable the rerun functionality. The rerun icon will no longer appear next

to steps that can be rerun.

Rerun a flow from a specific step

1. In Studio, mark the step as a rerun point in Studio. See "Enable Rerunning a Flow from a Specific Step" in the *Operations Orchestration Studio Authoring Guide* .

2. In the **Run Management** workspace, select the desired flow.

3. Double click on the flow or click the **Drill down** ❯ button at the end of the row to drill down into the flow steps.

4. Click on the step from which you want to start the flow rerun.

5. Click the rerun icon ▷ next to the step.

   The Rerun window opens:

   Run Flow   ? ✕

   Run Name   Restart Service

   Persistence Level   Standard   ▾

                                    Cancel   Run

6. Select the Persistence Level from the dropdown list and click **Run**.

For details on how to define the persistence levels, see "Default Persistence Level for the Run Log " on page 145.

Drill down into a Rerun Step

1. Click on the arrow next to the rerun step ▸ to drill down into the details of the rerun step.

   For example:

Step Name	Transition Message
▼ ✖ rerun-L3	failure
✔ UUID Generator	success
✔ UUID Generator	success
⚙ Error : failure	
✔ UUID Generator ▷	success
▼ ✔ rerun-L2	success
✔ UUID Generator ▷	success
▼ ✖ rerun-L3 ▷	failure
✔ UUID Generator	success
✔ UUID Generator ▷	success
⚙ Error : failure	

The substeps of the rerun step are shown.

2. Click Step Details in the right pane to see information on the selected substep.

Reference Material

For reference information about the **Run Explorer** tab, see "Tracking and Managing Flow Runs" on page 244.

Scheduling Flow Runs

Schedules allow you to control when to run your flows. You can specify a schedule to run for a specific flow. You can also set up recurring schedules for a task that needs to be repeated.

For example, you need to regularly check whether a number of servers are online. In this case, you can define a flow to check the IP address and then create a schedule to run that flow.

Schedules are displayed in the **Scheduler** tab. The display of schedules depends on the permissions that you have been assigned in your role:

- If you have been assigned a role with the **View Schedules** permission, you will be able to view the OO schedules and their details.

- If you have been assigned a role with the **Manage Schedules** permission, you will be able to view and edit schedules.

> **Note:** The Central Scheduler does not work when SAML authentication is used.

Schedule Owner

Each schedule has an owner. This is the user who created the schedule. The name of the owner appears in the **User** column in the Scheduler

- In the **Scheduler** tab, the **User** column indicates the schedule owner.

- In the **Run Explorer** tab, the **User** column indicates the owner of this flow execution (the user who performs the scheduled run).

If another user updates the schedule (for example, the recurrence, name, or time zone), the schedule owner changes to this user. This user is displayed as the schedule owner in the Scheduler and as the flow executor in the Run Explorer.

> **Note:** The new owner appears only for runs that started after the change. For runs that started before the change, the original owner is displayed in the **User** column.

If the schedule is created or updated by a user who is logged in as an anonymous user, the **User** column will display N/A.

> **Note:** If the schedule is updated by a user without run permissions on the scheduled flow, this flow will not run.

> **Note:** If an LDAP configuration exists and the configured LDAP server is not responding, this will affect all types of schedules. This includes both schedules that are run with a LDAP user and schedules that are run with an internal user.

Schedule Name

Each schedule has a name. The flow name is entered by default, but you may prefer to use a different name, for example, if you have multiple schedules using the same flow.

> **Note:** If you give the schedule a different name, this is the name that will appear in the Run Explorer, rather than the flow name.

Scheduler Troubleshooting

A scheduler that fails to trigger is known as a misfire.

You can set the action (instruction) to take place upon a scheduler misfire in the file **<OO installation folder>\central\tomcat\webapps\oo\WEB-INF\classes\META-INF\spring\quartzContext.xml**.

In this file, set the **p:misfireInstructionName** attribute for simple trigger and cronTrigger beans. The default instruction is "MISFIRE_INSTRUCTION_DO_NOTHING"

In the case of a misfire, the following log message opens:

```
There was a misfire in the scheduler <name_of_the_scheduler>
```

> **Note:** The logging of a scheduler misfire will occur only when the Central service is running.

What do you want to do?

View existing schedules

To view the **Scheduler** tab, you must be logged in with a user account that is assigned a role with **View schedules** permissions.

1. Click the **Run Management** button to display the Run Management workspace.

2. Click the **Scheduler** tab. A table displays all the current schedules, including the following information:

   - **State**: Displays whether the schedule is enabled ⊘ or disabled ⊗.

   - **Schedule Name**: Displays the name that was given to this schedule. This value can be edited in the Edit Schedule dialog box.

   - **Flow Name**: Displays the name of the flow that the schedule relates to. This value is taken from the flow and cannot be modified in the schedule.

   - **Next Run**: Displays the time and date when the next run is scheduled to occur. The time and date displayed is in the your time zone.

   - **Previous Run**: Displays the time and date when the previous run occurred, if this exists.

   - **User**: Displays the name of the user who created or modified this schedule.

Display the details of a schedule

You can drill down to see the details of a schedule in the Schedule Details dialog box. This functionality is useful for users with **View Schedules** permission, but without permission to edit schedules.

The details include information that is not displayed in the table in the **Scheduler** tab, such as recurrence pattern, range of recurrence, and the timezone in which the schedule runs.

To display the details of a schedule:

1. Click the **Scheduler** tab.

2. Select the check box next to the schedule that you want to view and click the **Details** ⓘ button in the **Scheduler** toolbar. The details of the selected schedule are displayed in the Schedule Details dialog box.

   > **Note:** The **Details** button is only available if a single schedule is selected.

3. When you have finished viewing the details of the schedule, click **OK**.

Edit schedules

To edit a schedule, you must be logged in with a user account that is assigned a role with **Manage schedules** permission.

To edit a schedule:

1. Click the **Scheduler** tab.

2. Select the check box next to the schedule that you want to edit and click the **Edit** ✏ button in the **Scheduler** toolbar.

3. In the Edit Schedule dialog box, make the required changes to the schedule, and then click **Save**.

> **Note:** If you try to set the end date for the schedule to the current date (or earlier), the Save button is disabled and the following warning message appears:
>
> `The end date must be after the start date.`

Clone schedules

You can create an identical copy of a schedule that appears in the **Scheduler** tab. You can make any required changes to this schedule in the Clone Schedule dialog box.

To clone a schedule:

1. Click the **Scheduler** tab.

2. Select the check box next to the schedule that you want to clone, and click the **Clone** 🗗 button in the **Scheduler** toolbar.

   > **Note:** The **Clone** 🗗 button is only enabled if a single schedule is selected.

   The Clone Schedule dialog box opens. All the fields in the dialog box are populated with the values from the selected schedule.

3. If required, edit the schedule details in the Clone Schedule dialog box. You may wish to give the clone a different name.

4. Click **Save**.

> **Note:** If you try to set the end date for the cloned schedule to the current date (or earlier), the Save button is disabled and the following warning message appears:
>
> `The end date must be after the start date.`

Delete schedules

You can delete individual or multiple schedules.

To delete schedules:

1. Click the **Scheduler** tab.

2. Select the check boxes next to the schedules that you want to delete and click the **Delete** ✕ button in the **Scheduler** toolbar.

3. In the confirmation dialog box, click **Yes**.

The selected schedules are deleted from the **Scheduler** tab.

Enable a schedule

By default, when a new schedule is created, it is enabled. If a schedule has been disabled, you need to enable it before it will run. You can enable individual or multiple schedules.

To enable a schedule for a flow run:

1. In the **Scheduler** tab, select the check box next to each disabled schedule that you want to enable.

2. In the **Scheduler** toolbar, click **Enable** ⊘.

   > **Note:** The **Enable** button is only available if at least one disabled schedule is selected.

3. In the confirmation dialog box, click **Yes**.

The icon in the **State** column changes to **Enabled** for this schedule.

Disable a schedule

You can disable individual or multiple schedules. These schedules remain in the **Scheduler** tab, but they will not run until they are enabled.

To disable a schedule for a flow run:

1. In the **Scheduler** tab, select the check box next to each enabled schedule that you want to disable.

2. In the **Scheduler** toolbar, click **Disable** ⊘.

   > **Note:** The **Disable** button is only available if at least one enabled schedule is selected.

3. In the confirmation dialog box, click **Yes**.

The icon in the **State** column changes to **Disabled** for this schedule.

Filter the display of schedules

You can filter the schedules that are displayed in the **Scheduler** tab. This can help you locate schedules quickly, when there is a large number of schedules.

For example, if you are looking for schedules of flows relating to connectivity, you could enter "Connect" as the filter term in the **Flow Name**column.

State	Schedule Name	Flow Name	Next Run	Previous Run	User	Persistence Leve
▼			[Jan 5 2016 1:57 ▼	[Jan 6 2016 1:57 ▼		

> **Note:** The filter is not case-sensitive.

1. Click the **Scheduler** tab.

2. In the text boxes under the column titles, enter the information to filter by.

   If you do not know the exact name, enter part of the name.

Refresh the display of schedules

If someone creates, edits, deletes, enables, or disables a schedule in another client, this does not show in the **Scheduler** tab until you have refreshed the page.

To refresh the display of schedules:

1. Click the **Scheduler** tab.

2. In the **Scheduler** toolbar, click **Refresh** .

   > **Note:** It is also possible to refresh the page using a browser refresh command (for example, using key combinations such as F5 or Ctrl+R), but this loads the entire page. Clicking the **Refresh** button is faster, because it only refreshes the display of schedule data .

Sort the display of schedules

If there are multiple schedules displayed in the **Scheduler** tab, you can sort them according to any of the column headings.

1. Click the **Scheduler** tab.

2. Click the column header that you want to use to sort the schedules.

Run a Scheduled Flow Now

You can trigger an ad-hoc run of a scheduled flow.

Central users that can perform this action must have the following permissions enabled for their role:

- View Schedules and Manage Schedules permissions. For more information see "Setting Up Security – Roles" on page 66.

- View and Run content permissions. For more information, see "Setting Permissions for Content" on page 181.

> **Note:** If a schedule is set to end after a specific number of occurrences, running this scheduled flow ad-hoc will not be counted as an occurrence. Only runs that are initiated by the schedule's recurrence pattern will be counted in the number of occurrences.

1. Click the **Scheduler** tab.

2. Select the schedule from the list.

3. In the Scheduler toolbar, click the Run-now button .

Create a schedule

To create a schedule, you must be logged in with a user account that is assigned a role with **View schedules** and **Manage schedules** permissions.

1. You can create a new schedule either from the Scheduler or from the Flow Library:

   - To create a schedule from the Flow Library:

     i. Click the **Content Management** button to display the Content Management workspace.

     ii. Click the **Flow Library** tab.

     iii. Select a flow from the Flow Library.

     iv. Click the **Schedule** button. The Create Schedule dialog box opens.

     v. Click the browse button to browse for and select the flow for which you want to create the schedule.

   - To create a schedule from the Scheduler:

     i. Click the **Run Management** button to display the Run Management workspace.

     ii. Click the **Scheduler** tab.

     iii. In the **Scheduler** toolbar, click the **Create** button.

      iv.   Select the flow for which you want to create the schedule, and click **OK**. The Create Schedule dialog box opens.

2. In the **Schedule Name** box, enter a meaningful name for this schedule. The flow name is entered by default, but you may prefer to use a different name, for example, if you have multiple schedules using the same flow.

> **Note:** If you give the schedule a different name, this is the name that will appear in the Run Explorer, rather than the flow name.

Create Schedule

Flow

Flow Path:	* Library/Accelerator Packs/Network/Simple SMTP Che ...
Schedule Name:	* Simple SMTP Check
User:	No Data Available
Persistence Level:	Taken from flow settings (currently Standard) ▼

☐ Use empty value for prompts

4. From the **Persistence Level** list, select the persistence level for the scheduled run, to control the level of detail that is saved to the Run Log. The options are:

- **Standard** - Includes raw results, step results, and big inputs/outputs.

- **Extended** - Includes the complete raw results, step results, subflow outputs (from step results), and big inputs/outputs.

- **Taken from flow settings** - whatever was selected as the default persistence level for the flow.

3. Select the **Use empty value for prompts** check box if you want the schedule to skip over prompts for inputs. For any prompt that has no value assigned, it will use blank, and the scheduled flow will run without stopping to prompt for values.

However, note that if any of these are required inputs, the flow will fail.

This option is selected by default after an upgrade from version 9.x to 10.x. This is to enable backward compatibility, because this is how all schedules worked in 9.x.

If you do not select this check box, the run is stopped and prompts for inputs, even if they are not required inputs.

4. In the **Recurrence pattern** section, select from one of the following recurrence frequencies for the schedule: daily, weekly, monthly.

   ○ **Daily**: To run the flow on a daily basis, select **Daily** and one of the following:

   Recurrence pattern

   - **Every**: To run the flow every day at regular intervals, select **Every**, and then enter the value and select **minutes** or **hours**.

     Note: If you select **Every**, the interval field cannot be left empty, and the value must be at least **1**.

   - **Every day**: Select to run the flow once every day of the week.

     Note: The flow run will start at the time set in the **Start time** field in the **Range of recurrence** section.

   - **Every weekday**: Select to run the flow once a day, on weekdays only.

     Note: If the working week is not Monday-to-Friday, use the **Weekly** option instead, and select the working days.

   ○ **Weekly**: To run the flow on a weekly basis, select **Weekly** and then select the days of the week on which you want the flow to run.

Recurrence pattern

○ Daily   ☐ Sun   ☑ Mon   ☑ Tue   ☐ Wed   ☐ Thu

◉ Weekly   ☐ Fri   ☐ Sat

○ Monthly

○ Yearly

> **Note:** If you select **Weekly**, you must select at least one day.

○ **Monthly**: To run the flow every month, select **Monthly** and one of the following:

Recurrence pattern

○ Daily   ○ Day [1 ▾] of every [1 ▾] month(s)

○ Weekly   ◉ The [first ▾] [Monda ▾] of every [3 ▾] month(s)
                                day

◉ Monthly   Sunday

                                  Monday

○ Yearly   Tuesday

                                  Wednesday

- To run the flow on a particular date in each month, select **Day** and select the date. For example, select **15** to run the flow on the 15th of every month.

- To run the flow once a month on a particular day of the week, select the day of the week and whether it is the first, second, third, fourth, or last time that this day occurs in the month. For example, to run the schedule on the third Friday of each month, select **third**, then select **Friday**, and then select **1** month(s).

- To run the flow at intervals of several months, select a number of months. For example, to run a schedule once a quarter, on the first Monday of the quarter, select **first**, then select **Monday**, and then select **3** months.

  For example, in a schedule starting on April 1, 2013, and ending on December 20, 2013, with a recurrence pattern of the third day of every 2 months, the flow will run on June 3, August 3, October 3, and December 3.

- To run the flow on the last day of the month, select **last** and then select **day** instead of a specific day of the week.

○ **Yearly**: To run the flow once a year, select **Yearly** and one of the following:

- To run the flow on a particular date each year, select **Every** and then select the month and the date. For example, every July 4.

- To run the flow once a year on a particular month and on a particular day of the week, select the month, then select the day of the week and whether it is the first, second, third, fourth, or last time that this day occurs in the month. For example, the second Monday of each November.

5. In the **Range of recurrence** section, in the **Start date** box, enter the date on which you want the flow schedule to start.

   Note: If a schedule with an earlier date undergoes content upgrade, the start date is changed to the current date.

   Note: The date format appears in accordance with the locale of the client browser.

6. In the **Start time** boxes, enter the time at which you want the flow schedule to start, in hours and minutes.

7. From the **Timezone** list, select the time zone where you are located.

   Note: Time zones around the world are expressed as positive or negative offsets from Coordinated Universal Time (UTC). When you create a schedule, a time zone is selected by default from the alphabetically-ordered list that is currently available. Operations Orchestration selects the first time zone from the list that has the offset equal to the client

offset (computed with daylight savings, if relevant).

Note that this timezone might not be geographically correct. For example, if you are located in a location where the time zone is UTC-09.00, the default time zone will be **America/Anchorage**. If desired, you can select the geographically correct location from the **Timezone** list.

8. Select when you want the schedule to end:

    ○ Select **No end date** if you do not want to set an end date for the schedule.

    ○ To limit the schedule to a specific number of runs, select **End after** and enter the number of runs after which the schedule will stop. This option is only available for daily schedules that have been set to run at intervals of a number of hours or minutes.

    ○ To end the schedule on a specific date, select **End by** and enter the date after which schedule will stop.

    ○ **Note:** If you try to set the end date for the schedule to the current date (or earlier), the Save button is disabled and the following warning message appears:

        The end date must be after the start date.

9. If the flow requires inputs, enter them in the **Inputs** section on the right side. If a red star appears next to an input box, this means that the input is required, and the **Save** button will not be enabled unless an input value is provided.

10. If you want to enter additional inputs that are not flow inputs, click **Add Input** at the bottom of the **Inputs** section, and enter the name and value of the input.

11. Click **Save**.
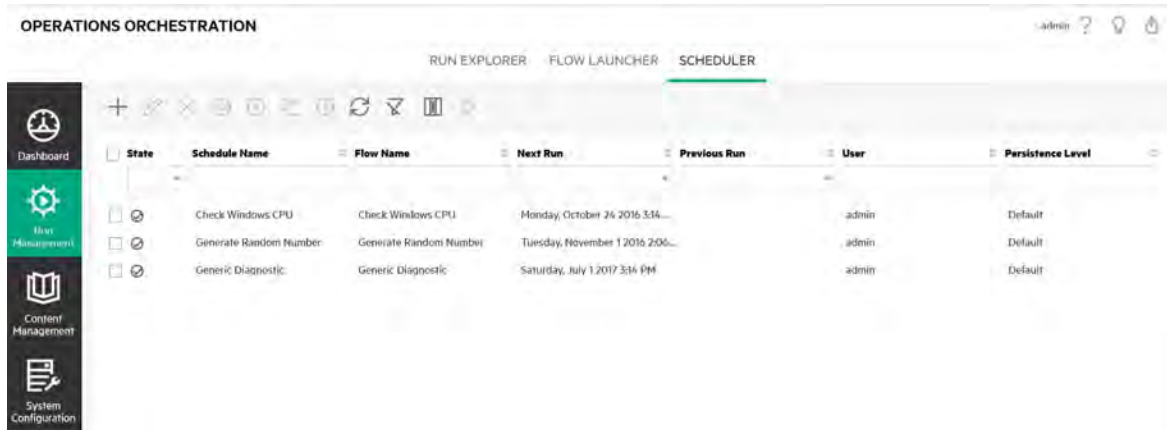
    Note: If the **Save** button is not enabled, this means that a required input has not been

> provided, or there is a mistake in one of the fields in the dialog box. For example, the **Save** button is not enabled if a non-existent date is entered, such as February 31, or if no days of the week are selected for a weekly recurrence pattern.

Reference Material

Scheduler tab

GUI item	Description
New ➕ button	Opens the Create Schedule dialog box, so that you can add a new schedule.
Edit 🖊 button	Opens the Edit Schedule dialog box, so that you can edit the selected schedule. The **Edit** button is only available if at least one schedule is selected.
Delete ✕ **button**	Deletes the selected schedules. The **Delete** button is only available if at least one schedule is selected.
Enable ⊘ **button**	Enables the selected schedules. The **Enable** button is only available if at least one disabled schedule is selected.
Disable ⊘ **button**	Disables the selected schedules. The **Disable** button is only available if at least one enabled schedule is selected.
Clone ⧉ **button**	Creates an identical copy of the selected schedule. The **Clone** button is only enabled if a single schedule is selected.
Details ⓘ **button**	Displays the details of the selected schedule. The **Details** button is only enabled if a single schedule is selected.

Refresh button	Click to refresh the display of schedules.
Select Columns button	Click to display the column picker, to choose which columns to display.
Clear Filters button	Click to remove the filters and display all the schedules.
Run now button	Click to trigger an ad-hoc run of the scheduled flow. **Note:** The "Run Now" button is only enabled if a single schedule is selected.
Filter boxes	In the filter boxes at the top of the columns, enter text to filter the displayed schedules.
State	Displays whether the schedule is enabled or disabled .
Schedule Name	Displays the name that was given to the schedule. In the filter box under **Schedule Name**, enter the name of a schedule or part of a name, to filter the displayed schedules.
Flow Name	Displays the name of the flow that is run in the schedule. In the filter box under **Flow Name**, enter the name of a flow or part of a name, to filter the displayed schedules.
Next Run	Displays the time when the next run is scheduled.
Previous Run	Displays the time when the previous run in the schedule occurred.
Persistence Level	Displays the persistence level, which controls the level of detail that is saved to the Run Log: • **Standard** - Includes raw results, step results, and big inputs/outputs. • **Extended** - Includes the complete raw results, step results, subflow outputs (from step results), and big inputs/outputs. • **Taken from flow settings** - whatever was selected as the default persistence level for the flow. For more information, see "Monitoring and Controlling Database Size" on page 144.
User	Displays the name of the user who created or modified the schedule.

	If the schedule is created or updated by a user who is logged in as an anonymous user, the **User** column will display N/A.

Create Schedule dialog box

Create Schedule

Flow
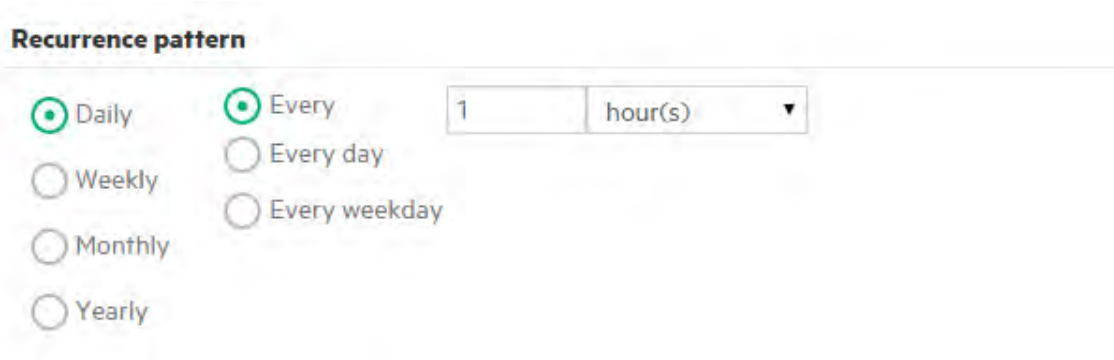
Flow Path: * Library/Accelerator Packs/Network/Simple SMTP Che   **...**

Schedule Name: * Simple SMTP Check
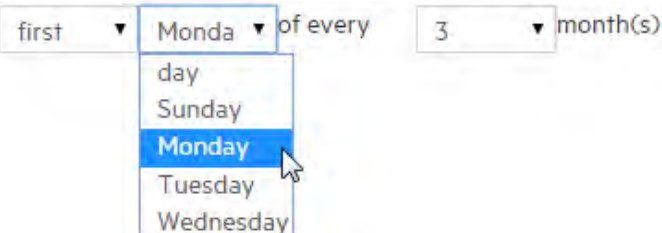
User: No Data Available

Persistence Level: Taken from flow settings (currently Standard)   ▼

☐ Use empty value for prompts

GUI item	Description
Flow Path	Displays the path to where the flow is located.
Schedule Name	Enter a meaningful name for this schedule.
Use empty value for prompts	Select this check box if you want the schedule to skip over prompts for inputs. For example, when you just want the schedule to run without you, and you don't want to wait to see the prompt and enter the input. However, note that if input is required for that flow, the flow will fail. This option is selected by default after an upgrade from version 9.x to 10.x. This is to enable backward compatibility, because this is how all schedules worked in 9.x
Persistence Level	Select the persistence level, in order to control the level of detail that is saved to the Run Log. The options are: • **Standard** - Includes raw results, step results,

	and big inputs/outputs.
	• **Extended** - Includes the complete raw results, step results, subflow outputs (from step results), and big inputs/outputs.
	• **Taken from flow settings** - whatever was selected as the default persistence level for the flow.
	For more information, see "Monitoring and Controlling Database Size" on page 144.
Recurrence pattern section	In the **Recurrence pattern** section, you define the frequency at which the schedule recurs. The **Recurrence pattern** section changes, depending on whether you select **Daily**, **Weekly**, **Monthly**, or **Yearly**.

Recurrence pattern

⦿ Daily	⦿ Every [1] [hour(s) ▾]
◯ Weekly	◯ Every day
◯ Monthly	◯ Every weekday
◯ Yearly	

Daily - Every <n> minutes/hours	To run the flow every day at regular intervals, select **Daily**, select **Every**, enter the value, and select **minutes** or **hours**. **Note:** If you select **Every**, the interval field cannot be left empty, and the value must be at least **1**.
Daily – Every day	To run the flow once a day, every day of the week, select **Daily**, and then select **Every day**. **Note:** The flow run will start at the time set in the **Start time** field in the **Range of recurrence** section.
Daily – Every weekday	To run the flow every weekday, select **Daily**, and then select **Every day**.

	Note: If the working week is not Monday-to-Friday, use the **Weekly** option instead, and select the working days.

Weekly	To run the flow on a weekly basis, select Weekly and then select the days of the week on which you want the flow to run. **Note:** If you select **Weekly**, you must select at least one day.

Monthly – Day <n> of every month	To run the flow on a particular day of each month , select **Monthly**, select **Day** and then select the date. For example, select **15** to run the flow on the 15th of every month.
Monthly – The <nth> <day of week> of every <n> month(s)	To run the flow once a month on a particular day of the week, select the day of the week and whether it is the first, second, third, fourth, or last time that this day occurs in the month. For example, to run the schedule on the third Friday of each month, select **third**, then select **Friday**, and then select **1** month(s).

	To run the flow at intervals of several months, select a number of months. For example,to run a schedule once a quarter, on the first Monday of the quarter, select **first**, then select **Monday**, and then select **3** month(s). To run the flow on the last day of the month, select **last**, and then select **day** instead of a specific day of the week.

Recurrence pattern

Daily

Weekly

Monthly

Yearly

Every | January ▼ | 1 |

The | second ▼ | Tuesday ▼ | of | April ▼ |

Yearly – Every <month> <n>	To run the flow on a particular date each year, select **Yearly**, select **Every**, and then select the month and the date. For example, every July 4.
Yearly – The <nth> <day of week> of <month>	To run the flow once a year on a particular month and on a particular day of the week, select **Yearly**, select the month, select the day of the week and whether it is the first, second, third, fourth, or last time that this day occurs in the month. For example, the second Friday of each November.
Range of recurrence section	In the **Range of recurrence** section, you define when the schedule starts and ends.

Range of recurrence

Start date: 12/27/2015

Start time: 09 : 15

Timezone: (UTC-08:00) America/L ▼

No end date

End after 10 occurrences

End by 12/31/2016

| **Start date** | Enter the date on which you want the schedule to start.

Note: If a schedule with an earlier start date |

	undergoes content upgrade, the start date is changed to the current date.
Start time	Enter the time of day at which you want the schedule to start.
No end date	Select if you want the schedule to run continuously.
End after <n> occurrences	To limit the schedule to a specific number of runs, select **End after** and enter the number of runs after which the schedule will stop. This option is only available for daily schedules that have been set to run at intervals of a number of hours or minutes.
End by <date>	To end the schedule on a specific date, select **End by** and enter the date after which schedule will stop.
Timezone	From the **Timezone** list, select the time zone where you are located.
	Note: Time zones around the world are expressed as positive or negative offsets from Coordinated Universal Time (UTC). When you create a schedule, a time zone is selected by default from the alphabetically-ordered list that is currently available. Operations Orchestration selects the first time zone from the list that has the offset equal to the client offset (computed with daylight savings, if relevant).
	Note that this timezone might not be geographically correct. For example, if you are located in a location where the time zone is UTC-09.00, the default time zone will be **America/Anchorage**. If desired, you can select the geographically correct location from the **Timezone** list.
	(UTC-09:00) America/Anchorage (UTC-09:00) America/Juneau (UTC-09:00) America/Nome (UTC-09:00) America/Sitka (UTC-09:00) America/Yakutat (UTC-09:00) Pacific/Gambier

Inputs		
host:	*	
domain:		
	Value	×

Inputs	If the flow requires inputs, enter them in the **Inputs** section. This section will look different, depending on whether inputs are required.
Save	Click to save the schedule.

Tracking and Managing Flow Runs

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

The Run Explorer is auto-refreshed. The view is updated whenever a new run is started or if the run status, duration, or user is updated.

> **Note:** The name that appears under **Run Name** in the Run Explorer is not always the flow name:
>
> - If you gave the run a name, this is the name that appears.
>
> - For a scheduled run, if you gave the schedule a name, this is the name that appears.

You can use the Run Explorer to manage flow runs when there are multiple flows running. You can perform actions on one or more flow runs, such as pausing, resuming, reassigning, and canceling them.

The display of runs depends on the content permissions that your user has been assigned. You can only see runs of flows you have permission to view.

The actions that you can perform on flow runs depends on permissions that you have been assigned in your role:

- If you have been assigned a role with the **Manage Others' Runs** permission, you will be able to perform actions on the runs triggered by all users.

- If you have been assigned a role without this permission, you will be able to perform actions on the runs that you have triggered.

If you are unable to locate a flow run, you can use the filter capabilities of the Run Explorer to find the flow run that you need. For more information about filtering, see "Adjusting the Display of Flow Runs" on page 249.

What do you want to do?

View basic details about a flow run

1. Click the **Run Management** button to display the Run Management workspace.

2. Click the **Run Explorer** tab.

3. View the basic details about each run, in the **Runs** table.

Select multiple runs

You can perform the following actions on multiple runs: Cancel, Pause, Reassign, Resume.

There are two ways to select multiple runs in the Run Explorer:

- Click in the check box next to each run that you want to select

- Hold down the **Shift**/**Ctrl** keys and click anywhere in the row to select a run:

  ○ Use **Shift** to select multiple runs that are adjacent. Select the first run, and hold the **Shift** key while you select the last run.

  ○ Use **Ctrl** to select multiple runs that are not adjacent.

Display advanced details about a flow run

1. Click the **Run Explorer** tab.

2. Select the flow run that you want to look at in detail.

3. Click the **Drill down** ❭ button at the end of each row to display more information about the selected run.

   > **Note:** Alternatively, you can double-click a row to drill down to see this information.

   For more information, see "Testing and Troubleshooting a Flow Run" on page 256.

   > **Note:** The **Drill down** ❭ button is only displayed when you select a row in the table.

Pause a flow run

1. Click the **Run Management** button to display the Run Management workspace.

2. Click the **Run Explorer** tab.

3. Select one or more flows that are currently running.

4. Click the **Pause** ❚❚ button in the Run Explorer toolbar.

   > **Note:** The **Pause** ❚❚ button is only available if all selected runs are currently running.
   >
   > Note also that a run with **Pending pause** status will be timed out if the timeout threshold is reached.

The run status is changed to **Pending pause** and then to **Paused**.

Resume a flow run that was paused

You can resume a run that has **Paused – User Paused** status. If you have been assigned a role with the **Manage Others' Runs** permission, you can resume runs that were paused by others. Otherwise, you can resume runs that you paused.

1. Click the **Run Management**> **Run Explorer** tab.

2. Select one or more runs that were paused.

3. If the **Resume** ❚▷ button is available in the Run Explorer toolbar, click it to resume the run (or runs).

If the **Resume** button is not available in the Run Explorer toolbar, this means that you need to interact with the system in order to complete these runs individually:

1. Click the **Drill down** button at the end of the row to display more information about a run.

2. Click the **Expand** button to expand the prompt message.

3. Enter the required information.

4. Click **Resume** .

5. Repeat for the next run that needs to be resumed.

> **Note:** The **Resume** button is only available if all selected runs are currently paused.

Cancel a flow run

1. Click the **Run Management**> **Run Explorer** tab.

2. Select one or more flows that are currently running.

3. Click the **Cancel** button in the Run Explorer toolbar.

4. In the confirmation dialog box, click **Yes** to confirm that you want to cancel the flow run (or runs).

> **Note:** The **Cancel** button is only available if all selected runs can be canceled: if they are running or paused but not finished.

Reassign ownership of a flow run

Each run has an owner. By default, this is the person who started the run, but you can change the owner during execution by assigning the run to a different user.

> **Note:** The owner of a run is the user who appears in the **User** column.

You can reassign multiple runs at the same time.

Reassignment is usually triggered as a result of a gated transition, where the run is paused because it doesn't have permission to proceed, or as a result of a handoff, where the author marked a transition as needing to be handed off to another user. However, you can also reassign a flow at any point in the flow run. If the flow is running, it will become paused so that it can be reassigned.

> **Note:** Reassignment does not validate that the reassigned user exists.

> **Note:** If you reassign a flow with the status of **Paused – Action Required**, note that the assigned user will not be able to resume it via the Run Explorer. The assigned user will need to drill down into the run and resume it from there.

1. Click the **Run Management** > **Run Explorer** tab.

2. Select one or more flows that are currently running or paused.

3. Click the **Reassign** button.

4. In the Reassign a Run dialog box, enter the user name of a different user and the domain to which this user belongs.

5. Click **Reassign**.

> **Note:** The **Reassign** button is only available if the selected runs are running or paused.

Hand off a flow run

Some flows contain gated transitions, requiring the current user to hand off the run to another user. The run pauses, and a dialog box enables the current user to send an email to another user, asking them to continue the run.

1. Click the **Run Management** > **Run Explorer** tab.

2. Select a run that is currently paused at the gated transition.

3. Click the **Drill down** button at the end of the row to display more information about the selected run.

4. Click the **Expand** button to expand the prompt message.

5. In the prompt message, click the **Hand off to another user (email this run)** link to create an email message to send to the user who needs to complete the flow run.

6. An email message opens, with a message for the person to whom you are handing off the flow run and a link to the flow run. Enter the email address of this person and send the message.

Refresh the runs in the Run Explorer
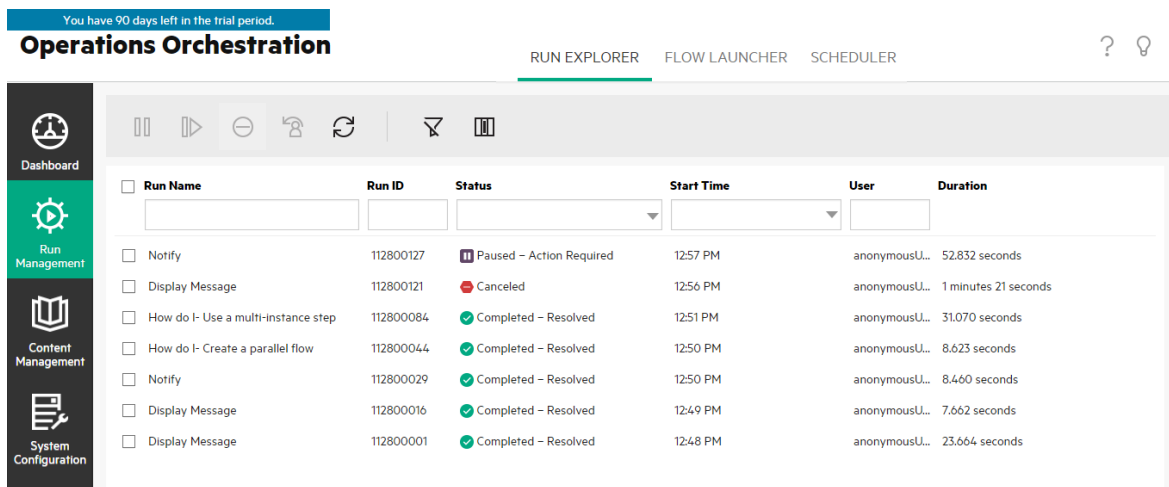
The **Refresh** button refreshes the view that is currently below in the Run Explorer—whether this is the run list or the drill down. For information about the drill down, see "Testing and Troubleshooting a Flow Run" on page 256.

- Click the **Refresh** button in the Run Explorer toolbar.

> **Note:** It is also possible to refresh the page by pressing F1 on your keyboard, but this loads the entire page. Clicking the **Refresh** button only refreshes the data display, so this is much faster.

Reference Material

Run Explorer

Adjusting the Display of Flow Runs

Adjusting Filters

The Run Explorer can display up to 200 flow runs in a single page. So if you have many flows running at the same time, it is recommended to use filters to locate the flow that you need.

You can create a filter to view runs that contain a specific word in their name, runs with a specific status, runs that were started by a specific user, or runs that started within a defined time range.

You can apply multiple filters at the same time. For example, you can filter the Run Explorer to display only failed runs with the word "Ping" in their name, which were started on the first of June.

To apply a filter, select or enter text in the filter boxes in the Run Explorer toolbar. Once you have entered the criteria, the filter is applied—there is no need to click a button to apply it.

It is also possible to filter runs via API.

Adjusting Columns

You can also adjust the width and position of a column, and decide which columns to show and which to hide.

Changes that you make to these settings (column width, column order, hidden columns, and applied filter) are saved in the local cache of the browser.

> **Note:** Changes that you make to the columns will affect all users on this machine, regardless of which user is logged in.

> **Note:** When you hide a column, if you were using a filter, this deletes the filter.

What do you want to do?

Create a filter according to run status

1. Click the **Run Management** button to display the Run Management workspace.

2. Click the **Run Explorer** tab.

3. From the **Status** list, select the status that you want to filter by. You can select more than one status.

    The Run Explorer only displays runs with one of the selected statuses.

    > **Note:** After you have selected a status for the filter, the name of this status appears as the title of the **Status** list.

    Status

    Completed – Resolved ▼

Create a filter according to start time

○ Last Day ○ Last 7 Days ○ Last 30 Days ● Custom Range

1. In **Run Management**, click the **Run Explorer** tab.

2. In the Run Explorer toolbar, select the arrow ⌄ next to **Start Time**.

3. Select one of the following options:

    ○ **Last Day** – to display only runs that started within the last 24 hours.

    ○ **Last 7 Days** – to display only runs that started within the last week.

- ○ **Last 30 Days** – to display only runs that started within the last 30 days.

- ○ **Custom Range**, and then select the start and finish of the time range that you want to filter by. You can select both the date and the time of day. The calendar on the left is for the start of the time range and the calendar on the right is for the end of the time range.

  For example, you may want to display only runs that started between 9:00 and 10:00 AM on Monday August 20, 2013.

Create a filter according to user name

Filters are not case-sensitive, so you can type in upper or lower case.

1. In **Run Management**, click the **Run Explorer** tab.

2. In the **User** text box, enter part or all of the user name.

   The Run Explorer displays only runs that were started by the specified user. For example, entering your user name displays only the flow runs that you have run.

Create a filter according to run name

Filters are not case-sensitive, so you can type in upper or lower case.

1. In **Run Management**, click the **Run Explorer** tab.

2. In the **Run Name** text box, enter part or all of the run name.

   The Run Explorer displays only runs that contain this text in their name. For example, typing "network" returns all the rows where the filtered field contains the word "network".

Create a filter according to run ID

1. In **Run Management**, click the **Run Explorer** tab.

2. In the **Run ID** text box, enter part or all of the run ID.

   The Run Explorer displays only the run with this run ID.

Remove a filter

1. In **Run Management**, click the **Run Explorer** tab.

2. To remove an individual filter, clear the relevant filter text box.

3. To remove all the filters, click the **Clear Filter** > button.

   The filters are no longer applied.

Change the position of a column

1. In **Run Management**, click the **Run Explorer** tab.

2. In the Run Explorer toolbar, drag a column name to the left or the right, to adjust the column's position in the Run Explorer.

Adjust the width of a column

1. In **Run Management**, click the **Run Explorer** tab.

2. In the Run Explorer toolbar, drag the edge of a column name, to adjust its width.

Hide/display a column

1. In **Run Management**, click the **Run Explorer** tab.

2. In the Run Explorer toolbar, click the **Select Columns** button to display the column picker.
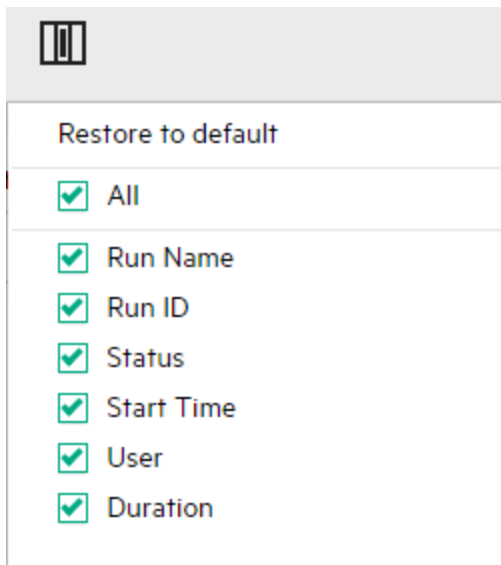
3. To hide a column, clear the check box next to its name.

Reference Material

Run Explorer toolbar - Filter options

GUI item	Description
Filter by: Status	From the **Status** list, select the status or statuses that you want to filter by.
Filter by: User	Enter part or all of the user name.
Filter by: Start Time	Select the arrow ⌄ next to **Start Time** to filter according to the date and time that the runs were started. Select one of the options:

	Last Day – to display only runs that started within the last 24 hours. **Last 7 Days** – to display only runs that started within the last week. **Last 30 Days** – to display only runs that started within the last 30 days. **Custom Range**, and then select the start and finish of the time range that you want to filter by. You can select both the date and the time of day. The calendar on the left is for the start of the time range and the calendar on the right is for the end of the time range. For example, you may want to display only runs that started between 9:00 and 10:00 AM on Monday August 20, 2013.
Filter by: Run Name	Enter part or all of the run name.
Filter by: Run ID	Enter part or all of the Run ID.
Clear Filter button	Click to remove the filters and display all the flows.
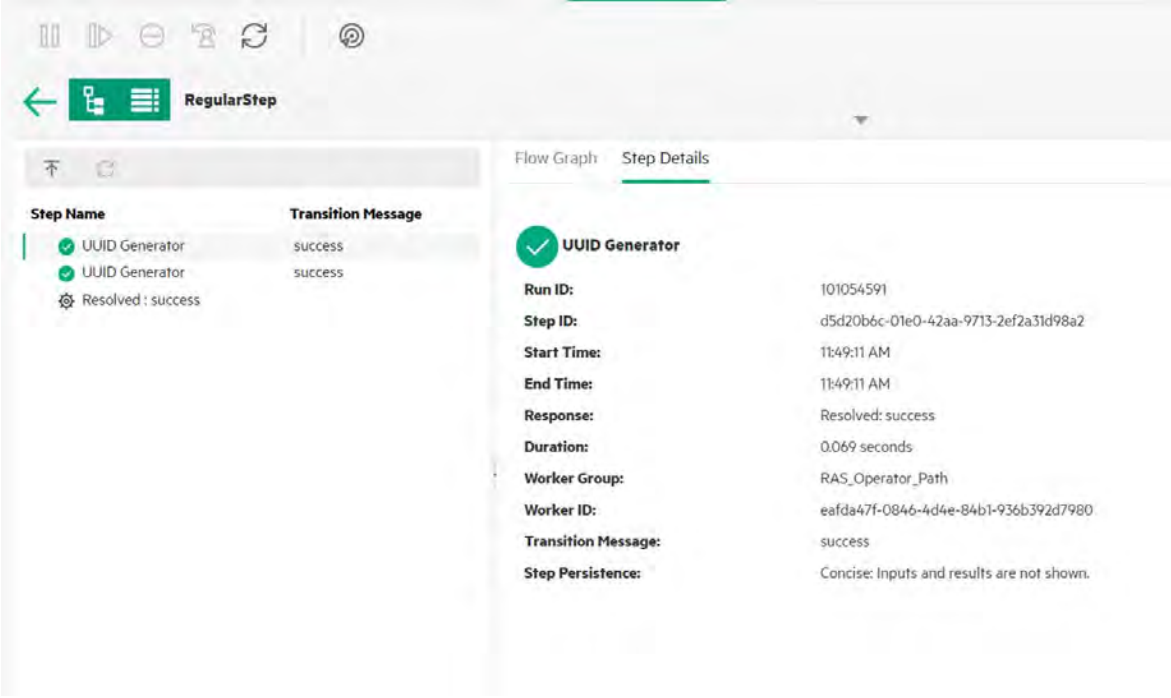
Testing and Troubleshooting a Flow Run

After you have triggered a flow run, you can drill down into the run and investigate its progress in detail. This is particularly helpful if a flow run fails.

When you click the **Drill down** button at the end of a row in the Run Explorer, this displays detailed information about the selected run.

> **Note:** Alternatively, you can double-click a row to drill down to see this information.

To the left, the Run Tree displays the steps in the run and the transition messages. The Run Tree highlights the currently running step and displays information about the progress of the run. If a flow fails, you will see at which step the failure occurred. The steps and transition messages are automatically refreshed as the run progresses. While a flow is running, if you click a step, the auto-refresh will pause. In order to resume you need to click the **Resume Refresh** button.

There are a number of collapsible views that you can use to display different kinds of information:

- **Information about the flow**—UUID, flow name, path, description, flow inputs, flow outputs, and so on.

  Click the **Down** arrow in the center of the Run Explorer drill down view toolbar.

- **Information about a selected step in the flow**—start time, end time, step result, duration, inputs, results, worker group, the ID of the worker that executed the step, and so on. Click the **Step Details** tab.

  The **Step Details** tab displays details about the step that is currently running. If you select a step in the Run Tree, the **Step Details** tab displays details about the selected step.

  The Step Details tab may contain Concise or Detailed information about the step. The information shown depends on the option selected for the step in the Advanced tab of the Step Inspector. See "Setting Step Results" on page 520 "**Step Inspector > Advanced tab**" in the *OOStudio Authoring Guide* for details.

  > **Note:** If any of these details were defined as sensitive data, they will be encrypted and will appear as asterisks.
  >
  > All sensitive data is encrypted in-memory and in the database during the run and later. The event is persistent and is presented as asterisks in the UI.
  >
  > This property is propagated between elements. For example, if a step result was marked as sensitive data, the step result passes this behavior to every input/step result that is assigned from it.
  >
  > This includes sensitive data returned by a scriptlet.
  >
  > For information about how to define data as sensitive, see "Creating Outputs and Results" in the *OO Studio Authoring Guide*.

  > **Note:** You can drag the slider on the left edge of the **Flow Info** pane to adjust its width.

- **A graphical view of the flow**, with the currently running step highlighted. Click the **Flow Graph** tab.

  The Flow Graph highlights the step that is currently running. If you select a step in the Run Tree, the Flow Graph highlights the selected step.

  > **Note:** You can adjust the way that the flow graph is displayed, by dragging it across the screen to view hidden item, or by using the zoom buttons to adjust the size of the image.

- **The Run Log**—a table with information about the entire run (steps, start time, duration, inputs, results, and so on), with the option to find a specific step.

Use

Click the **Run Log** button to display the **Run Log** pane.

> **Note:** If the run includes subflows, the steps in the subflows are indented in the run log.

Click the **Run Tree** button to collapse the **Run Log** pane and show the Run Tree.

> **Note:** You can click the button in the toolbar to toggle between viewing a selected step in the Run Tree and in the Run Log.

> **Note:** There are also cases where a run has no drill down details:
>
> - If a run comes from a content upgrade from version 9.x.
> - If the run details were cleared in the system.

What do you want to do?

Display details about the steps in a flow run

1. Click the **Run Management** button to display the Run Management workspace.
2. Click the **Run Explorer** tab.
3. Select the flow run that you want to look at in detail.
4. Click the **Drill down** button at the end of the row to display the drill down information about the selected run.

   To the left, the Run Tree displays the steps in the run and the transition messages. The Run Tree highlights the currently running step and displays information about the progress of the run. If a flow fails, you will see at which step the failure occurred.

   > **Note:** The steps and transition messages are automatically refreshed as the run progresses. While a flow is running, if you click a step, the auto-refresh will pause. In order to resume you need to click the **Resume Refresh** button.

5. When you click a step in the Run Tree, you display information about that step in the **Run Info** pane on the right.

In the **Run Info** pane, click the **Step Details** tab to display information about the currently running step, such as the start time, duration, inputs, results, description, worker group, the ID of the worker that executed the step, and so on.

If you select a step in the Run Tree, the **Step Details** tab displays details about the selected step.
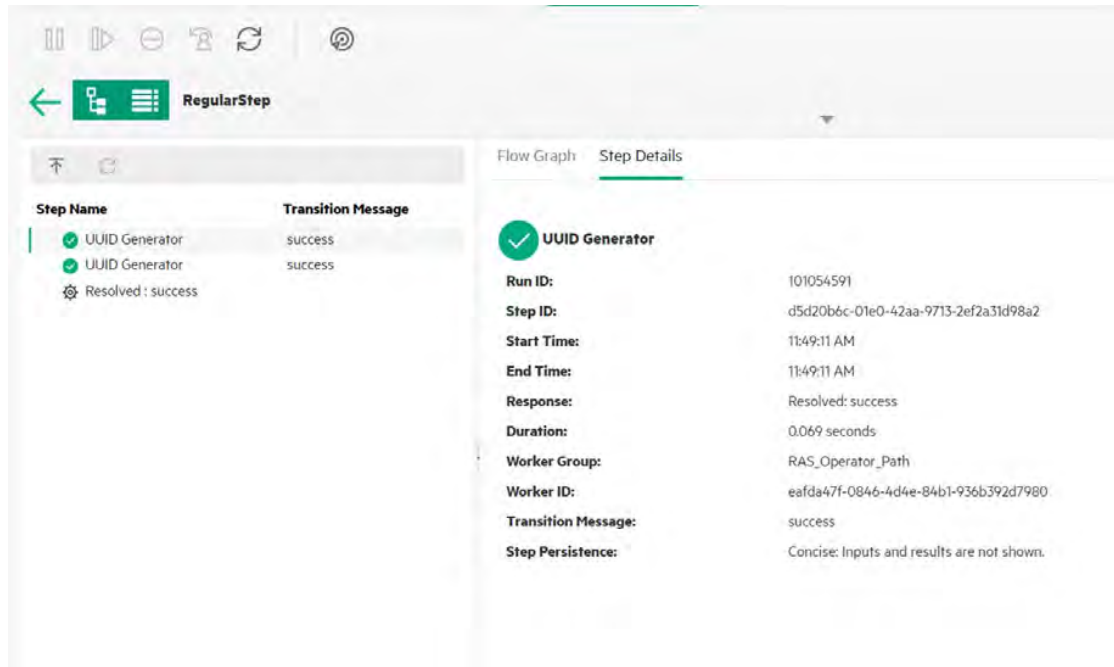
> **Note:** The fields displayed in this section will vary between different steps, depending on whether they include inputs, results, and so on.

6. (Optional) You can click the button in the toolbar to view the selected step in the Run Log.

7. Click the **Back** button to return to the main **Run Explorer** window.

Display the graph of a flow run

You can display a graphical image of a flow run, and highlight the position of the currently running (or selected) step.

1. Click the **Run Management** button to display the Run Management workspace, and click the **Run Explorer** tab.

2. Select the flow run that you want to look at in detail, and click the **Drill down** button at the end of the row.

3. Click a step in the Run Tree on the left.

4. In the **Run Info** pane on the right, click the **Flow Graph** pane to display a graphical view of the flow, with the currently running step highlighted.

   If you select a step in the Run Tree, the Flow Graph highlights the selected step.
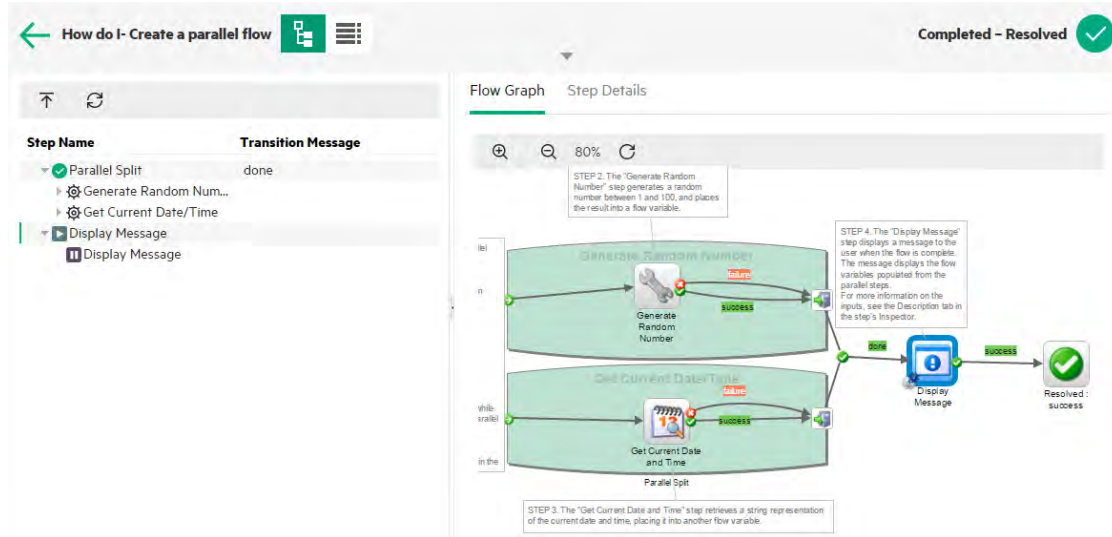
5. You can adjust the way that the flow graph is displayed:

   ○ Drag the flow graph across the screen, to display elements that are not visible.

   ○ Click the zoom buttons to increase or decrease the size of the flow graph.

   ○ Click the **Reset** button to reset the view to the default size and position.

Display the Run Log with details of the entire run

The Run Log is a table with information about the entire run (steps, start time, duration, inputs, results, and so on). This information is useful for identifying issues about a run, while troubleshooting.

The Run Log displays a maximum of 500 runs in one page, in order to meet performance requirements. Runs with over 500 steps are displayed in multiple pages. You can use the paging functionality to navigate through the pages.

1. Click the **Run Management** button to display the Run Management workspace, and click the **Run Explorer** tab.

2. Select the flow run that you want to look at, and click the **Drill down** button at the end of the row.

3. Click the **Run Log** button to display the **Run Log** pane, with details about the entire run.

> **Note:** If the run includes subflows, the relevant steps are indented.

4. For runs of over 500 steps, which are displayed in multiple pages, navigate through the pages:

   ○ Click the arrow buttons

   ○ Enter the number of a specific page

5. To hide or display columns, click the **Select Columns** button to display the column picker. Select the check box next to a column name to display that column or clear the check box to hide it.

   > **Note:** When you hide a column, this does not affect the results if you run a search.

6. You can click the **Run Tree** button to collapse the **Run Log** pane and show the Run Tree. Or click the button in the toolbar to view the selected step in the Run Tree.
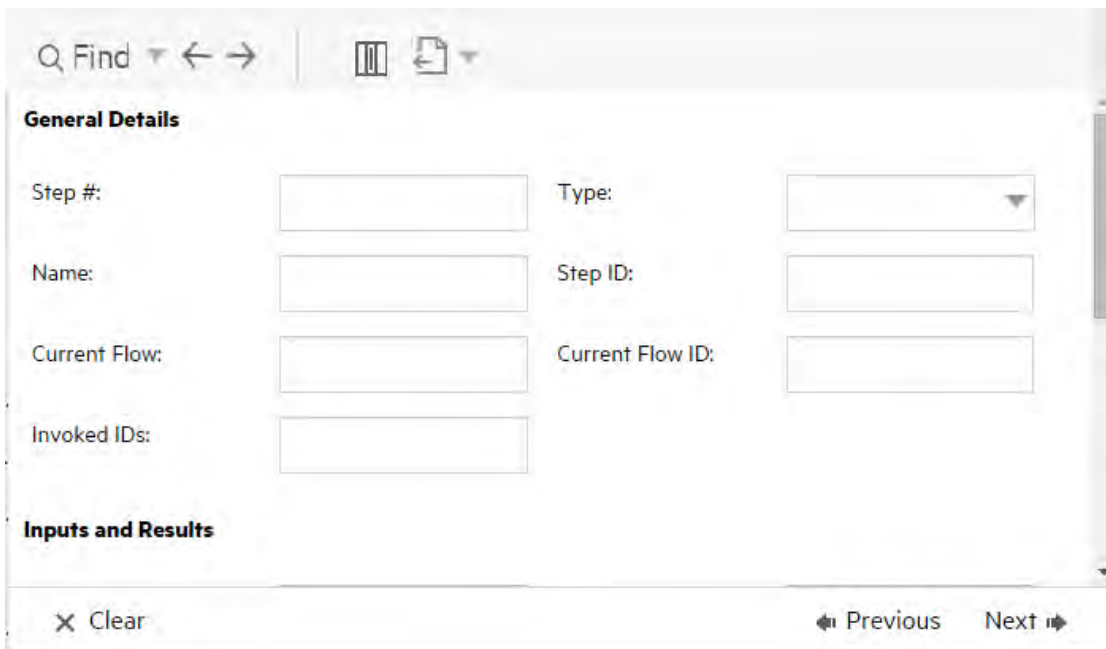
Find a step in the Run Log

In a long run, it may be difficult to find a particular step. Use the **Find** button to help locate the step that you want to see.

> **Note:** Searching in the Run Log is only supported if the storage space taken up by the searched field is no more than 4,000 bytes. Note that Japanese and Chinese take up about 3 bytes per character, while other (non-English) languages take up about 1.5 bytes per character. Also note that in the case of transition descriptions, the 4,000 bytes are shared among all of the locales in which the description is available.

1. Click the **Run Management** button to display the Run Management workspace, and click the **Run Explorer** tab.

2. Select the flow run that you want to look at, and click the **Drill down** ❯ button at the end of the row.

3. Click the **Run Log** button to display the **Run Log** pane.

4. Click the **Find** button to display the Find dialog box.

5. Enter the search criteria, scrolling down to see those that are hidden.

   > **Note:** This is not a filter. Steps that correspond to the search criteria are highlighted, but the unselected steps are displayed, so that you can see the context of the highlighted steps.

6. Click **Next** to highlight, in turn, each step that corresponds to the search criteria.

7. If desired, you can close the Find dialog box and use the **Next** and **Previous** ← → arrows in the toolbar to navigate across the search results.

Toggle between viewing a step in the Run Tree + Flow Graph and in the Run Log

You can move in either direction:

- While a step is selected in the Run Tree, you can jump to that same step in the Run Log to display more information about that step.

- While a step is selected in the Run Tree, you can jump to that same step in the Run Log and Flow Graph, to see how the step fits into the overall picture of the flow.

1. Click the **Run Management** button to display the Run Management workspace, and click the **Run Explorer** tab.

2. Select the flow run that you want to look at, and click the **Drill down** ❯ button at the end of the row.

3. When you select a step in the Run Tree, click the ⊚ button to display the Run Log with the selected step highlighted.

4. When you select a step in the Run Log, click the ⊚ button to display the Run Tree with the selected step highlighted. The selected step is also highlighted in the Flow Graph.

Collapse the Run Tree

If the Run Tree is expanded, you can collapse the expanded steps.

1. Click the **Run Management** button to display the Run Management workspace, and click the **Run Explorer** tab.

2. Select the flow run that you want to look at, and click the **Drill down** ❯ button at the end of the row.

3. Click the steps in the Run Tree, in order to expand them, and display subflows and lanes.

4. Click the **Collapse All** ⊼ button to collapse the Run Tree and display only the top level of the steps.

Adjust the width of the Run Info pane

You can use the slider at the edge of the **Run Info** pane to adjust its size.

1. Click the **Run Management** button to display the Run Management workspace, and click the **Run Explorer** tab.

2. Select the flow run that you want to look at, and click the **Drill down** ❯ button at the end of the row.

3. Drag the edge of the slider to adjust the width of the **Run Info** pane.
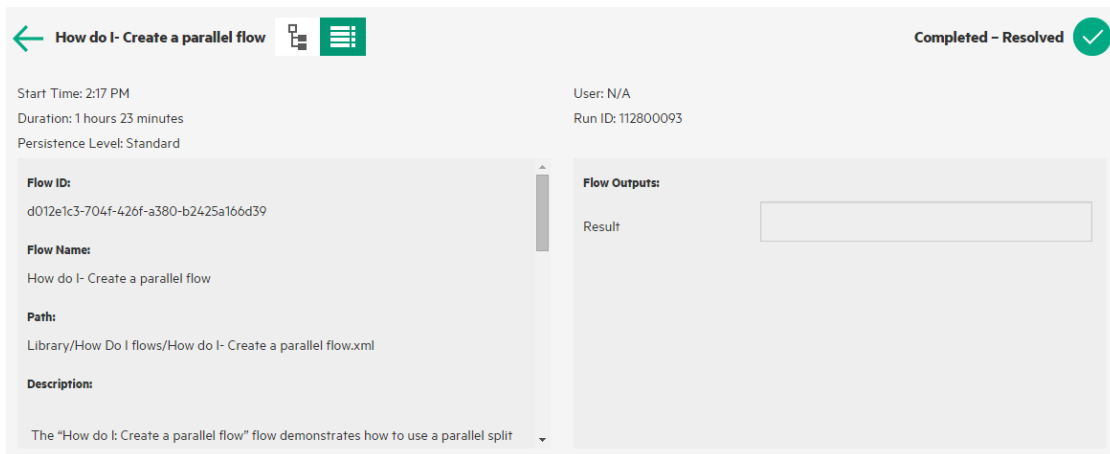
4. Use the open/close button in the slider to toggle between expanding and collapsing the pane.

Display Details about the flow

While drilling down to a flow run, you can view details about the original flow, such as flow UUID, flow name, path, description, flow inputs, flow outputs, and so on.

1. Click the **Run Management** button to display the Run Management workspace, and click the **Run Explorer** tab.

2. Select the flow run that you want to look at, and click the **Drill down** ⟩ button at the end of the row.

3. Click the **Down** arrow
   ▼ to the right of the flow run name, to expand the **Flow Info** pane.

4. If you can't see all the text in the pane, use the slider to display the text that is lower down in the pane.

5. Click the **Up** arrow ▲ ⌃ to collapse the **Flow Info** pane.

Export the Run Log as a text file

You can export the Run Log into a text file, listing all the steps' logs, which occurred in the flow run. This can help identify the causes of a problem. It may be useful to send the this file to the flow author, when a flow needs fixing.

1. Click the **Run Management** button to display the Run Management workspace, and click the **Run Explorer** tab.

2. Select the flow run that you want to look at, and click the **Drill down** ❯ button at the end of the row.

3. Click the **Run Log** button to display the **Run Log** pane.

4. Click the **Export** button and from the dropdown, select .

    A button with the name of the text file appears at the lower edge of the OO window. You can click it to open the file in your default text editor.

5. You can send the text file to the flow author, when requesting a change to a flow.

Export the Run Log as a CSV file

You can export the Run Log into a CSV file, listing all the steps' logs, which occurred in the flow run. This can help identify the causes of a problem. It may be useful to send the this file to the flow author, when a flow needs fixing.

> **Note:** In order to open the CSV file in Excel, make sure that Excel is configured correctly for character encoding. For more information, see http://superuser.com/questions/280603/how-to-set-character-encoding-when-opening-excel.

1. Click the **Run Management** button to display the Run Management workspace, and click the **Run Explorer** tab.

2. Select the flow run that you want to look at, and click the **Drill down** ❯ button at the end of the row.

3. Click the **Run Log** button to display the **Run Log** pane.

4. Click the **Export** button and from the dropdown, select .

    A button with the name of the CSV file appears at the lower edge of the OO window. You can click

    it to open the CSV file in Excel.

5. You can send the CSV file to the flow author, when requesting a change to a flow.

Reference Material

Run Explorer drill down view toolbar

When you drill down to see the details of a flow run, the toolbar at the top remains constant, and the details displayed below vary according to your selections.
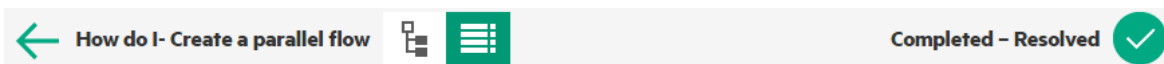
GUI item	Description
Back ← **button**	Click to return to the main **Run Explorer** window.
Run Tree [icon] **button**	Click to display the **Run Tree** pane, when the Run Log is displayed.
[icon] **button** **Run Log**	Click to display the **Run Log** pane, when the Run Tree is displayed.
Run name	Displays the name of the flow run.
Down arrow ▼	Click to expand the **Flow Info** pane.
Status	Displays the status of the flow run: Success [✓ icon] or Error [✗ icon]

Run Tree

When you drill down to see the details of a flow run, the Run Tree appears on the left. The Run Tree displays all the steps in the flow run and their transition messages. The action toolbar at the top enables you to perform actions on the run.

GUI item	Description
Action Toolbar	The action toolbar shows different buttons, depending on whether the Run Log or the Run Tree is displayed.
Pause button	Click to pause the selected run. The run will stop until it is resumed. This button is only available for runs that are currently running.
Resume button	Click to resume the run. This button is only available for runs that are paused.
Cancel button	Click to cancel the run.
Reassign button	Click to reassign the run to a different user.
Refresh button	Click to refresh the run.
Go to Log Tree button	While a step is selected in the Run Tree, jump to that same step in the Run Log.
Collapse All button	Collapse the Run Tree and display only the top level of the steps.
Resume Refresh button	While a flow is running, if you click a step, the autorefresh will pause. In order to resume you need to click the **Resume Refresh** button.

Step Name	Displays the name of each step in the flow run. Next to each step, an icon displays the step's status. Click a step to display its details in the **Flow Info** pane to the right.
Transition Message	Displays the transition message for the step, if one exists.

Run Info > Step Details tab (Concise Step Persistence)

Click the **Step Details** tab to display information about the currently running step. If you select a step in the Run Tree, the **Step Details** tab displays details about the selected step.

The fields displayed in the **Step Details** tab may vary between different types of step.

GUI item	Description
Run ID	Displays the UUID of the selected run.
Step ID	Displays the UUID of the selected step.
Start Time	Displays the time when the selected step started.
End Time	Displays the time when the selected step ended.
Response	Displays the response of the selected step, if one exists.
Duration	Displays the duration of the selected step.

Description	Displays the description of the selected step, if one exists.
Workers Group	Displays the worker group of the selected step, if one exists.
Worker ID	Displays the UUID of the worker that executed the selected step, if one exists.
Transition Message	Displays the message that appears during the transition of the selected step, if one exists.
Step Persistence	Displays the persistence level for this step as defined in the Advanced tab in the Step Inspector in Studio. For Concise, the following text appears: `Concise: Inputs and results are not shown`

Run Info > Step Details tab (Detailed Step Persistence)

Click the **Step Details** tab to display information about the currently running step. If you select a step in the Run Tree, the **Step Details** tab displays details about the selected step.

The fields displayed in the **Step Details** tab may vary between different types of step.
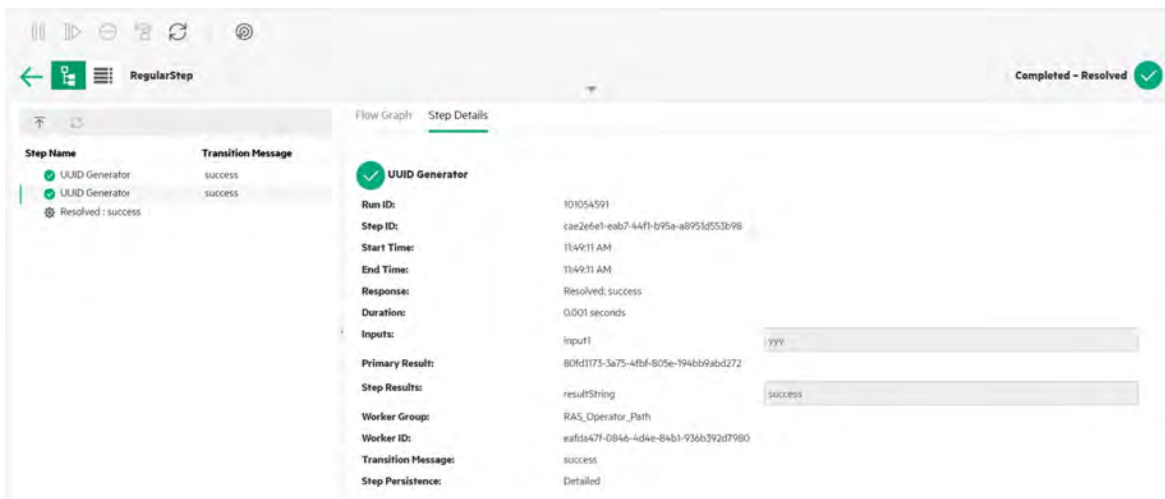
GUI item	Description
Run ID	Displays the UUID of the selected run.
Step ID	Displays the UUID of the selected step.
Start Time	Displays the time when the selected step started.
End Time	Displays the time when the selected step ended.
Step Response	Displays the response of the selected step, if one exists.
Duration	Displays the duration of the selected step.

| Inputs | Displays the inputs of the selected step, if these exist.

Note: If these inputs were defined as sensitive data, they will be encrypted and will appear as asterisks. |
| --- | --- |
| Primary Result | Displays the primary result of the selected step, if one exists.

Note: If the raw result is marked as sensitive, then the primary result will be encrypted and will appear as asterisks. |
| Step Results | Displays the results of the selected step, if any exist.

Note: If these results were defined as sensitive data, they will be encrypted and will appear as asterisks. |
Worker Group	Displays the worker group of the selected step, if one exists.
Worker ID	Displays the UUID of the worker that executed the selected step, if one exists.
Transition Message	Displays the message that appears during the transition of the selected step, if one exists.
Step Persistence	Detailed.

Run Info > Flow Graph tab

Click the **Flow Graph** tab to display a graphical view of the flow, with the currently running step highlighted. If you select a step in the Run Tree, the Flow Graph highlights the selected step.

GUI item	Description
Zoom In button	Click to increase the size of the flow graph.
Zoom Out button	Click to decrease the size of the flow graph.
Reset button	Click to reset the graph to the default size and position.

Run Log

Click the **Run Log** button to display the **Run Log** pane with details about the entire run. The action toolbar at the top enables you to perform actions on the run.

Ste...		Name	Start Time	Duration	Inputs	Raw Results		Transition	ROI	Current Flow
1		▾ Parallel Split	2:17 PM	1.303 seco...			✓	done	0	How do I- Cr...
2		▾ Generate Random Num...	2:17 PM	0.104 seco...						
3	⚙	Generate Random N...	2:17 PM	0.103 seco...	(max=100), (min=1)	(response=success), (returnRe...	✓	The "Generat...	0	How do I- Cr...
4		▾ Get Current Date/Time	2:17 PM	0.081 seco...						
5	⚙	Get Current Date an...	2:17 PM	0.080 sec...	(Field_3=${in3}), (Field_1=${in...	(response=success), (returnRe...	✓	The "Get Cur...	0	How do I- Cr...
6	⛬	▾ Display Message	2:17 PM	1 hours 23 ...	(message=Parallel split step c...		✓	The "Display ...	0	How do I- Cr...
7	⚙	Display Message	2:17 PM	1 hours 23 ...	(Field_3=), (field4=), (Field_4=...	(response=success), (returnRe...	✓	success	0	Display Mess...
8	↵	Resolved: success	3:40 PM	0.009 sec...						Display Mess...
9	↵	Resolved : success	3:40 PM	0.009 sec...						How do I- Cr...

GUI item	Description
Action Toolbar	The action toolbar shows different buttons, depending on whether the Run Log or the Run Tree is displayed.
Pause ‖ **button**	Click to pause the selected run. The run will stop until it is resumed. This button is only available for runs that are currently running.
Resume ▷ **button**	Click to resume the run. This button is only available for runs that are paused.
Cancel ⊖ **button**	Click to cancel the run.

Reassign button	Click to reassign the run to a different user.
Refresh button	Click to refresh the run.
Go to Run Tree view button	While a step is selected in the Run Log, jump to that same step in the Run Tree.
Find	Click to open the Find dialog box, to locate a step.
← →	After running a search, close the Find dialog box and use the arrows in the toolbar to navigate across the search results.
Export All to Text/CSV button	Click to export the run log into a text/CSV file, listing all the events that occurred in the flow run.
Step #	Displays the step number of each step
Icon	Displays the type of step. Icons represent operation, subflow, return step, and other.
Name	Displays the name of the step. If the step is part of a subflow, the step name is indented.
Start Time	Displays the time when each step started.
Duration	Displays the duration of each step.
Inputs	Displays the inputs of each step, if these exist.
Raw Results	Displays the results of each step. **Note:** If these results were defined as sensitive data, they will be encrypted and will appear as asterisks.
Response	An icon represents the response of each step: Resolved, Error, Diagnosed, No Action Taken.
Transition	Displays the description of the outgoing transition, or its name if there is no description.
ROI	Displays the ROI of the outgoing transition.
Current Flow	Displays the name of the flow containing the step. If there are subflows, this contains the name of the subflow containing the step.

User	Displays the run owner at the time the step was run.
Worker	Displays the UUID of the worker that executed the step.

Run Log > Find dialog box

Use the **Find** button in the Run Log to help locate a step that you want to see. Enter one or more search criteria, scrolling down to see those that are hidden. Steps that correspond to the search criteria are highlighted.

GUI item	Description
Clear	Click to clear the search criteria boxes.
Next	Click to highlight the next step that corresponds to the search criteria.
Previous	Click to highlight the previous step that corresponds to the search criteria.
← →	You can close the Find dialog box and use the arrows in the toolbar to navigate across the search results.

Flow Information

Click the **Down** arrow

▼ to the right of the flow run name, to expand the **Flow Info** pane.
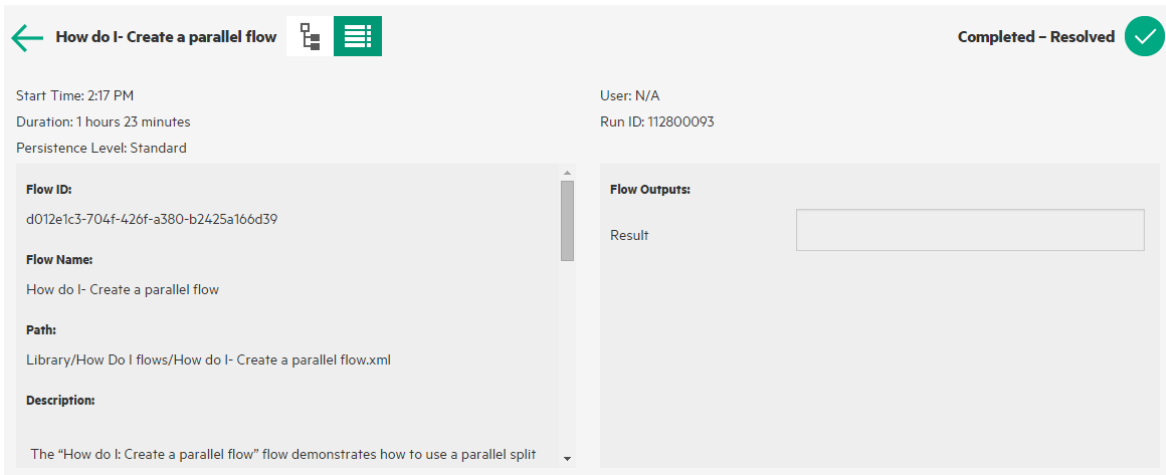
GUI item	Description
General details	Displays basic information about the run, such as start time, duration, user, run ID, and the persistence level, which specifies how the run history is saved to the database. For information about the persistence level, see "Monitoring and Controlling Database Size" on page 144.
Flow ID	Displays the UUID (unique identifier) of the flow.
Flow Name	Displays the name of the flow.
Path	Displays the path to where the flow is located.
Description	Displays the description of the flow, if one exists. If you can't see all the text in the pane, use the slider to display the text that is lower down in the pane.
Flow Inputs	Displays the flow inputs (if any exist).
Flow Outputs	Displays the flow output.
Up arrow ▲	Click to collapse the **Flow Info** pane.

Use

Embedding Central Views in External Web Pages

OO gives you the option to embed parts of the Central UI into an external application. You can embed:

- Flow Run view - for running and interacting with flows
- Drill Down view - for advanced tracking of running flows

To embed the view in your web page, use an iFrame element in the HTML with the appropriate URL.

In the example below, the **999** in **drilldown-ex.html#999** stands for the **<run_id>** value. For more information, see Run a flow in an embedded Flow Run view.

```
<!DOCTYPE html>

<html lang="en-US">

<head>

<style>

 iframe {

 width : 800px;

 height : 600px;

 }

</style>

<title>HTML iframe tag</title>

</head>

<body>

<iframe src="http://localhost:8080/oo/drilldown-ex.html#999"></iframe>

</body>

</html>
```

Important! If you are going to embed the Drill Down window into an external HTML page, it is recommended to have an SSO solution in place (for example, SAML 2.0). For more information, see "Setting Up Security – LWSSO" on page 110 and "Setting Up Security – SAML Authentication" on page 104.

> If an SSO solution has not been configured, an OO login screen will appear in the iframe, when it first loads inside the page. On some browsers (for example, IE11), the default privacy settings will need to be updated to allow third party cookies before users will be able to log into OO via the iframe.

What do you want to do?

Run a flow in an embedded Flow Run view

The Flow Run view lets you run a specific flow and perform minimal tracking as it runs.

After the flow is run, the basic tracking view appears.
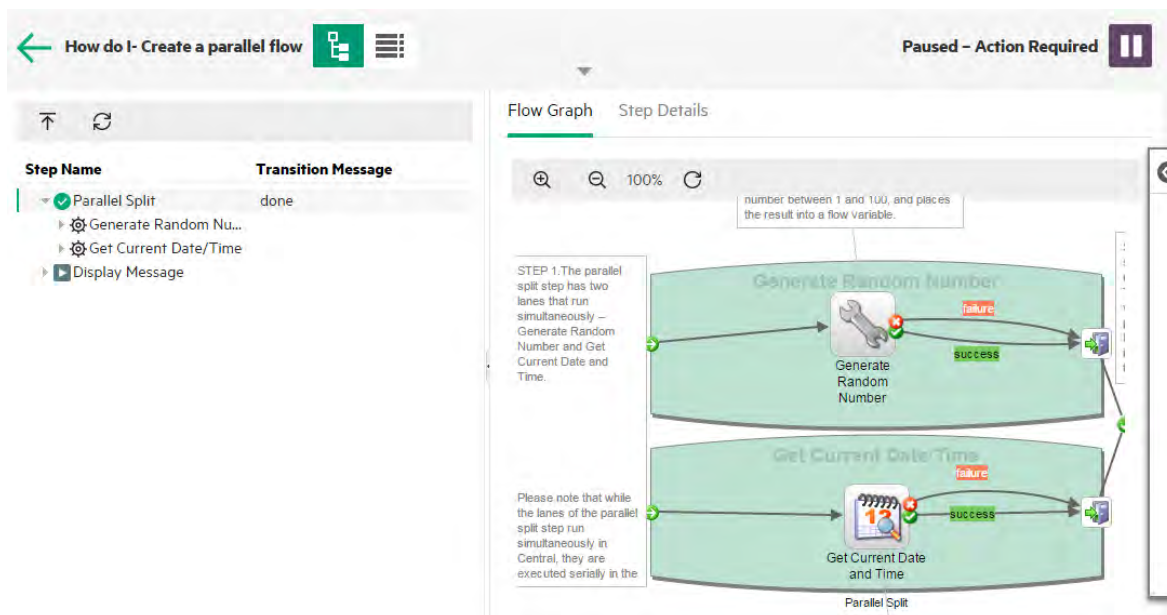
To run a flow in an embedded Flow Run view, use the following URL pattern:

```
http://<hos_name>:<port>/oo/trigger.html#<flow_uuid>
```

Identify the flow to run via the `<flow_uuid>`. You can find it in the flow's **Run Link** section in the Flow Library.

If required, use the `input_` and `runName` prefixes to edit the run link. You can get it to pass inputs or to change the name of the flow run.

The format is `input_<inputName>=<inputValue>` and `runName=<runNameValue>`. These values should come after the `?` and concatenate with `&`.

For example:

```
http://<hos_name>:<port>/oo/trigger.html#7a1fc3c7-1c5b-4868-a1c7-
2932d878897e?input_host=someHost&input_username=someUsername
```

Track a flow in an embedded Drill Down view

The embedded drill down view lets you track and interact with a running flow.

To track a flow in an embedded Drill Down view, use the following URL:

```
http://<hos_name>:<port>/oo/drilldown-ex.html#<run_id>
```

Identify the flow to track via the `<run_id>`. You can find it in the flow's **Run Link** section in the Flow Library.

Typical Central Errors

Typical Deployment Errors

Failed to read the file <cp jar name>. The file might be corrupt.

This error message appears when OO is unable to read a content pack.

Solution:

If the file was copied from another machine, try copying it again.

Otherwise, go to Studio and select **Create content pack**, and redeploy the content pack.

Failed to read the file <cp jar name>. The file is empty.

This error message appears when the content pack file is empty.

Solution:

If the file was copied from another machine, try copying it again.

Otherwise, go to Studio and select **Create content pack**, and redeploy the content pack.

Wrong file format. <cp jar name> should contain the Lib and Content root folders.

This error message appears when a content pack is missing the required **Lib** and **Content** folders.

Solution:

If the file was copied from another machine, try copying it again.

Otherwise, go to Studio and select **Create content pack**, and redeploy the content pack.

> **Note:** Do not edit the contents of the content pack manually.

Wrong file format. <cp jar name> is missing the 'contentpack.properties' file.

This error message appears when a content pack is missing a required file.

Solution:

If the file was copied from another machine, try copying it again.

Otherwise, go to Studio and select **Create content pack**, and redeploy the content pack.

> **Note:** Do not edit the contents of the content pack manually.

Missing '<flow or operation/type of configuration item>' with UUID '<uuid>', which is required by the '<flow/operation>': '<flow path/operation uuid>'.

This error message appears when there is a missing flow, operation, or configuration item.

Solution:

1. Go to Studio and search for the flow, operation, or configuration item with the given UUID in either the content packs or other projects in your repository.

2. Redeploy the deployment unit with the content pack that contains the flow, operation, or configuration item that you found.

Missing '<flow or operation/type of configuration item>' with UUID '<uuid>', which is required by the '<flow/operation>': '<flow path/operation uuid>' and <number of additional dependents> more . See the server log for the full list of dependents.

This error message appears when there is a missing entity (flow, operation, or configuration item) that other flows or operations depend upon.

Solution:

1. Check the server log to see all of the flows and operations that depend on this missing entity.

2. Go to Studio and search for a flow, operation, or configuration item with the given UUID in either the content packs or other projects in your repository.

3. Redeploy the deployment unit with the content pack that contains the flow, operation, or configuration item that you found.

An exception occurred during deployment. Check the server log for more details.

This error message may appear in a number of different situations, leading to an exception during deployment. This includes conflicts between two deployed configuration items with the same type and path.

Solution:

Check the server log to see the details of the exception. In the case of a configuration item path conflict, check the **general.log** file for the most detailed information.

The content pack that you are deploying has been deployed previously. In the previous deployment, this content pack contained an entity ('<entity type>' named '<entity name>' with UUID <uuid>), which is used by other deployed content packs (<names of one or 2 cps>). The current content pack that is being deployed is missing that entity.

This error message appears when you are deploying a content pack that has been deployed previously, and this content pack originally contained an entity that is used by other deployed content packs. However, the version that you are deploying is missing that entity.

Solution:

Option 1: Go to Studio and search for the flow, operation, or configuration item with the given UUID in a previous version of the content pack that you are deploying, and add it to the current version of the content pack.

Option 2: Go to Studio and in the dependent content pack(s), remove the dependency on the given flow, operation, or configuration item, and redeploy that/those content packs as well.

The content pack that you are deploying has been deployed previously. In the previous deployment, this content pack contained an entity (('<entity type>' named '<entity name>' with UUID <uuid>), which is used by other deployed content packs (<name of one of the cps> and <number of other cps> more. The current content pack that is being deployed is missing that entity. See the server log for the full list of dependent content packs.

This error message appears when you are deploying a content pack that has been deployed previously, and this content pack originally contained an entity that is used by other deployed content packs. However, the version that you are deploying is missing that entity.

Solution:

Check the server log to see all of the content packs that depend on this missing entity.

Option 1: Go to Studio and search for the flow, operation, or configuration item with the given UUID in a previous version of the content pack that you are deploying, and add it to the current version of the content pack.

Option 2: Go to Studio and in the dependent content pack(s), remove the dependency on the given flow, operation, or configuration item, and redeploy that/those content packs as well.

A flow with UUID <uuid> has <number of scheduled tasks> scheduled tasks that will be deleted.

This warning message appears when a flow is being deleted by deploying a new version of an already deployed content pack that contained flows with scheduled tasks.

Solution:

There is nothing to fix.

Schema validation error in '<flow name>'. <the xml parser exception>.

This error message appears when the schema is not correctly validated.

Solution:

Go to Studio and select **Create content pack** and redeploy the content pack.

> **Note:** Do not edit the contents of the content pack manually.

A flow with a duplicated path is being deployed. This path already exists either in a flow in another content pack in the deployment unit, or in a flow that was deployed previously. To see the error details, run the deployment again with the server log level set to DEBUG.

This error message appears when you are deploying a content pack with a flow that has the same path as another flow in another content pack in the deployment unit.

Solution:

1. To see the details about the flow path, flow UUID, and content pack name, run the deployment again with the server log level set to DEBUG. You will receive one of the following errors:

   ○ A flow with a duplicated path: '<flow path>' is being deployed. A flow with this path has previously been deployed.

   ○ A flow with a duplicated path: '<flow path>' is being deployed. A flow with this path exists in the content pack '<name of content pack>'.

2. In Studio, modify one of the content packs so that there is no longer a duplication, and redeploy the content pack. For more information about the required changes, see the sections about these error messages, below.

A flow with a duplicated path: '<flow path>' is being deployed. A flow with this path has previously been deployed.

This error message appears when you are deploying a content pack with a flow that has the same path

as a flow that was previously deployed.

Solution:

In Studio, change the path of the flow (change the name of one of the folders or the flow name) and redeploy the content pack.

A flow with a duplicated path: '<flow path>' is being deployed. A flow with this path exists in the content pack '<name of content pack>'.

This error message appears when you are deploying a content pack with a flow that has the same path as another flow in another content pack in the deployment unit.

Solution:

In Studio, change the path of the flow (change the name of one of the folders or the flow name) and redeploy the content pack.

A '<flow or operation/type of configuration item>' with a duplicated UUID: <uuid> is being deployed. This UUID already exists in a '<flow or pperation/type of configuration item>'in the content pack '<name of content pack>'.

This error message appears when you are trying to deploy a content pack with a flow, operation, or configuration item that has the same UUID as an item in another content pack that is deployed in the deployment unit.

Solution:

1. In Studio, search for the flow, operation, or configuration item with the given UUID.
2. Copy the flow, operation, or configuration item into the same project.
3. Delete the original copy of the flow, operation, or configuration item, in that same project.
4. Select **Create content pack**, and redeploy the content pack.

Note: Do not edit the contents of the content pack manually.

The property <content.pack.name/content.pack.version> is missing in the properties file contentpack.properties.

This error message appears when the content pack name or version are missing from the content pack's properties file.

Solution:

Go to Studio and select **Create content pack**, and redeploy the content pack.

Service unavailable. The server may be down, too busy, or experiencing network problems. Check the server log for the deployment status.

This error message appears when the server is either too busy to handle the request or is down for maintenance, or when some network error has prevented the response from arriving to the client.

Solution:

Check the server log to see if the deployment didn't start or end successfully, or failed.

During deployment the file name became corrupted. Please try to redeploy.

In some cases, the file name arrives in a corrupt state, and causes an exception.

Solution:

Deploy again.

Typical Central Errors

You do not have permission to perform this action.

This error message appears when you are logged in as a user that does not have permission to perform the action.

Solution:

Log in as a user with the appropriate permission. Alternatively, ask the administrator to give your user permission to perform the action.

An error has occurred. Check the server log for more details.

This error message appears when there is an unexpected error status from the server, but no error status text was received.

Solution:

Check the server logs to see if there was an exception.

liquibase: Waiting for changelog lock.

After a few attempts to log into the system, the server does not start and this error message appears.

Solution:

In the **DATABASECHANGELOGLOCK** table, set the value of the **LOCKED** column to **0**.

<server response status (for example, Internal Server Error)>. Check the server log for more details.

This error message appears when there is an unexpected error status from the server (for example, "Out of memory").

Solution:

Check the server logs to see if there was an exception.

Authoring Using Studio

This document shows you how to use OO Studio to author flows and create content packs.

Welcome to Operations Orchestration Studio Authoring Guide

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

Visual Overview of OO Studio
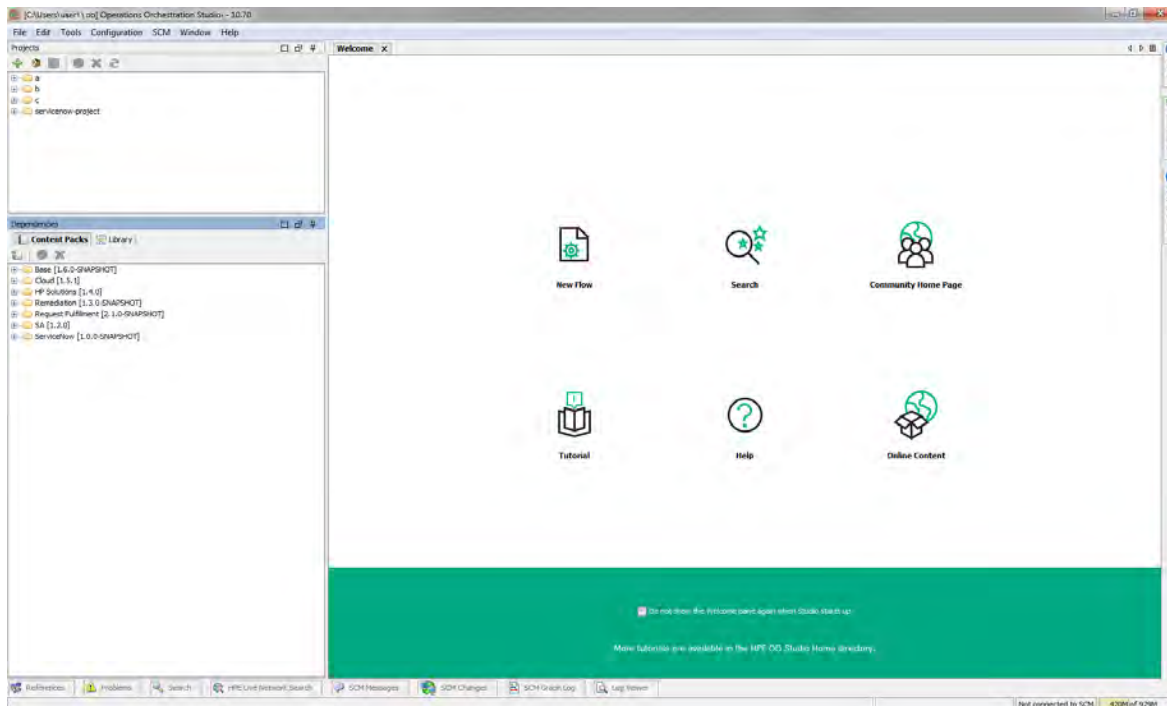
The main elements of Studio are:

- **Projects** pane (on the left), which shows the project you're working in, and displays the editable flows, operations, and other OO objects that you can use in the project. Projects are listed in alphabetical order.

- **Dependencies** pane (on the left), which includes the imported content packs. In this pane, you can import, delete and close the content packs. The **Dependencies** pane contains two tabs:

  - **Content Packs** - Displays multiple trees, for multiple content packs. From this view, it is possible to close, delete, or import a content pack.

  - **Library** - Displays a single tree, with all of the content merged together under a general **Library** folder. From this view, it is possible to import a content pack.

- **Authoring** pane (in the center). When a flow is opened in the authoring pane, the following three tabs are available at the bottom of the authoring pane:

  - **Design** tab, where you can work on the flow diagram

  - **Properties** tab, where you can set the properties for flows, operations, and configuration objects

  - **Inspector** tab, where you can set the properties for individual steps and transitions (only available when the **Design** tab is open)

- **Welcome** tab (in the center). When you first open Studio, the **Welcome** tab is displayed in the authoring pane.

- **Icons** pane (on the right), which contains collections of icons that you use for operations or steps. Open this pane by clicking the **Icons** tab.

- **Bookmarks** pane (on the right), where you can store shortcuts to favorite operations and flows. Open this pane by clicking the **Bookmarks** tab.

- **Flow Variables** pane (on the right), which shows the flow variables that are used in the flow and lists and describes how each flow variable is used. Open this pane by clicking the **Flow Variables** tab.

- **References** pane (on the bottom), which shows how flows and operations are used in existing flows. Open this pane by clicking the **References** tab.

- **Problems** pane (on the bottom), which displays problems with a selected flow or operation. Open this pane by clicking the **Problems** tab. See "Problems pane" on page 298 for more information.

- **SCM Messages** pane (on the bottom), displays source control related messages. Open this pane by clicking the **SCM Messages** tab. See "Working with Source Control in Operations Orchestration Studio" on page 320 for more information.

- **SCM Changes** pane (on the bottom), displays the latest source control changes. Open this pane by clicking the **SCM Changes** tab. See "Working with Source Control in Operations Orchestration Studio" on page 320 for more information.

- **SCM Repository Log** pane (on the bottom), displays a log of the changes made to the SCM repository (for Git only). Open this pane by clicking the **SCM Repository Log** tab. See "Working with Git Source Management System " on page 346 for more information.

- **Log Viewer** pane (on the bottom), displays all the errors that have occurred in the current user session. Open this pane by clicking the **Log Viewer** tab. See "Viewing Studio Errors in the Log Viewer " on page 691 for more information.

  Note: The **Log Viewer** tab appears only after you enable it by selecting **Studio Log Viewer** from the **Window** menu.

- **Search** pane (on the bottom), which enables you to search for a flow, operation or configuration item. Open this pane by clicking the **Search** tab.

- **Live Network Search pane** (on the bottom), allows you to search for relevant information and content on ITOM Marketplace based on their profile. access permissions directly from Studio. See "Searching Content on ITOM Marketplace from Studio" on page 589 for more information.

- **Status bar** (on the bottom), shows the source management system the current project is connected to (Git or SVN) and its current state. In addition, the total heap size available and current heap size usage are shown.

> **Note:** If you cannot see the entire Studio screen, as it appears above, this may be a screen resolution issue. When working in Studio, you should set your screen resolution to a minimum of 1280x1024 pixels.

Reference Material

Projects pane

The **Projects** pane contains the project tree—a hierarchical folder structure containing the editable content of your project:

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

  > **Note:** You can create folders in all the configuration items under the existing Configuration Item folder structure.

- The project properties

GUI item	Description
New Project ✚	Creates a new project.
Import Project	Browse to and import an existing project from a different workspace.
Create Content Pack	Creates a content pack from the selected project.
Delete ✖	Permanently deletes the selected project from the workspace.
Open	Opens the currently selected closed project.
Close	Closes the currently selected project, so that it is grayed out.
Refresh	Refreshes the files in the currently select project.

For information about working with projects, see "Working with Projects" on page 305.

Dependencies pane

The **Dependencies** pane displays the available content packs, with folders that contain the operations and flows.

The Dependencies pane includes two view options: **Content Packs** tree view and **Library** which is an aggregated view of all the imported content packs.

When you switch the view to the **Library** pane, all the items are merged under the same tree. In addition all the configuration items are merged under a general **Configurations** folder. All the items in the tree will be merged into the same folder if they have a common path.

The next time you open Studio, it automatically opens the last view selected.

- **Content Packs**: Displays multiple trees, for multiple content packs. From this view, it is possible to close, delete, or import a content pack. When you right-click on an item, a drop-down menu with the available options for this view appears.

- **Library**: Displays all the library and configuration items, (including all the folders and sub-folders). From this view, it is possible to import a content pack. Content packs that were removed or deleted from the **Content Packs** view will automatically update the **Library** view. When you right-click on an item, a drop-down menu with the available options for this view appears.
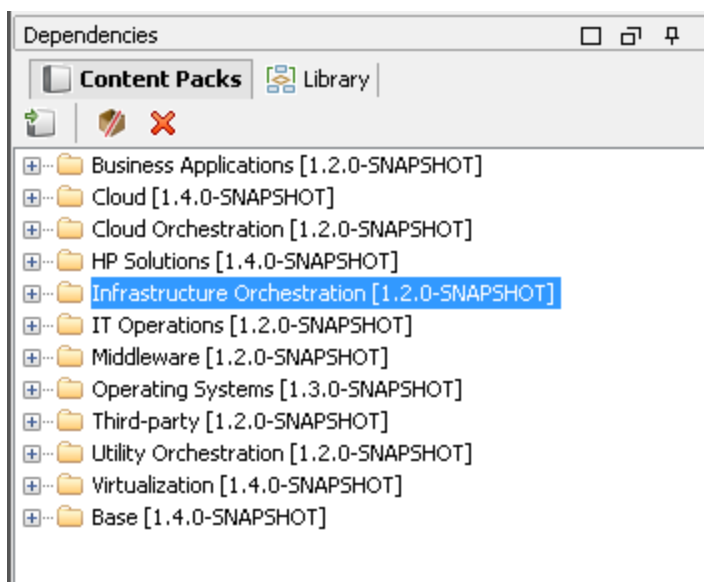
Authoring pane

The authoring pane is the central area in Studio, where you work on flow diagrams, adding steps and the connections between them, and setting properties that determine how flows work.

When a flow is opened in the authoring pane, the following three tabs are available:

- **Design** tab, for working on the flow diagram, adding steps and the connections between them

- **Properties** tab, to display the **Properties** sheets, where you can set the properties for flows and operations, as well as configuration objects such as selection lists, filters, and scriptlets.

- **Inspector** tab, to display the Inspector, where you can set the properties for individual steps and transitions
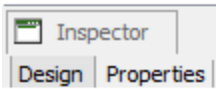
Authoring pane toolbar

When a flow is opened in the authoring pane, and the **Design** tab is open, the authoring pane toolbar is available.

The authoring pane toolbar buttons provide shortcuts for a number of tasks.
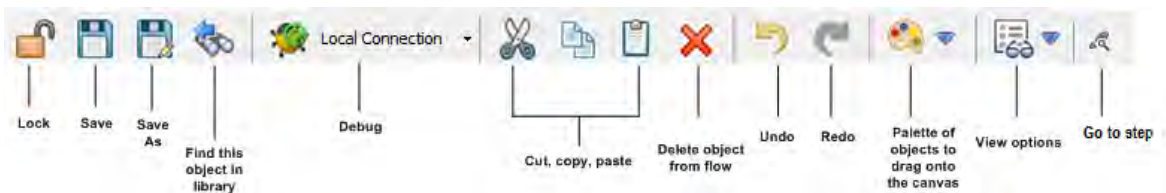
Button	What it does
Step Palette	Opens the **Step** palette to drag step objects onto the canvas
View Options	Opens the **View Options** palette
Find this object in Library	Expands the Library tree to select the flow or operation that you're working on
Debug Flow	Adds or edits Central debugger connections.
Go to Step	Lets you jump to a specific step in the flow. Type the name of the step to jump to it, or the first letters of the step to select it from a list.

For information about working with the authoring pane, see "Authoring a Flow – Basics" on page 451.

Welcome tab

When you first open Studio, the **Welcome** tab is displayed in the authoring pane. If it has been closed, select **Help** > **Show Welcome Panel** to reopen it.
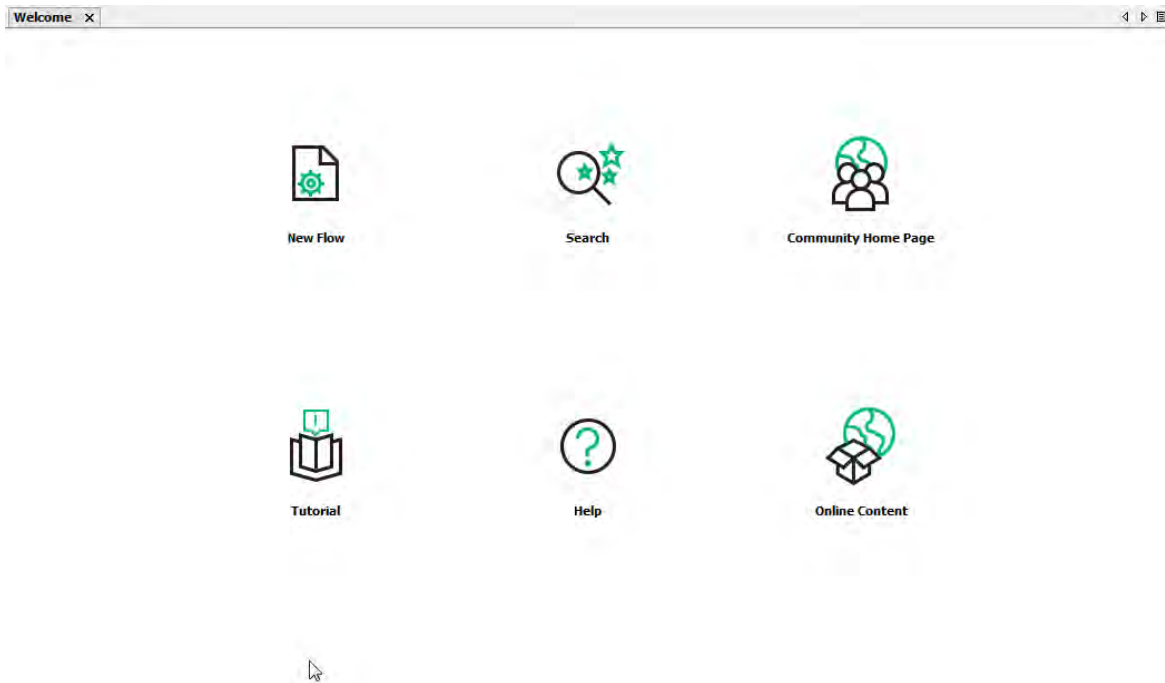
Button	Description
New Flow	Click to create a new flow from a predefined template.
Search	Click to open the **Search** pane, so you can search for flows from the content repository.
Community Home Page	Click to go to the ITOM Marketplace Community Home Page.
Tutorial	Click to view the OO tutorials.
Help	Click to open the OO Help.
Online Content	Click to go to the OO Content Catalog download page on ITOM Marketplace.
Do not show the Welcome pane again when Studio starts up	Select this check box to specify that the Welcome page will not be shown the next time Studio is opened.

Note: Links to the OO Help, ITOM Marketplace Community home page and the online content are also available from the **Help** menu.

For more information about creating a new flow from a predefined template, see "Creating a New Flow" on page 455.

For more information about searching for a flow, see "Finding a Flow or Operation" on page 642.

Step palette

The **Step** palette contains buttons for dragging return steps, parallel split steps, multi-instance steps, and callouts onto the flow. Display the **Step** palette by clicking the **Step Palette** button from the authoring pane toolbar.

Button	Description
Success	Lets you drag a **Success** return step to the flow.
Diagnosed	Lets you drag a **Diagnosed** return step to the flow.
No Action Taken	Lets you drag a **No Action Taken** return step to the flow.
Failure	Lets you drag a **Failure** return step to the flow.
Parallel Split Step	Lets you drag a parallel split step to the flow.
Multi-instance Step	Lets you drag a multi-instance step to the flow.
Callout	Lets you drag a callout to the flow, providing information to users.
Docking bar	Click to dock and undock the palette.

For information about working with return steps, see "Creating Return Steps" on page 553.

For information about parallel split steps and multi-instance steps, see "Advanced Authoring" on page 557.

View Options palette

The **View Options** palette contains buttons for changing the appearance of the flow on the authoring pane. Display the **View Options** palette by clicking the **View Options** button from the authoring pane toolbar. Using the Zoom bar, you can zoom into the flow up to 300%.

Button	Description
Show/Hide Labels	Shows or hides response labels on objects
Show/Hide Connected Response Icons	Shows or hides response icons on objects
Align Selection Horizontally	Aligns selected steps horizontally
Align Selection Vertically	Aligns selected steps vertically
Bring to Front	Moves the selected object to the front of the stack
Send to Back	Moves the selected object to the back of the stack
Show/Hide Grid	Reveals the authoring pane grid, which you can use for arranging steps. When you stop dragging a step, it snaps to the nearest position on the grid.
Docking bar	Click to dock and undock the palette.

For information about working with the view options, see "Adjusting the Appearance of a Flow" on page 464.

Object Properties sheets

The **Properties** sheets for flows, operations, and configuration objects are the editors in which you add, remove, or change values for the objects. For most objects in the Library, the **Properties** sheet is the interface you use to work with the object. In addition to the fields that you can edit, **Properties** sheets

provide the UUID and information about the version of the object.

After you modify the properties of an operation in the **Properties** sheet, the changes affect all steps that you create from this operation, including steps that were created earlier from this operation.

- To display the **Properties** sheet for a flow, open the flow in the authoring pane and click the **Properties** tab.

- To display the **Properties** sheet for an operation or configuration object, right-click the operation or object in the Library and select **Properties**.

Input	Required	Type	Assign From	Otherwise		Assign To	
host	✓	⬭	host	Prompt user from the text	A…	host	
baseDN	✓	⬭	baseDN	Prompt user from the text	A…	baseDN	
DN	✓	⬭	DN	Use the constant: CN=Users,${baseDN}	A…	DN	
groupDN	✓	⬭	groupDN	Use the constant: CN=Domain Admins,CN…	A…	groupDN	
daysOld	✓	⬭	daysOld	Use the constant: 30	A…	daysOld	
username		⬭	username	Prompt user from the text	A…	username	
password		⬭	password	Prompt user from the text	A…	password	

For information about working with the **Properties** sheet, see "Creating Input" on page 472 and "Setting Operation Outputs" on page 515.

Step Inspector

The Step Inspector is similar to the **Properties** sheet for an operation, but it relates to a single step in a flow. If you modify the properties of a step in the Step Inspector, the changes only affect this step, which is an instance of the operation.

For information about working with the Step Inspector, see "Creating Input" on page 472.

Transition Inspector

The Transition Inspector is used to configure the transitions between steps. To display the Transition Inspector, right-click the line that leads between two steps and select **Properties**.

For information about working with the Transition Inspector, see "Creating Transitions" on page 496.

Bookmarks pane

The **Bookmarks** pane, which you open with the **Bookmarks** tab in the upper-right of the Studio window, makes it easier to find and use the operations and flows that you use frequently.

You can add flows and operations to the **Bookmarks** pane by dragging them from the Library. You can also drag flows and operations from the **Bookmarks** pane to the Projects pane, to copy them to a project.

For more information about bookmarks, see "Bookmarking Flows and Operations" on page 666.

Icons pane

The **Icons** pane, which you open with the **Icons** tab in the upper-right of the Studio window, contains libraries of icons that you can use to clarify what a step does. You can use one of these icons to replace the default icon on a flow or step.
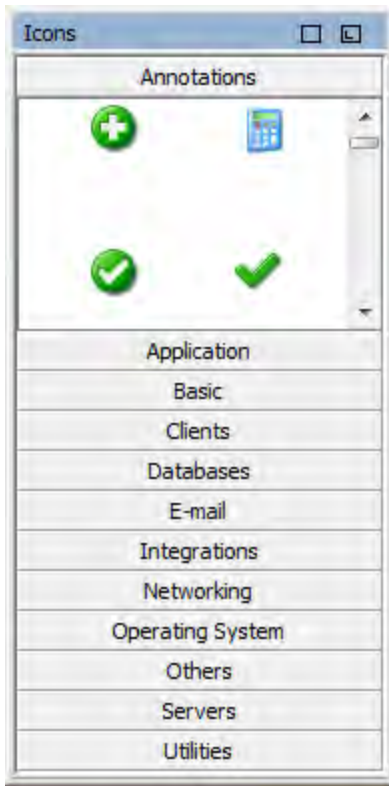
For information about working with the **Icons** pane, see "Modifying a Flow" on page 466.

Flow Variables pane

The **Flow Variables** pane, which you open with the **Flow Variables** tab in the upper-right of the Studio window, lists the flow variables used in the flow and describes how they are used.

For information about working with the **Flow Variables** pane, see "Working with Variables" on page 547.

References pane

The **References** pane, which you open with the **References** tab on the lower edge of the Studio window, shows how an operation or flow is used in existing flows. The pane can display two kinds of references:

- **What uses this?** - Identifies flows that have a step created from the operation or flow.

- **What does this use?** - Identifies objects (selection lists, permissions, system filters) that the operation or flow makes use of. In the case of flows, this includes operations and subflows from which the flow's steps were created.
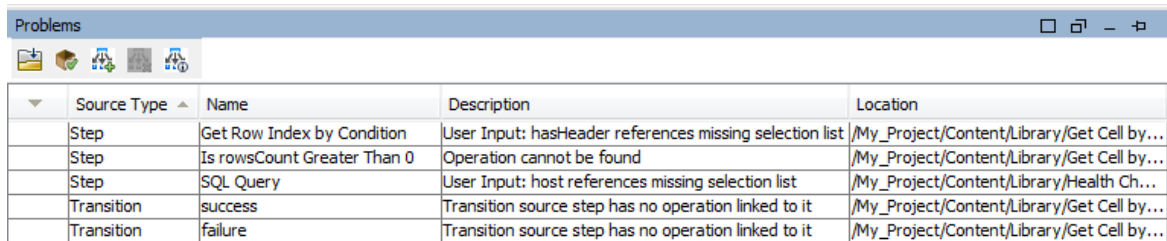
For information about working with the **References** pane, see "Finding Out How Flows and Operations are Used" on page 655.

Problems pane

The **Problems** pane, which you open with the **Problems** tab on the lower edge of the Studio window, lets you check whether a selected flow or operation is valid. This pane displays problems with a

selected flow or operation, with their locations and descriptions.
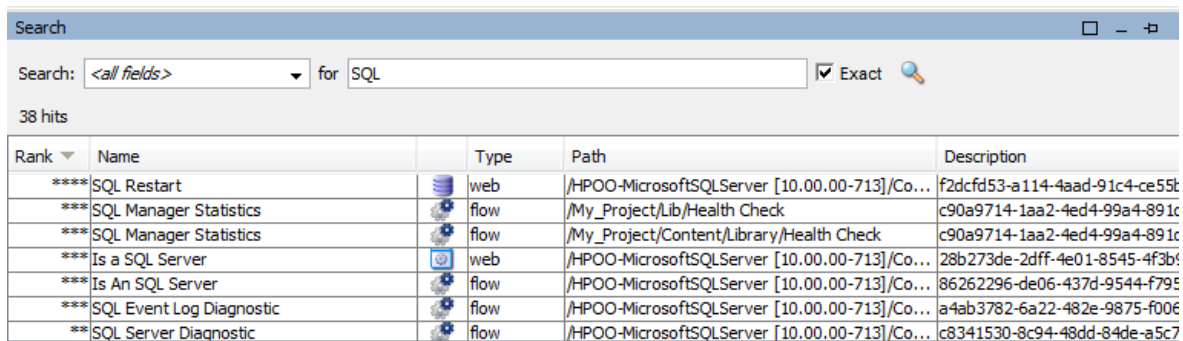
	Source Type ▲	Name	Description	Location
	Step	Get Row Index by Condition	User Input: hasHeader references missing selection list	/My_Project/Content/Library/Get Cell by...
	Step	Is rowsCount Greater Than 0	Operation cannot be found	/My_Project/Content/Library/Get Cell by...
	Step	SQL Query	User Input: host references missing selection list	/My_Project/Content/Library/Health Ch...
	Transition	success	Transition source step has no operation linked to it	/My_Project/Content/Library/Get Cell by...
	Transition	failure	Transition source step has no operation linked to it	/My_Project/Content/Library/Get Cell by...

For information about working with the **Problems** pane, see "Validating Flows in the Problems Pane" on page 593.

Search pane

The **Search** pane, which you open with the **Search** tab on the lower edge of the Studio window, enables you to search for a flow or operation. The Studio Search engine uses the Apache Lucene syntax.

Search: `<all fields>` for SQL   ☑ Exact

38 hits

Rank ▼	Name		Type	Path	Description
****	SQL Restart		web	/HPOO-MicrosoftSQLServer [10.00.00-713]/Co...	f2dcfd53-a114-4aad-91c4-ce55b
***	SQL Manager Statistics		flow	/My_Project/Lib/Health Check	c90a9714-1aa2-4ed4-99a4-891c
***	SQL Manager Statistics		flow	/My_Project/Content/Library/Health Check	c90a9714-1aa2-4ed4-99a4-891c
***	Is a SQL Server		web	/HPOO-MicrosoftSQLServer [10.00.00-713]/Co...	28b273de-2dff-4e01-8545-4f3b9
***	Is An SQL Server		flow	/HPOO-MicrosoftSQLServer [10.00.00-713]/Co...	86262296-de06-437d-9544-f795
***	SQL Event Log Diagnostic		flow	/HPOO-MicrosoftSQLServer [10.00.00-713]/Co...	a4ab3782-6a22-482e-9875-f006
**	SQL Server Diagnostic		flow	/HPOO-MicrosoftSQLServer [10.00.00-713]/Co...	c8341530-8c94-48dd-84de-a5c7

For information about working with the **Search** pane, see "Finding a Flow or Operation" on page 642.

Getting Started with OO Studio – Major Steps in the Workflow

This topic briefly describes the major steps involved in working with OO Studio. Click the links to see more detailed information about each step.
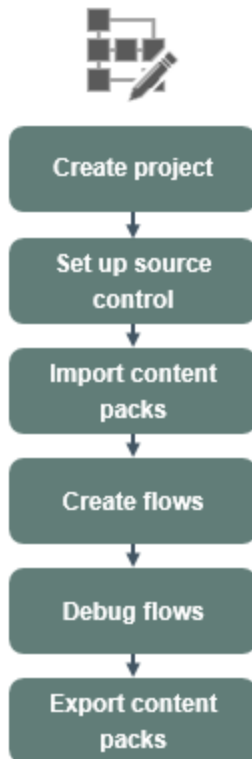
Click a button to navigate to the relevant topic.

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

   See "Managing Projects" on page 309.

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

   See "Working with Source Control in Operations Orchestration Studio" on page 320.

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

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

See "Importing Content Packs to a Project" on page 391.

4. **Create a flow** - Put together the operations, inputs, transitions, responses, and return steps that make up your flow.

   See "Creating a Flow – Step-by-Step" on page 452 and "Advanced Authoring" on page 557.

5. **Run and debug the flow** - Validate your flow in the Debugger.

   See "Testing and Debugging a Flow" on page 595.

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

   See "Exporting a Content Pack" on page 619.

Adjusting the Appearance of the Operations Orchestration Studio Window

You can set the panes in Studio to the following states:

- **Docked** - set in a permanent position in the Studio window
- **Floating** - free to be repositioned in the Studio window
- **Pinned** - hidden at the side of the Studio window so that only the tab is visible, and you have more room for your workspace

What do you want to do?

Float a pane

By floating a pane, you make it possible to move it to a different position in the Studio window.

1. Click the **Float** button 🗗 in the upper-right-hand corner of a docked pane.

2. Move the pane to a new position in the Studio window.

Dock a pane

If a pane has been floated to a new position in the Studio window, docking returns it to its permanent position in the Studio window.

Click the **Dock** button 🗖 in the upper-right-hand corner of the floating pane. The pane returns to its docked position.

Maximize a pane

To maximize a pane, so that it expands to the size of the entire OO window, click the **Maximize** button
⬜.

Restore a pane to its original size

To restore the pane to its size before you maximized it, click the **Restore** button 🗗.

Pin a pane to the side of the Studio window

Click the **Pin** button 📌 to pin the pane to the side of the Studio window so that only the tab is visible.
You can display the pane by clicking the tab.

Unpin a pane

After a pane has been pinned, click the **Pin** button 📌 again to unpin it. Unpinning a pane returns it to its
open, docked position in the Studio window.

Adjust the size of a pane

Drag the edge of a pane to make it larger or smaller.

Reset the Studio window to the default layout

To return the Studio window to the default layout, select **Window** > **Reset Window Layout**.

> **Note:** When you reset the Studio window, this will only be after creating or importing a new
> project.

Working with Different Languages in OO Studio - Localization

Localization refers to the adaptation of software for specific environments (countries or regions). OO
Studio can be localized into the following languages:

- French (fr)
- German (de)
- Japanese (ja)
- Spanish (es)
- Chinese (zh)

The following Studio text strings are localized:

- Flow descriptions and callouts

- Transition descriptions

- Configuration item descriptions

- Step descriptions and prompt texts (from the **Display** tab)

- Input prompts

- Folder descriptions

Language Codes

The following codes are valid:

Language locale code (user.language)	Region locale code (user.region)	Language
fr	FR	French
de	DE	German
ja	JP	Japanese
es	ES	Spanish
zh	CN	Regional Chinese

Localizing Content Packs

All the text strings contained in content pack flows and operations are shown in the current Studio locale.

The localization files are saved with the name **cp_<locale>{_<region>}.properties**. The **region** parameter is optional.

For example:

- The default content pack properties file is named **cp.properties**

- The French file is named **cp_fr.properties**

- The Chinese file is named **cp_zh_CN.properties**

Localizing Projects

Projects are not localized the same way as content packs. You can only edit the default language of a project, which is saved as the default file (**cp.properties**) during the packaging of the project into a content pack.

This has two implications:

- Project texts are not really localized. For flows, operations and configuration items, the text is stored inside the XML file, and for folders, the descriptions are stored in multiple **.properties** files, one for each folder. When creating a content pack, the texts are always written to the same **cp.properties** file.

- If a content pack is unpacked and loaded as a project in Studio, the localized properties file will not be used, only the default file. If one does not exist, all text strings will be empty.

When you create a content pack from a project in OO Studio, the resource bundle folder contains only the **cp.properties** file.

> **Note:** After changing the configuration of a content pack, you must restart Studio.

When the project is created or saved, the **cp.properties** file is saved in the project file system location under **\resource-bundles\cp.properties**.

You can also add your own resource files to the **resource-bundles** directory. These files will be bundled along with the other resource files into the content pack's jar file.

What do you want to do?

Change the current Studio locale

By default, Studio is installed in the default language of the computer.

You can override the current locale by changing its configuration:

1. Open the **C:\Users\<logged in user name>\.oo\studio.properties** file.

2. Search for the following line:

   `user.language=`

3. Add the language locale, using the **user.language** codes listed in the Language Codes table.

   For example, `user.language=ja`

> **Note:** In order to implement a different language interface, you must assign the same valid language locale from the table to both the **user.language** field and **user.region** fields (where relevant). If an invalid value is assigned, or if the field is left blank, Studio uses the default system locale.

Set the Studio display language

Studio takes the display language from the **user.language** property. This property must be valid (and

user.region must be valid as well). If it is not valid, Studio uses the system locale.

Currently, Studio supports the following display languages:

Language	Language code
French	fr
German	de
Japanese	jp
Spanish	es
Regional Chinese	cn

In addition, you can localize content packs. See "Localizing Content Packs" on page 303.

In versions 10.5x and later, you can have multiple workspaces. Therefore, each workspace will have its own **studio.properties** file under the **.oo** folder.

- If you want to use the same properties in all new workspaces, you can edit the **studio.properties.template** file located under **Studio\conf**.

- To configure different settings for each workspace, you can still edit each of the **studio.properties** files individually. These are located under the **.oo** folder inside each workspace.

Working with Projects

A project is a logical unit that can include flows, operations, folders, domain terms, selection lists, and other items for a business purpose. A project may be developed by a single author or as a collaboration project.

Projects in Studio use a file directory structure, like that of an operating system, which enables easy integration with a source control application.

Managing Workspaces

A "workspace" in Studio is the folder that contains the OO projects. The default workspace location is under **C:\Users\<logged in user name>\.oo**, but you can specify a different location.

Using Multiple Workspaces

There may be cases in which you need to work with more than one workspace—for example, when working with source control.

You can work in SVN in one workspace, and then switch to another workspace to work in Git, without detaching from the SVN workspace and without losing any work you did in SVN, even if you have not yet committed the changes.

Using multiple workspaces in Studio allows you to easily save custom workspace-specific settings such as the proxy settings, debugger connections, the projects you are working on, and the content packs you use.

Additionally, you can work simultaneously on several completely distinct projects and switch between projects as necessary. In this way, you can organize your work more efficiently especially when projects are shared with other authors.

Note: Every workspace contains its own **logs** folder with its own log file.

Note: When creating multiple workspaces in Studio, it is recommend that each user creates workspaces only under folders to which they have read and write permissions.

Workspaces created under public folders may be accessible to all users, making them prone to tampering and disclosure of sensitive information.

What do you want to do?

Define the workspace on startup

When you start Studio, you are prompted to select your workspace in the Select Workspace dialog box.
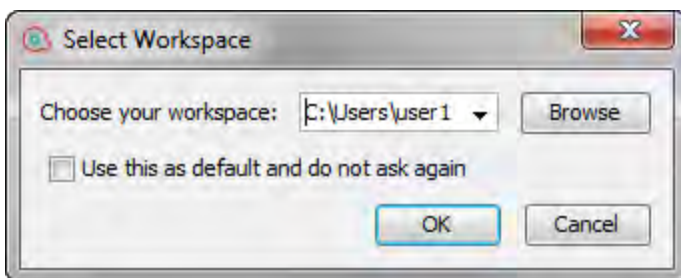
1. Do one of the following:

   ○ Select a workspace from the list.

   ○ Enter the workspace path.

> **Note:** The path must not include **.oo**.

- Click **Browse** to select a workspace from a different location.

2. (Optional) Select **Use this as default and do not ask again** if you do not want to be prompted again at Studio startup, and to always open the most recently used workspace.

   The next time Studio starts up, it will use the workspace that was last used, and the **Select Workspace** dialog box will not appear.

   If you switch to another workspace using **File** > **Switch Workspace** and then close Studio, the next time Studio starts up, it will use that workspace.

3. Click **OK**.

   If the path represents an existing valid workspace, Studio loads the settings, projects and content packs and connects to SVN or Git as configured previously in this workspace. It also uses the proxy, ITOM Marketplace, debugger settings as configured previously in this workspace.

   If the path you provided does not represent an existing workspace, Studio creates a new workspace.

   > **Note:** If this workspace is already in use (by another instance of Studio), a warning message is displayed.

Switch to another workspace

You can also switch to another workspace while working in Studio.

1. From the **File** menu, select **Switch Workspace**.

2. Do one of the following:

   - From the list, select a workspace location.

   - Click **Other** to browse to a workspace from a different location.

     > **Note:** If you select a folder that does not contain a workspace, Studio asks if you want to create a new workspace in this folder:

Use
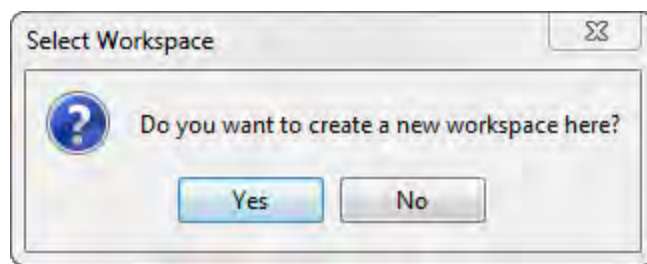
Click **Yes** to create a new workspace.

Studio opens the selected workspace.

Reference Material

File menu
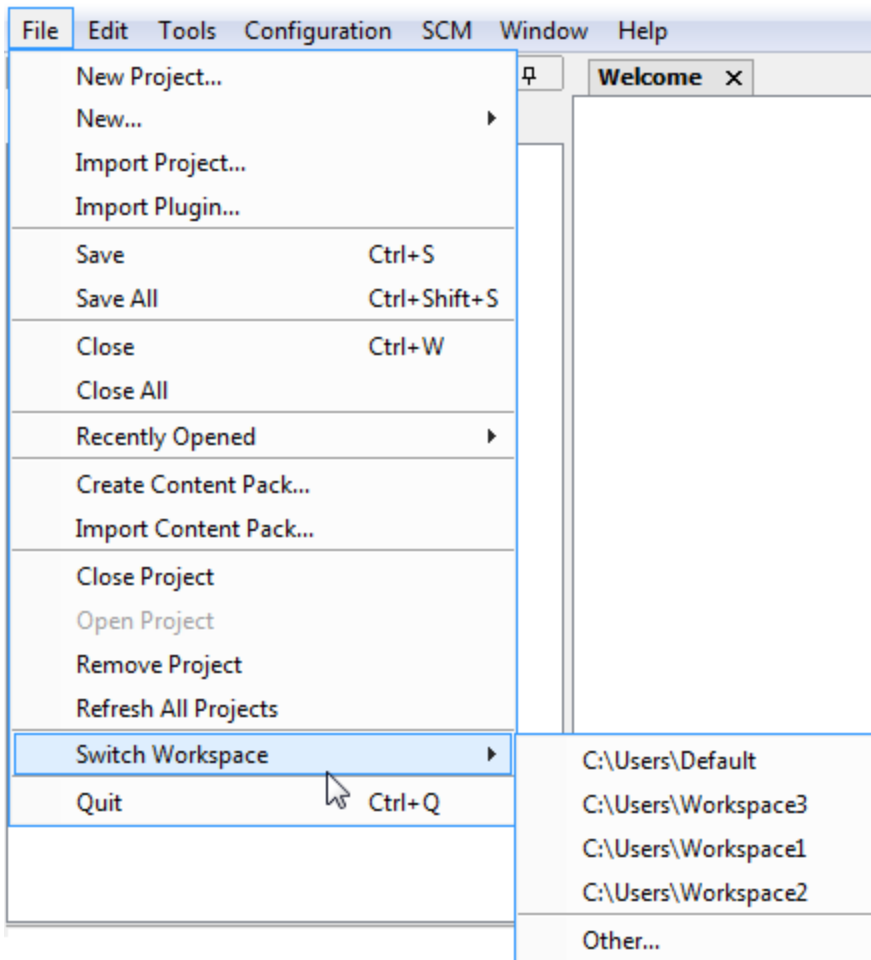
Menu item	Description

Switch workspace	Select the workspace you want to switch to from the list or select **Other**... to browse to a different workspace.
	Note: If you select a folder that does not contain a workspace, Studio asks if you want to create a new workspace in this folder.

Managing Projects

Before you can start creating flows in OO Studio, you must be working in a project.

Note: Creating the project does not have to be the first step in the workflow. It is possible to import content packs before creating the project.

In Studio 10.x, projects are stored locally on the author's file system. In order to have source control capabilities, you should use an external source control management tool. For more information about source control management, see "Working with Source Control in Operations Orchestration Studio" on page 320.

Handling Duplicate UUIDs

In Studio, each flow, operation, and configuration item must have a unique UUID. However, it is possible to accidentally end up with items of the same type and subtype with duplicate UUIDs. For example, this may occur if you are doing several revert operations in SCM or when resolving conflicts while working with branches in Git.

If there are items of the same type and subtype with duplicate UUIDs in a single project, in multiple projects (two or more), in multiple custom content packs or between content packs and projects, they appear as problems in the project tree (in red and underlined), and are displayed in the **Problems** pane.

Note: The same type and sub-type means flows with flows, operations with operations, configuration items with configuration items of the same sub-type.

The same type and sub-type means flows with flows, operations with operations, configuration items with configuration items of the same sub-type.

Items with duplicate UUIDs have the following characteristics:

- They are read-only, and you cannot move or rename them.
- They cannot be dragged to the canvas.
- You cannot move or rename the folders containing them.

- They are excluded from an exported content pack.

- You cannot debug flows containing duplicate UUIDs. You will get a validation error.

- You cannot use references. The "What uses this?" and "What does this use?" features are disabled.

- They cannot be committed. The folders and projects containing them can be committed, but you are prompted with an error messages informing you that you are about to commit items with duplicate UUIDs.

To resolve duplicate UUIDs, delete the item with the duplicate UUID and leave the original item.

If you want to keep both items, use the **Copy/Paste** or **Duplicate** commands to save a copy of one of the items with a different UUID, and then delete the item that you copied from.

To resolve duplicate UUIDs in content packs, delete or close the content pack.

What do you want to do?

Create a project

1. Select **File** > **New Project**.

    > **Note:** Alternatively, you can click the **New Project** button in the **Projects** pane.

2. In the **Name** box, enter a name for the project.

3. In the Create Project dialog box, verify that the path in the **Location** box is the correct path to the **Workspace** folder that you set up in the previous task. If the path does not lead to the **Workspace** folder, browse to and select the **Workspace** folder.

4. Click **OK**.

5. If you are working with a source control management tool, add the project folder to your shared repository. For more information, "Working with Source Control in Operations Orchestration Studio" on page 320.

Import a project

To open a project that was created in a different location, you need to import it to Studio. Once a project has been imported, it appears in the **Project** pane.

If you are working with a source control tool, you may need to check out the project folder, so that you have a local working copy, and then import the project to Studio. For more information, see "Working with Source Control in Operations Orchestration Studio" on page 320.

You can validate the project from the **Tools > Validate** menu, and as a result the **Problems** pane displays any issues in that project. For example, If the project contains items with duplicate UUIDs, they are displayed in the Problems pane.

1. Select **File** > **Import Project**.

   > **Note:** Alternatively, you can click the **Import Project** button in the **Projects** pane.

2. In the Select Project Directory dialog box, browse to locate the project that you want to import. You can import multiple projects into Studio at once.

3. Click **OK**. The project appears in the **Projects** pane in Studio.

Close an open project

If you close a project, it is visible in the **Projects** pane, but is grayed out and not available.

1. Select the project that you want to close.

2. Select **File** > **Close Project**.

   > **Note:** Alternatively, you can click the **Close** button in the **Projects** pane.

Open a closed project

After a project has been closed, you can open it, to work with it again.

1. Select the closed (grayed out) project that you want to open.

2. Select **File** > **Open Project**.

   > **Note:** Alternatively, you can click the **Open** button in the **Projects** pane.

Delete a project

Deleting a project is different from closing it, in that a deleted project is permanently removed from the workspace.

1. Select the project that you want to delete.

2. Select **File** > **Delete Project**.

   The Confirmation dialog opens:
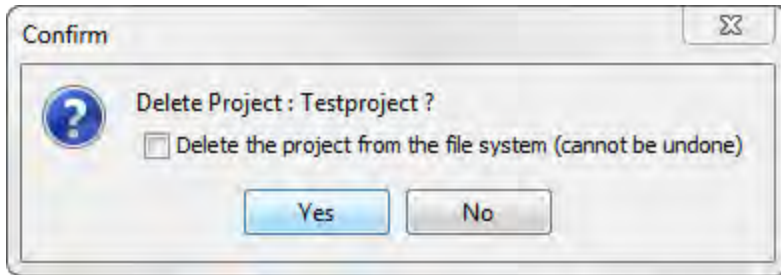
> **Note:** Alternatively, you can click the **Delete** ✖ button in the **Projects** pane.

3. Select the check box to delete the project from the Studio workspace and the file system. If you do not select the check box, the project is removed from Studio workspace but not from the file system. If required, you can reimport the project at a later stage.

4. Click **Yes** in the confirmation dialog box.

Move flows, operations, and configuration items

You can move flows, operations, and configuration items to a different folder within a project or to a new project.

> **Note:** It is not possible to drag and drop flows and operations at the same time as configuration items. It is not possible to drag and drop a flow or operation into the **Configuration** folder, and vice versa.

1. In the **Project** pane, select the flows or operations that you want to move. You can select multiple items using the SHIFT or CONTROL key.

2. Drag and drop the items or cut and paste them into another folder or another project.

Copy flows and operations from a content pack into a project

The flows and operations in the **Dependencies** pane are read-only. You can create editable copies of these flows and operations by copying them into the project.

1. In the **Dependencies** pane, select the flow or operation that you want to copy.

2. Select **Edit** > **Copy**.

3. Select the location in the project tree where you want to paste the copy, and select **Edit** > **Paste**. The flow or operation is treated as a new object and is detached from the content pack it arrived with.

> **Tip:** A quick way to copy multiple items from the Dependencies pane is to select them, holding down the **Shift** or **Control** key, and drag and drop them into the project.

> **Note:** If you also want copy the operations that make up a flow into the project, select **Edit** > **Copy Deep**. For more information, see "Copying Flows and Operations" on page 652.

Display the project properties

Do one of the following:

- Double click the **Properties** option in the **Projects** pane.

- Right-click the project and select **Properties**.

- Right-click the project on the **Properties** node in the **Projects** pane and select **Open**.

The Properties window displays information about the project. The project's Properties window allows you to set the **Publisher**, **Version** and the **Description** of the project as well as the dependencies of the project.

You can also perform standard SCM operations from the **Properties** node in the Projects pane (as well as from the **SCM Changes** pane. You can add, commit, revert to old versions, update and see the history of the properties.

This means that you can commit only the project metadata changes (publisher, version or description changes) without committing the entire project's changes. Similarly, you can commit dependency changes without committing the entire project's changes.

For full details on these options, see "Working with Subversion Source Control Management " on page 323 or "Working with Git Source Management System " on page 346.

Show the project location in Windows explorer

1. Select the **Properties** option in the **Projects** pane.

2. Right-click and from the menu, select **Show in Explorer**.

Windows Explorer opens, showing the folder containing the project files.

Display the properties of an object in a project

- Double-click a flow, operation or other object in the **Projects** pane. The Properties window for the object opens.

- If a flow is open on the authoring canvas, click the **Properties** tab in the lower left corner of the authoring canvas, to display the Properties window for the flow.

Reference Material

File menu

Menu item	Description
New Project	Creates a new project.
New	Creates a new file, folder, or operation.
Import Project	Browse to and import an existing project from a different workspace.
Import Plugin	Browse to and import an action plugin, to use as the basis of a new operation. See "Creating Operations" on page 632.
Save	Saves the selected element.
Save All	Saves all updated elements.
Close	Closes the currently selected flow or Properties window.
Close All	Closes all open flows or Properties windows.
Recently Opened	Lists recently opened items. This includes flows and

	operations that were open in the authoring canvas and Properties windows that were recently viewed.
Create Content Pack	Creates a content pack from the selected project.
Import Content Pack	Browse to and import a content pack in .jar format. See "Importing Content Packs to a Project" on page 391.
Close Project	Closes the currently selected project, so that it is grayed out.
Open Project	Opens the currently selected closed project.
Remove Project	Removes the selected project from the workspace.
Refresh All Projects	Refreshes all the projects in the Projects pane.
Quit	Close the Projects pane without saving changes.

Display the project properties

Do one of the following:

- Double click the **Properties** option in the **Projects** pane.

- Right-click the project and select **Properties**.

- Right-click the project on the **Properties** node in the **Projects** pane and select **Open**.

The Properties window displays information about the project. The project's Properties window allows you to set the **Publisher**, **Version** and the **Description** of the project as well as the dependencies of the project.

You can also perform standard SCM operations from the **Properties** node in the Projects pane (as well as from the **SCM Changes** pane. You can add, commit, revert to old versions, update and see the history of the properties.

This means that you can commit only the project metadata changes (publisher, version or description changes) without committing the entire project's changes. Similarly, you can commit dependency changes without committing the entire project's changes.

For full details on these options, see "Working with Subversion Source Control Management " on page 323 or "Working with Git Source Management System " on page 346.

Show the project location in Windows explorer

1. Select the **Properties** option in the **Projects** pane.

2. Right-click and from the menu, select **Show in Explorer**.

Windows Explorer opens, showing the folder containing the project files.

Display the properties of an object in a project

- Double-click a flow, operation or other object in the **Projects** pane. The Properties window for the object opens.

- If a flow is open on the authoring canvas, click the **Properties** tab in the lower left corner of the authoring canvas, to display the Properties window for the flow.

Reference Material

File menu

Menu item	Description
New Project	Creates a new project.
New	Creates a new file, folder, or operation.
Import Project	Browse to and import an existing project from a different workspace.
Import Plugin	Browse to and import an action plugin, to use as the basis of a

	new operation. See "Creating Operations" on page 632.
Save	Saves the selected element.
Save All	Saves all updated elements.
Close	Closes the currently selected flow or Properties window.
Close All	Closes all open flows or Properties windows.
Recently Opened	Lists recently opened items. This includes flows and operations that were open in the authoring canvas and Properties windows that were recently viewed.
Create Content Pack	Creates a content pack from the selected project.
Import Content Pack	Browse to and import a content pack in .jar format. See "Importing Content Packs to a Project" on page 391.
Close Project	Closes the currently selected project, so that it is grayed out.
Open Project	Opens the currently selected closed project.
Remove Project	Removes the selected project from the workspace.
Refresh All Projects	Refreshes all the projects in the Projects pane.
Quit	Quits Operations Orchestration.

Create Project dialog box
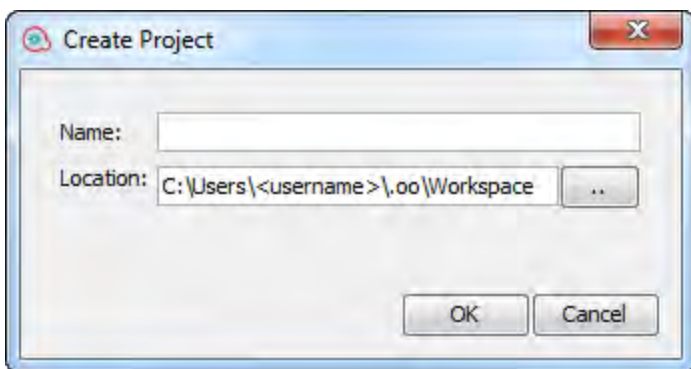
GUI item	Description
Name	Type a name for the new project.
Location	Enter the location of the workspace, in which to store the new project. • Type the path to the location. • Click the browse button ⬚ to browse for the location.

Projects pane

The **Projects** pane contains the project tree—a hierarchical folder structure containing the editable content of your project:

- The **Library** folder, which holds flows and operations.

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

  > **Note:** You can create folders in all the configuration items under the existing Configuration Item folder structure.

- The project properties

GUI item	Description
New Project ✚	Creates a new project.
Import Project 🗐	Browse to and import an existing project from a different workspace.
Create Content Pack 🗐	Creates a content pack from the selected project.

Delete ✖	Permanently deletes the selected project from the workspace.
Open 📦	Opens the currently selected closed project.
Close 📦	Closes the currently selected project, so that it is grayed out.
Refresh 🔄	Refreshes the files in the currently select project.

Managing Folders in the Project Pane

In the **Projects** pane, you manage the folders in the project—adding, deleting, copying, and renaming folders.

What do you want to do?

Create a folder

1. Select **File** > **New** > **Folder**.

2. Type a name for the new folder and click **OK**.

> **Important!** If you need to create an element (folder, flow, operation, or configuration item) inside your project, make sure to do this from within Studio, and not by creating the item in the file system.

> **Note:** Names can be a maximum of 128 characters long, and are not case-sensitive.

Rename a folder

Right-click a folder in the project tree, and select **Rename**.

> **Important!** If you need to rename an element (folder, flow, operation, or configuration item) inside your project, make sure to do this from within Studio, and not by renaming the item in the file system.

Delete a folder

Right-click a folder in the project tree, and select **Delete**.

> **Important!** If you need to delete an element (folder, flow, operation, or configuration item) inside your project, make sure to do this from within Studio, and not by deleting the item in the file

system.

> **Note:** You can delete a folder/hierarchy of folders that contains duplicate items, but cannot cut it for pasting at a later stage.

Copy and paste a folder

1. Right-click a folder in the project tree, and select **Edit** > **Copy**.

2. Right-click the position in the project tree where you want to paste the folder, and select **Edit** > **Paste**.

Drill down to subfolders

Click the expand button ⊞ to expand a folder and display the subfolders inside it.

Display the folder descriptions

Right-click a folder in the project tree, and select **Properties**. The properties of that folder are displayed.

Working with Source Control in Operations Orchestration Studio

This section describes the common tasks that are used with the Source Control Management (SCM) tools and more advanced tasks that you, as author, may encounter when projects and items are shared with multiple authors.

What is Source Control in OO Studio?

OO Studio includes two built-in Source Control Management tools that allow authors to work in their local environment and then synchronize changes with the public version:

- **Subversion:** For details, see "Working with Subversion Source Control Management " on page 323.

- **Git:** For details, see "Working with Git Source Management System " on page 346.

> **Note:** You must select only *one* of these tools as they cannot be used concurrently. If you are not sure which to use, consult with your system administrator.

What do you want to do?

Select a source control tool

To select a source control tool:

1. Select **SCM > Connection**

2. In the **SCM Connection** dialog box, select the SCM type: **Subversion** or **Git**.

Reference Material

SCM Connection

Menu item	Description

Type	Select the source control management tool you want to use: **Subversion** or **Git**.
SCM Repository: URL	Enter the URL of the Subversion or Git repository. **Note:** The HTTP and HTTPS protocols can be used for the URL.
Clone / Detach	**For Git only:** Clones the main Git repository to the working copy on the local machine in the directory of the current workspace . After cloning, the Git path is disabled, and this button changes to **Detach**.
Detach	Detaches the main Git repository from the local directory. Use the **Detach** option if you are using more than one Git repository and want to switch to a different repository. After selecting **Detach**, the following message opens: `Before detaching from Git, make sure that all your changes are committed and then pushed to the main Git repository. You can clone the Git repository again at a later stage to retrieve your changes.` **Note:** After detaching, you can no longer run any Git commands.
Checkout / Detach	**For Subversion only:** Copies sources from the SVN repository to the working copy on the local machine in the directory of the current workspace. After checking out, the SVN path is disabled and this button changes to **Detach**.
Trust server certificate	If you are working with a secure server (SSL/SSH), select this check box, otherwise you will be unable to access the server.
Credentials: Use Windows Authentication	When checked, performs authentication using the currently logged in user. To perform authentication with a different user, uncheck this field, and enter the user name and password in the fields below.
User name Password	Enter the user name and password to use as credentials for authentication.

Working with Subversion Source Control Management

Terminology

Working Copy

The SCM repository holds all versioned data on a source control server. The Source Control Management tool in Studio manages local copies of the versioned data, known as the working copy. SCM accesses its repository across networks. Multiple users can access the repository at the same time.

Checkout

Checkout is used to download sources from the repository to the working copy. If you want to access files from the source control server, checkout is the first operation you should perform. When you check out, a working copy is created, allowing you to edit, delete, or add contents. You can check out a file, directory, trunk or whole project. To check out, you need the source control server URL of the components you want to check out.

Commit: Save changes to the repository

When you make changes to your local working copy, they are not automatically saved in the source control server. To make the changes permanent, commit the changes.

Add: Add a new file to SVN repository

The **Add** command allows you to add new files or directories to the repository. The repository shows the newly added file after you commit the changes.

Delete: Removing a file from repository

The **Delete** command deletes an item from the working copy (or repository). Files are deleted from the repository after you commit the changes.

Move: Rename file or directory

The **Move** command moves a file from one directory to another or renames a file. The file is moved on your local sandbox immediately, as well as on the repository after committing.

Update: Update the working copy

The **Update** command transfers changes from the repository into your working copy. It is recommended that you update your working copy before you start working, so that all the latest changes available in repository are available in your working copy.

> **Note:** The Source Control Management tool does not include historical data for flows, such as

date, time, and comment.

Creating an Initial Source Control Repository

In a production environment it is recommended to use a dedicated SVN server, accessed by the protocols http, https or svn.

Note: Go to the subversion web site, http://subversion.apache.org/packages.html, for a list of SVN distributions for different operation systems.

Example 1: Using VisualSVN in a Multi-Author Scenario

The following example illustrates how to configure and use a VisualSVN server in a multi-author scenario.

Before beginning to work with SVN, it is recommended that you get accustomed with its concepts from the following link: https://subversion.apache.org/docs/.

1. Download a 1.9.x SVN compatible version of VisualSVN.

2. To configure the new SVN server users, right-click and select **Users > Create User…**.
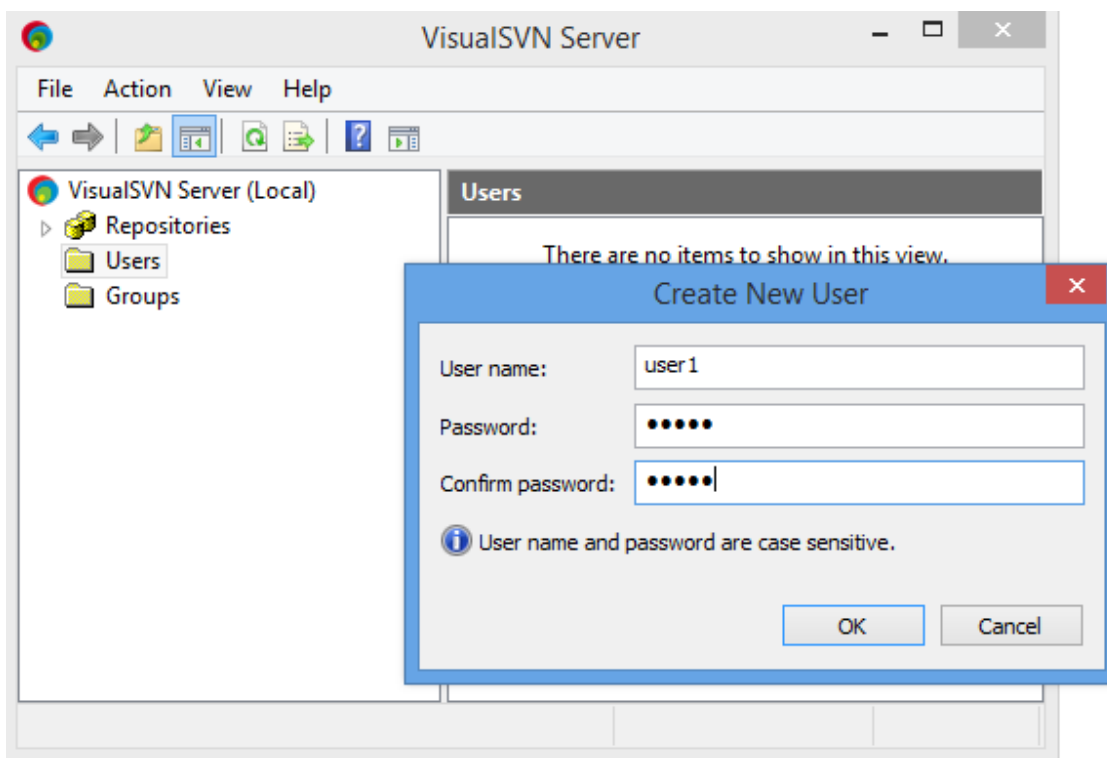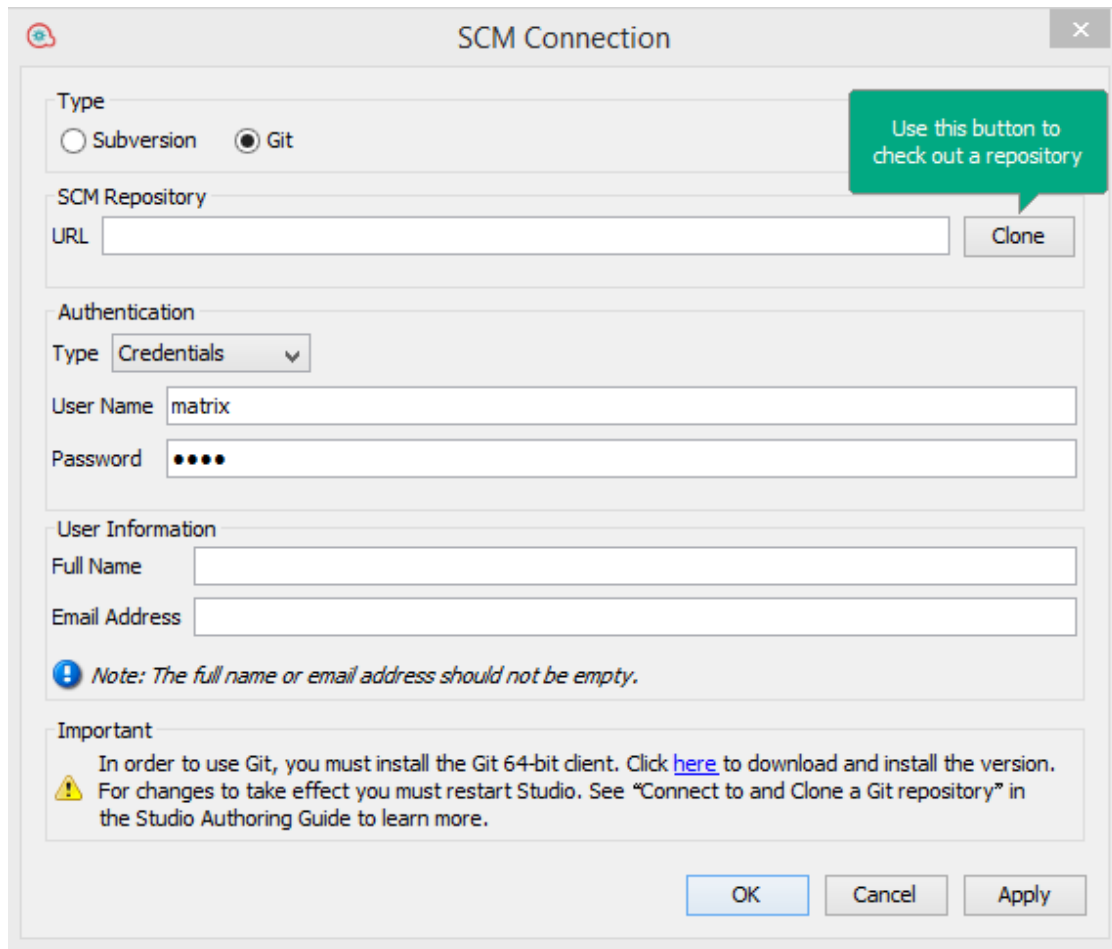
Note: Every Studio persona is identified by a Visual SVN user.

3. To create a SVN repository, right-click on the Repository folder, select **Create New Repository** and create an empty repository.
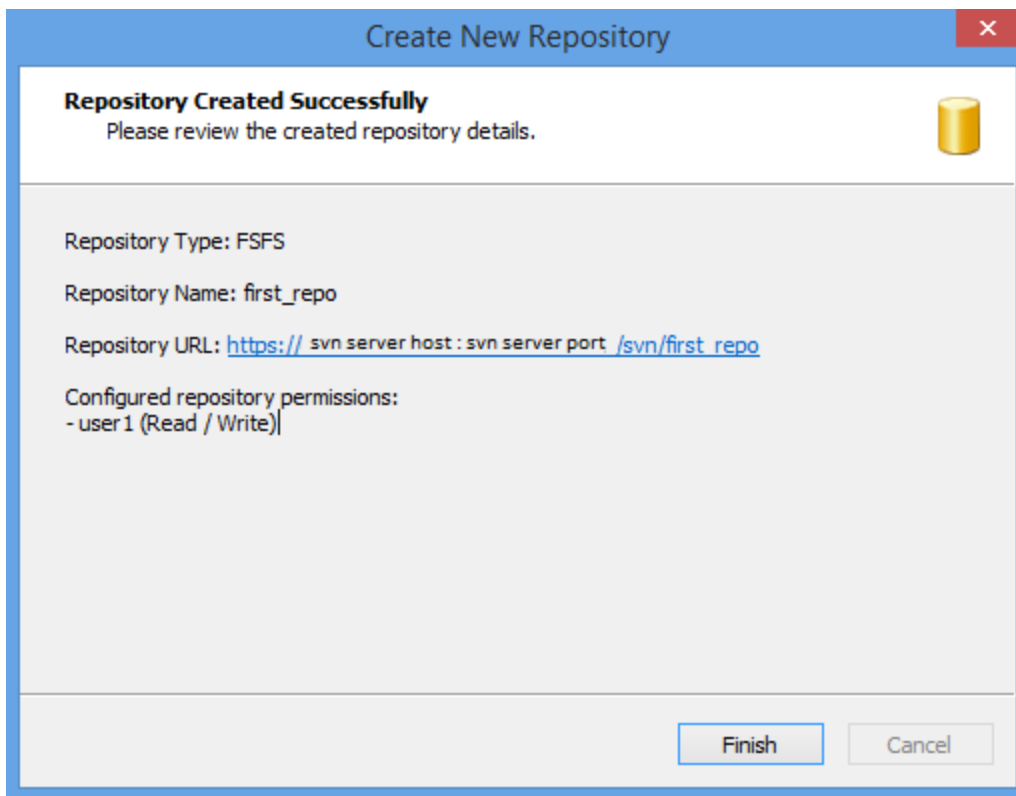
4. In Studio, from the **SCM** menu, select **Settings**.

5. Select the **Subversion** radio button.

6. Enter the SVN user name credentials - **User Name** and **Password**.

7. For an HTTPS protocol, select the **Trust server certificate** check box.

8. Check out the repository.

   If you have direct access to the VisualSVN console application you can obtain the repository URL as follows:

   a. Right-click on the repository location in VisualSVN.

   b. Select **Copy URL to Clipboard**.
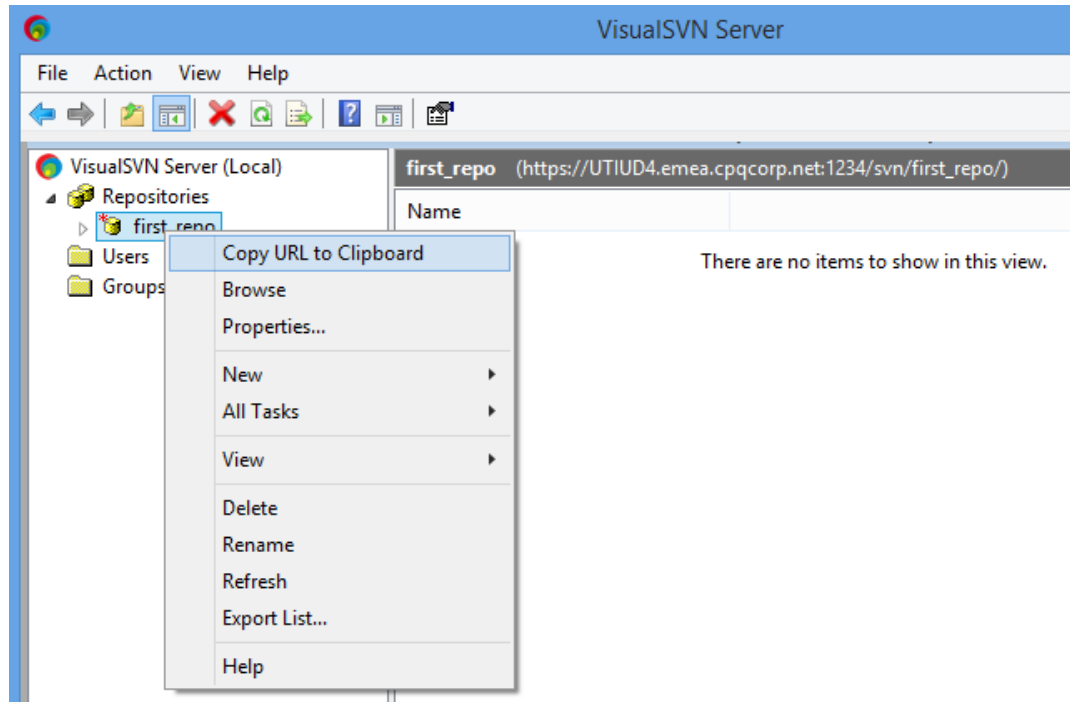
If you do not have access to a VisualSVN console application, ask your SVN administrator to provide you with the repository URL.

9. In Studio, from the **SCM** menu, select **Connection**.
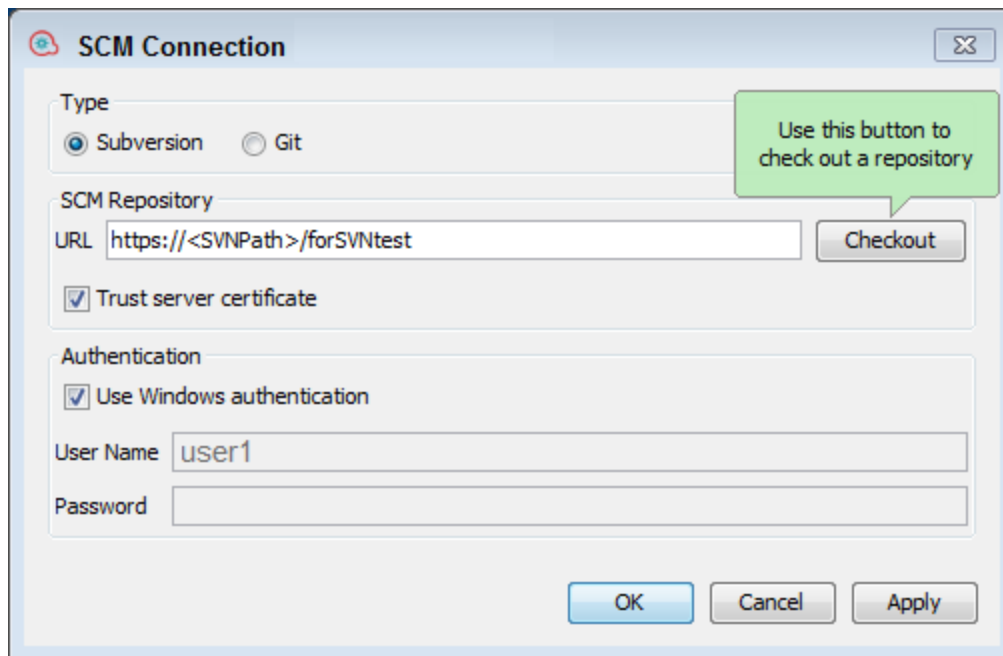
10. In the **SCM Repository** area, **URL** field, paste the URL using **Ctrl+V** and click **Checkout**.

A Checkout message appears in the **SCM Messages** pane.

Example 2: Single-User Scenarios

The following example is only for single-user scenarios, testing, and for debugging purposes.

Before you begin working with source control in Studio, you need to setup an initial repository.

> **Note:** It is recommended not to use a file system-based (or network share-based) repository in production.

This repository is intended for local, single-user access only, particularly testing and debugging, as when shared over the network, there are security concerns and possible performance problems.

> **Note:** If your company already has a source control repository, this section is not relevant.

To create an SVN repository:

1. Create a shared folder in Windows, and assign it with read and write permissions, for the authors that will work on the repository.

2. Open command line (`cmd`) in `<studio_installation_folder>\studio\tools`.

3. In the command window, type: **manageScmRepository.bat** and the full path to the local folder.

   For example:

   ```
   c:\<studio_installation_folder>\Studio\tools> manageScmRepository.bat
   -c LOCAL_FOLDER_FULL_PATH
   ```
   The following appears:
   ```
   Repository was successfully created at <LOCAL_FOLDER_FULL_PATH>
   ```

The shared repository folder is now ready for use with Studio.

Working with Multiple Authors in SVN

When multiple authors work on a common project, there is a possibility that two authors will modify the same item simultaneously. Studio attempts to merge all these changes without causing conflicts. Locking an item prevents other authors from working on the item at the same time. Items that are locked by another user can still be edited, but not committed. It is highly recommended to lock any item before editing it in order to prevent conflicts during updates. If you are unable to lock an item, then it is recommended that you do not edit the item in order to avoid conflicts.

A flow is automatically locked if the author tries to delete, move, rename or revert a flow. Once the flow is locked, other authors are prompted that the flow is locked when they try to perform one of these

actions. This occurs when the **Enforce locking policy** is enabled. See "Enforce Locking Policy" below for more information.

> **Note:** Make sure you perform an **Update**, before locking, modifying, and committing the item. Committing changes to a locked item automatically releases the lock.
>
> You cannot lock items that do not exist in the SCM repository.

If an item is unable to lock, a warning (yellow) message appears, and the icon will not change.

Committing changes to a locked item will automatically release its lock.

The lock button is disabled for items that are added, but not committed yet, or for items that are in projects that are not under Studio's workspace.

> **Note:** A lock is associated to a workspace and not to a particular SCM user. Therefore, if you have locked items and your workspace is destroyed, after recreating the workspace, you will not be able to continue working on your previously locked items nor will you be able to unlock them. In this case, you will need to either contact your SVN administrator, or to use an external SVN tool to break the locks on your items.

Enforce Locking Policy

This options prevents the flow author from making any changes to an item (flow or configuration item) unless the item is locked. This ensures that only one author can edit an item.

This option is automatically selected once the author checks out an SCM repository.

When the items is locked, the author can choose to unlock the item manually.

The author will then be prompted to select one of the following options after making changes to the item and unlocking it:

- Commit your changes.
- Unlock the item and keep editing.
- Revert the item.

In the following cases the author cannot obtain a lock:

- Another author has already locked the item.
- Network issues.
- The item version is out-of-date.

In the above cases, the **Failed to Acquire Lock** message appears prompting you if there are any local changes. Click **Yes** to continue editing, or **No** to discard the changes.

When **Enforce Locking Policy** is enabled, Studio automatically obtains locks on the affected items of all operations that change the project's structure. For example, renaming, moving or deleting flows, operations or folders.

What do you want to do?

After you have installed Studio with a local repository, you are ready to start authoring in Studio and use the source control in your local environment. In the following sections, you can learn about common tasks that you perform with SCM.

Set the authentication settings with the source control server

The first step in working with SCM is to set the user authentication with the source control server.

1. From the **SCM** menu, select **Connection**.

2. Under **Type**, select the **Subversion** radio button.

3. Select one of the following options, depending on your source control server:

   ◦ **Use Windows Authentication**: This option is the default option selected, and performs authentication using the currently logged in user. This is applicable for file-based source control repositories.

   ◦ Clear, the **Use Windows Authentication** option. Enter the user name and password defined on the source control server. If you are working with a secure server (SSL/SSH), select **Trust server certificate**. Otherwise, you will be unable to access the server.

   > **Note:** The credentials are only used to access the SCM itself. If the system is based on a simple network fileshare, OO expects the operating system to be able to create the network connection to share. This means that OO does not pass on the credentials to the operating system for a network connection.

Check out a repository

The repository contains all your projects and can be shared by multiple authors. A repository can be hosted either on a file system or on a web server.

You can check out the repository directly from Studio and then add projects to it for sharing with other authors.

To check out all projects:

1. From the **SCM** menu, select **Connection**.

2. In the **SCM Repository** area, **URL** field, paste the URL using **Ctrl+V** and click **Checkout**. This checks out the files from the URL and into Studio's Workspace directory.

   The first time you select **Checkout**, you are prompted to enter the URL of the shared folder you created when you setup the local repository. If you are working on a local repository enter the URL using the file URL scheme.

   For example:

   ○ Web-based URL:

   ```
   http://svn.samplehost.com/repo/trunk
   ```

   ○ If the repository folder is not shared, connect with the following:

   ```
   file:///c:/temp/repo/trunk
   ```

   ○ If the repository is shared and you have all the permissions:

   ```
   file://myshared/repo/trunk
   ```

   A Checkout message appears in the SCM Messages pane.The **SCM Messages** pane displays the shared folder with the location where the files are checked out to. If projects already exist in the checked-out repository they can now be imported into Studio and worked on.

   > **Note:** In the **SCM Messages** pane, the messages coming from the Source Control Management client tool may contain an encoded URL. This is the repository URL in a standard encoded form. The message is coming from an external SVN client tool used by Studio (SlikSVN). Note that a non-encoded version of the URL is also logged in the **SCM Messages** pane.

3. The next step is to either create a new project or import an existing project. From the **Project** pane, either click the ✚ button to add a new project or click 🔧 to import an existing project.

Detach the Workspace from SVN

After detaching a repository, the Studio workspace is detached from the source control server. This is useful if you made a mistake and checked out the wrong repository. Detaching does not delete anything from the repository.

1. From the **SCM** menu, select **Connection**.

2. In the SCM Repository area, click **Detach**.

3. Click **OK** to close the window.

   > **Note:** Detaching from an SVN repository will unlock all the flows, operations and configuration items which were already locked by the user.

Commit changes to SVN

After making local changes, use the **Commit** menu option to check them into the repository. There are a number of ways to commit changes:

- From the **SCM** menu, select **Commit All** to commit all the changes.

- In the **SCM Changes** pane, click the **Commit** button to commit all the changes.

- In the **Projects** pane, right-click an updated item and select **SCM > Commit**.

- In the **SCM Changes** pane, right-click an updated item and select **SCM > Commit**.

View a history of SVN operations in the repository

The **History** option shows you:

- A list of commits in which the selected project/item was affected

- All affected files for a particular commit

The **History** option also allows you to revert to a specific revision of the project/item. See "Revert to a previous revision of a project/item " on the next page for details.

1. Right-click on a file/folder and select **SCM > History**.

   The Revision History dialog box opens:

Revision	Date/Time	Author	Comment
71259*	06/03/15 14:07	leah.hakim	Third revision
71257	06/03/15 14:06	leah.hakim	
71256	06/03/15 13:55	leah.hakim	first version
71254	06/03/15 13:39	leah.hakim	
71252	06/03/15 13:35	leah.hakim	
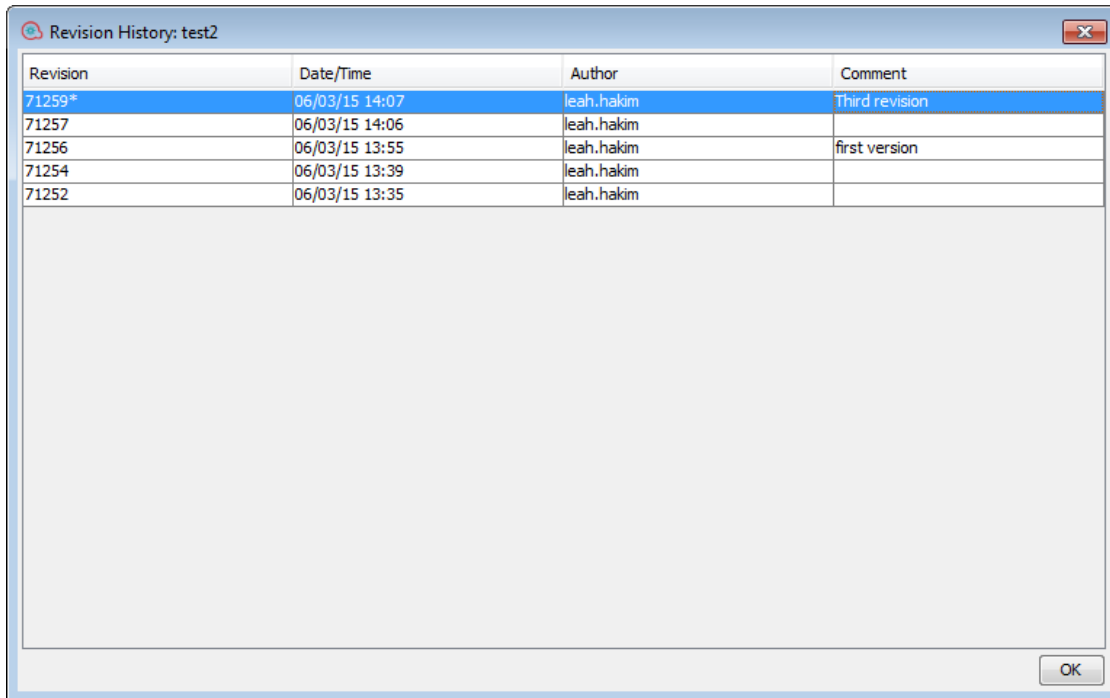
Revert to a previous revision of a project/item

To revert to a previous revision:

1. Right-click on a file/folder and select **SCM > Revert**.

   The Revision History dialog box opens.

2. Right-click on the commit revision you want to revert to.

3. Select **Revert to this version**.

Note: You cannot revert to a revision number earlier than the initial revision given when the project was first committed to SVN.

Update to a tagged version or revision of a project/item

There may be cases for which you need to recreate a faulty Central content pack/project in order to troubleshoot a specific flow inside the content pack. If the content pack is not available (it has been lost or been overwritten by a new version), you can restore a specific version of the project/item using the **Update to...** operation.

Note: In SVN, you can perform a **Update to...** operation on individual folders and repositories.

You can use the **Update to...** option if you have already checked out the repository and you want to update your workspace to a specific revision from the SCM tag.

1. Select the content pack/project you want to update.

2. Right-click and select **SCM > Update To...**.

   The Update Options window opens:
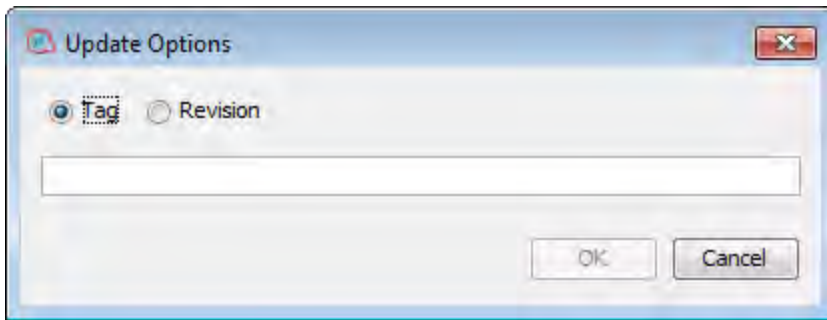
3. Select **Tag** and type in the SCM tag you received from the Central administrator. By default, the SCM tag is in the format `[Project Name]-[Project version]`.

   or:

   Select **Revision** and type in the revision number you received from the Central administrator.

4. Click **OK**.

Move/cross-move a versioned item

You can move an item from one location to another within the same project or cross-move an item between two different projects.

To move an item:

1. In the **Project** pane, select the items that you want to move. You can select multiple items using the **Shift** or **Ctrl** key.

2. Drag and drop the items into another folder or another project.

   or:

   Cut the items using **Ctrl+X** and paste them at the new location using **Ctrl+V**.

   The item is marked in the **SCM Changes** pane in the old location with the status **deleted** and in the new location with a status **moved**.You can see the original location in parentheses.

> **Note:**

- When moving a versioned item, the item must also be scheduled for moving in the SVN repository.

- When cross-moving an item (between a source project and a destination project), the item must be scheduled for moving, deletion or addition in the appropriate projects.

- Where relevant you can commit the result of the move/cross-move action to SVN.

Clean up the working copy

In some cases, you will need to clean up the working copy in Studio's workspace. For example, if a Studio process crashes or if there is an IO error, and the working copy remains locked.

If you need to clean up the working copy, you will see an error message.

To clean up the workspace:

- From the **SCM** menu, select **Subversion > Cleanup**.

Lock an Item

- In the top left corner of the items editor window, click the **Opened Lock** icon.
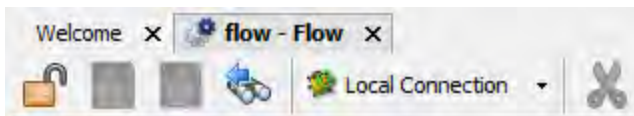
If the lock was successful, the icon changes to a closed lock and displays an SCM message displaying the full path to the locked file and with the user details.
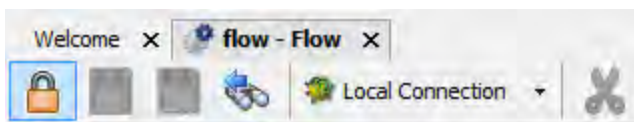
06/09/13 17:04:33 - Locking C:\Users\SHARONDO\.oo\Workspace\Project1\Content\Library\flows\flow.xml
'flow.xml' locked by user 'sharondo'.

> **Note:** After you lock an item, its status is shown in the SCM Changes pane as "locked". If an item is locked and has been modified (not recommended), its status is show as "modified". For

example:
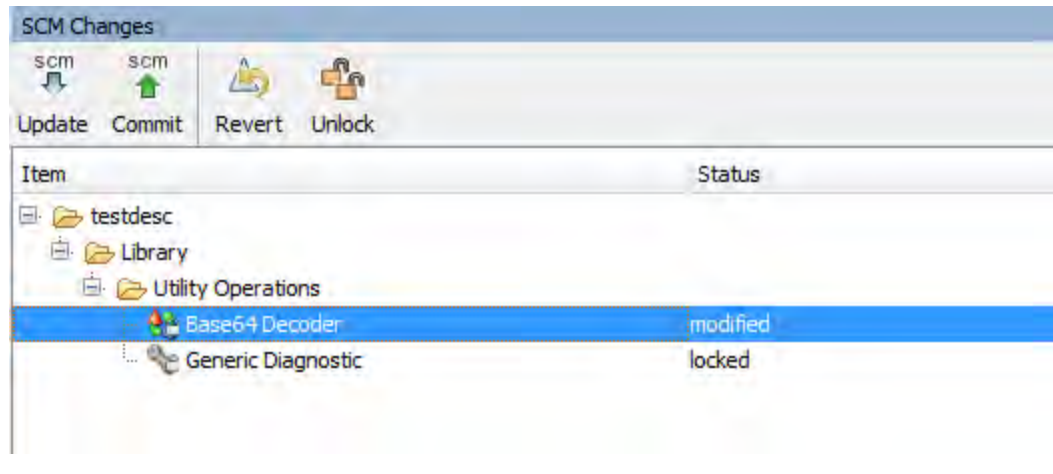
Unlock an Item

- In the top left corner of the items editor window, click the **Closed Lock** icon.
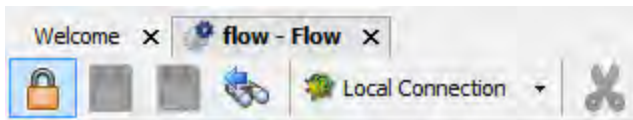
If the unlock was successful, the icon changes to an Open Lock and displays an SCM message displaying the full path to the unlocked file and with the user details.
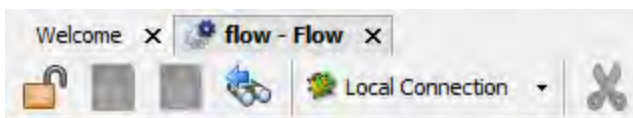
06/09/13 17:05:06 - Unlocking C:\Users\SHARONDO\.oo\Workspace\Project1\Content\Library\flows\flow.xml
'flow.xml' unlocked.

Set the Enforce Locking Policy

- From the **SCM** menu, select **Subversion > Enforce Locking Policy**.

Reference Material: SCM Changes Pane

The **SCM Changes** pane displays all that was changed in the working copy, compared to the working copy's revision. For example, editing a flow will cause that flow to appear in the **SCM Changes** pane. This pane also shows a list of deleted projects (projects marked for deletion), if such projects exist.

> **Note:** Projects that are outside of Studio workspace are not added/committed to SCM. They are shown as unversioned (in brown) in the **Projects** pane, and are not shown in the **SCM Changes** pane.

Types of changes

You can see how different types of changes are shown:

- Deleted flow, named **flow**, and colored gray.

- Added folder, named **deprecated**, colored green

- Unversioned project, named **Project2**, colored brown.

- One deleted project.

- The changed item **MathOp**, in blue, also shows a lock indicator, which means it was locked for editing by the user.

Renamed and Moved Items

Renamed and moved items have a special label with the location where the item was moved or renamed from. In the following example, **windows** was renamed **linux**, and the folder **flow** was moved.

SCM Changes toolbar

	Update all: Updates the entire Studio workspace.
	Commit all changes: Commits all the changes that show in the **SCM Changes** pane. Only available when there are changes.
	Revert all changes: Reverts all changes that show in the **SCM Changes** pane. Only available when there are changes.
	Unlock all: Unlocks any locked item.

Color codes

Studio shows the following color indications for items:
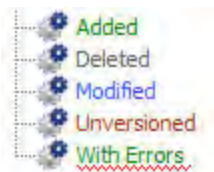
- Black: Normal item with no changes (not available in the changes pane)

- Green: Added

- Grey: Deleted (not available in the projects tree)

- Blue: Modified

- Brown: Unversioned

- Zig-zag underscore: Includes errors.

Note: By default, SVN has a list of file patterns (shown below) that are ignored when non-versioned files are added to SCM:

```
*.o *.lo *.la *.al .libs *.so *.so.[0-9]* *.a *.pyc *.pyo
```

This means that a non-versioned project or folder that matches one of these patterns is shown as black (as if it is versioned) in the **SCM Changes** pane.

However, you can still add the project or folder to SCM and work with them as usual.

SCM Message Pane

The **SCM Message** pane displays the message results from the SCM actions. Every action results in a message and is color-coded. In addition, a pop-up message appears informing the user of the result of the SCM action.Clear the **SCM Message** pane by right clicking anywhere inside the pane and then clicking the **Clear All** button.

Examples of SCM Messages

In the following example, a new operation called **MathOp** was added:

After each addition, the new items are automatically marked to be added. The following SCM message shows that a folder, oo-base-plugin, and the operation's XML file, MathOp.xml were marked to be added.
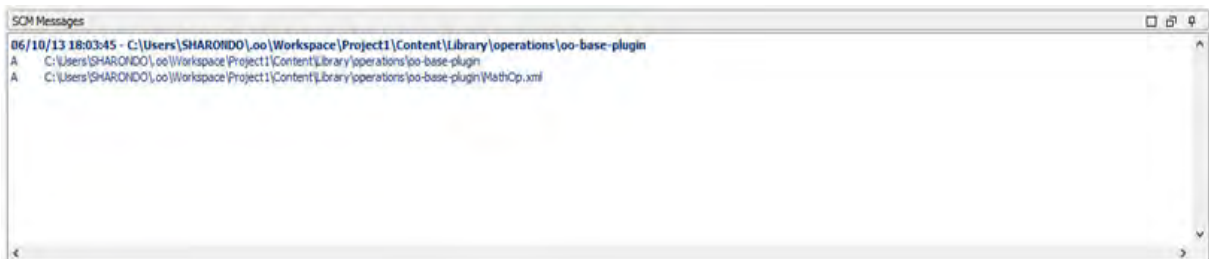
After you commit an item, the following message appears. This message shows that your previously added items were now successfully committed to the server.

After making a change to **MathOp**, for example, adding a new input, revert the changes. This message shows that a change to MathOp.xml was reverted.
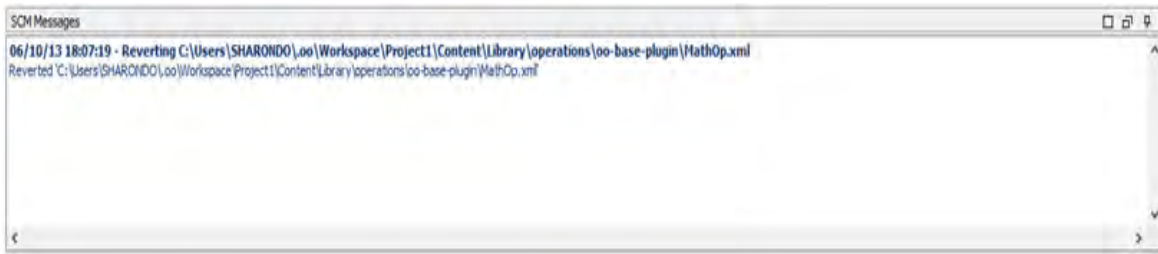
Error messages

In certain cases, errors appear. These are displayed in red. In this example, an author attempted to commit these changes, but the file was out-of-date.

Reference Material

Revision History

OO Studio also offers version history control. The **Revision History** pane displays the SCM history. It is divided into four columns and contains one line for each commit.

- **Revision**: The revision for the commit. The revision of the local copy is marked with an asterisk. After checkout, the asterisk always marks the latest revision, but after changes have been committed by another user, and before updating the asterisk marks a revision that is not the latest. The screen shot above shows that the local copy is at revision 1953, but there was a commit to the file in revision 402.

- **Date/Time**: The date and time in which the revision was committed.

- **Author**: The author who committed the revision. This displays the user's name.

- **Comment**: The comment added during the commit by the user.

When you right-click on a history item, the context menu displays the following options:

- **Update to revision**: Updates the file to the selected revision. Using this option causes some items in the project to be in a different revision than others. In the following example, right-click on the revision above the asterisk (*).
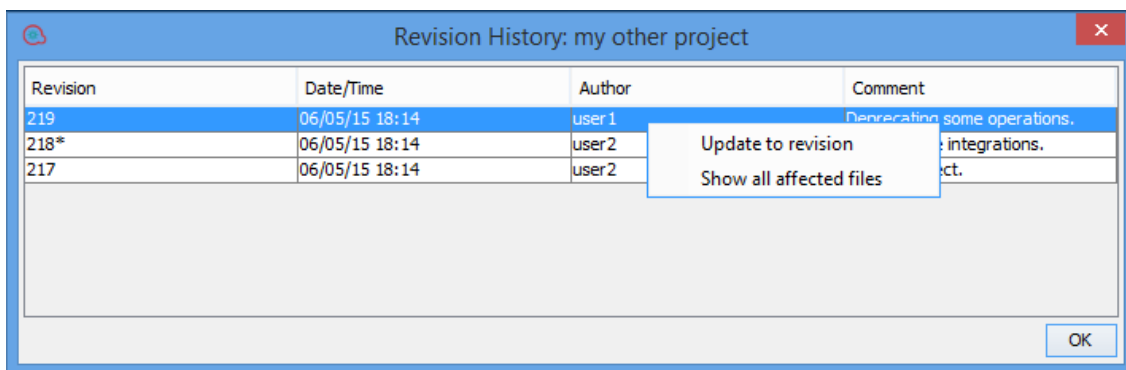
- **Revert to this revision**: Locally changes the object to the way it was in the selected revision. If for example a flow contains two steps in revision 333, and three steps in revision 337 then reverting it to revision 333 causes it to have two steps again. In order to make the changes visible to other authors, you must commit the flow. In the following example, right-click on the revision below the asterisk (*).
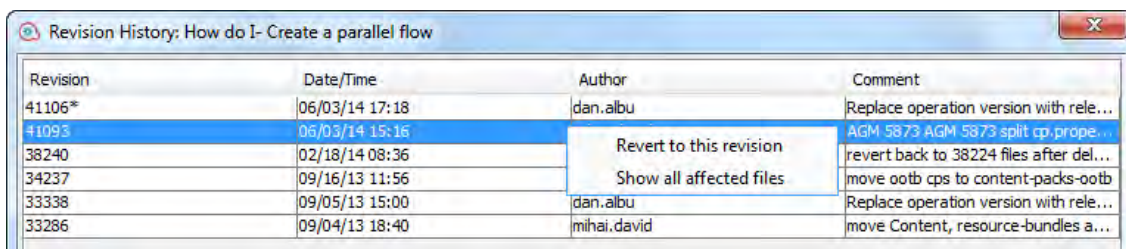
- **Show all affected files**: Shows all the files that were changed in the selected revision.

> **Note:** When running a **Revert to this revision** operation on a folder with changed items or on a changed flow/operation/configuration item, the changes can be lost and there could be tree conflicts that are resolved using the local version. In these cases, the following message appears:
>
>
>
> It is recommended to revert the changes first, and then to try to run **Revert to this revision** again.
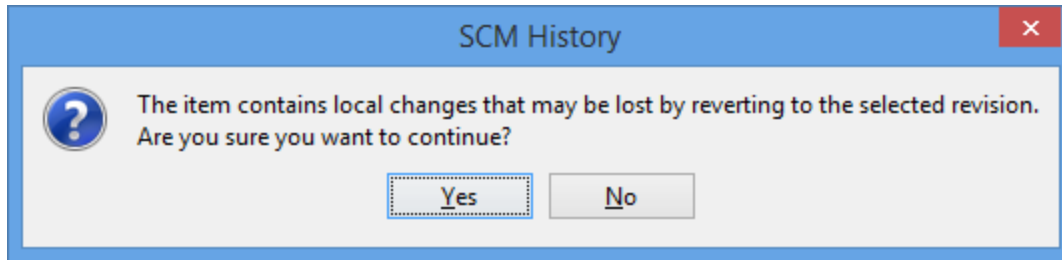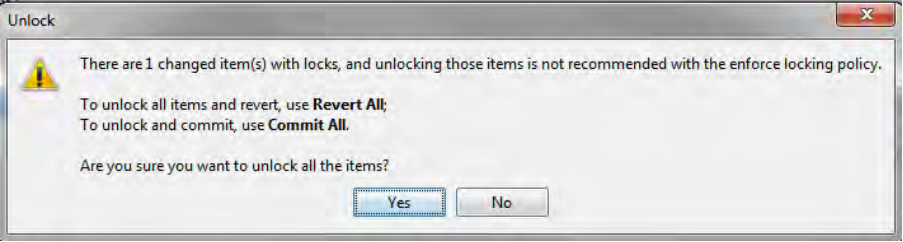
Source Control Menu

Contains operations that affect the entire workspace:

Option	Description
Update All	Updates the workspace directory, but does not import projects into Studio. Projects that are already imported will reflect any updates that were received from the server. Since this command updates the entire workspace, new projects committed by other authors, may be retrieved during update. These projects can be imported into Studio and worked on now.
Commit All	Commits all changes from the active projects. When you click **Commit**, you can add a comment for the commit.
Revert All	Reverts all changes from the active projects.
Subversion > Enforce Locking Policy	This options prevents the flow author from making any changes to an item (flow or configuration item) unless the item is locked. This ensures that only one author can edit an item. See Enforce Locking Policy above for more information.

Option	Description
Subversion > Unlock all	Unlocks any locked items in the workspace. A warning message appears. For example: ![Unlock dialog box: "There are 1 changed item(s) with locks, and unlocking those items is not recommended with the enforce locking policy. To unlock all items and revert, use **Revert All**; To unlock and commit, use **Commit All**. Are you sure you want to unlock all the items?" with Yes and No buttons] Select **Yes** to unlock all the items or use the other options presented in the message.
Subversion > Cleanup	Clean up the working copy in Studio's workspace.
Connection	Allows you to change the way authentication is performed with the source control server, to select the SCM type and to check out/detach a repository. **Windows Authentication**: Performs authentication using the currently logged in user. This is applicable for file-based source control repositories. **Username** and **PasswordAuthentication**: Performs authentication using the supplied user name and password. **Trust server certificate**: If you are working with a secure server (SSL/SSH), select this option, otherwise you will be unable to access the server.

Projects pane

Shows the project you're working in, and displays the editable flows, operations, and other OO objects that you can use in the project.

Context Menu from the Projects pane
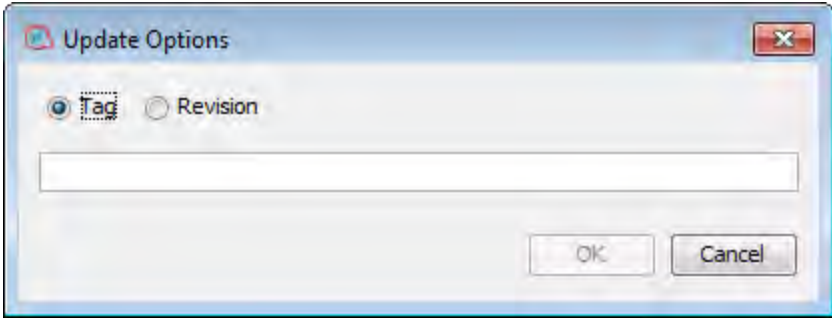
The context menu shows all actions that are available for the selected object, according to its state.

Following are the options available when you right-click on a flow:

If the object was to be changed by an author, for example by adding a step, then it would have local changes. In this case, **Commit** and **Revert** are available.

Option	Description
Add	Marks an item (flow, operation, configuration item or folder) to be added to source control. In Studio **Add** includes all ancestor and descendant objects. So if a folder is added, any child flows and parent folders are also added. Sibling items are not added. Items that are created from within Studio, are added automatically and committed on the next commit.
Commit	Commits local changes to the server. This option is available for changed items and for folders that have changed children items. A commit works recursively, so when you commit a folder, all of its child items are also committed. After you commit, you can add a comment for the commit.
Update	Updates the selected item. This option is unavailable only for items that are locally added but not committed yet. Works recursively and updates all child items.
Update to...	Updates the selected item to a specific version. When this option is selected, the Update Options window opens:

Option	Description
	Enter the tag name (from the deployment page in Central) or the revision number (from the Revision History window) and click **OK**.
Revert	Reverts all local changes in the selected items. Changes in flows, operations and configuration Items are reverted. Items that were deleted are restored. **Important**: When you add an item it is automatically marked as added, but when you revert changes on it only the addition mark is removed, but the item still exists in Studio and in the file system. You can re-add the item using **Add** in the menu, or delete it.
History	The **Revision History** window, displays the SCM history. See Revision History for more information.

Deleting a Project

- Select the project that you want to delete, and click the **Delete** button. When you click **Delete**, the project is deleted from Studio (un-imported). If the **Delete project from file system (cannot be undone)** option is selected, the project is marked for deletion from the repository, and the deletion occurs on the next commit.

Working with Git Source Management System

Git is a Source Control Management (SCM) system that provides you with an alternative solution to source control, in parallel with the Subversion – SVN solution. Git is a distributed version-control system, with certain concepts that differ from Subversion.

The methodology of working with Git in Studio is very similar to the methodology of SVN.

The initial version of Git support for 10.5x and later targets the basic use cases and provides ease-of-use, similar to the Git support integrated to other IDE tools (such as IntelliJ IDEA or Eclipse).

Git Terminology

Add: Add a new file to a Git repository

The **Add** command allows you to add new files or directories to the local repository. The repository shows the newly added file after you commit the changes.

Clone

Clone is used to download sources from the Git repository to the working copy. If you want to access files from the source control server, clone is the first operation you should perform. When you clone a repository, a working copy is created, allowing you to edit, delete, or add contents. To clone, you need the source control server URL of the Git repository you want to clone.

Commit: Save changes to the repository

When you make changes to your local working copy, they are not automatically saved in the source control server. To make the changes permanent, you must commit the changes. The changes are saved in the local Git repository. You can then push them to the remote Git repository.

Delete: Removing a file from repository

The **Delete** command deletes an item from the working copy (or repository). Files are deleted from the repository after you commit the changes to the remote repository.

Downstream/Upstream

In terms of source control, you're downstream when you copy (clone, checkout, etc) from a repository. Information flowed "downstream" to you.

When you make changes, you usually want to send them back "upstream" so they make it into that repository so that everyone pulling from the same source is working with all the same changes. This is mostly a social issue of how everyone can coordinate their work rather than a technical requirement of source control. You want to get your changes into the main project so you're not tracking divergent lines of development.

Ignore

There are files that do not need to be version-controlled in Git. These files are handled by the Git Ignore mechanism that operates behind the scenes, and is automatically executed when adding a project. By default, the files to ignore are (relative to the project home): *.idx, *.tmp, *.lock, and Content/.metadata.

Merge

The **Merge** command incorporates changes from the commits (since the time their histories diverged from the current branch) into the current branch.

Move

The **Move** command moves a file from one directory to another or renames a file. The file is moved on your local folder structure immediately, as well as on the repository after committing.

Pull

Pull refers to when you are fetching in changes and merging them. For example, if someone has edited the remote file you're both working on, you'll want to pull in those changes to your local copy so that it is up to date.

Push

Pushing refers to sending your committed changes from the local repository to a remote repository. For instance, if you change something locally, you'd want to then push those changes so that others may access them.

Rebase

Rebase take all the changes that were committed on one branch and replays them on another branch.

Stash

Stash saves your working directory and index to a safe place and restores your working directory and index to the most recent commit. After stashing, you can then work on other branches, make commits, etc. and when you're ready to get back to where you were, run a **git stash apply** (unstash) operation to restore your working directory.

Working Copy

The SCM repository holds all versioned data on a source control server. The Source Control Management tool in Studio manages local copies of the versioned data, known as the working copy. SCM accesses its repository across networks. Multiple users can access the repository at the same time.

Update: Update the working copy

The **Update** command transfers changes from the remote repository into your local working copy. It is recommended that you update your local copy before you start working, so that all the latest changes available in the remote repository are available in your local copy.

> **Note:** Some of the terminology in this glossary is taken from the github site
> https://help.github.com/articles/github-glossary/.
>
> For more information on Git, go to one of these sites: https://help.github.com/ or http://git-scm.com/.

Installing and Configuring the Studio Git Client

In order to use Git, you must install the Git 64-bit client version 2.9.2 (and all its references).

1. Download the Git client from the following URL: https://github.com/git-for-windows/git/releases/download/v2.9.2.windows.1/Git-2.9.2-64-bit.exe.

2. Save the Git client to **<oo_installation_folder>/studio/Git**, so that the **bin** folder is directly under **<oo_installation_folder>/studio/Git**. In the Git installation wizard, use the default options.

Alternatively, if you already have a Git client installation on your local disk, point Studio to use that Git installation by performing the following steps:

1. Close Studio.

2. Go to the user home folder **C:\Users\<user>\.oo** (the Studio workspace location) and locate the **Studio.properties** file.

3. Modify the **Studio.properties** file by adding the following property at the end of the file:

   ```
   studio.git.installation.location=<git_installation_folder>
   ```

   For example:

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

   The **bin** folder should be directly under **C:/Program Files/Git**. Note that **/** should be used as a path separator.

4. Save the **Studio.properties** file and start Studio.

> **Note:** If you opted for this second alternative, you need to consider the following:
>
> If you are using multiple workspaces and you want the Git location property to be added in each new workspace, you should edit the template properties file located in **Studio\conf\studio.properties.template**. Otherwise, each time you switch to a new workspace, you will have to set the Git location in the new workspace in the **.oo\Studio.properties** file.

If you have another version of the Git client installed, note that you must use version 2.9.2 of Git with Studio. This is the version that was validated with Studio. While other versions might still work correctly, they are not officially supported.

Getting Started with Git in Studio

To begin working with Git, you must connect to the remote Git repository and clone it. This way, you have a copy of the repository that you can work on, and then commit changes to the local repository and push your changes to the remote Git repository.

Have a look at the following movies to get a better understanding on how to work with Git in Studio.

Movie	What you will learn
Connecting to Git and cloning a Git repository	In this movie, you will be introduced to some basic Git operations. You will learn how to clone a Git repository, commit changes to the local repository and push them to the remote repository.
Working with Git – Basic Operations	In this movie, you can see how to use a different type of authentication (SSH) to clone the Git repository, learn how to stash local changes, revert to an earlier committed version and unstash.
Conflict Handling	In this movie, you will see how to handle conflicts that may occur in Git when two authors make changes to the same Studio workspace and try to commit and push to Git.
Git branching functionality	In this movie, you will learn how to work with different branches in Git.
Multi-authoring	In this movie, you will learn about advanced multi-authoring capabilities in Git.

Git Branch support

Branching is one of the strong points of Git, one that is a differentiator when compared to Subversion. One most popular branching model is explained on this blog: http://nvie.com/posts/a-successful-git-branching-model/.

Branch management is an important part of the Git workflow. Branches make it easier to track changes and collaborate with other people. You can manage branches in the Git repository directly from Studio. You can see more information on Git branching here:

https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow/

For details on creating a new branch, see "Create a new branch in the Git repository" below.

Understanding the Git Repository Log

The **Git Repository Log** shows a graphic representation of the latest commits and updates that have been made to the remote Git repository. This functionality is similar to the logs provided in other Git tools.

The repository log shows the following information:

- Important pointers (such as master, HEAD, origin/master, branches, tags etc.), that allow you to identify the branch from which the commit was done type of commit and whether or not it was pushed to a remote branch

- A commit graph showing the relationship between the different commits/updates (based on their ancestry) and showing how merges between the commits/updates were done

- The commit message, as entered in the Commit window

- The author of the commit (name and email address)

- The date and time of the commit (in chronological order)

Each node in the graph represents a commit/update operation. Depending on the update strategy selected (see "Git Update Strategies " below), the path will branch off into a new branch.

> **Note:** Every workspace contains its own **logs** folder with its own log file.

Git Update Strategies

When updating files from the remote Git repository, you can choose the update strategy you want to use. Depending on the update strategy selected, the path will branch off into a new branch.

- **Update strategy: Branch Default** The default update strategy (which is always either merge or rebase) is configurable by the Git repository configuration using one of the following Git keys: **pull.rebase** and **branch.<name>.rebase**. For more information, see http://git-scm.com/docs/git-config.

- **Update strategy: Merge** When using the update strategy merge, Git preserves the original hierarchy of the local commits, and creates a special "merge" commit that merges the two diverged "branches" (the local branch and the remote branch) back together.

  Example: Consider a simple example of a repository that has the following graph before updating.
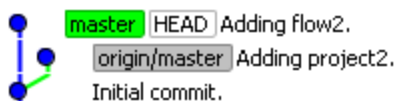
  In this case, the local repository and the remote repository have diverged. The local repository contains commits that the remote repository does not include (the commit with the message **Adding flow2**), and the remote repository contains commits that the local repository does not include (the commit with the message **Adding project2**).

  After merging the two commits, a special merge commit **Merge branch 'master' of ….** is created:
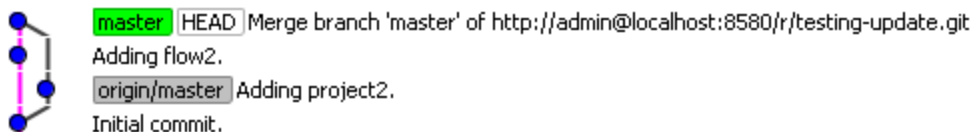
  If conflicts occur during the merge operation, the merge commit is not created automatically. The merging stops, and you must resolve the conflicts in order to complete the merge process.

In this state, all the remote changes are shown as local changes, visible in the **SCM Changes** pane. It is your responsibility to resolve the conflicts, review all the changes, and conclude the merging by running a **Commit All Changes** operation from the toolbar in the SCM Changes pane. See "Resolve conflicts during an Update operation" on page 362 for details.

If there are multiple conflicts from multiple local commits, all the conflicts are resolved at once, at the same time.

- **Update strategy: Rebase** Rebasing is an alternative method of integrating changes that is sometimes preferable over merge. Using merge makes the repository history grow horizontally (as information of all the separate "branches" are kept) and it inserts new merge commits, making it grow vertically as well. This makes it difficult to get an overall picture of the changes made to a repository.

  Rebasing changes all the local commits to appear as if they were based on the most recent changes from the remote repository, making the repository log graph appear as a straight line. Rebasing is an iterative operation: first it rebases the oldest (least recent) local commit on top of the latest from the remote repository, then it rebases the second commit on top of the previously rebased commit, and so on, until it rebases all the local commits.

  If there are conflicts during the rebase of a certain commit, rebasing stops and you can select to resolve the conflicts and continue rebasing, or to skip the commit (discarding all its changes). In any event, you can choose to abort the rebase operation and return to the initial state, with all previously rebased commits disappearing.

  An important implication of rebasing is that you may need to resolve conflicts multiple times (when rebasing different commits), possibly on the same items.

  Example:

  Both repositories (local and remote) are identical until the author commits **Introducing Utility Operations**. Then, the author of the local repository adds the operations **Base64 coders**, **UUID Generator**, **Remote Command Execution** and **Groovy Scripts**.

  In the meantime, another author pushes his commits to the remote repository, adding **Search and Replace**, his own version of the **Remote Command Execution** and **Remote File Transfer** operations. The **Remote Command Execution** operation results in a conflict as it was added by both authors with different content.

  After running an **Update All** operation with rebase selected as the Update strategy, the following events occur:

  ○ Studio displays the Conflicts dialog box with the following message: **"Merge of the current commit: 4d10a29 detected conflicts. Resolve them before continuing the rebase."** The item in conflict is a flow named **Remote Command Execution**.

Resolve manually is chosen for the item.

○ The SCM Messages pane shows that two commits were successfully rebased (**Adding UUID Generator** and **Adding Base64 coders**), rebasing stopped due to conflicts, and the current commit is **Adding Remote Command Execution**.

At this stage, the repository log looks as follows:

The commits **Adding Base64 coders** and **Adding UUID Generator** appear twice: first, at their original location based on **Introducing Utility Operations**, and second, at the top of the list based on the latest commit from origin/master: **Adding Remote File Transfer**. Note that the current branch ("master") is still intact. if the author chooses to abort rebasing, the workspace will go back to the master branch.

Let as assume that the user resolves the conflict and chooses to continue rebasing. At this point, then, the SCM Messages explains that it further rebased two commits (**Adding Remote Command Execution** and **Adding Groovy Scripts**), and rebasing completed successfully. The Repository log looks like this:
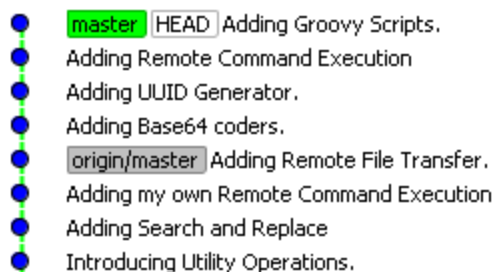
Notice that after rebasing, the graph the log is a straight line. After rebasing, the author can push the newly rebased commits to the remote repository.

> **Note:** When there are conflicting changes on moved or renamed items (during Update), Studio automatically detects these and tries to resolve them automatically.
>
> For example, user1 moves a flow and makes some changes to it, and user2 makes some non-conflicting changes on the same flow. Studio automatically detects this situation, and tries to merge the changes to the flow.

See for details on how to perform a Git Update operation.

When is the Git Repository Log refreshed?

Studio automatically updates the Git Repository Log when:

- Studio starts up (and is connected to Git)
- After checking out a Git repository
- After committing files to the remote Git repository
- After updating files from the remote Git repository
- After detaching from the remote Git repository
- After you click (a manual refresh in the Git Repository Log)

Conflict Handling in Git

In Git, a change that you made is referred to as "ours", while a change made by a different user is referred to as "theirs".

When you try to update your local repository, a conflict may occur between a local copy of a file (added/edited/deleted by "us") and the file in the remote Git repository (added/edited/deleted by "them") in the following cases:

- A file was deleted by you ("us") and was modified by another user ("them").
- A file was deleted by another user ("them") and was modified by you ("us").
- An item was added by both you ("us") and another user ("them") with different contents.
- The same item was modified by both you ("us") and another user ("them"), and it is not possible to automatically merge the changes.

Different Types of Conflict

There are three situations in which an author may come across conflicts – all these can be encountered on an **Update** operation, but one of them can also be seen on an **Unstash** operation. In each situation, he type of conflict that occurred is shown in the Conflicts dialog.

- **Merge:** Conflicts during a merge operation. In an **Update** operation, a merge is also made if the update strategy selected in the Update window was **Merge**). For details on how to resolve this type of conflict, see "Resolving Conflicts: When to use Accept Our/Theirs and when to use Prefer

Ours/Theirs" on the next page

- **Rebase**: Conflicts during rebase. In an **Update** operation, a rebase is also made if the update strategy selected in the Update window was **Rebase**.

  > **Note:** In this situation, the meaning of "ours" and "theirs" are opposite to the Merge case. So, for example, if the author selects "Accept/prefer ours", he is accepting/preferring the change that comes from the remote branch, and if he selects "Accept/prefer theirs", he is accepting/preferring the change from the local branch.

- **Unstash**: Conflicts during unstash. These occur if before the Update operation, the author has uncommitted changes that are in conflict with items that are located on the remote repository.

  These conflicts behave exactly like conflicts that occur during rebase (as described "Rebase: Conflicts during rebase. In an Update operation, a rebase is also made if the update strategy selected in the Update window was Rebase. "). For details, see "Manually Unstash Changes" on page 367.

See "Resolve conflicts during an Update operation" on page 362 for details on how to resolve conflicts.

Resolving Conflicts: When to use Accept Our/Theirs and when to use Prefer Ours/Theirs

Studio provides two methods of resolving conflicts: **Accept** and **Prefer**.

The **Accept Ours/Theirs** resolution uses the entire file from us/them. There is no merging of files, even if there are changes in some of the files.

The **Prefer Ours/Theirs** resolution automatically merges as many changes as possible into the selected file. If merging is not possible, the system selects changes from the preferred side. All other changes that cannot be automatically resolved, can be manually merged later in Studio. See "Resolve conflicts during an Update operation" on page 362 for an example of automatic and manual resolution of conflicts.

For example, two users, user1 and user2, made changes on the same flow.

- user1 changed a certain step and added a flow-level input named **input1**.

- user2 changed the same step and added a flow-level output named **output2**.

In this case, if user1 has to resolve the conflict, s/he has the following options:

- **Accept Ours** - user1's entire flow will be used: user1's version of the step, with **input1**, but without **output2**.

- **Accept Theirs** - user2's entire flow will be used: user2's version of the step, **output2**, but without **input1**.

- **Resolve Manually Preferring Ours** - the flow will contain user1's version of the step, and both **input1** and **output2**.

- **Resolve Manually Preferring Theirs** - the flow will contain user2's version of the step, and both **input1** and **output2**.

> **Note:** If there are conflicting changes on moved and renamed items during an **Update** operation, Studio detects these and tries to resolve them automatically as shown in the following example:
>
> 1. User1 moves a flow and makes some changes to it.
>
> 2. User2 makes some non-conflicting changes on the same flow.

Studio automatically detects these changes and merges them automatically. There is no conflict and both users can see their changes to the flow.

Authentication Options

When connecting to a Git repository, you can choose to connect to the Git server using one of two authentication schemes: Credentials or SSH Private Key.

- **Git Repository Credentials: Password authentication**

- **Private key file:** To use this scheme, you need to fill in the Private Key File and Private Key Passphrase. You can generate a **public key-private key** pair by running the following command from the git command line:

```
$ ssh-keygen -t <algorithm_type> -C "your_email@example.com"
```

where `algorithm_type` may be `rsa1` for protocol version 1 and `rsa` or `dsa` for protocol version 2.

> **Note:** The private key must be in OpenSSH format. For public keys, the RSA and DSA algorithms are supported. A public key must have an extension of **.pub** only.

In addition, you can configure known host parameters in the **studio.properties** file:

```
git.ssh.known.hosts.policy=allow|strict|add
```

The policy for known hosts. If set to `add`, a connection to a host that is not in the **known_hosts** file will be allowed and the host and connection will be added to the file. If set to `allow`, Studio allows SSH connections to any host. If set to `strict`, only allow connections to hosts specified inside the **known_hosts** file are allowed.

```
git.ssh.known.hosts.file
```

The location of the **known_hosts** file. By default, the value is `${user.home}/.ssh/known_hosts`.

You can select the authentication type in the SCM Connection dialog. For details, see "Connect to a private Git repository" below.

> **Connecting to a Git repository using a proxy**
>
> All Git operations that are available in Studio and interact with the Git server can be done using `http/https` connection URLs through a proxy. The proxy can be without authentication or using basic authentication scheme.
>
> See "Set the Proxy for Debugging on a Remote Central " on page 614 for details.

What do you want to do?

> **Note:** SCM operations cannot be performed on items that have duplicate UUIDs.
>
> To resolve this:
>
> 1. View the problematic items in the **Problems** pane.
>
> 2. Delete the duplicated item.
>
> 3. Leave the original item as is.
>
> If you want to keep both items, use the **Copy/Paste** or **Duplicate** commands to save a copy of one of the items with a different UUID, and then delete the item that you copied from.

Connect to a private Git repository

You must use personal access tokens in Git to connect to a private Git repository. To do this:

1. In the Git repository, go to the **Settings** tab.

2. Click **Personal access tokens** > **Generate new token** > **Copy token**.

3. In Studio, select **SCM** > **Connection**.

4. In the SCM Connection dialog:
   a. In the **Authentication** area, select **Credentials** from the **Type** drop-down list.

   b. In the **User Name** field, type your user name .

   c. In the **Password** field, type the value of the token you generated in the Git repository.

   d. In the **User Name** area, type your user name and email ID.

   e. Click **OK**.
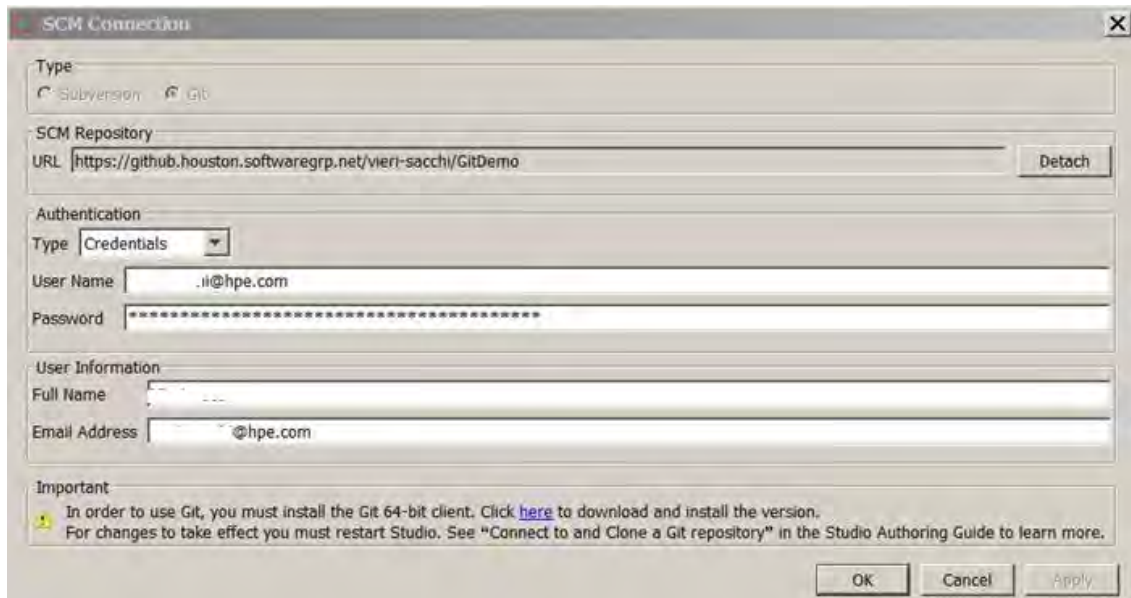
Connect to and Clone a Git repository

In order to work with the Git source control system, you must first connect to the Git repository and clone it to your local file system. When you clone a repository, Studio performs the following steps:

- Copies the repository into your Studio workspace

- Creates remote-tracking branches for each branch in the cloned repository

- Creates and checks out an initial branch that is forked from the cloned repository's currently active branch.

You can then work locally, commit your changes and then push the changes to the Git remote branch.

> **Note:**
>
> - Upon cloning, the default branch is always checked out.
>
> - If the Git server verification fails, an error message opens. You can choose to continue or abort the operation.

1. Select **SCM > Connection…**.

   The **SCM Connection** dialog box opens.

2. In the **Type** area, select the SCM type: **Git**.

3. In the **SCM Repository** area, in the **URL** field, type or paste the URL of the Git repository.

   > **Note:** The URL may include an **http**, **https**, **file** or **ssh** protocol.

> When using **https**, If the server certificate is not trusted, a message appears asking whether you want to trust this certificate. You can also import the certificate automatically as described in ""Import Certificates Automatically with a Remote Debugger Connection " on page 615.

4. **Authentication**: There are two types of authentication available:**Credentials** or **SSH Private Key**.

   ○ **Credentials**: In the **Type** dropdown, select **Credentials** to use Windows credentials, and enter the user name and password in the User Name and Password fields.

   ○ **SSH Private Key**: In the **Type** dropdown, select **SSH Private Key** to use SSH private key authentication, and enter/select the private key file and passphrase in the **Private Key File** and **Private Key Passphrase** fields.

   > **Note:** Click on the [icon] icon to select a private key file from a location on the disk.

5. **User Information**: Enter the user name and email address in the **Full Name** and **Email Address** fields. Both these fields must be filled in as this information is attached to every Git commit operation.If you do not fill in these fields, Git will use the computer name as the full name, and the `<computer name>@<full computer name>` as the email address.

6. Click **Clone**.

   A message in the following format appears in the SCM messages pane:

   ```
   <date> <time> - Clone https://<full_path>/gitTestRepo.git

   Cloning from https://<full_path>/gitTestRepo.git
   to C:\Users\<username>\.oo\Workspace was done successfully.
   ```

> **Note:**
>
> • In order for the **Clone** command to run correctly with a destination directory that is not empty, Studio creates a blank directory (under *C:\...\.oo\Workspace\.git*) and copies the entire Git repository into the new directory.
>
> • The first time you try to connect to Git, if you have filled in the Windows credentials, Git will use these to configure and populate the Git library.

> **Note:** There is no Lock functionality in Git. This is because there is no need to lock files in Git, as there is the Git Conflict management. Therefore, the **Lock** button in the editor for items (flows, operations, configuration items) is always disabled – the same behavior as when you are not connected to any SCM repository.

Add files to the local Git repository

1. From the projects tree, select the project whose files you want to add to the repository.

2. Right-click and select **SCM > Add**.

3. A message appears in the **SCM messages** pane. For example:

   ```
   Add C:\Users\<username>\.oo\Workspace\Project\Content\Configuration\Scriptlets
   ```

   Note: You cannot add projects that are located outside the current workspace.

Move/cross-move a versioned item

You can move an item from one location to another within the same project or cross-move an item between two different projects.

To move an item:

1. In the **Project** pane, select the items that you want to move. You can select multiple items using the **Shift** or **Ctrl** key.

2. Drag and drop the items into another folder or another project.

   or:

   Cut the items using **Ctrl+X** and paste them at the new location using **Ctrl+V**.

   The item is marked in the **SCM Changes** pane in the old location with the status **deleted** and in the new location with a status **moved**.You can see the original location in parentheses. For

example:
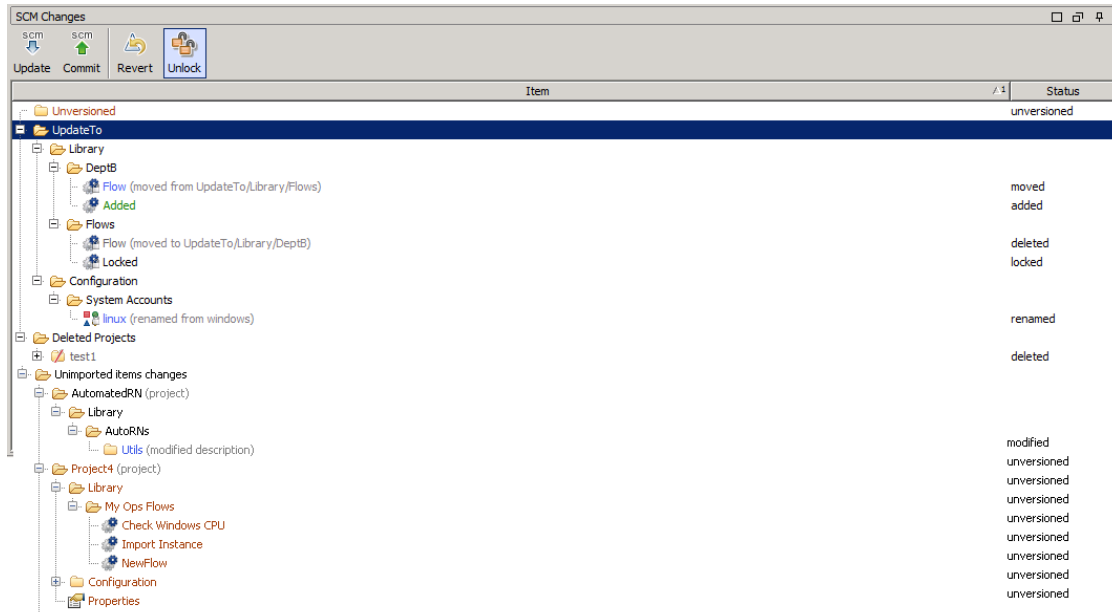
Note:

- Where relevant you can commit the result of the move/cross-move action to Git.

Update files from the remote Git repository

The **SCM Update** operation performs three Git operations, Stash, Pull and Unstash.

Stashing takes the current state of your local working workspace — that is, your modified tracked files and staged changes — and saves it on a stack of unfinished changes that you can reapply at any time.

If changes exists in your local workspace, and you have committed them, Studio tries to merge them automatically with the files in the remote Git repository. If there are conflicts, the Conflicts dialog opens and you can resolve the conflicts as described in "Resolve conflicts during an Update operation" on the next page.

Note: You can also perform manual **Stash/Unstash** operations only using the **Stash** and **Unstash** options from the SCM > Git menu. For details, see "Manually Stash Local Changes" on page 365 and "Manually Unstash Changes" on page 367.

1. Right-click and select **SCM > Update**.
2. The following window opens:
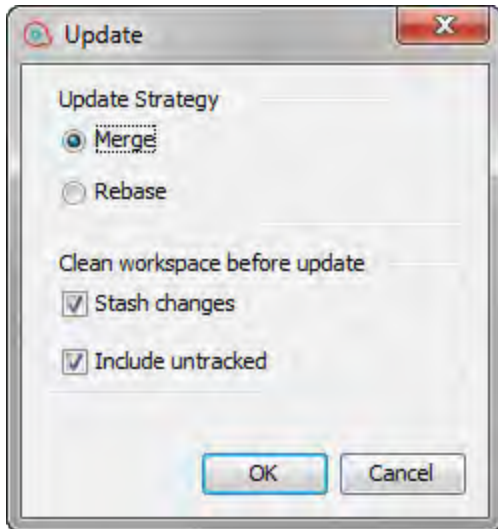
3. Select the Git Update strategy you want to use:

   ○ **Merge:** When using the update strategy Merge, Git preserves the original hierarchy of the local commits, and creates a special "merge" commit that merges the two diverged "branches" (the local branch and the remote branch) back together.

   ○ **Rebase:** The Rebase strategy changes all the local commits to look like they were made based on the most recent changes from the remote repository. This shows in the repository log as a straight line, preventing the horizontal growth that is present when using Merge.

   > **Note:**
   >
   > - You can change the default Update strategy using the **pull.rebase** and **branch.<name>.rebase** keys. For more information, see http://git-scm.com/docs/git-config.
   >
   > - When performing a Rebase operation, there may be several conflicts for the same file.

4. Select the **Stash changes** check box to automatically stash local changes before updating.

5. Select the **Include untracked** check box to include the untracked items (the unversioned items marked in brown) in the stash.

> **Note:** If you try to update from the remote Git repository and it is currently empty, Studio displays the following error message:

```
Cannot update the current branch as it does not exist in the remote
repository.
```

Resolve conflicts during an Update operation

Conflicts during an Update operation may appear in one of the following situations:

- **On merge:** If the author selects the Merge strategy on update and there are conflicting differences between the commits on the local branch and the commits coming from the remote branch.

- **On rebase:** If the author selects the Rebase strategy on update and there are conflicting differences between the commits on the local branch and the committed items coming from the remote branch.

- **On stash:** If the author has local changes that were not committed but have been stashed, and there are conflicting differences between the these items and the committed items that come from remote branch. See "Manually Stash Local Changes" on page 365 for details.

You can see any conflicts that arise in the Conflicts dialog box. For example:
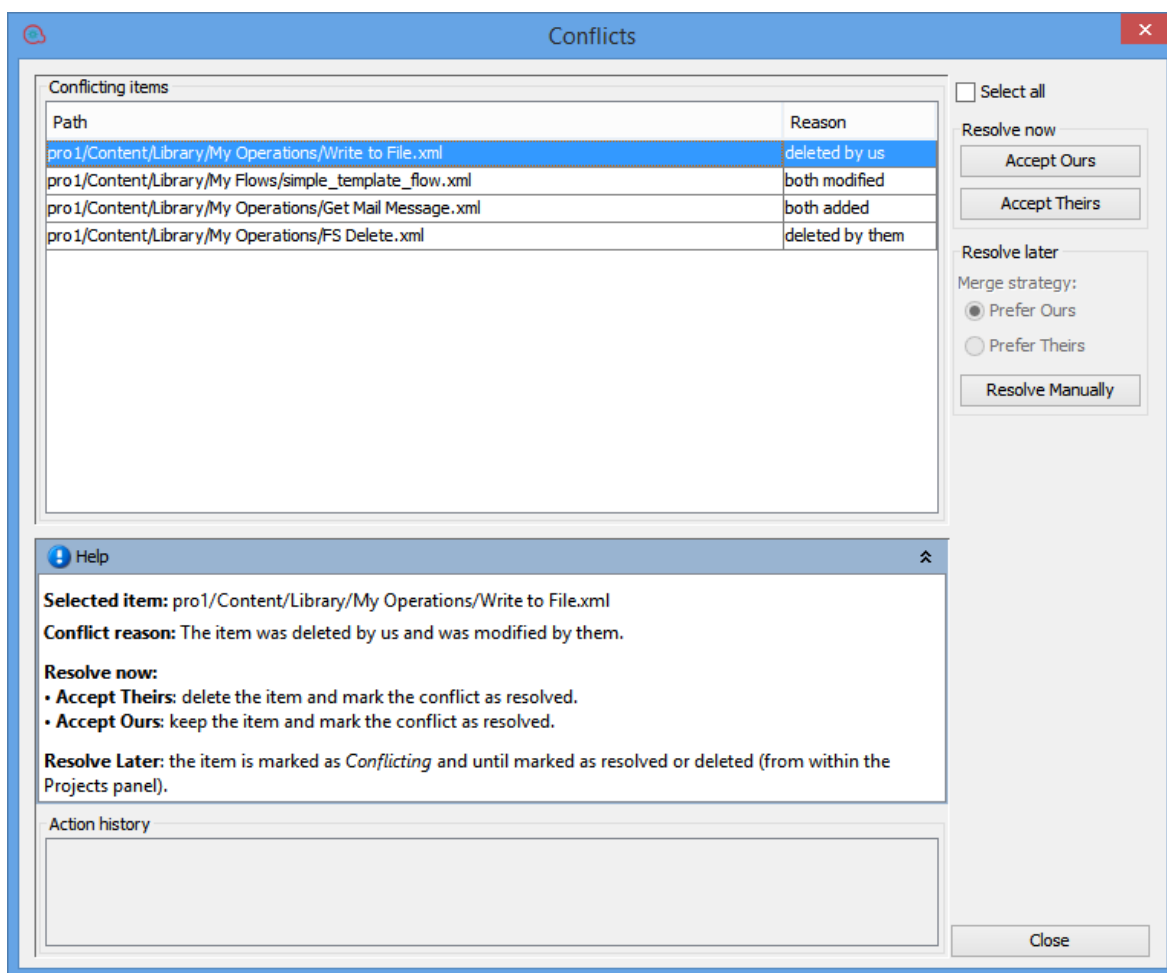
Note:

- There may be several conflicts in the same item. You must resolve all the conflicts in order to mark the item as resolved

- When the author updates his workspace all the added, unversioned projects are stashed, among other added/unversioned flows/operations/CIs. If they are not stashed the update cannot be done.

  Another solution is to commit the local changes (added/unversioned items). The information about all the stashed items appears in the SCM messages pane. If the update results in a conflict, after resolving the conflict the author must manually unstash the stashed items.

For each conflict, you can choose to resolve the conflict now or later.

Resolve Now

- From the right pane, select **Accept Ours** to apply the changes made by "us" (the ones that are committed on our local repository).

  or:

  Select **Accept Theirs** to apply the changes made by "them" (the ones received by pulling from the remote repository).

Resolve Later (Manually)

If you choose to resolve a conflict at a later stage, Studio prepares the item for manual conflict resolution. All non-conflicting changes are automatically merged into the item, and conflicting changes are replaced based on your selected preference (ours or theirs).

The original file is marked as **Conflicting** until marked as resolved or deleted (from within the **Projects** pane).

If the conflict type is not "deleted by us" or "deleted by them", Studio automatically creates two new items named `<filename> - ours` and `<filename> - theirs` shown in red:

- `<filename> - ours`: The original version of the item (from the local repository)

- `<filename> - theirs`: The original version of the item (from the remote Git repository)

For example:
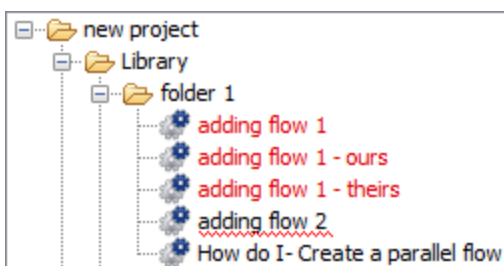
These files are read-only and cannot be modified or deleted/cut. After the conflict is resolved, Studio automatically deletes the "ours" and "theirs" files.

> **Note:**
>
> - When there are unresolved merge conflicts, there may be some operations that you will not be able to perform on the conflicting items, for example, committing them to the Git repository. In addition, you cannot move or rename conflicting items or the folders containing them.
>
> - Conflicting files are shown in the Problems pane.
>
> - The new file, created by merging conflicts, automatically or manually, may be in an invalid state.

To manually resolve the conflict:

1. From the right pane, select **Prefer Ours** to apply the changes made by "us" (the ones that are committed on our local repository).

   or:

   Select **Prefer Theirs** to apply the changes made by "them" (the ones received by pulling from the remote repository).

2. Select **Resolve Manually**.

3. Right-click on a conflicting item from the project tree, and select **SCM > Mark Conflict as Resolved**, **SCM > Resolve Conflict Accepting Ours** or **SCM > Resolve Conflict Accepting Theirs**.

4. Commit and push all the changes to the remote repository.

Manually Stash Local Changes

Stash enables you to create a "patch" of your local changes saved inside your local workspace, that can be accessed later, at any time.

When you creates a stash, all the changes in the workspace are "stashed", and the local changes are *reverted*, i.e., removed from the workspace.

> **Note:** If you try to stash changes while connected to an uninitialized, empty repository, Studio displays the following error message: "Before stashing, you must initialize your repository by a first commit".

> **Note:** The **Stash** operations always include all changes made in the workspace.

You can also choose whether or not to include the untracked files.

> **Note:** If you have unimported projects in your workspace, when you perform a **Stash** operation, these projects are included. After stashing, these files are stored in memory and not in the workspace.
>
> However, the empty folders still remain. At any point, you can unstash your changes and the stashed files will return to their original location.
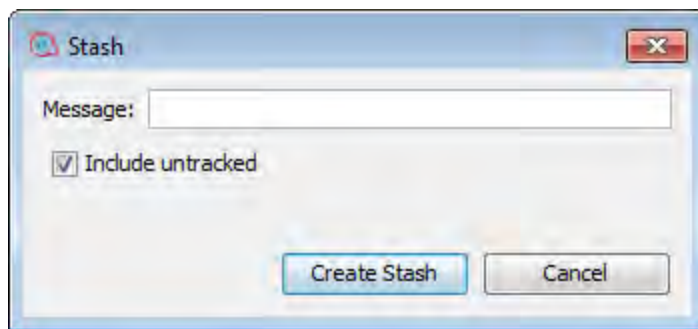
To stash local changes:

1. From the SCM menu, select **Git > Stash**.

   or:

   Click ![stash icon] in the SCM Change Pane toolbar.

   The Stash dialog box opens:

2. In the Message field, type a message to identify the Stash operation. For example, `Changes to the Integration Operations`.

3. Select the **Include untracked** check box to include the untracked files in the stash.

4. Click **Create Stash**.

   When the Stash operation has completed, you can see the operations that have been performed in the **SCM Messages** pane. For example:

   ```
   04/14/15 15:13:30 - Stash
   Stashing changes in the workspace including the untracked files.
   Removed project(s): oo-virtualization-project
   Stashed paths:
   added: oo-virtualization-project/.gitignore
   added: oo-virtualization-project/Content/Library.properties
   added: oo-virtualization-project/Content/Library/Accelerator Packs.properties
   added: oo-virtualization-project/Content/Library/Accelerator
   Packs/Virtualization.properties
   added: oo-virtualization-project/Content/Library/Accelerator
   ```

```
Packs/Virtualization/Get All Snapshots.xml
added: oo-virtualization-project/contentpack.properties
added: oo-virtualization-project/pom.xml
```

Manually Unstash Changes

The Unstash operation enables you to apply previously stashed changes. This operation is equivalent to a **git stash apply** operation. The Unstash dialog box shows a list of stashes along with the details of the currently selected stash.

> **Note:** Take care when running an Unstash operation as it may result in conflicts.
>
> Conflicts can occur during an Unstash operation if the author has uncommitted changes that are in conflict with items that are located on the remote repository.

To unstash changes:

1. Select the stash commit you want to unstash.

2. From the SCM menu, select **Git > Unstash**.

   The Unstash dialog box opens:
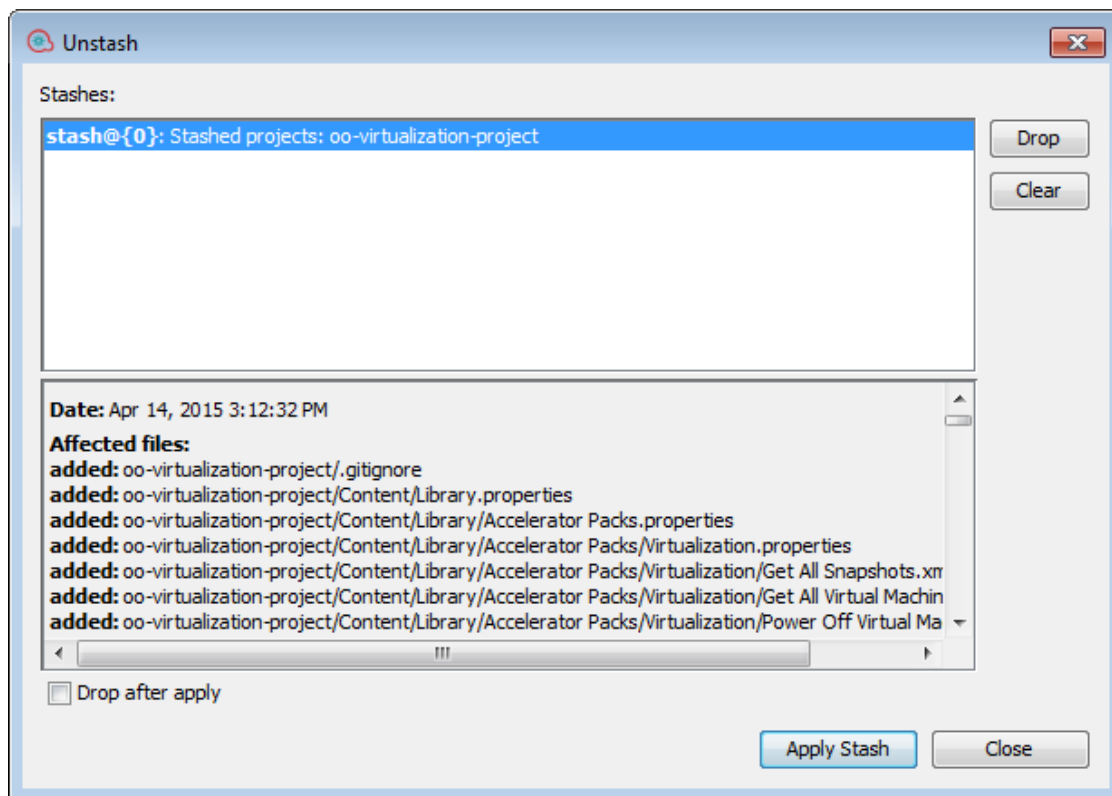
3. Select **Drop** to delete the currently selected stash.

4. Select **Clear** to delete all the stashes.

5. Select **Apply Stash** to implement the currently selected stash(es). If **Drop after apply** is selected, the **Apply Stash** operation will also delete the currently selected stash.

   Note: If the author has uncommitted changes that are in conflict with items that are located on the remote repository, a conflict may occur here.
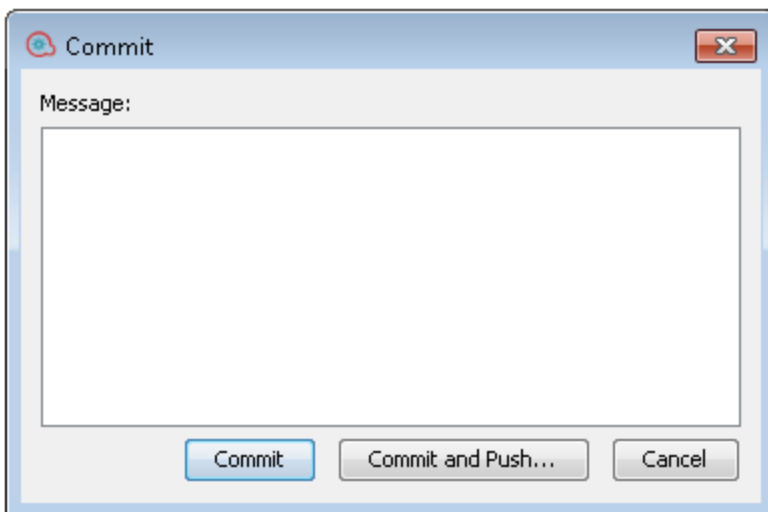
Commit and push files to the Git repository

After making changes in Studio, you must commit the files to the local Git repository in order to record the changes, and then push them to the remote Git repository on the Git server.

You can choose to commit changes, and then push them later, or to commit and push them in the same step.

Best practice: In order to ensure that you have the latest files, before committing it is highly recommended to update the files (pull) from the remote Git repository using the command **SCM > Update**.

1. From the projects tree, select the folder whose files you want to commit to the local repository.

2. Right-click and select **SCM > Commit**.

   The Commit dialog box opens:

3. Click **Commit** to commit the changes to the local repository.

   or:

Click **Commit and Push...** to commit the changes to the local Git repository then push (copy) them to the remote Git repository.

> **Note:** Before pushing, your entire SCM workspace must be up-to-date. The push operation pushes all committed files in the workspace to the remote Git repository. If there are uncommitted changes, they are not pushed to the remote repository.
>
> You can also push changes to the remote repository using the **SCM > Git > Push..** option from the main menu. See "Push files to the Git repository" below for details.

> **Note:** If there are items with duplicate UUIDs in your commit list, they are shown in underlined text and Studio warns you with the following message:
>
> ```
> There is at least one other item with the same UUID as one of the items
> included in the commit. Do you want to continue?
> ```
>
> Click **Yes** to ignore the duplicate UUIDs and proceed with the commit, or **No** to cancel the commit operation.

The following message appears in the SCM messages pane:

```
Pushed 1 commit(s) to origin/master
```

> **Note:** Git does not store empty folders in the repository. However, in Studio, you can have empty folders inside projects, as Studio automatically creates them as required..

Push files to the Git repository

After making changes in Studio, you must commit the files to the local Git repository in order to record the changes, and then push them to the remote Git repository on the Git server.

You can choose to commit changes, and then push them later, or to commit and push them in the same step. This procedure describes how to push changes to the remote Git repository.

> **Note:** When you push your changes to the remote server, *all* your commits are pushed.

> **Best practice:** In order to ensure that you have the latest files, before committing it is highly recommended to update the files (pull) from the remote Git repository using the command **SCM > Update**.

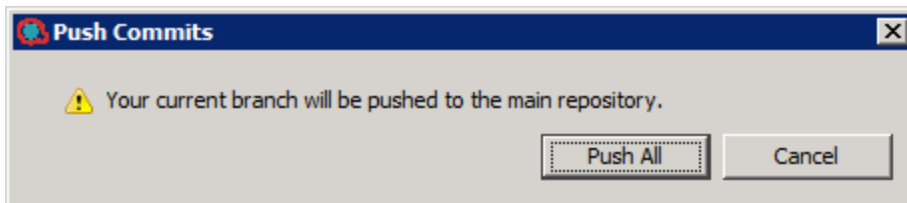1. From the main menu, select **SCM > Git > Push...**.

   or:

   From the **SCM Changes** pane, click .

The Push Commits dialog box opens showing the list of commits to be pushed (along with the Commit ID) and an aggregated list of affected files for each commit:
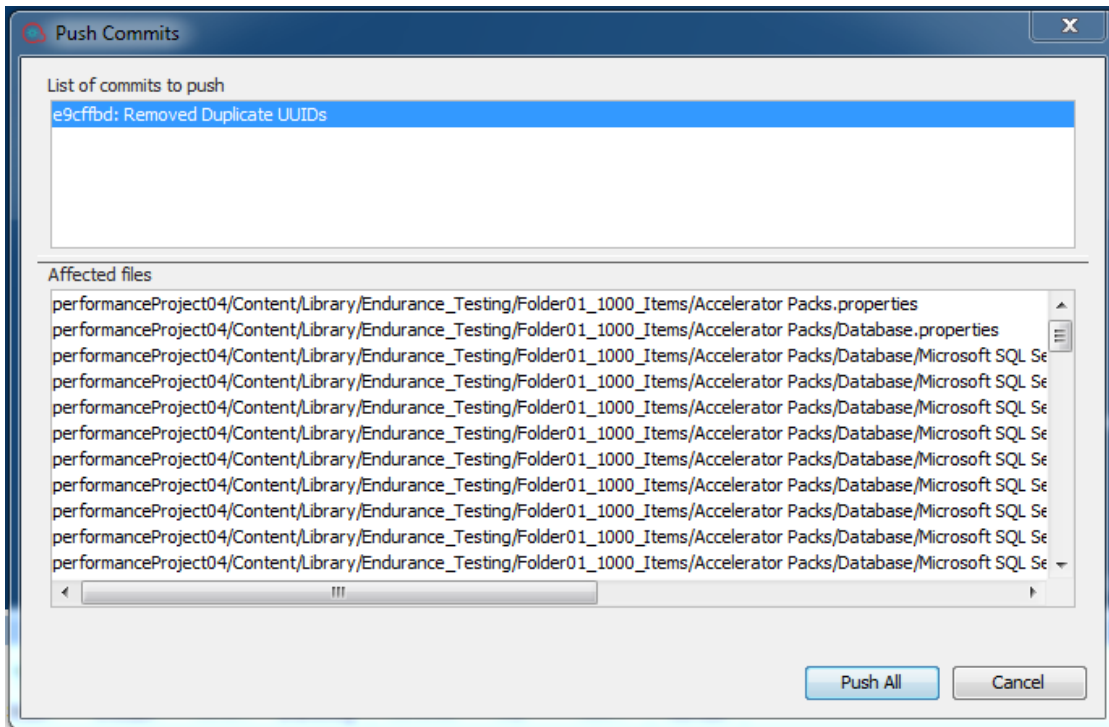
If the branch is not pushed on the remote repository, the Push window looks as follows:

2. From the **List of commits to push**, select a commit to see the list of affected files in the **Affected Files** section.

   Note:

   ○ Before pushing, your entire SCM workspace must be up-to-date. This means that there must not be any files in the workspace that have not yet been committed or updated.

   ○ If the remote Git repository is empty and you try to push to it without a local commit, Studio displays the following error message:

   ```
   Before pushing, you must initialize your repository by a first
   commit.
   ```

The list of affected files is shown in the **Affected Files** section.

3. Click **Push All** to push all the commit operations to the remote repository.

    A confirmation message appears in the SCM Messages pane. For example:

    ```
    04/14/15 13:19:19 - Push
    Push successful
    Pushed 3 commit(s) to origin/master
    ```

Abort Merging or Rebasing

There may be cases in which you do not want to accept changes from the branch you are merging/rebasing with. Aborting is common in the case of rebase.

When rebasing, you go through each commit from the branch you are rebasing and add it sequentially to the branch you are rebasing in. If you find a problem after several such commits, you can abort the rebase and then restart it.

In the case of merging, you may have a conflict in your workspace that you did not manage to resolve as you intended. In this case you can choose to abort the operation.

1. From the SCM menu, select **Git > Abort Rebasing or Merging**.

    or:

    From the SCM Changes toolbar, select

    A confirmation message appears in SCM Messages pane.

## View a history of Git operations in the repository

The **History** option shows you:

*   A list of commits in which the selected project/item was affected
*   All affected files for a particular commit

The **History** option also allows you to revert to a specific revision of the project/item. See "Revert to a previous revision of a repository" on the next page for details.

1. Right-click on a file/folder and select **SCM > History**.

    The Revision History dialog box opens:
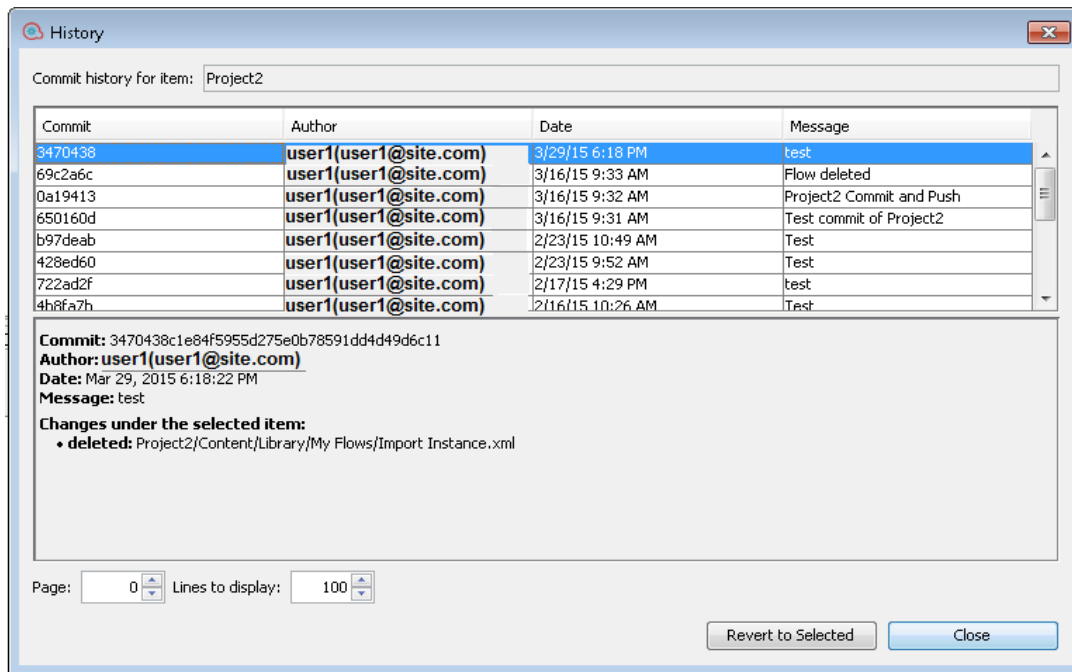
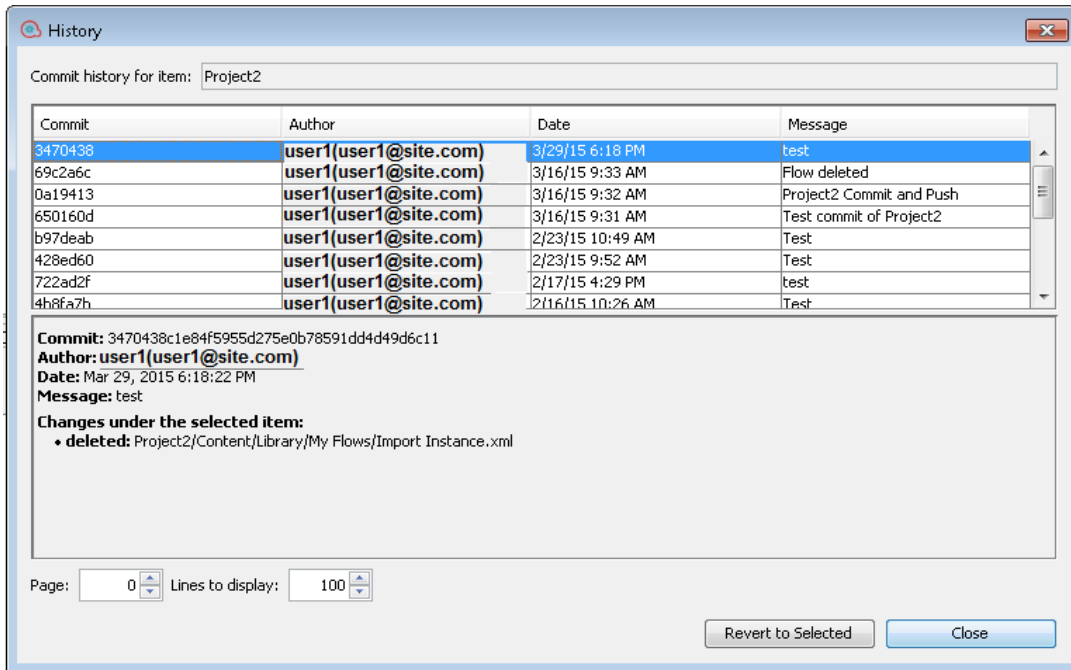## Revert to a previous revision of a repository

There may be cases for which you need to recreate a faulty Central content pack in order to troubleshoot a specific flow inside the content pack. If the content pack is not available (it has been lost or been overwritten by a new version), you can restore a specific version of the content pack using the **History** option.

> **Note:** In Git, you can perform a **Revert** operation only on repositories.

To revert to a previous revision:

1. Right-click on a file/folder and select **SCM > History**.

   The Revision History dialog box opens:

2. Select the commit revision you want to revert to.

3. Click **Revert to Selected**.

## Revert to a older version of a file/folder from the Git repository

Reverting removes the changes that the author made to the items (and in the case of folders, to all items they contain recursively), and returns the original state of the reverted items to the state they were in, the last time they were updated.

1. From the projects tree, select the requested folder/file(s).

2. Right-click and select **SCM > Revert**

   or:

   From the SCM Changes toolbar, select **Revert All Changes** .

   The following message appears:
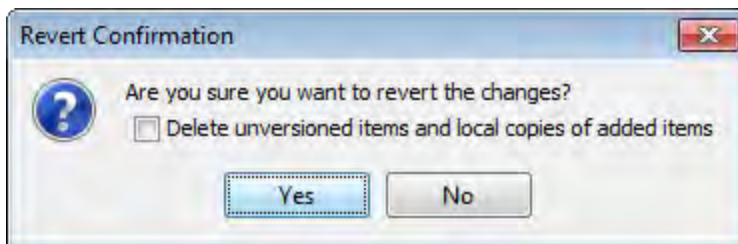
3. Select the check box if you want to delete all unversioned items and local copies of added items.

4. Select **Yes** to revert to the previous version.

Reset to a tagged commit of a project/item

There may be cases for which you need to recreate a faulty Central content pack/project in order to troubleshoot a specific flow inside the content pack. If the content pack is not available (it has been lost or been overwritten by a new version), you can restore a specific version of the project/item using the **Reset to tag...** operation.

You can use the **Reset to tag...** option if you have already checked out the repository and you want to update your workspace to a specific revision from the SCM tag.

> **Important!** The entire workspace reverts to the state it was in at the time it was tagged.

1. From the SCM menu, select **Git > Reset to tag...**.

2. Type in the SCM tag that you received from the Central administrator. By default, the SCM tag is `[Project Name]-[Project version]`.

   Note:

   ○ For Git, the SCM tag must not include blank spaces.

   ○ If you try to reset to a tag that is not on the current branch, an error message appears.

View the SCM repository log

To view the SCM repository log:

1. Select the **SCM Repository Log** tab at the bottom of the screen.

   The repository log opens. For example:

	Author	Date
master HEAD	user 1 <user 1>	2015-03-16 10:22:19
origin/master Commit.	user 1 <user 1>	2015-03-16 10:20:44
Merged.	user 1 <user 1>	2015-03-13 15:20:06
commit 2	user 1 <user 1>	2015-03-13 15:15:53
User 1 commit.	User 1 <User 1>	2015-03-13 15:14:49
merged	user 1 <user 1>	2015-03-12 17:38:48
user 2 changes.	user 2 <user 2>	2015-03-12 10:49:13
user 1 changes again,	user 1 <user 1>	2015-03-12 10:45:04
User 1 changes.	user 1 <user 1>	2015-03-12 10:42:43
Initial commit of my flows and operations (2).	user 1 <user 1>	2015-03-12 10:40:15
Initial commit of my flows and operations.	user 1 <user 1>	2015-03-12 10:40:04
v 100	user 1 <user 1>	2015-03-10 18:47:35

2. If the local and the remote repositories have different versions that have been merged, the split and merge are shown as two different paths in the repository log (as shown in the example above).

Add unversioned files to a Git repository

You can add unversioned files to the local Git repository at any stage. The files can be an entire project, a specific folder (either under "Library" or a configuration item folder), a flow, an operation or a configuration item.

> **Note:**
>
> - You cannot add files from projects located outside the current workspace.
>
> - When adding a file from an unversioned project, the entire project is added to the repository.

Added files appear in green in the **Projects** pane and the **SCM Changes** pane.

1. Right-click and select **SCM > Add**.

2. A confirmation message appears in SCM Messages pane.

Rename versioned files

When renaming versioned files, the files are also scheduled for renaming in the Git repository.

Renamed items appear in blue in the **SCM Changes** pane.

> **Note:** If the file is versioned but it has the status **Added**, the color of the file will be green even after a **Rename** action, both in the Projects Pane and in the **SCM Changes** pane.

If a Git error occurs during the operation, you can see it in the **SCM Messages** pane.

1. Right-click and select **Rename**.

2. Type a new name for the file and press Enter.

   A confirmation message appears in SCM Messages pane.

Delete versioned files

When deleting versioned files, the files are also scheduled for deletion in the Git repository.

Deleted items appear in gray in the **SCM Changes** pane. After you delete a file, you can still commit it to the repository from the **SCM Changes** pane. This is useful when working in multi-author environments.

If a Git error occurs during the operation, you can see it in the SCM Messages pane.

1. Right-click on the file(s) and select **Delete**.

   The file is deleted from the local repository and a confirmation message appears in **SCM Messages** pane.

2. You can now commit the deleted file(s) from the **SCM Changes** pane to remove it from the remote Git repository.

Detach from the Git repository

After you have finished working in the Git repository, it is a best practice to detach from the remote Git repository.

> **Note:** Before detaching, make sure that you have committed and pushed all your local changes to the remote Git repository.

If a Git error occurs during the operation, you can see it in the **SCM Messages** pane.

1. Select **SCM > Settings**

2. In the **SCM Repository** area, click **Detach**.

   An information message appears recommending that you to commit and push all your changes before detaching.

3. Click **OK**.

   Your local repository is detached from the remote Git repository.

Create a new branch in the Git repository

Creating a new (local) branch enables you to have a branch that initially points to the same commit as the current HEAD pointer. This means that the new branch will have the projects that Studio included when creating the branch. The newly created branch is immediately checked out and Studio connects automatically to the new branch). Any further commits are made to the current branch and you can choose later to push the branch to the remote repository.

> **Note:** When connected to an uninitialized repository, the local branch will be created only after an initial commit.

If a Git error occurs during the operation, you can see it in the SCM Messages pane.

1. From the **SCM** menu, select **Git > Create a new branch…**

   The Create New Branch dialog opens.

2. Type the name of the new branch in the field and click **OK**.

> **Note:** A Git repository name must not include spaces and must follow standard Git naming conventions.

The new branch is created in the local Git repository and a message is shown in the SCM Messages pane.

3. To make the branch available to other authors working with the same remote repository, you can push the current branch to the remote repository using the **Push** command.

Check out a branch in the Git repository

Checking out a branch allows you to connect to an existing branch. Connecting to an existing branch means that your workspace (along with the projects in it) will be changed to reflect the state of the newly checked out branch.

> **Note:**
>
> - Any project that exists in the newly checked out branch but does not exist in the current branch will be removed. The project data is not lost, however, as you can still check out that branch at a later stage).
>
> - Any project that exists in both branches will be changed and refreshed to reflect its state in the newly checked out branch.
>
> - Any project that exists only in the newly checked out branch will be copied to the disk, but Studio will not automatically import it.
>
> - Checking out from a branch that has local changes is not recommended.
>
>   If the same files that are modified locally are different in the branch to be checked out, the checkout will fail, keeping the current branch and the local changes.
>
>   In addition, local changes are lost when the branch is checked out.

1. From the **SCM** menu, select **Git > Check out the existing branch....**.

   The Checkout Branch dialog opens:
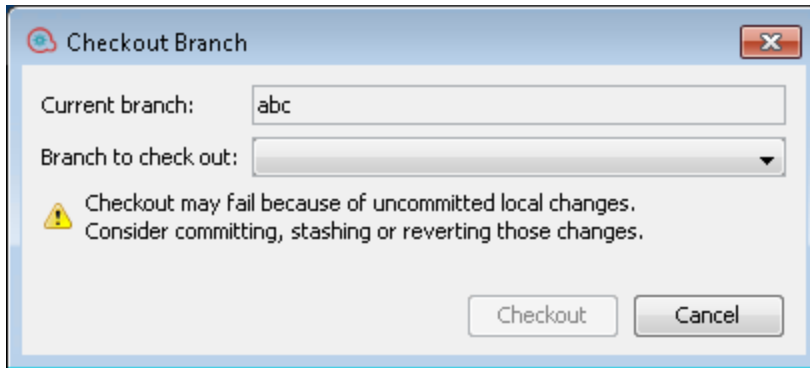
2. From the **Branch to check out** list, select the local branch you want to work on.

3. Click **Checkout**.

    The branch is checked out and ready for you to work on. You can see it the list of branches with a prefix of the remote repository's name. For example, `<master>-newbranch`.

    Note:

    ○ If a branch has not yet been pushed to the remote repository, it is listed with its name only under the Local Branches heading.

    ○ If you check out a remote branch that was never checked out locally (such as in the example below), Studio creates a local branch with the same name and configures its upstream to the remote branch.
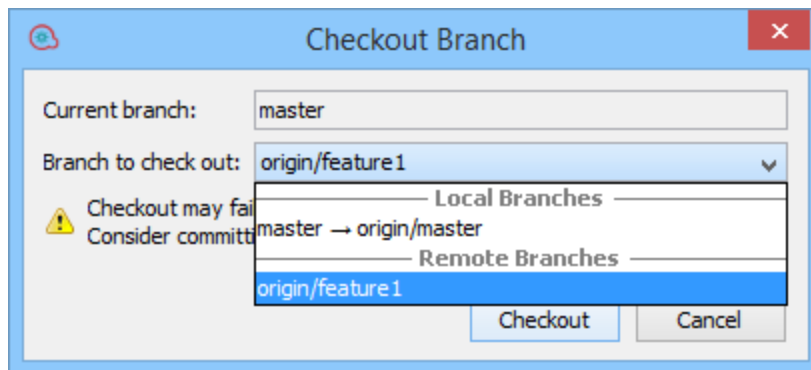
    The slash sign means the branch is on the remote repository. For example **origin/** refers to a branch named origin on the remote repository.

    The right arrow means that your local repository has previously been pushed to the remote repository and has a remote version. For example **master →origin/master** refers to a local branch **master** that has previously been pushed to the branch named **origin** on the remote repository.

> There may be more remote branches that are not visible as the information has not yet been updated from the remote repository. In order to update this information, you need to perform a Fetch operation.

After check out, the following message is displayed in the **SCM Messages** pane:

```
07/07/15 14:29:28 - Branching
Successfully checked out branch feature1 with upstream
origin/feature1.
```

View unimported items in the current workspace

In the **SCM Changes** pane, you can see items (files, folders or projects) in the current workspace that have been changed and have not been imported into Studio. The **SCM Changes** pane displays changes between the local copy of the repository (the workspace) and the SCM repository. If a file is not changed, it is not displayed.

This information is important when you have conflicts, or when merging/rebasing and you need to understand all the changes that were made.

Unimported items may have one of the following states:

- **unversioned:** The item is new in the workspace, and has not yet been pushed to the Git repository.

- **modified:** The item is already in the Git repository and has been changed.

1. Select the **SCM Changes** tab.

2. In the Item tree, select the **Unimported changes** node.

   A list of modified and unversioned files opens. For example:
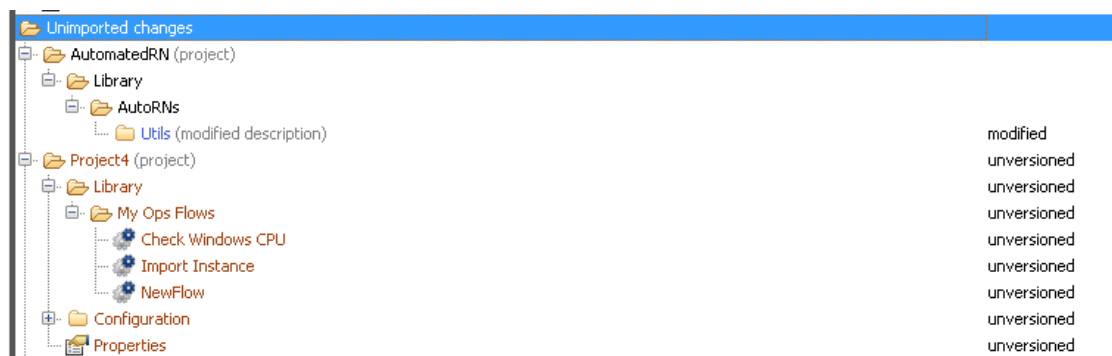
   From any item in the Unimported items changes node, you can perform the following operations:

   ○ **Show in Explorer:** Right-click on an item and select **Show in Explorer**. Windows Explorer is opened at the item's directory, with the item highlighted.

○ **Import Project** (from a project node only): Right-click on a project node and select **Import Project**. Studio imports the selected project, and you can see it in the workspace tree.

All other modified items that are not under the Studio Project structure can not be handled by Studio. You will need an external tool to handle them.

Reference Material: SCM Changes Pane

The **SCM Changes** pane displays all that was changed in the working copy, compared to the working copy's revision. For example, editing a flow will cause that flow to appear in the **SCM Changes** pane. This pane also shows a list of deleted projects (projects marked for deletion), if such projects exist.

> **Note:** Projects that are outside of Studio workspace are not be added/committed to SCM. They are shown as unversioned (in brown) in the **Projects** Pane, and are not shown in the **SCM Changes** pane.

Types of changes

You can see the different types of changes in the **Status** column.
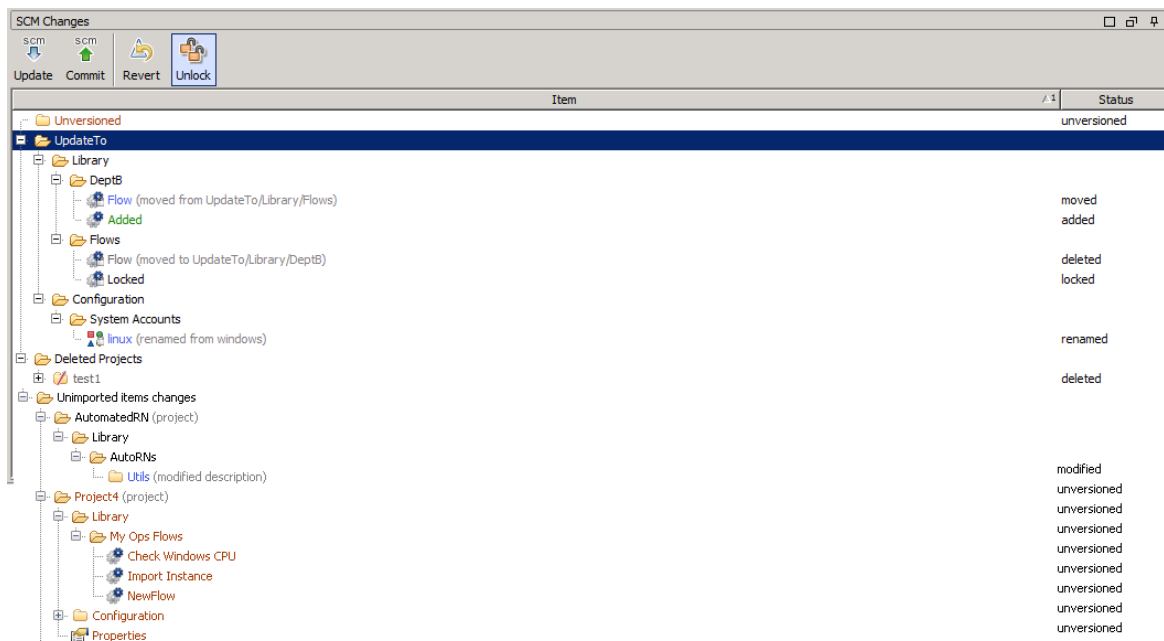
The types of changes are also reflected in the color coding. This may be:

Status	Color
added	green
renamed	blue
modified	blue
conflicted	red
moved (from/to)	blue
deleted	grey
unversioned	brown

In addition, you can see a text message in parentheses after the item name that describes the status of the item. For example, **moved from <old location>, moved to <new location>, renamed from <old name>**. You can also perform all standard SCM actions from these entries.

If there are items with duplicate UUIDs, they are shown in the SCM Changes pane in underlined text.

In the above example, you can see:

- A deleted project, named **test1**, and colored gray.

- An added flow, named **Added**, colored green.

- An unversioned folder, named **Unversioned**, colored brown.

- A list of unimported items that have been changed.

- Several changed items, shown in blue:

  - A modified flow, **adding flow 2**.

  - A renamed folder, **new folder a**.

  - A moved flow, **flow 1234567**.

  - A folder whose description has been changed, **Utils**.

Note: Renamed and moved items have a special label, shown in parentheses, with the location where the item was moved or renamed from.

State

The **State** area shows the state and the name of the current Git branch.

By default, the current Git branch is "master".

The state can be one of the following:

- **Ready:** the default state

- **Merging:** when you are in the middle of a merge operation and there are still unresolved conflicts

- **Merging - all conflicts resolved:** when you are in the middle of a merge operation and all conflicts have been resolved

- **Interactive rebase:** when you are in the middle of a rebase operation and there were conflicts

SCM Changes toolbar

GUI item	Description
scm	**Update all:** Updates the entire Studio workspace.
scm	**Commit all changes:** Commits all the changes in the **SCM Changes** pane. Only available when there are changes.
scm	**Push changes:** Pushes all the changes in the **SCM Changes** pane to the main Git repository.
	Stash: Takes the current state of your local working directory — that is, your modified tracked files and staged changes — and saves it on a stack of unfinished changes that you can reapply at any time.
	Unstash: Takes the stack of unfinished changes from the stash and reapplies them on the local working directory.
	Revert all changes: Reverts all changes that show in the **SCM Changes** pane.
	Branching: Presents the Git branching operations. Click the icon to see the additional options, **Create**, **Checkout**, **Merge** and **Rebase**.
	Create: Creates a new Git branch.
	Checkout: Connects to an existing branch. After checking out, your workspace (along with the projects in it) will be changed to reflect the state of the newly checked out branch.
	Merge: Git preserves the original hierarchy of the local commits, and creates a special "merge" commit that merges the two diverged "branches" (the local branch and the remote branch) back together.
	Continue rebasing: Rewrites the project history by creating new commits for each commit in the original branch.

▷❘	**Skip commit in rebasing:** Rewrites the project history without creating new commits.
◼	**Abort:** Aborts the merging (for a merge operation) or stops the project history rewriting (for a rebase operation).
🔄	**Refresh** Refreshes all the projects/items in the **SCM changes** pane.
State:	Shows the name of the current repository (e.g., master) and the rebasing stage. When connected to an empty repository, the text (**<empty>**) appears. For example: State: ⚙ Ready (<empty>) During rebasing, the text **Interactive rebase** appears. For example: State: ⚙ Interactive rebase (17b9ff7) When merging, the text **Merging** appears. For example: State: ⚙ Merging (master) When merging after all conflicts have been resolved, the text **Merging - all conflicts resolved** appears. For example: State: ⚙ Merging - all conflicts resolved (master) **Note:** To complete the merge process, you must perform a final commit to get the files into a state ready for committing and pushing to the main Git repository.

SCM Message Pane

The **SCM Message** pane displays the message results from the Git actions. Every action results in a message and is color-coded. In addition, a pop-up message appears informing the user of the result of the Git action. Clear the **SCM Message** pane by right clicking anywhere inside the pane and then clicking the **Clear All** button.

The type of messages that are shown depends on many factors, such as the update strategy, stashing, possible conflicts, or whether errors occurred when executing certain commands.

Examples of Git Messages

In the following example, the current workspace is updated. This process includes stashing, updating and adding files.


```
SCM Messages

05/06/15 13:39:54 - Update
Updating workspace started
Stash
Stashing changes in the workspace including the untracked files.
Removed project(s): test2
Stashed paths:
   modified: AutomatedRN/Content/Library/AutoRNs/Generate Release Notes(9x).xml
   modified: AutomatedRN/Content/Library/AutoRNs/Generate Release Notes.xml
   modified: AutomatedRN/contentpack.properties
   modified: AutomatedRN/pom.xml
   added: test2/.gitignore
   added: test2/Content/Library.properties
   added: test2/contentpack.properties
   added: test2/pom.xml
```

Error messages

In certain cases, errors appear. These are displayed in red. In this example,

05/07/15 10:55:16 - Push
http://admin@address.labs.company.com:8180/r/test01.git: cannot open git-receive-pack
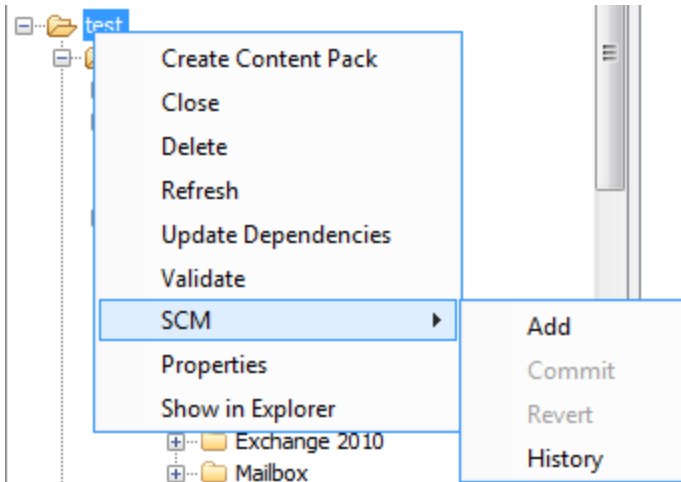There were 1 error(s).

Projects pane

Shows the project you're working in, and displays the editable flows, operations, and other OO objects that you can use in the project.
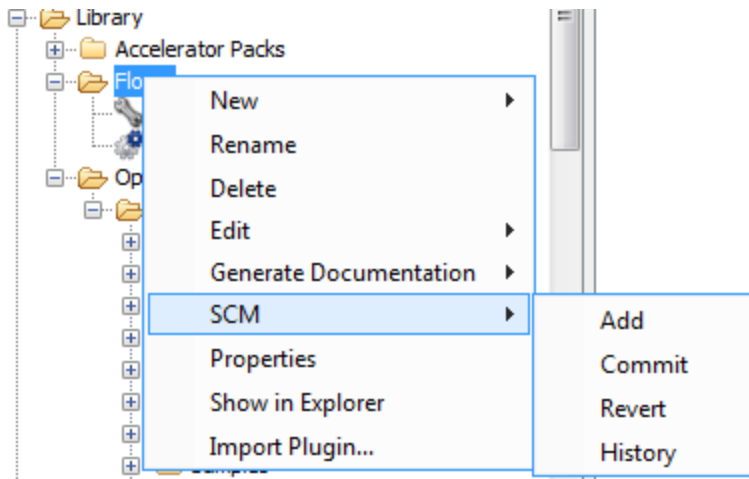
Context Menu from the Projects pane

The context menu shows all actions that are available for the selected object, according to its state.

Following are the options available when you right-click on a committed flow. In this case only **Add** and **History** are available:

If the object was to be changed by an author, for example by adding a step, then it would have local changes. In this case, **Commit** and **Revert** are available.
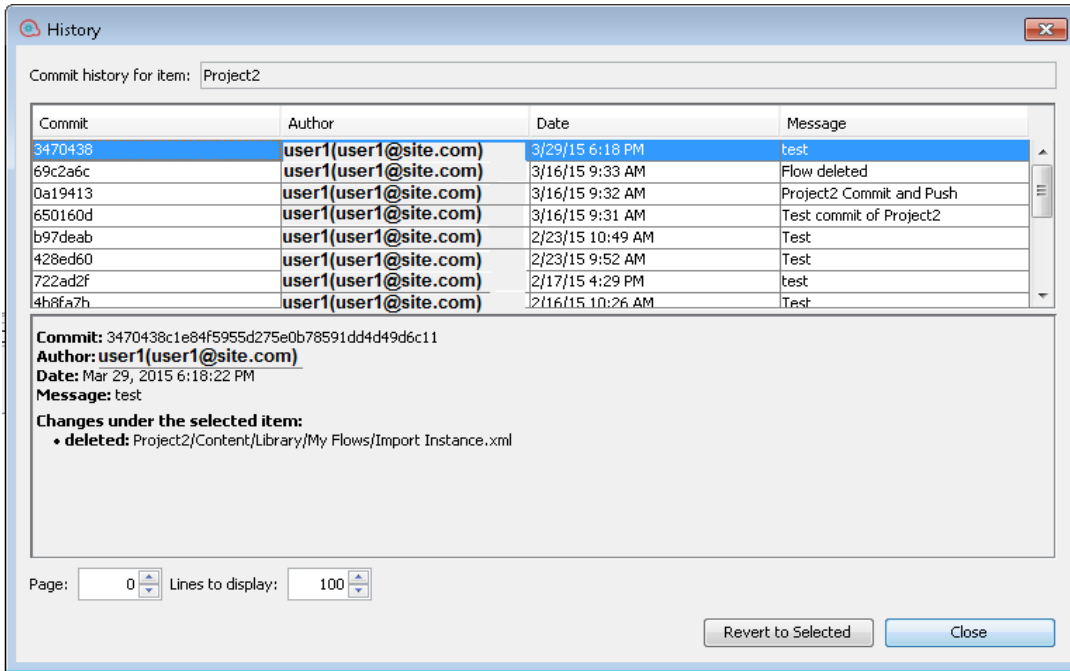

Option	Description
Add	Marks an item (flow, operation, configuration item or folder) to be added to source control. In Studio, **Add** includes all ancestor and descendant objects. So if a folder is added, any child flows and parent folders are also added. Sibling items are not added. Items that are created from within Studio, are added automatically and committed on the next commit.
Commit	Commits local changes to the server. This option is available for changed items and for folders that have changed children items. A commit works recursively, so when you commit a folder, all of its child items are also committed. After you commit,

Option	Description
	you can add a comment for the commit.
Revert	Reverts all local changes in the selected items. Changes in flows, operations and configuration Items are reverted. Items that were deleted are restored.
	Important: When you add an item it is automatically marked as added, but when you revert changes on it only the addition mark is removed, but the item still exists in Studio and in the file system. You can re-add the item using **Add** in the menu, or delete it.
History	The **Revision History** window, displays the SCM history. See "View a history of Git operations in the repository" on page 371.

History dialog box

The History dialog box is used to revert to a previous revision of a project/item.

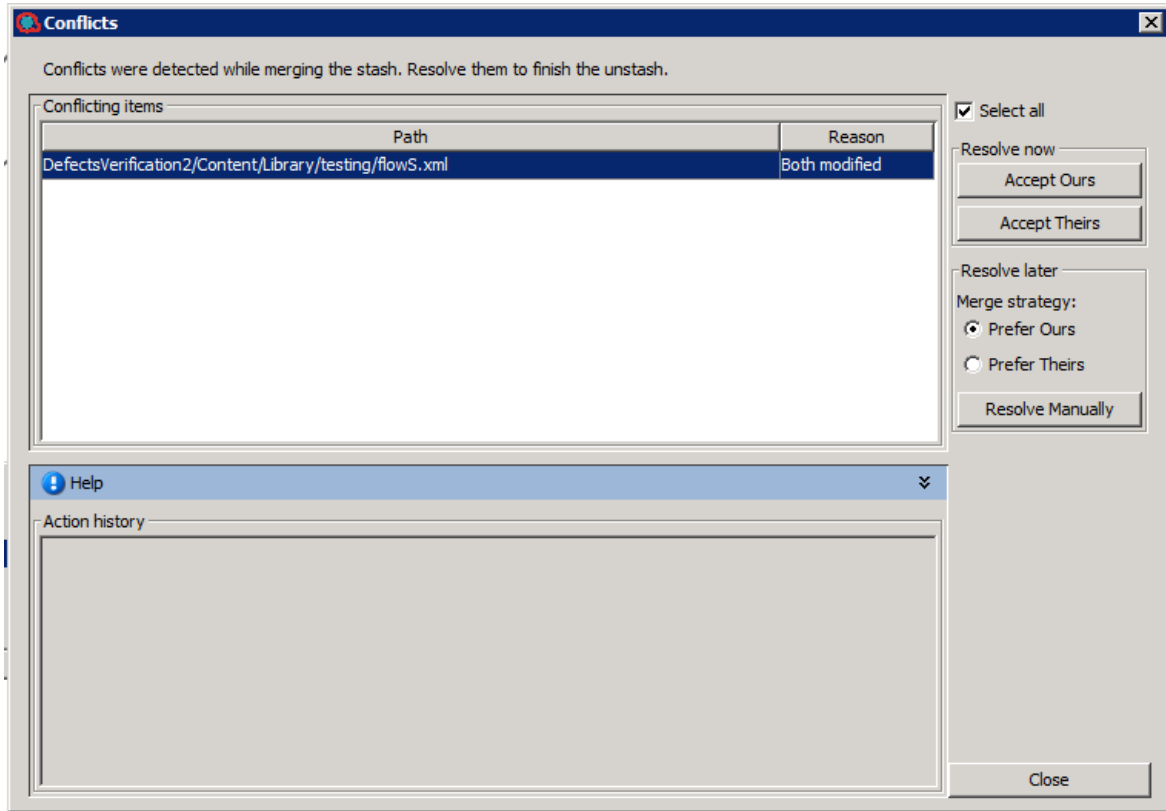| Option	Description
Commit history for item	Name of the project/item.

Option	Description
Commit	ID of the Commit operation.
Author	User name/id of the user who committed the changes.
Date	Date on which the Commit operation was done.
Message	Commit message written when the Commit operation was done.
Changes under the selected item:	Changes made and list of items changed for this commit.
Page	If the revision history is longer than one page, you can change the number of pages, and number of lines that are displayed on each page.
Lines to display	Enter the number of lines to be displayed on each page.
Revert to selected	Click **Revert to selected** to load the select commit of the project/item.

Conflicts dialog box

Merge Example
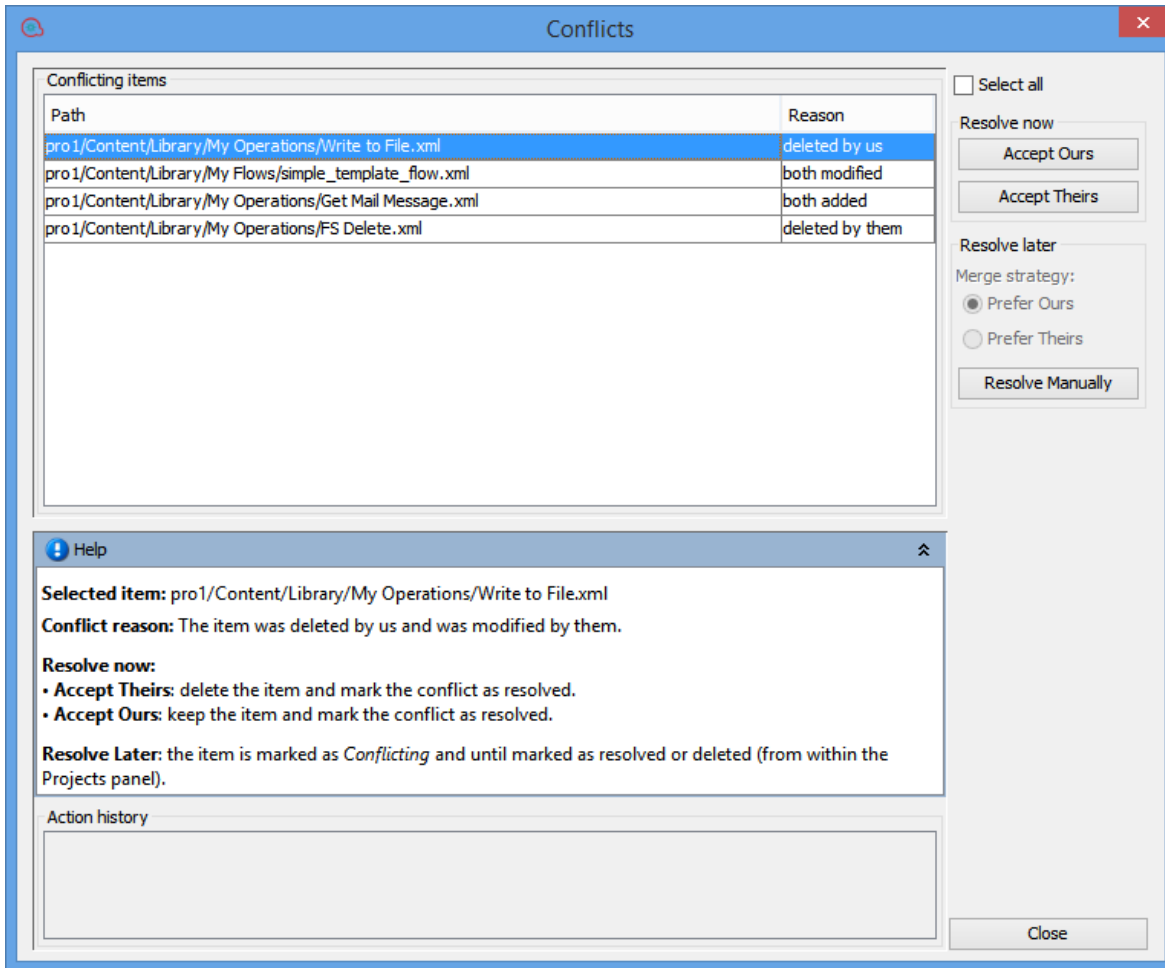
Stash Example

Rebase Example
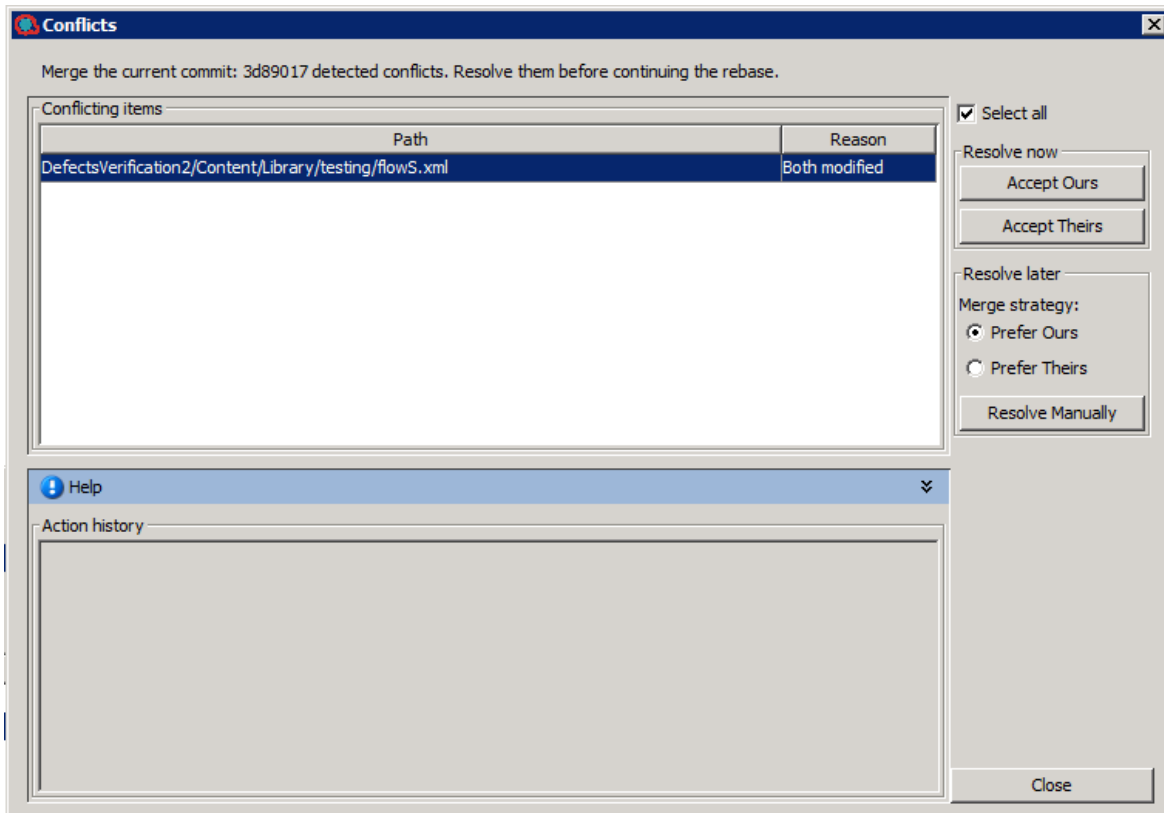
Option	Description
Conflicting Items	A list of the items in the project that caused the conflict and their paths. For each item, a reason for the conflict is shown.
Select All	Selects all the conflicting items from the list.
Resolve Now:	The conflicts are resolved immediately.
Accept Ours	Applies the changes made by "us" immediately.
Accept Theirs	Applies the changes made by "them" immediately.
Resolve Later:	The conflicts are resolved at a later stage.
Resolve Manually:	The original file is marked as **Conflicting** until marked as resolved or deleted (from within the Projects pane).
Prefer Ours	Changes made by us will be applied for the conflicts in the original file.

Option	Description
Prefer Theirs	Changes made by them will be applied for the conflicts in the original file.
Help	Shows the **Help** pane that provides additional details on the selected item.
Action History	Shows a log of all the executed operations. For example, the items that were resolved, how they were resolved, the items that will be resolved manually. You can scroll through the log using the scroll bar.

Resolve Later

If you choose to resolve a conflict at a later stage, Studio prepares the file for manual conflict resolution. All non-conflicting changes are automatically merged into the file, and conflicting changes are replaced based on your selected preference (ours or theirs).

The original file is marked as **Conflicting** until marked as resolved or deleted (from within the Projects pane).

In addition, Studio automatically creates two new files named `<filename> - ours` and `<filename> - theirs` representing the original versions from us and from them, respectively.

After the conflict is resolved, Studio automatically deletes the "ours" and "theirs" files.

Resolve Manually

To manually resolve the conflict:

1. From the right pane, select **Prefer Ours** to apply the changes made by "us".

Working with Content Packs

A content pack is a jar file containing operations, flows, actions, configuration items, and resource bundles. A content pack is the most granular entity that can be delivered for deployment.

You can download a content pack for your specific domain from ITOM Marketplace, and then import it into your project.

Importing Content Packs to a Project

This section describes how to import content packs to Studio.

A content pack is a collection of operations, flows, actions (Java-based or .Net based), configuration items (such as selection lists, domain terms, and so on) and resource bundles.

> **Note:** When you install Studio, you can select to import existing content packs in the Installation Wizard . These will load the first time you open Studio.

You need to import the Base content pack first, and then import the specific content packs that you need to create your custom flows, for your specific domain. You can download these content packs from ITOM Marketplace.

Once you have imported a content pack, the files are available in the **Dependencies** pane as read only.

> **Note:** You can import a content pack from any network drive.
>
> You cannot import a content pack directly from a remote hosting site. To access remote content packs, first download a copy of the files to your local system, or map to a network drive so that the File Chooser dialog box can navigate to the location.

You can reuse a flow from a different project, by importing a content pack that was created from that flow. For information about packaging a flow into a content pack for reuse, see "Exporting a Content Pack" on page 619.

You can download content packs from multiple sources:

- **My content** – Projects and content packs that are already in Studio.
- **My organization** – Content packs that were developed by various authors in the organization, and which reside in the artifact repository.
- **content** - Content that releases regularly, which resides on ITOM Marketplace.
- **Community content** - Content that was contributed to the community by other organizations, which also resides on ITOM Marketplace.

Content Pack Import Issues

Central provides a signature validation mechanism for content packs, giving the user indication about the signature status (unsigned, broken, expired, trusted, signed but not trusted).

The content packs may sometimes originate from a party that is external to the organization. In these cases it is recommended to verify the digital signature of the content pack manually using well known tools before importing it into Studio.

For details, see "Content Pack Validation - Digital Signatures" in the *OO Central Guide*.

What do you want to do?

Download the content packs

1. Open the ITOM Marketplace site and then navigate to the list of content packs.

2. Download the Base content pack to a location on your network drive.

3. Download any other content packs that you need for your specific domain to a location on your network drive.

Import the Base content pack when opening Studio for the first time

1. In the **Dependencies** pane, click the **Import** button 🔄 .

    Note: You can import a content pack from either the **Tree View** or **Merged View** tab in the **Dependencies** pane.

2. Browse to locate the content pack and click **Open**.

3. Click **OK**.

    Note: It may take a few minutes to import the Base content pack.

Import a content pack

1. In the **Dependencies** pane, click the **Import** button 🔄 or from the File menu, select **Import Content Pack**.

2. Browse to locate the content pack and click **Open**.

    Note: If required, you can select multiple content packs to import at once.

3. After selecting the content pack(s), Studio verifies the certificate signature of the content pack(s) and presents it to you.

4. Click **OK**.

    The imported content pack is displayed in the Dependencies pane and the following message appears:

    ```
    Content pack(s) imported successfully
    ```

Note: If a content pack has a broken signature (shown in red), you cannot import the content pack.

Extract a content pack and open it as a project

If you have upgraded content from previous versions of OO, the upgrade tool converts your content into

content packs. Follow these steps to open a content packs as an editable project.

1. In your file system (for example, Windows Explorer), unzip the content pack. The content pack is extracted into a project folder.

2. In Studio, select **File** > **Import Project**.

3. In the Select Project Directory dialog box, browse to locate the project that was created from the unzipped content pack.

4. Click **OK**. You can now edit this project in Studio.

> **Note:** If you opened the content pack in Studio before creating the project from it, you need to close the content pack in Studio. The project and content pack will have the same UUID, so they cannot be open in Studio at the same time.

Managing Content Packs and Dependencies in a Project

Once a content pack is imported, you can use its operations in your flows, but note that these operations are read-only. However, you can use the flows as steps as they are read-only also in the content pack.

If you want to modify operations from a content pack, you need to copy them into the project.

> **Note:** This is only recommended if you want to add responses or results. If only the inputs are used, you can modify them inside the steps.

After an operation is copied to a project, it is detached from the content pack, and becomes editable. You can drag and drop this new operation to a flow and modify its properties.

Dependencies

A dependency is a project or content pack that includes a reference to an item contained in the current project. The item may be one or more of the following:

- Operation soft copies that reference operations inside other content packs/projects
- An operation group alias that comes from another content pack/project
- An operation/flow/step scriptlet that comes from another content pack/project
- Operation/flow outputs filters that come from other content packs/projects
- Operation response rule evaluators that come from other content packs/projects
- Operation response rule filters that come from other content packs/projects

- Operations, flows or step inputs that have an Otherwise assignment from system accounts, selection lists or domain terms inside other content packs/projects.

- Operations, flows or step inputs evaluators and domain terms coming from input validation format and input record under respectively that are inside other content pack/projects.

- Flow steps referencing operations/flows from other content packs/projects

- Step results filters from other content packs/projects

- Transition role aliases from other content packs/projects

Dependencies are stored in the project's pom.xml file.

Dependency Conflicts

In general, the version of a declared dependency should match the version of dependencies that are automatically detected by Studio.

However, there are three types of conflicts that may occur between dependencies:

- **New conflict** – appears when the dependency is not declared in the project and Studio finds that this dependency is used. When resolving such a conflict, the dependency will be added with the version that exists in the current workspace.

- **Updated conflict** – appears when the detected version is different from the declared version. The conflict will only occur if the detected version is outside the range of the declared version. For example, if the declared version is a range 1.2 - 1.6 and the detected version is 1.3 there will not be a conflict as 1.3 is within the 1.2 - 1.6 range. If the declared version is a fixed version 1.5, and the detected version is a different fixed version 1.6, there will be a conflict. When **Apply Change** is checked for such a row in the Resolve Dependency Conflicts dialog box, the dependency will be updated in the list of dependencies of that project with the version taken from the Detected Version column.

- **Might not be used** – appears when the dependency is declared in the project but Studio does not detect that it is used. For example, you may have a project ProjectB that has been added as a dependency in your project ProjectA, but is not used in any of the flows, operations and configuration items from ProjectA. Or, you may have a dynamic reference to a system property inside ProjectB referenced from a flow inside ProjectA. In both these cases, ProjectB dependency will be tagged as a *Might not be used* conflict.

  In this case, the dependency will be removed from the dependency list in the current project if **Apply Change** is checked.

Updating Dependencies in a Project

If your project includes dependencies, the first step you must perform is to manually update all the dependencies as described in "Update dependencies for a project" below. You can then validate dependencies, add, edit or delete dependencies in the Dependency Editor.

What do you want to do?

Copy content pack objects to the project

1. Right-click the object to be copied in the **Dependencies** pane and select **Edit** > **Copy**. To select multiple objects, use the **Shift** and **Ctrl** keys.

2. Right-click the location in the **Projects** pane where you want to paste the object and select **Edit** > **Paste**.

   > **Tip:** You can also drag and drop an object from the **Dependencies** pane to the **Projects** pane.

Delete a content pack

Deleting a content pack is different from closing it, in that a deleted content pack is permanently removed from the workspace.

> **Note:** When you delete a content pack, it is deleted from the workspace, but is not deleted from the file system. If required, you can import it again.

1. Select the **Content Packs** tab in the **Dependencies** pane.

2. Select the content pack and click the **Delete** button .

3. Click **Yes** in the confirmation dialog box.

Close a content pack

If you close a content pack, it is visible in the **Dependencies** pane, but is grayed out and not available.

When to close a content pack:

- If you have two versions of a content pack in the workspace, you will need to close one before you can work with the other. The two contents packs will have the same UUID, so they should not be open in Studio at the same time.

- If you created a project from a content pack, you should close the original content pack, before working with the project. The project and content pack will have the same UUID, so they should not be open in Studio at the same time.

1. Select the **Content Packs** tab in the **Dependencies** pane.

2. Select the content pack that you want to close.

3. Right-click the content pack and select **Close**.

> **Note:** Alternatively, you can click the **Close** button in the **Dependencies** pane.

Open a closed content pack

After a content pack has been closed, you can open it, to work with it again.

1. Select the closed (gray) content pack that you want to open.

2. Right-click the content pack and select **Open**.

> **Note:** Alternatively, you can click the **Open** button in the **Dependencies** pane.

Display the properties of an imported content pack

Right-click a content pack in the **Dependencies** pane and select **Properties**. The Properties window for the content pack opens, in read-only mode.

Display the properties of an object in a content pack

Double-click a flow, operation or other object in the **Dependencies** pane. The Properties window for the object opens, in read-only mode.

Update dependencies for a project

There are three methods that you can use to update dependencies:

1. Right-click on the project name, and from the menu, select **Update Dependencies**.

   or:

2. In the Projects pane, right-click on **Properties**.

3. From the menu, select **Update Dependencies**.

   or:

   From the Dependencies Editor toolbar, click .

   If Studio finds a new dependency, it adds a new line corresponding to that dependency. If the dependency already exists, Studio updates it.

> **Note:** In the case of an update, the current version is kept as long as it does not conflict with

the dependency detected by Studio.

4. If there is a conflict between the current version and the version detected by Studio, the Resolve
   Dependency Conflicts dialog box opens:
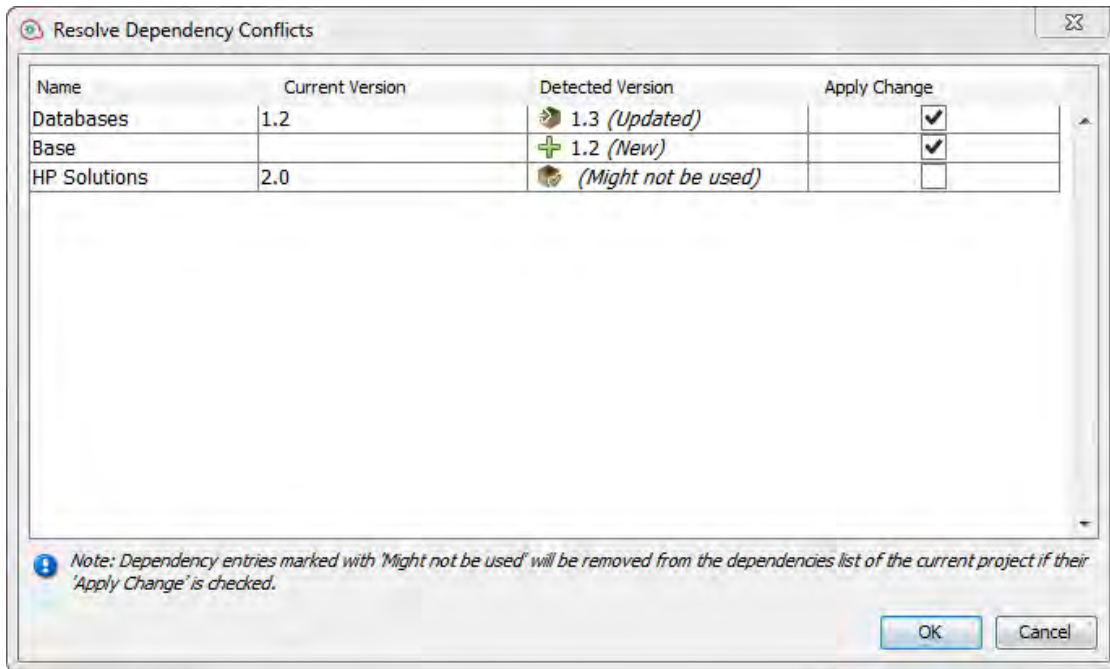
5. Check **Apply Change** for each row, in order to resolve the conflicts.

   Note: If a dependency is marked as *{Might not be used}*, this means that the dependency is
   declared in the project but Studio cannot identify whether or not it is used. For example, you
   may have an input in a flow that has a dynamic reference to a system property in another
   project. In this case, Studio cannot identify whether or not the system property is used in the
   flow.

   For these dependencies, **Apply Change** is unchecked by default. After checking **Apply
   Change** and clicking OK, the dependency will be removed from the dependency list in the
   current project.

6. Click OK.

   After clicking OK, Studio applies (updates, adds, removes) the selected dependencies (as marked
   in the **Apply Change** column).

   If there are no conflicts, or after all the conflicts have been resolved, the following message
   appears:
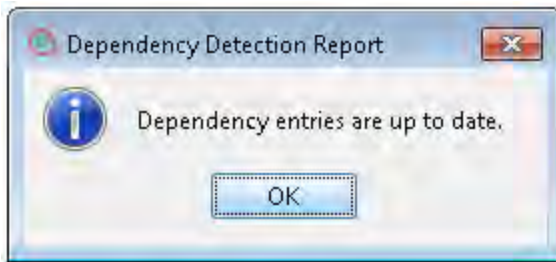
Validate dependencies for a project

This procedure shows how to validate dependencies manually. However, there are cases in which the dependency validation is triggered automatically:

- When the dependencies of a project are changed and saved

- When a project is updated from SCM either as individual project or as part of an **Update All** operation

- After a project is imported

- When a project is opened

> **Note:** The validation process checks the range of dependent items of a project and also checks that the version of dependent projects match the project version.

To manually validate dependencies:

1. In the **Projects** pane, right-click on **Properties**.

2. From the menu, select **Validate**.

If there are any dependency warnings or errors in the project, you can see them in the **Problems** pane. For example:

After the validation, the problems pane is updated with the new status of the dependencies.

Add a new dependency for a project

In addition to the dependencies that are automatically added to the project, you can add new dependencies, even if they do not exist in the workspace.

1. Click in the Dependencies Editor toolbar.
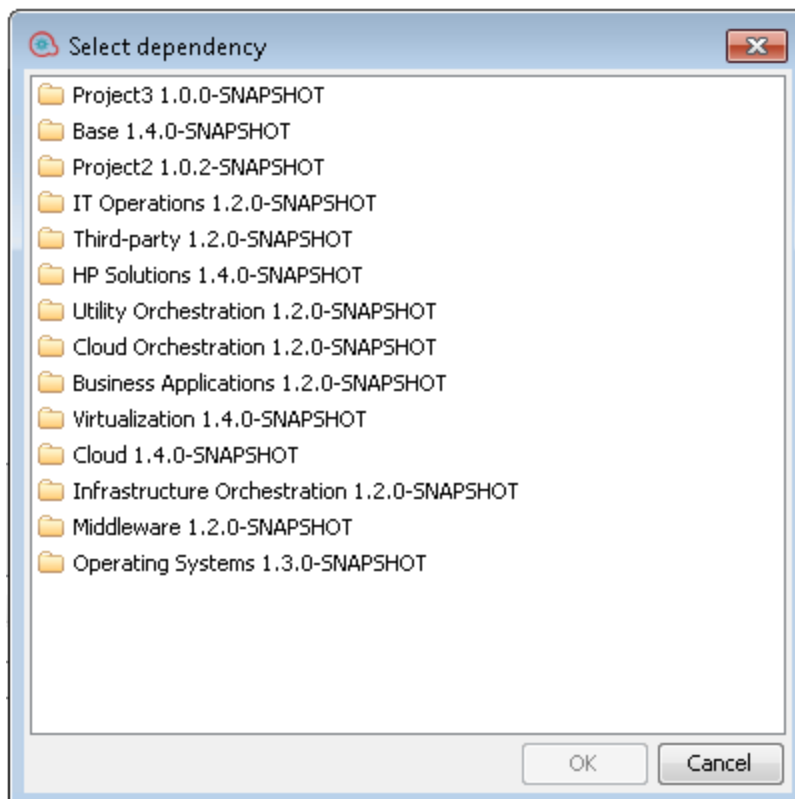
   The Add Dependencies dialog box opens.

2. In the **Name** field, type the name of the dependency you want to add or click and select the dependency from the list of current projects and content packs that have been imported into Studio:
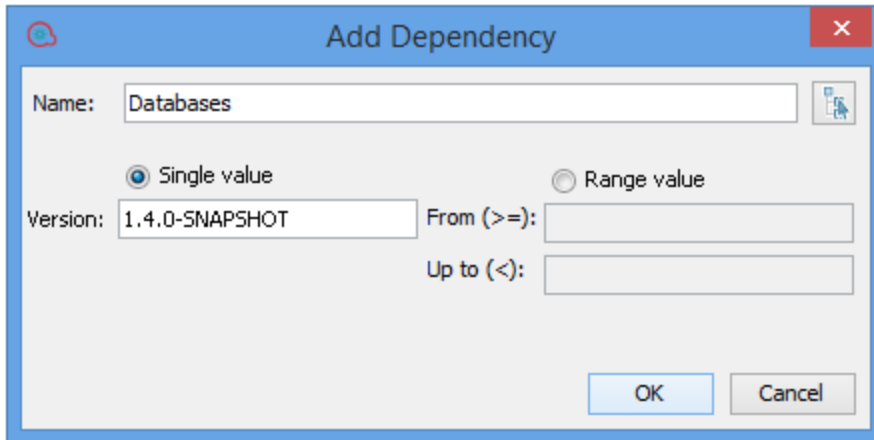
For example, if your project uses information from the Middleware content pack, click ![icon] and select type **Middleware 1.2.0-SNAPSHOT**.

3. In the **Version** field, type the version of the dependency you want to add. You can use a numeric value (up to 99999999) or a snapshot version, e.g., 1.4.0-SNAPSHOT .

> **Note:** If you selected from the list of dependencies , the **Version** field is automatically filled with the version number. For example:

In addition, if there is no official current version, the snapshot version is used.

4. To define a range of versions to be supported in the dependency, select **Range value**, and type the range in the **From** and **To** fields. For example, if you are currently using Base version 1.4.0, but you want your project to be backward compatible with version 1.2.0, type 1.2.0 in the **From** field, and 1.4.0 in the **To** field.

> **Note:** The **From** field includes the value entered, as well as values higher than it, while the **To** field does not include the value entered. For example, "from 1.2.0 to 1.4.0" means higher than and including 1.2.0, but lower than 1.4.0.

You can also specify all values lower than… by entering a value only in the **To** field, or all values greater than…. by entering a value only in the **From** field.

A range is marked by *(range)* in the Dependencies Editor.

> **Note:** In the inline editor, you can automatically define a range by typing the range in the **From** field. For example 1.2.0 - 1.4.0.

Edit a dependency

1. In the Dependencies Editor toolbar, select the dependency you want to edit.

2. Click .

3. Edit the dependency definitions as required.

Delete a dependency

You can delete dependencies from the project.

> **Note:** The dependency is only removed from the project, not from the file system.

1. In the Dependencies Editor toolbar, select the dependency you want to delete.

2. Click .

Show dependency details

From the Problems Pane, you can take a closer look at the dependencies that caused warning messages, and then update the dependencies as required.

1. In the Problems Pane, select the error/warning that you are interested in.

2. From the toolbar, click .

   The Dependency Details dialog box opens showing a list of items in the project that reference the selected dependency. For example:
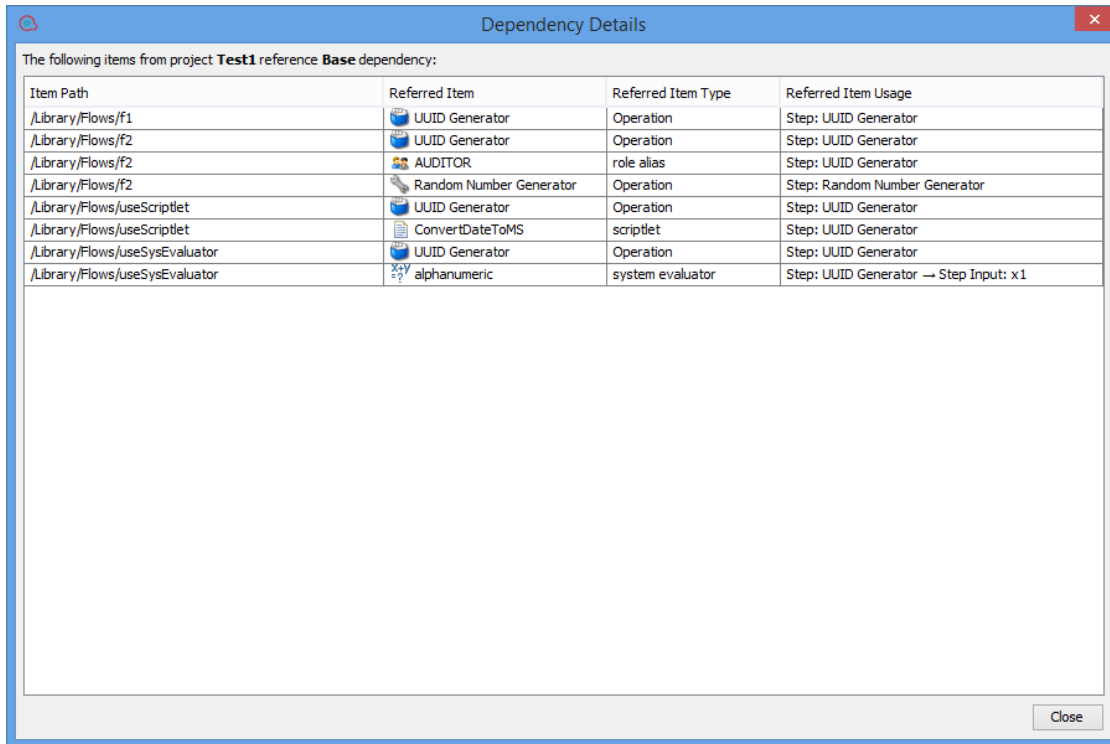
4. Double-click on an item to load it in Studio.

Reference Material

Dependencies pane

The **Dependencies** pane can display multiple content packs, each with its own hierarchical tree structure.

The **Dependencies** pane contains two tabs:

- **Content Packs** - Displays multiple trees, for multiple content packs. From this view, it is possible to close, delete, or import a content pack.
- **Library** - Displays a single tree, with all of the content merged together under a general **Library** folder and a general **Configurations** folder. From this view, it is possible to import a content pack.
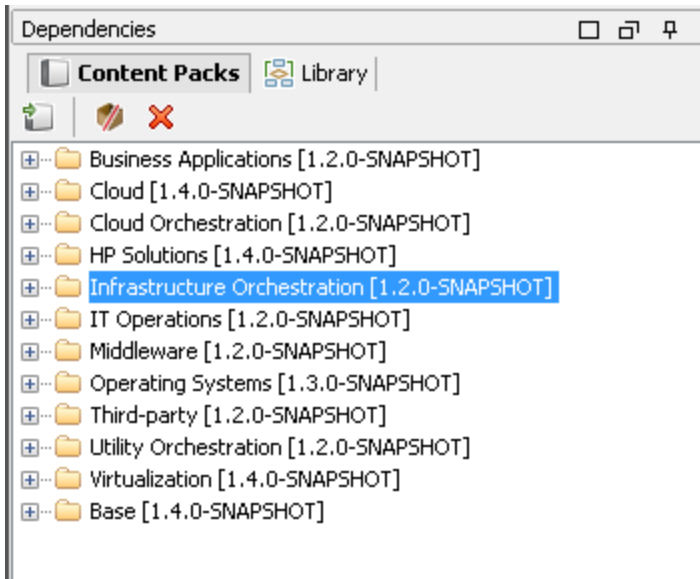
GUI item	Description
Import Content Pack	Opens the Import Content Pack dialog box, letting you select a content pack to be imported.
Delete	Deletes the selected content pack.
Open	Opens the currently selected closed content pack.
Close	Closes the currently selected content pack, so that it is grayed out.

Content Pack Properties

When you right-click a project and select **Properties** , the Properties sheet opens.

In the Properties sheet for a project, you can update all the properties. However, for a content pack, the properties are read-only.

You can show the same information from the Dependencies pane. The information is taken from the content pack jar file that was imported. If the content pack was created from an OO project, some of the information is taken from what was entered into the Create Content Pack dialog box.
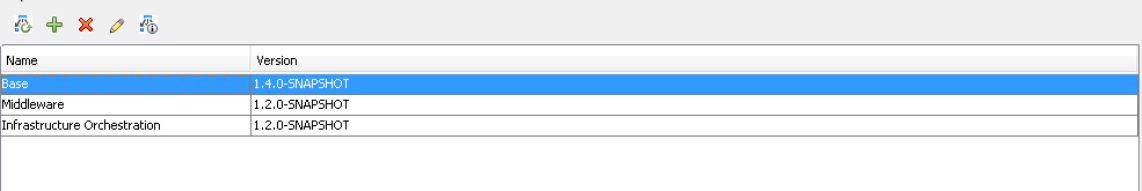
GUI item	Description
Name	The name of the content pack is taken from the project name. This field is read-only.
Uuid	A unique identifier for the content pack.
Version	The content pack version.
Publisher	Enter the publisher of the content pack. This information will appear in the content pack's **Properties** sheet.
Description	Enter a description of the content pack. This information will appear in the content pack's **Properties** sheet.

Dependencies Editor

At the bottom of the Content Pack/Project Properties page, the Dependencies Editor is shown.

In the Dependencies Editor, you can see all the installed content packs, their versions and their dependencies for the current project. You can edit these values as required.

GUI item	Description
Name	The content pack or project name
Version	The content pack version. You can edit this version with one of the following: • **A fixed version number:** For example, 1.2.0. This means that you can only use the exact version of the content pack. • **Interval:** For example, 1.2-. This means that you can use any version greater than or equal to 1.2. For example, -1.2. This means any version strictly less than 1.2. • **Interval with range:** For example, 1.2-3.0.1. This means that you can use any version from 1.2 up to 3.0.1 (up to but excluding 3.0.1).
Add ➕	Open the Add Dependency dialog to add a new dependency for the current project.

Delete ✖	Delete the selected dependency from the current project.
Edit ✏	Open the Edit Dependency dialog to change the definition of the selected dependency.
Update Dependencies ☑	Show and resolve any dependency conflicts in the current project.

Problems Pane toolbar

GUI item	Description
	Where relevant, open the flow related to the problem.
	Update dependencies in the current project.
	Add a dependency for the current project.
	Delete a dependency from the current project.
	Show details of dependencies in the current project.

Dependencies in the Problems Pane

If any issues occur when you validate dependencies, you can see them in the **Problems** pane.

There are three types of dependency issues that may occur. For each issue, a specific warning message is shown in the Problems pane:
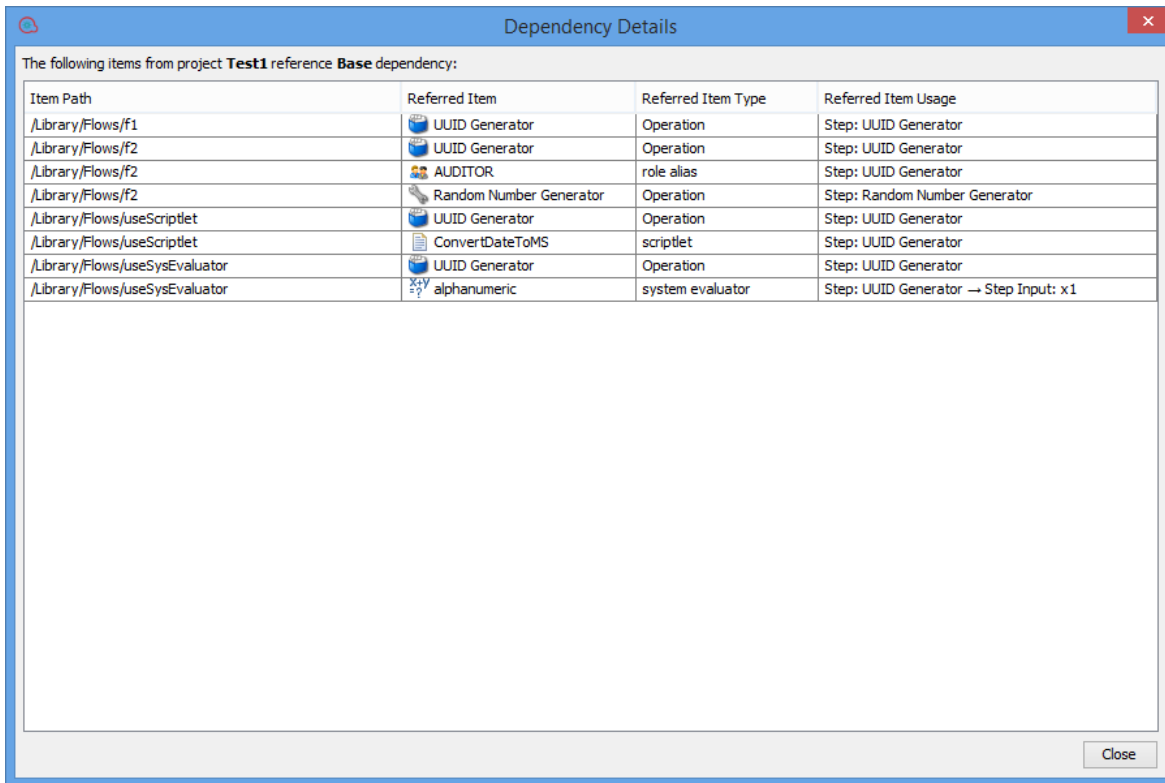
- **Missing Dependency:** The project contains links to items from another content pack/project but the content pack/project containing the items are not declared in the project's dependencies list.

- **Probably unused dependency:** The declared dependencies of a project contains a dependency that cannot be found by the Studio dependency detection rules. However, these rules do not detect dynamic references for system properties, so the declared dependency could exist for a system property dynamic reference, or may not exist at all.

- **Not in workspace:** The project links to a project or content pack that is not currently imported in Studio workspace.

The following screen shows examples of these three types :

GUI item	Description
Type	Type of issue: Warning or Error. You cannot debug a flow with errors, but you can edit it. **Note:** • It is recommended to fix all issues, warnings and errors. • In multi-author environments, it is important that no warnings are produced when running Validate Dependencies as modifications in the workspace by one author can be also used by another author. Therefore, it is important to have the necessary projects/content packs imported with the correct versions.
Source Type	The section in which the problem occurs. This may be a flow error, an operation error, a step error, a configuration item error or a dependencies warning.
Name	The name of the section.
Description	A description of the dependency issue.
Location	The location of the item.

Dependency Details dialog box

GUI item	Description
Item Path	The path of the item in the project that references the dependency.
Referred Item	The name of the item in the project that references the dependency.
Referred Item Type	The type of the item in the project that references the dependency.
Referred Item Usage	The name of the section in the flow in which the item is used.

Managing Configuration Items

The **Configuration** folder in a project contains a number of items that you can set up: domain terms, group aliases, system accounts, system properties, and so on.

If an authoring team are working in different projects and need the same configuration items, it is recommended to create a separate, shared project containing the configuration items.
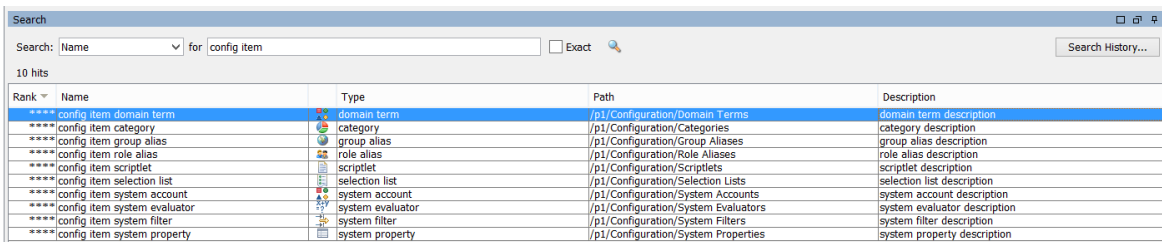
> **Important!** If you need to delete, create, or rename a configuration item inside your project, make sure to do this from within Studio, and not by deleting, creating, or renaming the item in the file system.

Working with Configuration Items

Searching for Configuration Items

You can search for configuration items by UUID, name, and description fields. The `<all fields>` type search and `<with Lucene query>` type search using name, description and/or UUID as fields inside the search query, will also search through the configuration items that belong to the currently opened projects and content packs.

Using the **Search** pane, you can perform a full-text search throughout the list of open projects and open content packs.

See "Search for an operation or configuration item" in "Finding a Flow or Operation" on page 642 for details on how to search for a configuration item.

Working with Configuration Item Folders

You can create folders in all the configuration items under the existing configuration item's folder structure. You can perform the following operations on the configuration item folders:

- Create folders in each configuration item section
- Copy and paste configuration Items folders from one project to another
- Move configuration items folders from one project to another
- Move items between folders and the top level section
- Delete configuration items folders.

- Rename configuration items folders

- Set descriptions for each configuration items

What do you want to do?

Search for a configuration item using the Navigate to item... option

You can use the **Navigate to item..** option to easily find and open a configuration item in your workspace.

1. Select **Tools > Navigate to item...** .

    The **Search items by name:** window opens.

2. Type in part of the configuration item name.

    The window is immediately populated with search results. For example, if you search for a item whose name contains the string "use", you may see the following results:
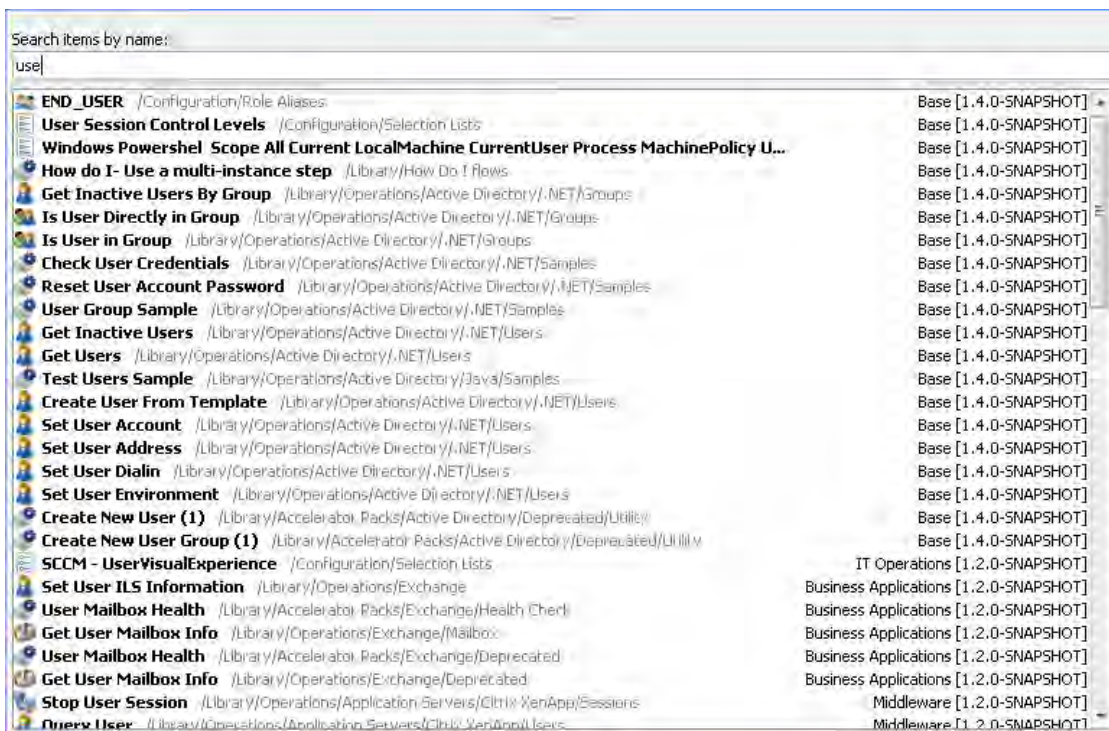
3. Use the Up and Down arrow keys to move in the list, then press Enter to open the selected item. Alternatively, you can open an item with a double-click.

4. You can create steps in the current Flow Editor by dragging and dropping operations or flows from the list.

Create a configuration item folder

1. In the **Projects** pane, expand the **Configuration** folder.

2. Right-click on a configuration item folder, and then click **New > Folder**.

3. Enter a name for the new folder and click **OK**.

   You can now add, copy or move configuration items into this folder.

Rename a configuration item/folder

1. In the **Projects** pane, expand the **Configuration** folder.

2. Right-click on a configuration item folder, and then select **Rename**.

3. Type in a new name and press Enter.

Copy and paste configuration items from one project to another

1. In the **Projects** pane, expand the **Configuration** folder.

2. Right-click on a configuration item folder, and select **Edit > Copy**.

3. Select the location for the configuration items in the desired folder and select **Edit > Paste**.

Move configuration item folders and items

1. In the **Projects** pane, expand the **Configuration** folder.

2. Right-click on a configuration item folder and select **Edit > Cut**.

3. Select the location for the folder in the second project and select **Edit > Paste**.

   The entire folder and its configuration items are moved to the new location.

Delete a configuration item folder

1. In the **Projects** pane, expand the **Configuration** folder.

2. Right-click on a configuration item folder, and then select **Delete**.

Note: Deleting the folder and its configuration items will break any dependencies that they had with other items.

Set a configuration item folder description
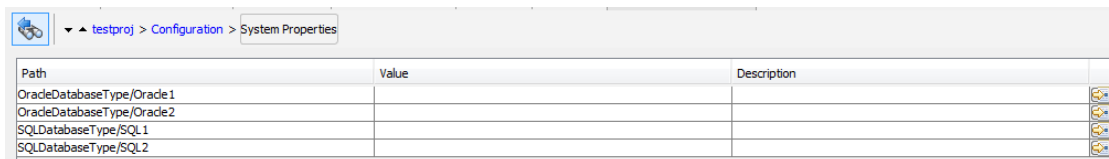
1. Right-click on a configuration item folder and select **Properties**.

2. In the **Description** box, type a description for the configuration item folder.

3. Click **Save**.

View all the configuration items and their folder structure

1. In the **Projects** pane, right-click on the configuration item folder.

2. Select **Properties**.

A list of all the configuration items along with their folder structure opens. For example, for System Properties the following list opens:

> **Note:** From this page, you can see the folder structure in read-only mode and cannot edit the names inline. However, you can edit the individual configuration items by double-clicking on a row, or by selecting the icon on the right of each row.

Configuring Categories

Categories are different classifications that can be assigned to a flow. A number of categories are installed with Studio, but you can also create your own categories.

Users might use categories to create reports that indicate the health of key infrastructure components. For example, if you assign the category **Server** to all the flows that check server health, then a report that finds only flows that were assigned the **Server** category could highlight the health of the servers on your network.

You can also use categories for filtering a search. For example, you can run a search that only looks for flows that have the **Security** category.

Categories are stored in the **Configuration\Categories** folder.

Use
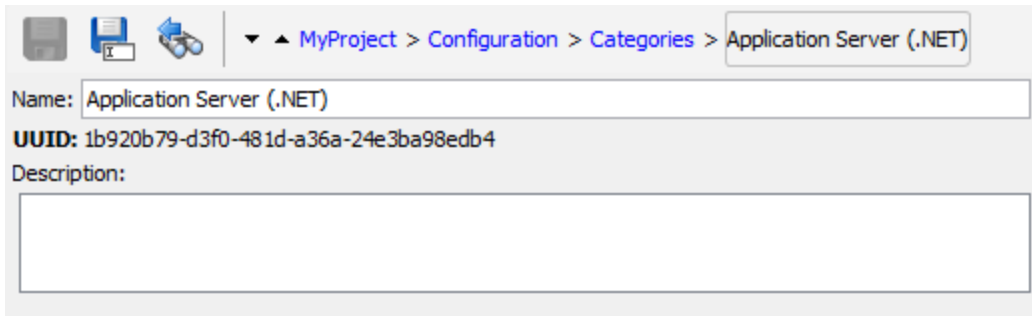
What do you want to do?

Create a category

1. In the **Projects** pane, expand the **Configuration** folder.

2. From the **Categories** folder, right-click and select **New > Category**.

3. In the dialog box that appears, type a name for the new category , and then click **OK**.

4. In the **Description** box, type a description of the new category .

5. Click **Save**.

Change a category

1. In the **Projects** pane, expand the **Configuration** and **Categories** folders, and double-click the category to open its editor.

2. Double-click the category you want to change and type the new value.

Rename a category

1. In the **Projects** pane, expand the **Configuration** and **Categories** folders, and double-click the category to open its editor.

2. In the **Name** box, type the new name for the category .

3. Click **Save**.

Copy a category

1. In the **Projects** pane, expand the **Configuration** and **Category** folders.

2. Right-click on the category that you want to copy.

3. Select **Edit > Copy**.

4. Move to the **Category** folder of the project you want to copy to.

5. Right-click and select **Edit > Paste**.

Delete a category

Before you delete a category , it is recommended to use the **What Uses This** function to check that other items do not depend on it. For more information, see "Finding Out How Flows and Operations are Used" on page 655.

1. In the **Projects** pane, expand the **Configuration** and **Categories** folders.

2. Right-click the category and select **Delete**.

3. Click **Yes** in the confirmation windows.

Reference Material

Categories editor

GUI item	Description
Name	The name of the category.
Description	(Optional) Description of the category.

Configuring Domain Terms

Domain terms are attributes that you can assign to flows and inputs. For example, you might create domain terms for the various kinds of servers in your system, and then you could get a step to run against a certain type of server only.

Domain terms can be used for specialized selection lists. For example, you can create a domain term for different kinds of actions. The values in this domain term could be **Restart**, **Reboot**, **Open**, and so on.

In another example, to specify that a flow run against certain classes of servers and not others, you can add domain terms for the various kinds of servers in your system, and provide a user prompt at the start of the flow, in which the user needs to select the classes of servers that you want to run a given flow against.

Domain terms can have values by default, obtain their values from the flow's inputs, or have the values that you specify for them.

Domain terms are stored in the **Configuration\Domain Terms** folder.

What do you want to do?

Create a domain term

1. In the **Projects** pane, expand the **Configuration** folder.

2. From the **Domain Terms** folder, select **New > Domain Term**.

3. In the dialog box that appears, type a name for the new domain term, and then click **OK**.

4. In the **Description** box, type a description of the new domain term.

5. Click **Add** to add a new domain term value.

6. In the **Name** column, enter the name for the domain term value.

7. (Optional) In the **Description** column, enter a description for the domain term value.

8. Click **Save**.

Remove a domain term value

1. In the **Projects** pane, expand the **Configuration** and **Domain Terms** folders, and double-click the domain term to open its editor.

2. Highlight the value and click **Remove**.

Change a domain term value

1. In the **Projects** pane, expand the **Configuration** and **Domain Terms** folders, and double-click the domain term to open its editor.

2. Double-click the value you want to change and type the new value.

Copy a domain term

1. In the **Projects** pane, expand the **Configuration** and **Domain Terms** folders.

2. Right-click on the domain term that you want to copy.

3. Select **Edit > Copy**.

4. Move to the **Domain Terms** folder of the project you want to copy to.

5. Right-click and select **Edit > Paste**.

Rename a domain term

1. In the **Projects** pane, expand the **Configuration** and **Domain Terms** folders, and double-click the domain term to open its editor.

2. In the **Name** box, type the new name for the domain term.

3. Click **Save**.

Delete a domain term

Before you delete a domain term, it is recommended to use the **What Uses This** function to check that other items do not depend on it. For more information, see "Finding Out How Flows and Operations are Used" on page 655.

1. In the **Projects** pane, expand the **Configuration** and **Domain Terms** folders.

2. Right-click the domain term and select **Delete**.

3. Click **Yes** in the confirmation windows.

Reference Material

Domain Term editor

GUI item	Description
Name	The name of the domain term.
Description	(Optional) Description of the domain term.
Add	Click **Add** to add a new item to the domain term list.
Remove	Click **Remove** to remove the selected item from the domain term list.
Name column	Enter the name of the item in the domain term list.
Description column	(Optional) Enter a description of the item in the domain term list.

Configuring Group Aliases

RAS Groups

A RAS group is a logical collection of RASes. Deployments can benefit from having more than a single RAS in a specific environment. For example, you are managing a remote data center in which you need two RASes to be able to withstand the action execution load, or simply for high availability of the RASes in that data center.

You can define a RAS group in the server using the RESTful API. For more information, see the *Operations Orchestration Application Program Interface (API) Guide*.

Group Aliases

Group aliases let you separate between assigning an operation to a RAS during authoring time and in the runtime environment.

1. At authoring time, the author defines the operation to execute on a group alias rather than a group.

2. At runtime, the administrator maps the alias to a RAS group in the runtime environment, using the Central RESTful API. There is no need for the administrator to dive into the flows and modify the RAS assignment manually.

As a fallback, if the group alias is identical to the group name, it is mapped automatically to that group.

Optionally, at triggering time, it is possible to override the group alias and map the operation to a different RAS group.
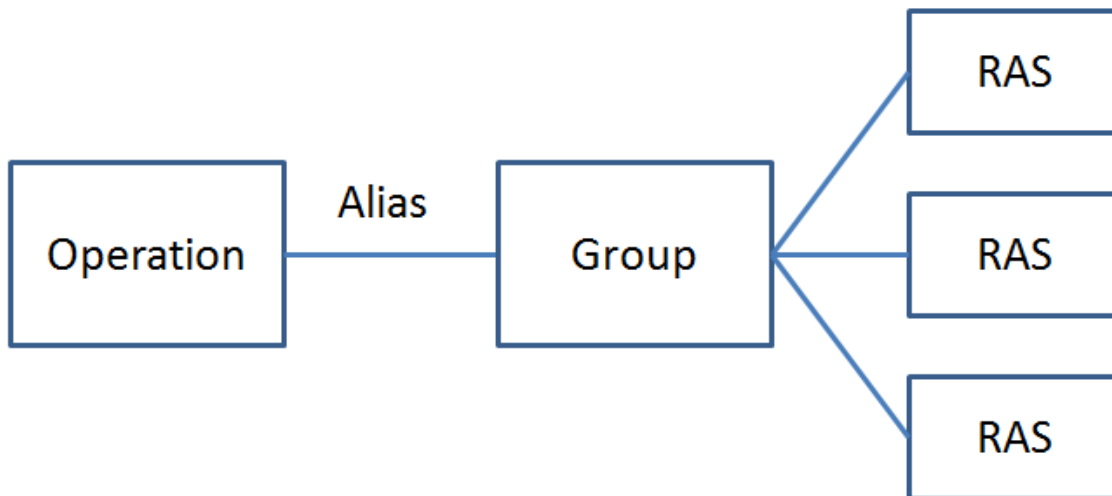
For example, you have a group of three RASes that run on the Oracle client. You create an operation that runs a query on Oracle. By using an alias for this group, you are telling Operations Orchestration that this operation needs to run on one of the RASes in this group. The choice of which RAS to use is determined at runtime, and does not need to be configured in the operation.

Group aliases are stored in the **Configuration\Group Aliases** folder.
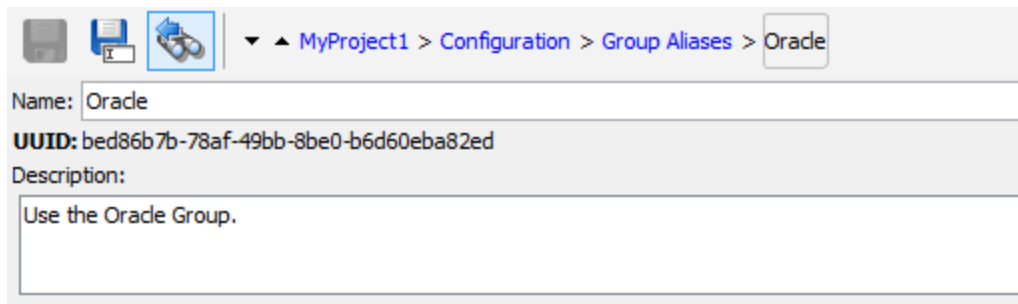
RAS Group Overrides

There are two methods of overriding RAS group details in Operations Orchestration:

- Static – as described in "Group Aliases" on the previous page

- Dynamic - Operations Orchestration assigns a value for the input at runtime. This can be done in one of the following ways:

  - The flow author can set the RAS using the **Override Group** field in the Operation fields in Studio's Input inspector. In the **Override Group** field, enter a different group, if you need to override the current group with a different one. This can be a static or a dynamic value. For example, ${overrideJRAS} where overrideJRAS is the name of a variable. During runtime, the variable can automatically be assigned a value using one of the methods described in "RAS Group Overrides" above.

○ From the Scheduler in Central, add an input field that will automatically be assigned during runtime. See "Scheduling Flow Runs" in the *Central User Guide*.

○ Add an input field to the flow in Studio. See "Creating Input" on page 472

Examples of RAS Group Override Use

- **Example 1:** Defining a dynamic run environment as a variable. This environment may be a test environment or a runtime production environment. This variable can be used as a flow input, step input or set context in Studio or as an input from the Scheduler.

- **Example 2:** Defining a SAP client with RAS. The client can be called from a Central flow as a SAP group.

What do you want to do?

Create a group alias

For example, you have created a RAS group of three RASes that run on the Oracle client. You need to create a group alias that will direct operations to work with this group, and thus to run on one of these RASes.

1. In the **Projects** pane, expand the **Configuration** folder.

2. From the **Group Aliases** folder, right-click and select **New > Group Alias**.

3. In the dialog box that appears, type a name for the new group alias, and then click **OK**.

4. In the **Description** box, type a description of the new group alias.

5. Click **Save**.

Rename a group alias

1. In the **Projects** pane, expand the **Configuration** and **Group Aliases** folders, and double-click the domain term to open its editor.

2. Right-click on the group alias and select **Rename**.

3. Type a new name for the group alias, and press Enter.

Copy a group alias

1. In the **Projects** pane, expand the **Configuration** and **Group Aliases** folders.

2. Right-click on the group alias that you want to copy.

3. Select **Edit > Copy**.

4. Move to the **Group Aliases** folder of the project you want to copy to.

5. Right-click and select **Edit > Paste**.

Delete a group alias

1. In the **Projects** pane, expand the **Configuration** and **Group Aliases** folders.

2. Right-click on the group alias to be deleted and select **Delete**.

Map the group alias to a group

Map the group alias to the group in the runtime environment using the Central RESTful API.

For more information, see the *Operations Orchestration Application Program Interface (API) Guide*.

Use the group alias in an operation

1. Create a new operation from an action plugin, as described in "Creating Operations" on page 632.

2. In the **Inputs** tab, in the **Operation Fields** section, notice the **Group Alias** button.

3. Go to **Configuration** > **Group Alias**, select the required group alias, drag it to the appropriate operation and drop it on the **Group Alias** icon (the globe based icon).

4. (Optional) In the **Override Group** box, enter a different group, if you need to override the current group with a different one. This can be a static or a dynamic value. For example, ${overrideJRAS} where overrideJRAS is the name of a variable. During runtime, the variable can automatically be assigned a value using one of the methods described in "RAS Group Overrides" on page 419.

5. Save the operation.

Remove the assignment of a group alias from an operation

After a group alias has been assigned to an operation, it is possible to remove this assignment.

1. Open an operation to which a group alias was assigned.

2. In the **Inputs** tab, in the **Operation Fields** section, right-click the link next to the **Group Alias** button.

3. From the right-click menu, select **Clear**.

Reference Material

Group Alias editor

GUI item	Description
Name	The name of the group alias.
Description	(Optional) Description of the group alias.

Configuring Role Aliases

Roles and permissions are assigned to users in Central. However, you can plan for the roles that will be used, by creating role aliases in Studio. For example, you may want to attach a role alias to a gated transition.

When the content pack is deployed to Central, role aliases such as ADMINISTRATOR, EVERYBODY, PROMOTER, SYSTEM_ADMIN, and END_USER are mapped to the corresponding roles in Central.

> **Note:** Note that some of the role aliases in the Base content pack (AUDITOR, LEVEL_ONE, LEVEL_TWO, and LEVEL_THREE) do not have a corresponding role in Central. These role aliases are considered deprecated.

Role aliases are stored in the **Configuration\Role Aliases** folder.

What do you want to do?

Create a role alias

1. In the **Projects** pane, expand the **Configuration** folder.

2. From the **Role Aliases** folder, right-click and select **New > Role Alias**.

3. In the dialog box that appears, type a name for the new role alias, and then click **OK**.

4. In the **Description** box, type a description of the new role alias.

5. Click **Save**.

Rename a role alias

1. In the **Projects** pane, expand the **Configuration** and **Role Aliases** folders.

2. Right-click on the role alias to be renamed and select **Rename**.

3. Type a new name for the role alias, and press Enter.

Copy a role alias

1. In the **Projects** pane, expand the **Configuration** and **Role Aliases** folders.

2. Right-click on the role alias that you want to copy.

3. Select **Edit > Copy**.

4. Move to the **Role Aliases** folder of the project you want to copy to.

5. Right-click and select **Edit > Paste**.

Delete a role alias

1. In the **Projects** pane, expand the **Configuration** and **Role Aliases** folders.

2. Right-click on the role alias to be deleted and select **Delete**.

Reference Material

Role Alias editor

GUI item	Description
Name	The name of the role alias.
Description	(Optional) Description of the role alias.

Configuring Scriptlets

Scriptlets (written in JavaScript) are optional parts of an operation that you can use to manipulate data.

You can use scriptlets to:

- Filter the results of an operation, flow, or step

- Determine the response of an operation

- Manipulate data in a subflow before passing the data to the parent flow

For example, the **divide by 1k** scriptlet performs a mathematical division of the input number (dividend) by 1024 (divisor) and returns the quotient rounded to the closest integer value.
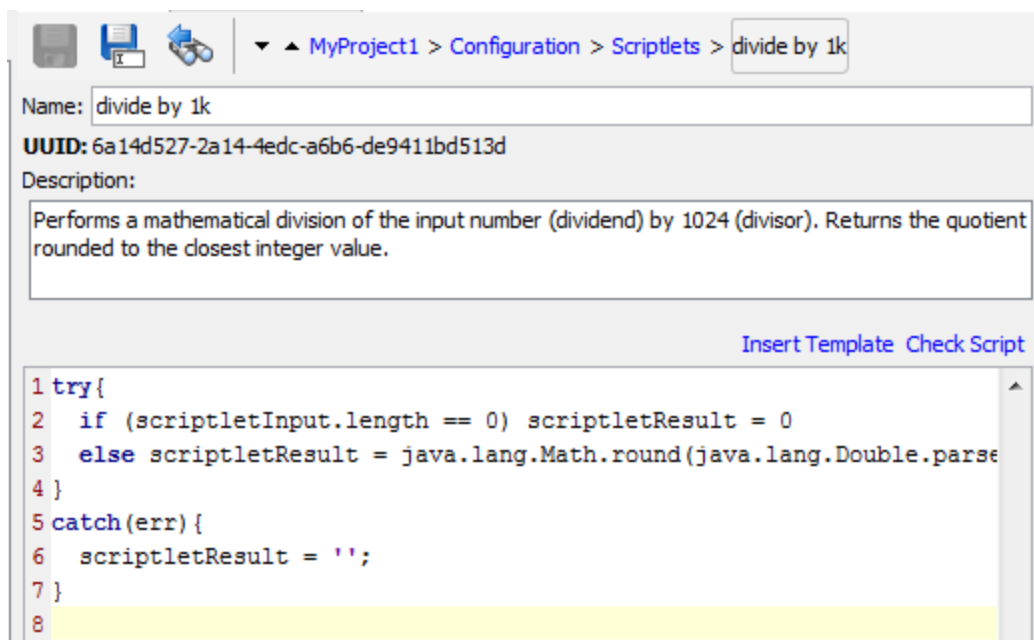
You can create a system scriptlet from scratch or take an existing scriptlet in an operation and save it as a shared system scriptlet. The resulting scriptlet is independent of the context for it was created and can be reused in any operation, flow, or step.

System scriptlets are stored in the **Configuration\Scriptlets** folder.

For more information about using scriptlets, see "Using Scriptlets in a Flow" on page 574.

What do you want to do?

Create a system scriptlet

1. In the **Projects** pane, expand the **Configuration** folder.

2. From the **Scriptlets** folder, right-click and select **New > Scriplets**.

3. Enter a name for the scriptlet and click **OK**.

   The Scriptlet Editor opens.
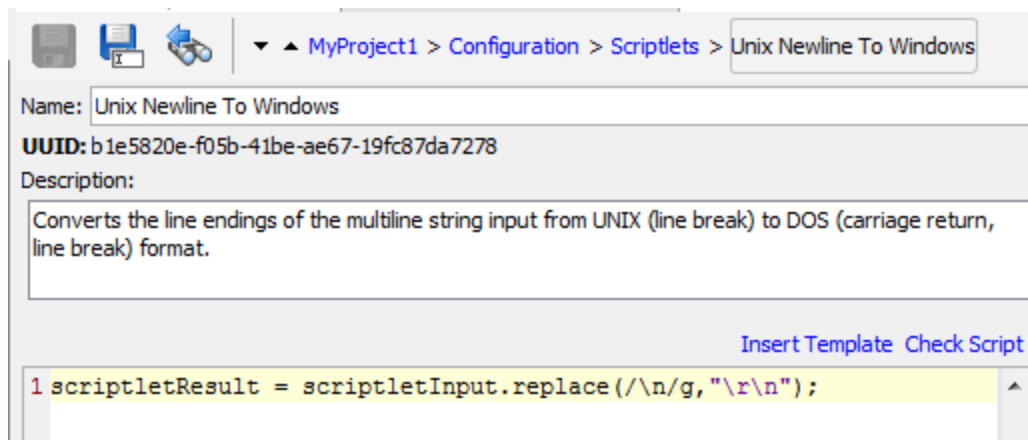
4. In the **Description** box, describe the purpose of the scriptlet.

5. Type the scriptlet in JavaScript.

   > **Note:** You can reference a variable that is stored in a folder using the format
   > `${variablename}` and a system property using the format `${path/system_property}`.

6. (Optional) Click **Insert Template** and follow the guidelines in the template to write the scriptlet.

7. Click **Check Script** to check for errors.

8. Click **Save**.

   The scriptlet is saved in the **Scriptlets** folder and is now available for use in any operation, flow, or step.

Save an existing scriptlet as a system scriptlet

1. Under the **Scriptlet** tab in the **Properties** sheet or Step Inspector, open the scriptlet that you want to save as a system scriptlet.

2. In the **Projects** pane, expand the **Configuration** and **Scriptlets** folders.

3. Drag the **Scriptlet** icon from the **Scriptlet** tab of the **Properties** sheet or Step Inspector to the

Configuration\Scriptlets folder.

4. To rename the new system scriptlet, right-click it, click **Rename**, and change its name.

Use a system scriptlet in an operation, flow, or step

1. Open the **Scriptlet** tab of the **Properties** sheet or Step Inspector for the operation, flow, or step on which you want to use the system scriptlet .

2. In the **Projects** pane, expand the **Configuration** and **Scriptlets** folders.

3. Drag the scriptlet from the **Scriptlets** folder to the **Scriptlet** icon in the **Scriptlet** tab of the **Properties** sheet or Step Inspector.

   The Scriptlet tab shows that there is now a reference to a shared scriptlet.

   Inputs | Outputs | Responses | Description | Scriptlet | Advanced |

   Reference to Shared Scriptlet: /MyProject1/Configuration/Scriptlets/divide by 1k

   Switch to a custom scriptlet
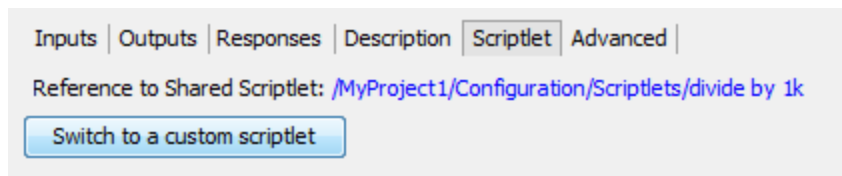
Edit a system scriptlet

1. In the **Projects** pane, expand the **Configuration** and **Scriptlets** folders.

2. Double-click the system scriptlet that you want to edit.

3. Modify the scriptlet and click **Save**.

Copy a scriplet

1. In the **Projects** pane, expand the **Configuration** and **Scriptlets** folders.

2. Right-click on the scriplet that you want to copy.

3. Select **Edit > Copy**.

4. Move to the **Scriplets** folder of the project you want to copy to.

5. Right-click and select **Edit > Paste**.

Rename a scriptlet

1. In the **Projects** pane, expand the **Configuration** and **Scriplets** folders.

2. Right-click on the scriplet to be renamed and select **Rename**.

3. Type a new name for the scriplet, and press Enter.

Delete a system scriptlet

Before you delete a system scriptlet, it is recommended to use the **What Uses This** function to check that other items do not depend on it. For more information, see "Finding Out How Flows and Operations are Used" on page 655.

1. In the **Projects** pane, expand the **Configuration** and **Scriptlets** folders.

2. Right-click the system scriptlet and select **Delete**.

3. Click **Yes** in the confirmation window.

Reference Material

Scriptlet Editor
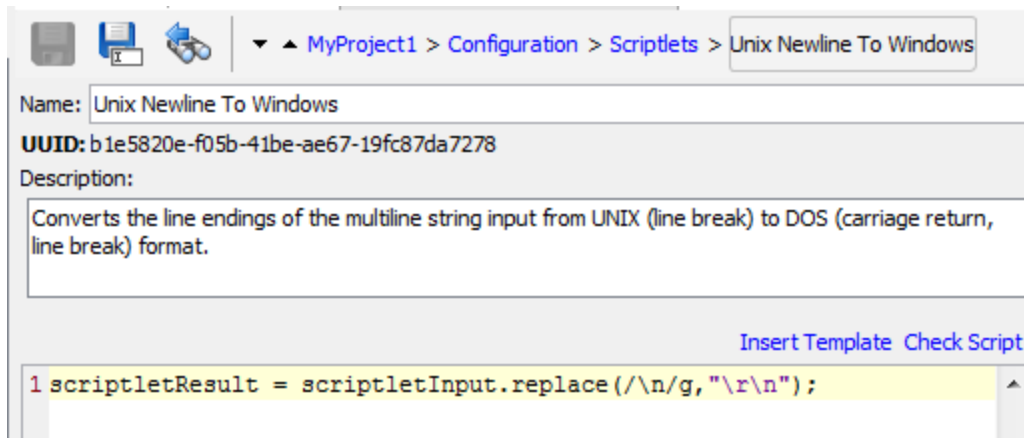
GUI item	Description
Name	The name of the scriptlet.
Description	(Optional) Description of the purpose of the scriptlet.
Insert Template	Click **Insert Template** to see guidelines to help you write your scriptlet.
Check Script	Click **Check Script** to check the scriptlet for errors.

Status Bar

- Displays the line column according to the location of the cursor.

- Displays the status of the **Insert** key, when the Insert mode characters are added. In **Overwrite** mode they are overwritten. Toggle between the two modes using the Insert key.

Keyword Completion

- While typing a keyword, press **Ctrl** and the **Space bar**, to display a drop-down list of options. Use the up and down arrow keys to navigate through the list and then select the appropriate word. After selecting an item the list disappears and the cursor is placed after the word.

```
1
2 var sysvalue_password = scriptletContext.get("var_sa_studio_central_default");
3 scriptletContext.put("field1", sysvalue_password);
4 v
5   volatile
    var
    void
```

Jump to Line

- Double-click on the line column item in the status bar or press **Ctrl + G** key. You can only jump to lines inside that have a number marker to the left of the scriptet window.

```
1
2 var sysvalue_password = scriptletContext.get("var_sa_studio_central_default");
3 scriptletContext.put("field1", sysvalue_password);
4
5
6
```

```
Go to line number:
1
(Between 1 and 6. Press ENTER to continue)
```

Configuring Selection Lists

Selection lists are lists of items that can be provided in user prompts in a flow.

For example, if the flow user needs to provide a step in the flow with the service status, you can create an input whose data source is a selection list and specify the Service Status selection list (whose members are Running, Stopped, and Paused).

Selection lists are stored in the **Configuration\Selection Lists** folder.

> **Note:** If you use the **Set Selection List** operation from the HPE Solutions content pack to add a custom value of a selection list, the custom value remains in Central even if the selection list itself is deleted (in the **oo_content_config_item** and **oo_config_item_values** tables).
>
> When redeploying, the new entity is reattached to the custom value.
>
> You can use one of the following workarounds to ensure that the custom value is deleted correctly:
>
> - After undeploying the old entity, but before deploying the new one, delete the configuration item using a REST call (with DELETE).
>
>   or:
>
>   Remove the custom value using a REST call (using PUT with null in the request body).

What do you want to do?

Create a selection list

1. In the **Projects** pane, expand the **Configuration** folder.

2. From the **Selection Lists** folder, right-click and select **New > Selection List**.

3. In the dialog box that appears, type a name for the new selection list, and then click **OK**.

4. In the **Description** box, type a description of the new selection list.

5. Click **Add** to add a new selection list value.

6. In the **Value** column, enter the name for the selection list value.

7. Click **Save**.

Remove a selection list value

1. In the **Projects** pane, expand the **Configuration** and **Selection Lists** folders, and double-click the selection list to open its editor.

2. Highlight the value and click **Remove**.

Change a selection list value

1. In the **Projects** pane, expand the **Configuration** and **Selection Lists** folders, and double-click the selection list to open its editor.

2. Double-click the value you want to change and type the new value.

Rename a selection list

1. In the **Projects** pane, expand the **Configuration** and **Selection Lists** folders, and double-click the selection list to open its editor.

2. In the **Name** box, type the new name for the selection list.

3. Click **Save**.

Copy a selection list

1. In the **Projects** pane, expand the **Configuration** and **Selection Lists** folders.

2. Right-click on the selection list that you want to copy.

3. Select **Edit > Copy**.

4. Move to the **Selection Lists** folder of the project you want to copy to.

5. Right-click and select **Edit > Paste**.

Delete a selection list

Before you delete a selection list, it is recommended to use the **What Uses This** function to check that other items do not depend on this selection list. For more information, see "Finding Out How Flows and

1. In the **Projects** pane, expand the **Configuration** and **Selection Lists** folders.

2. Right-click the selection list and select **Delete**.

3. Click **Yes** in the confirmation windows.

Reference Material

Selection List editor

GUI item	Description
Name	The name of the selection list.
Description	(Optional) Description of the purpose of the selection list.
Add	Click **Add** to add a new value to the selection list.
Remove	Click **Remove** to remove the selected value from the selection list.
Value	Enter the value in the selection list.

Configuring System Accounts

A system account is an object that contains an account's credentials (user name and password), while protecting the credentials from being viewed other than in the installation of Studio on which the system account was created.

Flow authors can use system accounts when creating a flow. For example, you can set an input source to be credentials from a system account. See "Specifying the Input Source " on page 482.

> **Note:** The system accounts defined here are for Studio only. System accounts also need to be set up for execution. This is done via API. For more information, see the *OO Application Program Interface (API) Guide*.

Users never see the system account name that provides a flow with user account credentials for access to a remote machine. Thus the credentials are protected from decryption, and the system account name is hidden from the user.

System accounts are stored in the **Configuration\System Accounts** folder.

> **Note:** The following characters cannot be used in a system account name: <>\\"/;%.

What do you want to do?

Create a system account

1. In the **Projects** pane, expand the **Configuration** folder.

2. From the **System Accounts** folder, right-click and select **New > System Account**.

3. In the dialog box that appears, type a name for the new system account, and then click **OK**.

4. (Optional) In the **Description** box, type a description of the system account.

5. In the **User Name** box, type the user name of the account that the system account represents, using the following syntax:

   `<domain>\<username>`

6. Click the **Assign Password** button.

7. In the **Password** box, type the password, and then in the **Confirm Password** box type it again.

8. Click **Save**.

Copy a system account

1. In the **Projects** pane, expand the **Configuration** and **System Accounts** folders.

2. Right-click on the system account that you want to copy.

3. Select **Edit > Copy**.

4. Move to the **System Accounts** folder of the project you want to copy to.

5. Right-click and select **Edit > Paste**.

> **Note:** You can also move/copy system accounts, or a folder that contains system accounts, from one project to another by dragging and dropping.

Edit a system account

1. In the **Projects** pane, expand the **Configuration** and **System Accounts** folders.

2. Double-click the system account that you want to edit.

3. Make your changes in the editor and click **Save**.

Delete a system account

1. In the **Projects** pane, expand the **Configuration** and **System Accounts** folders.

2. Right-click the system account and select **Delete**.

3. Click **Yes** in the confirmation windows.

> **Note:** If you delete a system account from a content pack and then redeploy the content pack, the system account is not removed from the database. If this occurs, you will need to remove the system account via REST API:
>
> Do DELETE on: `/oo/rest/system-accounts/<sa_name>`.

> For more information about working with REST APIs, see the *OO Application Program Interface (API) Guide.*

Reference Material

System Accounts editor

GUI item	Description
Name	The name of the system account. **Note:** The following characters cannot be used in a system account name: <>\\"/;%.
Description	(Optional) Description of the purpose of the system account.
User Name	The user name of the account that the system account represents, using the syntax `<domain>\<username>`.
Assign Password	Click to open the Enter Password dialog box, where you will need to type the password twice.

Configuring System Evaluators

System evaluators are string formats that flow authors can use to validate inputs for any data sources except system accounts.

For example:

- If the input is an email address, you can use an evaluator to check that the input is in the correct email format.

- If the input must be a numeric value greater than or equal to 1, you can use an evaluator to check that this is the case.

See "Evaluating Input Data" on page 494.

System evaluators can use any of the following:

- Simple operators such as =, !=, Begins with, Contains, Match All Words, and Match At Least One Word and so on

- Regular expressions – for more information, see "Using Regular Expressions in a Flow" on page 579

- Scriptlets – for more information, see "Using Scriptlets in a Flow" on page 574

System evaluators are stored in the **Configuration\System Evaluators** folder.

Name: MyEvaluator
UUID: 167a68dd-82d3-4e91-add4-ca48cdbe4a7d
Description:

Checks that the input value is less than 1

Evaluator Type: <
Compare the input against a string or number
Compare To: 1

Test Filter Input
Pattern Matches (1 occurrence)   Clear Quick Command

What do you want to do?

Create a system evaluator

1. In the **Projects** pane, expand the **Configuration** folder.

2. From the **System Evaluators** folder, right-click and select **New > System Evaluator**.

3. In the Select Evaluator dialog box, select an evaluator type, and then click **OK**.
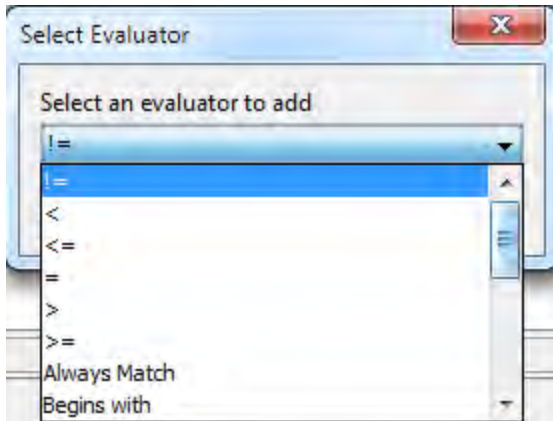
4. Enter a name for the evaluator and click **OK**.

   The Evaluator Editor opens. The appearance of the Evaluator Editor varies, depending on the type of evaluator you selected.

5. In the **Description** box, describe the purpose of the evaluator.

6. Enter the text, string, expression value, or scriptlet with which the evaluator should test the input.

7. Test the filter:

   a. Click **Clear** to clear the **Test Filter Input** box.

   b. Click **Quick Command**.

c. Type a command that generates the desired data.

d. Click **OK**. The output of the command appears in the **Test Filter Input** box.

For more information about testing filters, see "Filtering Output and Results" on page 529.

8. Click **Save**.

The evaluator is saved in the **System Evaluators** folder and is now available in the **Validation Format** list in the Input Editor.

Edit a system evaluator

1. In the **Projects** pane, expand the **Configuration** and **System Evaluators** folders.

2. Double-click the system evaluator that you want to edit.

3. Modify the text, string, expression value, or scriptlet with which the evaluator should test the input.

4. Click **Save**.

Copy a system evaluator

1. In the **Projects** pane, expand the **Configuration** and **System Evaluators** folders.

2. Right-click on the system evaluator that you want to copy.

3. Select **Edit > Copy**.

4. Move to the **System Evaluators** folder of the project you want to copy to.

5. Right-click and select **Edit > Paste**.

Delete a system evaluator

Before you delete a system evaluator, it is recommended to use the **What Uses This** function to check that other items do not depend on it. For more information, see "Finding Out How Flows and Operations are Used" on page 655.

1. In the **Projects** pane, expand the **Configuration** and **System Evaluators** folders.

2. Right-click the system evaluator and select **Delete**.

3. Click **Yes** in the confirmation window.

Reference Material

Evaluator Editor – Standard

The appearance of the Evaluator Editor varies, depending on the type of evaluator you selected. If you selected a simple operator in the Select Evaluator dialog box, such as such as `=`, `!=`, `Begins with`, `Contains`, `Match All Words`, and `Match At Least One Word`, the Evaluator Editor appears as follows:

GUI item	Description
Name	Displays the name of the system evaluator
Description	Enter a description of the evaluator
Compare To	Type the text, string, expression value, or scriptlet with which the evaluator should test the input.
Test Filter Input	This is where you place data in order to test that the filter works as expected. detailed information about the GUI items in this section, see "Filtering Output and Results" on page 529.

Evaluator Editor – Regular Expression

The appearance of the Evaluator Editor varies, depending on the type of evaluator you selected. If you selected **Regular Expression** in the Select Evaluator dialog box, the Evaluator Editor appears as follows:
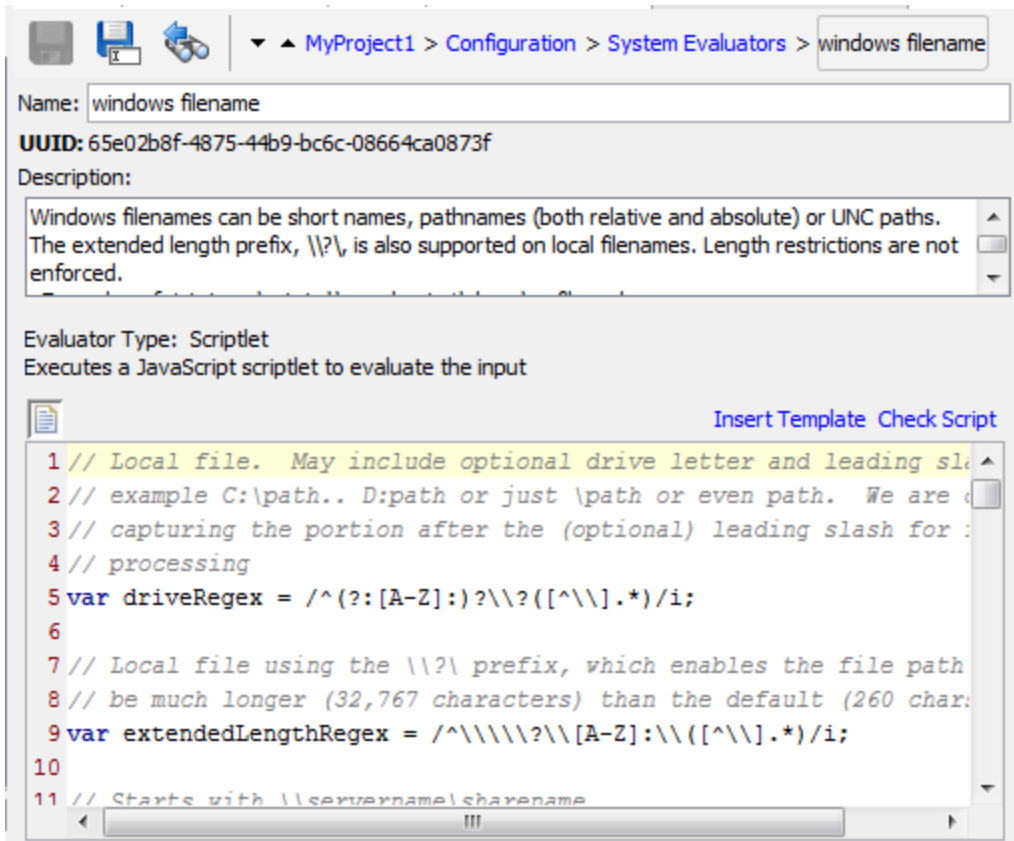
For information about creating regular expressions, see "Using Regular Expressions in a Flow" on page 579.

GUI item	Description
Name	Displays the name of the system evaluator
Description	Enter a description of the evaluator
Expression Type	Select **Java Style** as the type of regular expression. Do not use the other styles; they have been deprecated.
Expression Value	Type the regular expression.
Match the whole input	Select to apply the evaluator to the entire input.
Multi-line	Select to enable multiple lines in an expression.
Ignore Case	Select to make the regular expression not case-sensitive.

Evaluator Editor – Scriptlet

The appearance of the Evaluator Editor varies, depending on the type of evaluator you selected. If you selected **Scriptlet** in the Select Evaluator dialog box, the Evaluator Editor appears as follows:

For information about creating scriptlets, see "Using Scriptlets in a Flow" on page 574.

GUI item	Description
Name	Displays the name of the system evaluator.
Description	Enter a description of the evaluator.
Scriptlet icon 📄	Drag this icon to the **Configuration\Scriptlets** folder, to save a scriptlet there for reuse.
Insert Template	Click **Insert Template** to see guidelines to help you write your scriptlet.
Check Script	Click **Check Script** to check the scriptlet for errors.

Configuring System Filters

Filters are used to extract and modify parts of an operation's output or a step's result. A system filter is available system-wide, to be used in multiple steps and operations.

For example, the filters used in a Ping operation might be useful for other ping operations.

You can create a system filter from scratch or take an existing filter in an operation and save it as a system filter. The resulting system filter is independent of the operation for it was created and can be reused in any output or result.

System filters are stored in the **Configuration\System Filter** folder.
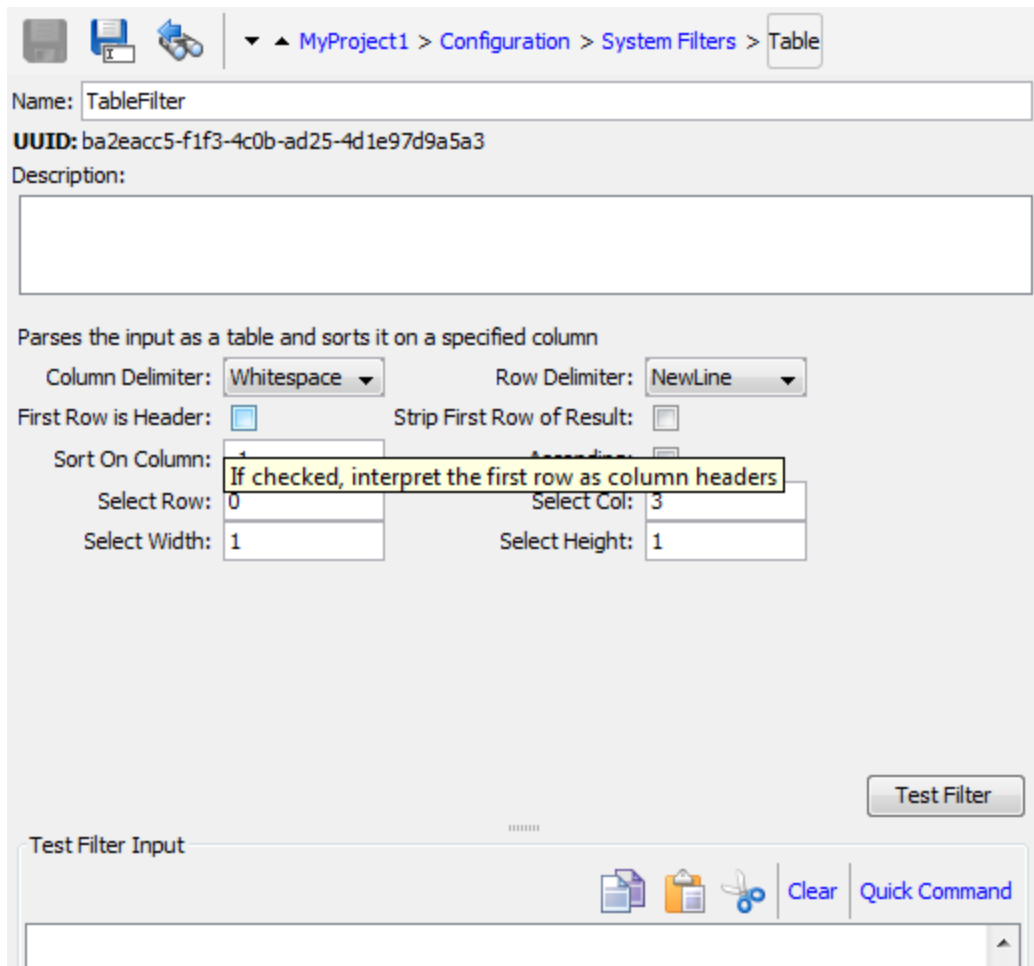
What do you want to do?

Create a system filter

1. In the **Projects** pane, expand the **Configuration** folder.

2. From the **System Filters** folder, right-click and select **New > System Filter**.

3. From the **Select Filter** list, select the type of filter.

4. Enter a name for the filter and click **OK**.

   The Filter Editor opens. The appearance of the Filter Editor varies, depending on the type of filter you selected.

5. In the **Description** box, describe the purpose of the filter.

6. Enter the text, string, expression value, or scriptlet with which to filter the output or result. For information about the different filter options, see "Filtering Output and Results" on page 529.

7. Test the filter:

   a. Click **Clear** to clear the **Test Filter Input** box.

   b. Click **Quick Command**.

   c. Type a command that generates the desired data.

   d. Click **OK**. The output of the command appears in the **Test Filter Input** box.

   For more information about testing filters, see "Filtering Output and Results" on page 529 .

8. Click **Save**.

   The filter is saved in the **System Filters** folder and is now available in the **Validation Format** list in the Input Editor.

> **Troubleshooting**: If you are using a localized Windows command line, after using Quick Command, you might get some strange characters in the results, because of the encoding of the command line.
>
> To avoid this:
>
> 1. Close Studio.
>
> 2. Open **<Install_folder>/studio/Studio.l4j.ini**.
>
> 3. Add the following:
>
>    ```
>    -Ddharma.windows.cmd.unicode=true
>    ```
>
> This will ensure that the characters read from the command line are interpreted in unicode.
>
> Note that using this parameter might cause performance problems if you have multiple runs of flows containing operations that use the command line. Remove it from the **Studio.l4j.ini** file after you finish testing the filters.

Save an existing filter as a system filter

1. Open the operation and, in the Filter Editor, select the filter you want to save as a system filter.

2. In the **Projects** pane, expand the **Configuration** and **System Filters** folders.

3. In the operation's Filter Editor, drag the filter from the **Filter** list to the **System Filters** folder.
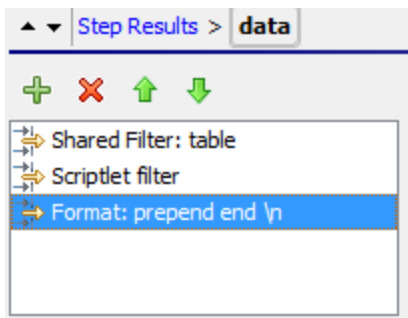
4. To rename the new system filter, right-click it, click **Rename**, and change its name.

Use a system filter in an output or result

1. Open the Filter Editor of the output or result on which you want to use the system filter.

2. In the **Projects** pane, expand the **Configuration** and **System Filters** folders.

3. Drag the filter that you want to use from the **System Filters** folder to the **Filter** list in the Filter Editor.
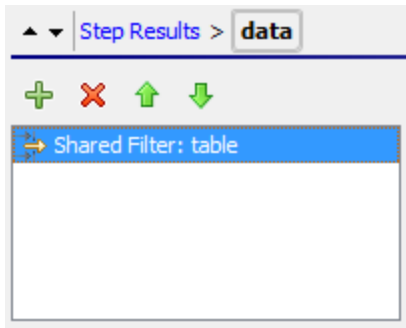
Edit a system filter

1. In the **Projects** pane, expand the **Configuration** and **System Filters** folders.

2. Double-click the system filter that you want to edit.

3. Modify the filter and click **Save**.

Copy a system filter

1. In the **Projects** pane, expand the **Configuration** and **System Filters** folders.

2. Right-click on the system filter that you want to copy.

3. Select **Edit > Copy**.

4. Move to the **System Filters** folder of the project you want to copy to.

5. Right-click and select **Edit > Paste**.

Delete a system filter

Before you delete a system filter, it is recommended to use the **What Uses This** function to check that other items do not depend on it. For more information, see "Finding Out How Flows and Operations are Used" on page 655.

1. In the **Projects** pane, expand the **Configuration** and **System Filters** folders.

2. Right-click the system filter and select **Delete**.

3. Click **Yes** in the confirmation window.

Reference Material

Filter Editor

The appearance of the Filter Editor varies, depending on the type of filter you selected. For information about the different options, see *Filter Options* in "Filtering Output and Results" on page 529
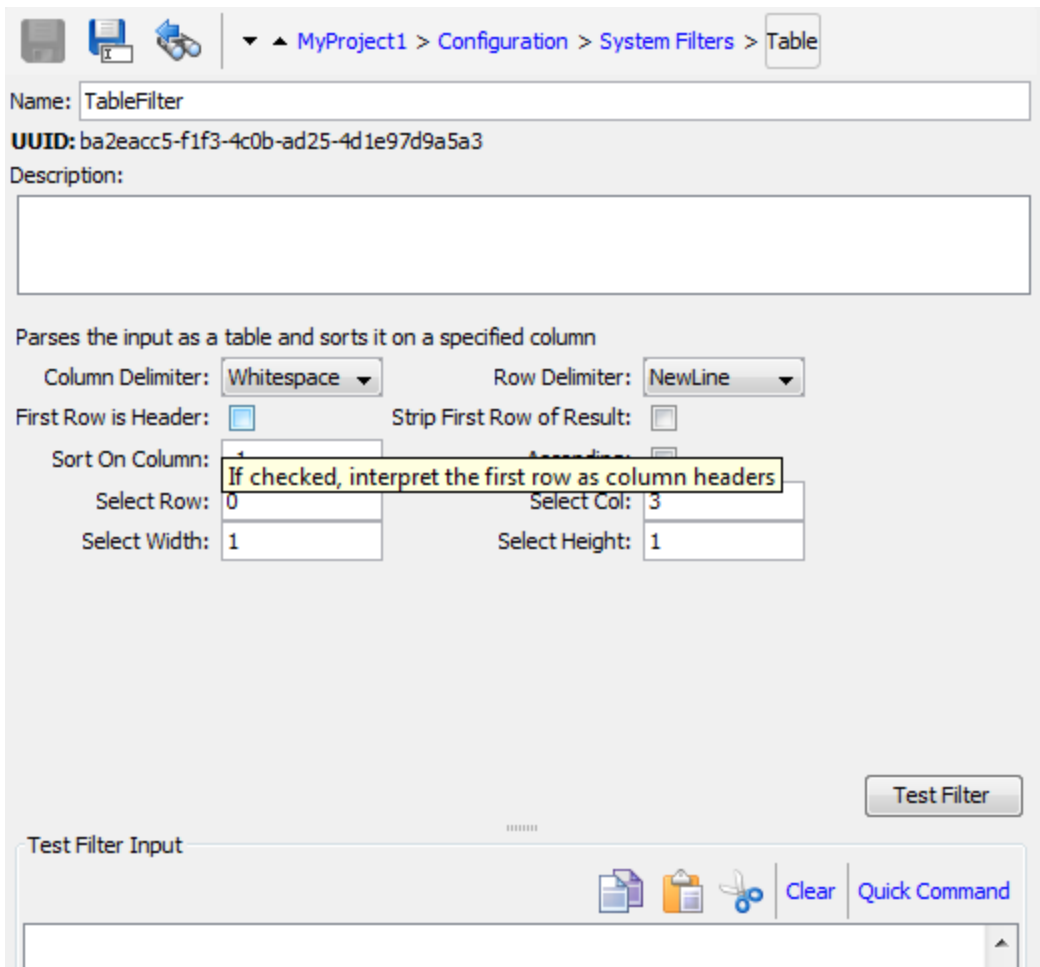
GUI item	Description
Name	Displays the name of the system filter
Description	Enter a description of the system filter

Configuring System Properties

System properties are global flow variables with values that never change and so can be used in any flow, saving you the time of recreating a flow variable each time you need to use it. Any reference to a system property obtains the system property's value.

For example, the **SQL Database Type** system property lists the different types of SQL databases.
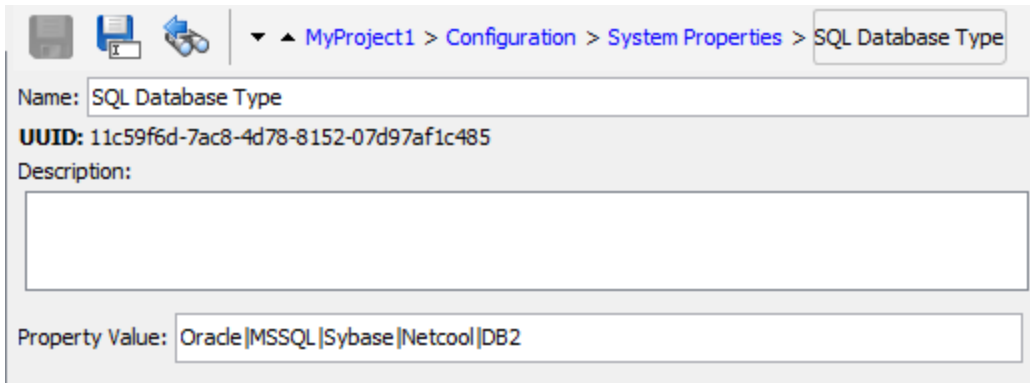
System properties are stored in the **Configuration\System Properties** folder.

What do you want to do?

Create a system property

1. In the **Projects** pane, expand the **Configuration** folder.

2. From the **System Properties** folder, right-click and select **New > System Property**.

3. Enter a name for the system property and click **OK**.

   The System Properties Editor opens.
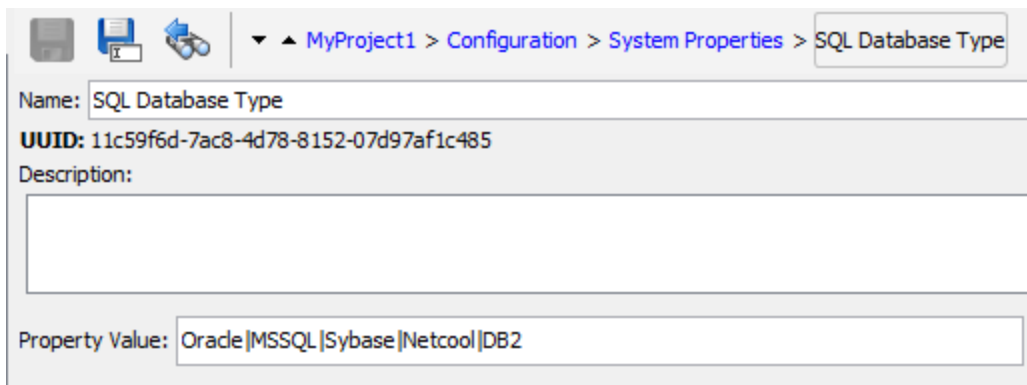
4. (Optional) In the **Description** box, type a description of the system property.

5. In the **Property Value** box, type the values for the system property, using **|** as a separator.

6. Click **Save**.

   The system property is saved in the **System Properties** folder and is now available for use in any flow.

Use the value of a system property in a flow

1. Specify the system property as the data source for a flow or step input.

2. In a scriptlet, use the appropriate command to obtain the system property's value.

   > **Note:** For information on the required command and its syntax, on the **Scriptlet** tab of an operation, click **Insert Template**. The template provides the necessary commands for working with the global context. For more information, see "Using Scriptlets in a Flow" on page 574.

Change the value of a system property

There are several ways to change the value of a system property:

- Change the system property in a scriptlet. This changes the value from the point at which the script is run. For information on the required command and its syntax, on the **Scriptlet** tab of an operation, click **Insert Template**.

- Create an operation that sets the system property's value.

- Open the system property in the **Configurations\System Properties** folder, and change the value.

Rename a system property

1. In the **Projects** pane, expand the **Configuration** and **System Properties** folders.

2. Right-click on the system property to be renamed and select **Rename**.

3. Type a new name for the system property, and press Enter.

Copy a system property

1. In the **Projects** pane, expand the **Configuration** and **System Properties** folders.

2. Right-click on the system property that you want to copy.

3. Select **Edit > Copy**.

4. Move to the **System Properties** folder of the project you want to copy to.
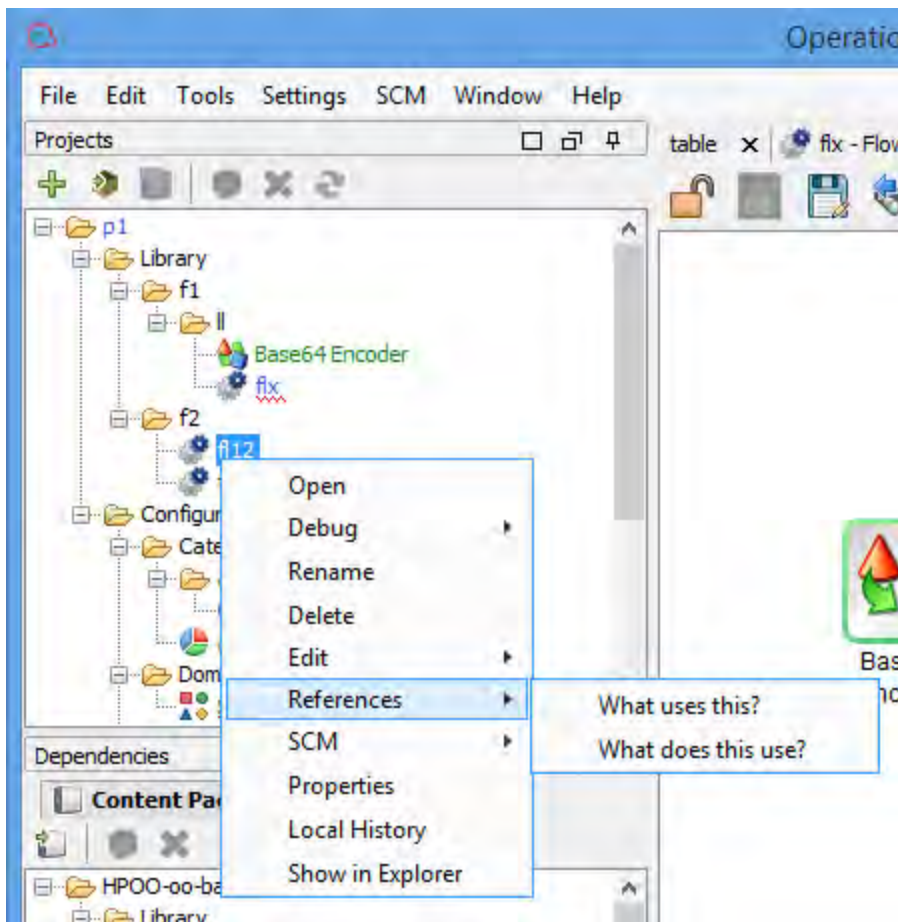
5. Right-click and select **Edit > Paste**.

Delete a system property

Before you delete a system property, it is recommended to use the **What Uses This** function to check that other items do not depend on it. For more information, see "Finding Out How Flows and Operations are Used" on page 655.

1. In the **Projects** pane, expand the **Configuration** and **System Properties** folders.

2. Right-click the system property and select **Delete**.

3. Click **Yes** in the confirmation window.

Identify what uses a system property

1. In the **Project** pane, right-click the system property, and select **References** > **What uses this?**.

The **References** pane opens, displaying the references to the system property.

2. Double-click a row in the **References** pane to open the item for editing.

> **Note:** To open multiple items, select them using the SHIFT or CONTROL keys, right-click, and select **Open**.

Identify what a system property uses

1. In the **Project** pane, right-click the system.

2. Select **References** > **What does this use?**.

    The **References** pane opens, displaying the references from the system property.

Reference Material

System Properties Editor

GUI item	Description
Name	The name of the system property.
Description	(Optional) Description of the system property.
Property Value	Type the values for the system property, using | as a separator.

Authoring a Flow – Basics

A flow is a set of actions that are linked by decision-making logic, to automate tasks.

For example, you want to verify that a page on your Web site contains the correct, current data, such as a certain piece of text. If the desired data is not on the Web page, you want to push new content to the site. You can create a flow to do these tasks automatically.

This chapter covers all the basic steps that need to be done to create a simple flow. For information about creating more complex flows, see "Advanced Authoring" on page 557.

Creating a Flow – Step-by-Step

This topic takes you step-by-step through the major steps involved in creating a flow. It demonstrates how to create a simple flow that checks whether a page on a Web site contains a certain piece of text and, if that text is not found, posts a page to the Web site.

Note that this is just a high level look at the Studio workflow, and there are many options that are not described here. For more detailed information about any of the steps, use the links to learn about the flow authoring options in-depth.

This topic presents the steps in a suggested order, but the steps do not have to be completed in the order shown.

Step 1: Create a flow

1. Open the project in which you want to create the flow.

2. Open the folder in which you want to create the flow.

3. Select **File** > **New** > **Flow**.

4. Type a meaningful name for the flow and click **OK**.

   > **Note:** Names can be a maximum of 128 characters long, and are not case-sensitive.

5. Open the flow in the authoring pane and click **Properties** (at the bottom of the pane), then click the **Description** tab.

6. Enter a description of the flow.

For more information and options, see "Creating a New Flow" on page 455.

Step 2: Add operations as steps

Drag an operation to the authoring pane, to use it as a step in your flow.

- If you want to modify the operation, copy and paste it into the **Projects** pane before dragging it onto the authoring pane.

   > **Note:** This is only recommended if you want to add responses or results, if only the inputs are used, then modify them inside the steps.

In our example, there are two steps. Step 1 uses the **Check Web Site** operation, which checks a Web page to see whether it contains specific text. Step 2 uses the **Post Page** operation to post a page to the Web site.

For more information and options, see "Creating Steps in a Flow" on page 458.

Step 3: Create return steps to end the flow

Create one or more return steps to end the flow. Return steps indicate four primary possible end states to the flow: **Success**, **Diagnosed**, **No action**, and **Failure**.

1. On the authoring pane toolbar, click the **Step Palette** button .

2. From the **Step** palette, drag the appropriate return step icons to the authoring canvas.

3. If required, change the flow responses that are assigned to the steps.

In our example flow, there are two end states: **Success** and **Failure**. There is no need to change the default flow responses.

For more information and options, see "Creating Return Steps" on page 553.

Step 4: Create transitions

Create connections between steps so that each response for a step takes the flow to another step or to an end step.

> **Note:** Every response icon on the step must be connected to another step; otherwise, the flow will show errors.

1. On the step that you want to connect to the next step, click the icon that represents one of the responses , and drag a line to the destination step for that response.

2. If you need to modify the transition, double-click the line to open the Transition Inspector.

3. Repeat the process for the other response icons on the step.

In our example:

- If the Web page cannot be found, the flow ends in failure. Drag from the red **Fail** response icon to the **Fail** end step.

- If the page is there but the text isn't present, go to the second step, of posting another page to Web site. Drag from the yellow response icon to the second step.

- If the page is there and the desired text is present, the flow ends in success. In both steps, drag from the green **Success** response icon to the **Success** end step.

For more information about transitions, see "Creating Transitions" on page 496.

For more information about setting the responses for an operation, see "Setting Responses" on page 501.

Step 5: Add inputs to the flow

1. Right-click on the flow and select **Properties**.

2. In the Operation Editor, click the **Inputs** tab.

3. Click **Add input** ✚ from the Inputs toolbar and enter a name for the input.

4. Click in the row's **Type** column and select one of the value assignment types from the list:

   ○ **Single Value** ⬭ ▼

   ○ **List of Values** ⬛ ▼ - to run an operation against multiple targets

5. In the **Assign From** field, define the data source for the input.

In our example, the **Check Web Site** step needs to know which page to check (mysite.com/mypage.htm) and what text to look for ("needed text"). In this case, you would create two single constant value inputs.

For more information and options, see "Creating Input" on page 472.

Step 6: Add results to the flow

By adding a result, you can capture a step's output and save it as a flow variable (to be used in other steps in the flow) or a flow output field (to be passed to a parent flow).

1. In the Step Inspector, click the **Results** tab.

2. Click **Add result** ✚ from the Step Inspector toolbar.

3. In the **Name** column, enter a name for the result, and press the Return key on your keyboard. This name will be used as the name of the flow variable or flow output field.

4. From the **From** list, select the source for the result. For example, select the primary output.

5. From the **Assign To** list, decide where the value will be stored: as a flow variable or output field.

6. From the **Assignment Action** list, select the appropriate action: overwrite, append, prepend, or one of the arithmetic assignment actions.

In our example, the **Check Web Site** step can be set to store the text that it found as a flow variable. In our simple two-step flow, the successful completion of the first step goes straight to the **Resolved: Success** end step. But it is possible to add another step, for example, a **Send Email** step, which includes the flow variable data in the body of the email.
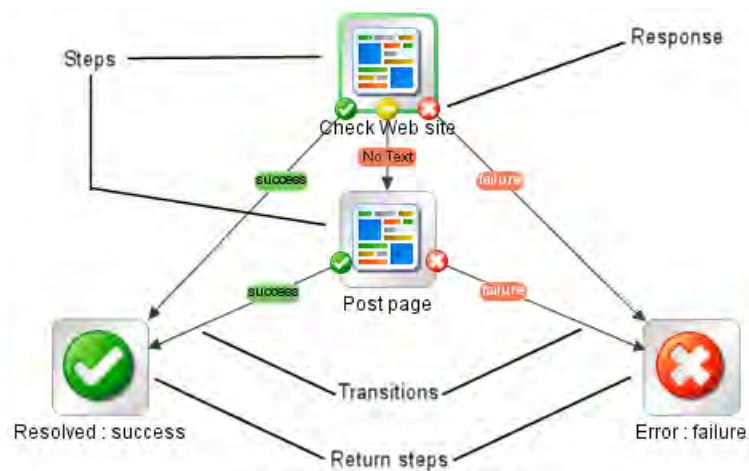
For more information and options, see "Setting Step Results" on page 520.

Step 7: Save the flow

Click the **Save** button.

Studio validates the flow. If a flow is not valid, it is saved and an error message appears.

The different flow elements are shown below:

Creating a New Flow

There are two options for creating a flow:

- Create the flow from scratch

- Use a predefined template as the basis of the flow

> **Important!** If you need to delete, create, or rename a flow inside your project, make sure to do this from within Studio, and not by deleting, creating, or renaming the item in the file system.

Create a flow from scratch

1. Open the project in which you want to create the flow.

2. Open the folder in which you want to create the flow.

3. Select **File** > **New** > **Flow**.

4. Type a meaningful name for the flow and click **OK**.

> **Note:** Flow names can be a maximum of 128 characters long, and are not case-sensitive.

Create a flow from a template

The templates provided with Studio provide flows that perform certain frequently used tasks. For example, there is a **Restart Service** template for creating a flow to restart a service.

1. Open the project in which you want to create the flow.

2. On the Welcome screen, click the **New Flow** button.

3. In the list of templates that appears, highlight a flow template to display its description. If required, drag the scroll bar to display the text.


4. Select the flow template that meets your needs, and then click the **Create** button.

5. The new flow is created in the **My Ops Flows** in the **Library** folder of the selected project. If you want to store the flow in a different folder, drag it to the desired folder, or use the **Edit** > **Cut** and **Edit** > **Paste** commands.

Tip: You can also get templates from the **Templates** folder in **OO-Base Content** content pack, in the **Dependencies** pane. Double-click the template to open it in the authoring pane. If you want to modify the flow, copy it into your project and modify the copy.

Note: You can create templates from other flows, by creating a **Templates** folder under **Library**, in your project, and storing flows in that folder. These flows will be displayed in the list of flow

templates.
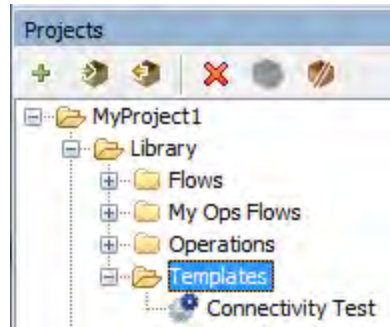
Add a description of the flow

1. On the authoring pane, right-click the flow, and click **Properties**.

2. Click the **Description** tab.

3. Enter a description of the flow and click **OK**.

Reference Material

Available Flow Templates

When you click the **New Flow** button on the Operations Orchestration Welcome screen, the **Available Flow Templates** list is displayed.

GUI item	Description
Name	Displays the names of the available templates. Highlight a name to display its description in the **Template Description** box.
Path	Displays the path to the folder where each template is stored.
Template Description	Displays a description of the selected template.

Creating Steps in a Flow

When you create a step from an operation, the step is an instance of the operation and so inherits the operation's inputs, outputs, references, and other characteristics.

To create a step from an operation, drag the operation to the authoring pane.

- If you drag an operation from the **Dependencies** pane, you will be able to modify the step, but note that the operation in the **Dependencies** pane is read-only.

- If you want to modify an operation before creating the step, copy it from the **Dependencies** pane and paste it into the **Projects** pane, before dragging it onto the authoring pane.

What do you want to do?

Create a step from an operation

1. In the **Projects** pane or **Dependencies** pane, select the operation that you want to add to the flow.

   Note: The operations in the **Dependencies** pane are read-only.

2. Drag the operation from the project tree to the authoring pane.

3. If required, rename the step to reflect its function within the flow (operation names may be too generic):

   a. Right-click the step that you want to rename and select **Rename**.

   b. Type the new name in the highlighted field.

4. If required, edit the step. For more information, see "Modifying a Flow" on page 466.

Copy a read-only operation into the project to make it editable

1. In the **Dependencies** pane, select the operation that you want to copy.

2. Select **Edit** > **Copy**.

3. Select the location in the project tree where you want to paste the copy, and select **Edit** > **Paste**. This operation is treated as a new object and is detached from the content pack it arrived with.

4. If required, edit the operation.

   Note: If you edit the operation, any steps that were created from this operation inherit the changes to the properties. If you edit the step, this does not affect the original operation.

5. Drag the operation from the project tree to the authoring pane.

Note: A operation that is copied from a content pack to a project is a "soft copy". This means that if the operation was originally created by importing an action plugin, the copied operation continues to reference the original operation. If the action plugin is upgraded and the original operation is updated to call the new version, the copied operation is updated automatically. For more information, see "Creating Operations" on page 632.

Copy a step from within the flow

To copy and paste a step, use any of the following tools:

- The **Copy** and **Paste** buttons on the authoring pane toolbar

  The **Edit**> **Copy** and **Edit** > **Paste** menu commands

- The right-click menu

- Keyboard combinations (**Ctrl+C**, **Ctrl+V**)

Give a step a description

1. On the authoring pane, right-click a step, and click **Properties**.

2. Click the **Description** tab.

3. Enter a description of the step and click **OK**.

   For best practices for writing a description, see Authoring Best Practices.

Give a step a callout

Callouts contain information about a step. They can greatly enhance the usability of a flow, by providing information such as:

- Data movement: how information is passed from one step to another

- Names of flow variables that store data

- Formats required for the input data

For the Windows Health Check, you need the machine name or IP address of the target machine and, if the machine is not local, user credentials for running the subflow on the machine.

Windows Health Check
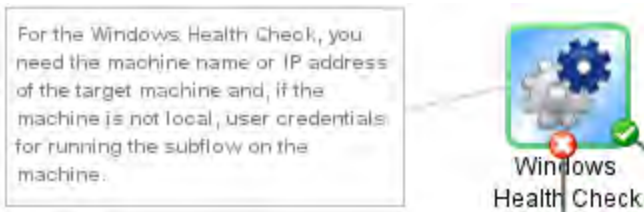
1. Display the **Step** palette by clicking the **Step Palette** button from the authoring pane toolbar.

2. Click the **Callout** button and drag the callout onto the authoring pane.

3. Type the text for the callout.

4. To connect the callout to a step, drag from the gray circle to the step.

5. Drag the corners of the callout text area to resize it.

Create a step from a flow (subflow)

A subflow is a flow within a flow. For more information, see "Creating a Subflow Within a Flow" on page 557.

1. Open the parent flow in the authoring pane.

2. In the Library, select the flow that you want to use as a step (or subflow).

3. Drag the flow from the Library to the parent flow in the authoring pane. The flow that you dragged becomes a step in the parent flow.

   Note: The **Accelerator Packs** folder contains flows that can run as parent flows.

Create a non-blocking step

A non-blocking step does not block the rest of the flow. While it is running, the flow run continues to carry out the steps that come after it.

You would do this when a flow's subsequent actions do not depend on the outcome of the step. For example, you want a flow to send a notification after the failure response of a step. But you don't need the flow to wait while the notification is being sent.

   Note: In a flat flow or subflow that contains a non-blocking step, the flow run will not end until the non-blocking step is finished. However, if it is part of a flow with multi-instance or parallel steps, the flow run will not wait for the non-blocking step.

> Note also that it is not possible for multi-instance or parallel split steps to be non-blocking.

1. With the flow open on the authoring canvas, right-click the step, and then click **Toggle Nonblocking**.

   An orange lightning bolt appears on the step's icon, and the step automatically acquires a single response, **done** .

2. Connect the **done** response to the next step.

   > **Note:** To change a non-blocking step back into a regular step, right-click the step and click **Toggle Nonblocking** again.

Create a manual step

A manual step is one that offers a choice of actions. The user will need to select an action at runtime.

To create a manual step, you copy the manual operation template from the base content and define the actions that will be made available to the user.

1. In the **Dependencies** pane, select the manual operation template .
2. Select **Edit** > **Copy**.
3. Select the location in the project tree where you want to paste the copy, and select **Edit** > **Paste**. This operation is treated as a new object and is detached from the content pack it arrived with.
4. Drag the operation from the project tree to the authoring pane.

5. In the step, add the actions that will be available to the user.
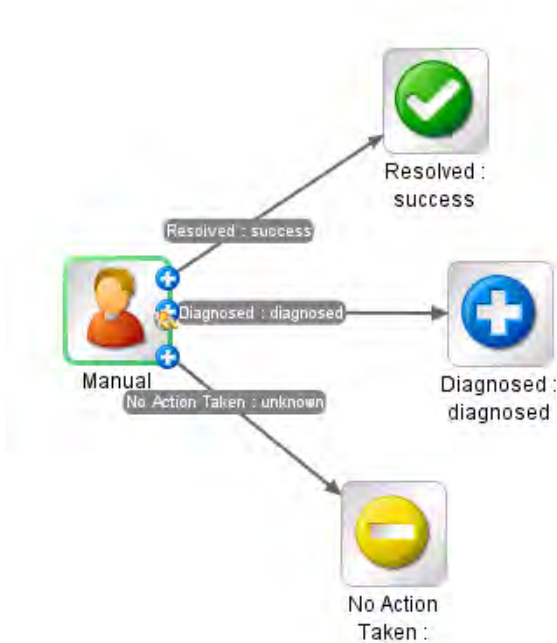
> **Note:** It is also possible to add the actions in the operation properties, rather than in the step. If you do this, you will be able to use the operation in other flows.

Create a display step

A display step is one that displays information in a pop-up prompt message, but does not perform any other action. The user will just need to click **Continue** at runtime.

To create a display step, you copy the display operation template from the base content and define the information that will be displayed to the user.

The prompt message can include variables. For example, to tell the user what time the preceding step concluded, you could include a date/time variable (${dateTime}) in the message.

1. In the **Dependencies** pane, select the display operation template .

2. Select **Edit** > **Copy**.

3. Select the location in the project tree where you want to paste the copy, and select **Edit** > **Paste**. This operation is treated as a new object and is detached from the content pack it arrived with.

4. Drag the operation from the project tree to the authoring pane.

5. Open the Step Inspector for the step and click the **Display** tab.

6. Select the **Always prompt user before executing this step** check box.

7. In the **Prompt Title** box, type the prompt's label (up to 128 characters).

8. In the **Prompt Width** box, type the width of the prompt in pixels.

9. In the **Height** box, the height of the prompt in pixels.

10. In the **Prompt Text** box, type a message to the user.

11. Click **OK**, and save your changes.

> **Note:** It is also possible to add the display information in the operation properties, rather than in the step. If you do this, you will be able to use the operation in other flows.

Reference Material

Step Inspector > Display tab

In the **Display** tab of the Step Inspector, you can create a user prompt that is displayed to the user.

| GUI item	Description
Always prompt user before executing this step	Select the checkbox if you want the prompt window to appear every time this step is run.
Prompt Title	Type the label that will appear in the title bar of the prompt window.
Prompt Width	Type the width of the prompt window in pixels.

Height	Type the height of the prompt window in pixels.
Prompt Text	Type the message that will appear in the body of the prompt window. You can include variables in the message. For example, `${dateTime}`.

Adjusting the Appearance of a Flow

When creating a flow, you can use the **View Options** palette to neaten up the flow and adjust its appearance on the authoring pane.

Display the **View Options** palette by clicking the **View Options** button from the authoring pane toolbar.
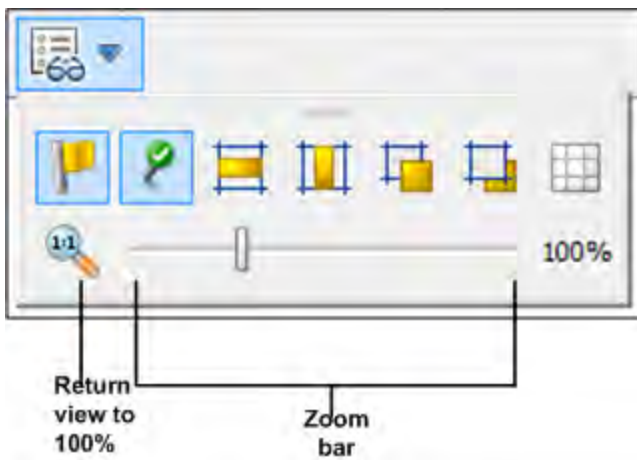
What do you want to do?

Snap steps to the grid

Snapping objects to the grid is a quick way to keep them aligned and neat.

1. If the grid is not visible in the background of the authoring pane, click the **Show/Hide Grid** button in the **View Options** palette. When you drag an operation to the authoring pane, it snaps to the nearest line in the grid.

2. To move a step from one line in the grid to another, move the step slightly and release the mouse.

> **Note:** By default, the grid is not displayed in the authoring pane. If you set the grid to display, using the **Show/Hide Grid** button , this state is not persistent after Studio is closed. To change the

default behavior, so that the grid is displayed by default, you can open the **Studio.properties** file and set the property `dharma.studio.ui.activegrid=true`.

Align steps

1. To align selected steps horizontally, select one or more steps and then select **Align Selection Horizontally** in the **View Options** palette.

2. To align selected steps vertically, select one or more steps and then select **Align Selection Vertically** in the **View Options** palette.

Show or hide response labels and icons

If your flow is looking overcrowded because of response labels and icons on operations, you can choose to hide these.

1. To show or hide response labels, click the **Show/Hide Labels** button to toggle between hiding and showing the response labels.

2. To show or hide the response icons, click the **Show/Hide Connected Response Icons** button to toggle between hiding and showing the response icons.

Move objects to the front or back

Flows may include objects that are stacked together in the authoring pane. This may occur in long flows where there are many items in the flow. In such cases, you need to bring the most important objects to the front of the stack.

1. To move an object to the front of the stack, select it and click **Bring to Front** .

2. To move an object to the back of the stack, select it and click **Send to Back** .

Reference Material

View Options palette

The **View Options** palette contains buttons for changing the appearance of the flow on the authoring pane.

Button	Description
Show/Hide Labels	Shows or hides response labels on objects
Show/Hide Connected Response Icons	Shows or hides response icons on objects
Align Selection Horizontally	Aligns selected steps horizontally
Align Selection Vertically	Aligns selected steps vertically
Bring to Front	Moves the selected object to the front of the stack
Send to Back	Moves the selected object to the back of the stack
Show/Hide Grid	Reveals the authoring pane grid, which you can use for arranging steps. When you stop dragging a step, it snaps to the nearest position on the grid.

Modifying a Flow

After a flow has been created, you can modify the flow. For example, you might copy a flow that you created earlier and adapt it for a slightly different use. Or you might take one of the out-of-the-box flows

provided in Operations Orchestration, such as those in the **Accelerator Packs** folder, and adjust it for your needs.

Best Practices

Always make a copy of a flow before modifying it.

Before making changes to a flow, use **References** > **What uses this?** to check whether other flows use it.

If you are copying a flow and you think that you may need to modify the properties of the operations, it is best to use the **Copy Deep** command. This makes a copy of the operations along with the flow, so that you can modify them without affecting the originals. See "Copying Flows and Operations" on page 652.

If you copy a flow using the **Copy Deep** command, create a new folder for the flow and its operations.

> **Caution:** Make sure that you understand the difference between modifying a step and modifying an operation.
>
> - When you modify the properties of a step (in the Step Inspector), this only affects the individual step.
>
> - When you modify the properties of an operation (in the **Properties** sheet), this affects all the flows that use this operation as a step. You need to be extremely careful about modifying an operation's properties—doing so can break other flows that use it.

What do you want to do?

Open a flow for editing

- Double-click a flow in the **Project** pane to open it for editing in the authoring pane.

- To open multiple flows in the authoring pane, select them using the SHIFT or CONTROL keys, right-click, and select **Open**.

Jump to a step to edit it

If you have a complex flow with many steps, use the **Go to Step** button to quickly jump to the step that you need to edit.

1. Open the flow in the authoring pane.

2. In the **Authoring pane** toolbar, click **Go to Step** .

3. Type the name of the step to jump to it, or the first letters of the step to select it from a list.

4. If desired, use the **Up** and **Down** arrows on your keyboard to navigate through the list of steps.

Change the start step

If you add a new step at the start of a flow, it will appear with a warning icon, because the start step has not been defined.

Right-click the step that you want to use to start the flow and select **Set Start Step**.

Rename a step

1. Right-click the step that you want to rename and select **Rename**.

2. Type the new name in the highlighted field, press the ENTER key, and save your work.

   Best practice: Make sure that the name clearly describes the purpose of the step.

Rename a flow or operation

If you are renaming an operation, check whether this operation is used in other flows. If so, it is better to make a copy of the operation and rename the copy.

1. In the **Project** pane, right-click the flow or operation that you want to rename and select **Rename**.

2. Type the new name in the highlighted field, press the **Enter** key, and save your work.

   Best practice: Make sure that the name clearly describes the purpose of the flow or operation.

Move a group of steps in the flow diagram

1. Hold down the **Shift** or **Ctrl** key to select a group of steps.

2. Click and drag the steps as a group.

Change an icon

You can change the icon on a step, operation, or flow to one that gives a clearer visual cue to what the element does.

1. To open the **Icons** pane, click the **Icons** tab.
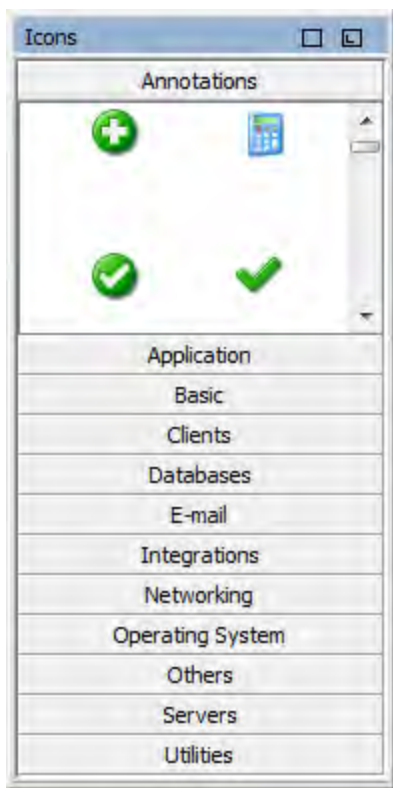
2. Select the icon group name that describes the icon you need.

3. Select the icon, and drag it onto the step, operation, or flow.

   > **Note:** By holding down the **Ctrl** key, and then dragging an icon, you can place the icon on top of an existing icon, in layers.

   Best practice: If you have classified flows according to type, use specific icons for each flow type.

Modify a step in the Step Inspector

1. Right-click a step and select **Properties**. The Step Inspector opens.

2. In the Step Inspector, you can modify the step:

   - Add or edit inputs in the step. For details, see "Creating Input" on page 472.

   - Add or edit results in the step. For details, see "Setting Operation Outputs" on page 515.

   - Add or edit a description of the step. For details, see "Creating Steps in a Flow" on page 458.

- Add or edit a user prompt for the step. For details, see "Display a user prompt for the step " below.

- Add or edit a scriptlet in the step. For details, see "Using Scriptlets in a Flow" on page 574.

> **Tip:** To keep the Inspector open so that you can shift its focus from step to step without having to close and reopen the Inspector, click the **Pin** button ⚐ at the right end of the Inspector's title bar.

Display a user prompt for the step

You can create a user prompt that is displayed before a step is run. The prompt message can include variables. For example, to tell the user what time the preceding step concluded, you could include a date/time variable (`${dateTime}`) in the message.

1. Right-click a step and select **Properties**.

2. Click the **Display** tab in the Step Inspector.

3. Select the **Always prompt user before executing this step** check box.

4. In the **Prompt Title** box, type the prompt's label (up to 128 characters)..

5. In the **Prompt Width** box, type the width of the prompt in pixels.

6. In the **Height** box, the height of the prompt in pixels.

7. In the **Prompt Text** box, type a message to the user.

8. Click **OK**, and save your changes. The step acquires a blue arrowhead, indicating the display prompt.

Change which operation a step is based on

For example, you need an existing flow step to be associated with a different operation, but you want to keep the existing transitions to and from that step.

1. Right-click a step and select **Properties**.

2. In the Step Inspector, click the **Advanced** tab.

3. Under **Source Operation**, click the **Select** button.

4. In the Select Source Operation dialog box, navigate to and select the operation that you want to

base the step on, and then click **OK**.

5. Rename the step to reflect the change in the operation.

6. Review and make any changes necessary to the value assignments for inputs, to reflect any differences between the old operation's inputs and those of the new one.

Reference Material

Step Inspector > Display tab

In the **Display** tab of the Step Inspector, you can create a user prompt that appears before a step is run.

GUI item	Description
Always prompt user before executing this step	Select the check box if you want the prompt window to appear every time this step is run.
Prompt Title	Type the label that will appear in the title bar of the prompt window.
Prompt Width	Type the width of the prompt window in pixels.
Height	Type the height of the prompt window in pixels.
Prompt Text	Type the message that will appear in the body of the prompt window. You can include variables in the message. For example, `${dateTime}`.

Creating Input

Inputs specify how and when the steps in a flow obtain the data that they need. For example, in a **Network Check** flow, the first step pings a server, so it needs the IP address of the server to ping. The IP address is provided via an input.

Each input is mapped to a variable, whose value can be set in the following ways:

- Create a user prompt, so that the value will be entered by the person running the flow, at the start of the flow.

- Set the input's value to a specific, unchanging value.

- Set the value to be obtained from another step.

- Assign a flow variable to the input. A flow variable is of a collection of variables and data values that are available to the entire flow.

You can create an input for a flow, operation, or step.

Caution: Make sure that you understand the difference between modifying a step and modifying an operation.

- When you modify the properties of a step (in the Step Inspector), this only affects the individual step.

- When you modify the properties of an operation (in the **Properties** sheet), this affects all the flows that use this operation as a step. You need to be extremely careful about modifying an operation's properties—doing so can break other flows that use it.

Note:

- An input that appears in red is a step input coming from a required operation input that was deleted for that step.For example:

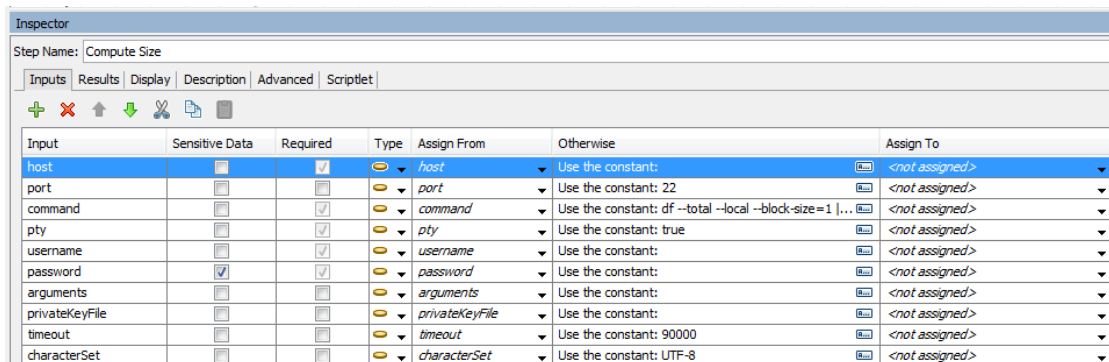| queryValuesList | ☐ | 🟠 ▾ | queryValuesList ▾ | Prompt user fr... 🅰 | queryValuesList ▾ |
| delimiter | ☐ | 🟠 ▾ | delimiter ▾ | Prompt user fr... 🅰 | delimiter ▾ |
| url | ☑ | 🟠 ▾ | url ▾ | Prompt user fr... 🅰 | url ▾ |

- An input that appears in gray is a step input coming from an optional operation input that was deleted for that step.

- You can resize each pane by dragging the divider bar between the two panes. Use the scroll bars at the bottom of each pane to show the unseen fields. The next time you open Studio, the divider bar is in the same position.

- There is always one input highlighted in the Inputs pane. After deleting an input, the first input in the Inputs pane is automatically highlighted.

- You can move columns in the Input pane by dragging and dropping.

- Inputs are processed according to their position in the inputs list, from top to bottom.

What do you want to do?

Create an input

1. Open the **Properties** sheet or Input Inspector:

   - To add an input to an operation, right-click the operation in the **Project** pane and select **Properties**.

   - To add an input to a flow, right-click the flow in the **Project** pane and select **Properties**.

   - To add an input to a step, double-click the step in the authoring pane.

2. Select the **Inputs** tab.

Input	Sensitive Data	Required	Type	Assign From	Otherwise	Assign To
host	☐	☑	⊖ ▾	host ▾	Use the constant:	<not assigned> ▾
port	☐	☐	⊖ ▾	port ▾	Use the constant: 22	<not assigned> ▾
command	☐	☑	⊖ ▾	command ▾	Use the constant: df --total --local --block-size=1 ...	<not assigned> ▾
pty	☐	☑	⊖ ▾	pty ▾	Use the constant: true	<not assigned> ▾
username	☐	☑	⊖ ▾	username ▾	Use the constant:	<not assigned> ▾
password	☑	☑	⊖ ▾	password ▾	Use the constant:	<not assigned> ▾
arguments	☐	☐	⊖ ▾	arguments ▾	Use the constant:	<not assigned> ▾
privateKeyFile	☐	☐	⊖ ▾	privateKeyFile ▾	Use the constant:	<not assigned> ▾
timeout	☐	☐	⊖ ▾	timeout ▾	Use the constant: 90000	<not assigned> ▾
characterSet	☐	☐	⊖ ▾	characterSet ▾	Use the constant: UTF-8	<not assigned> ▾

3. If there are existing inputs, select the row that you want the new input to follow. For example, if you select the first row, the new input will appear on the second row, and will be the second item in the list.

4. Click the ➕ button.

5. Enter the name of the new input and click **OK**. The input appears in a new row.

   > **Note:**
   >
   > - Do not name the input "service" or "sp". Doing so can create errors in flow runs in certain situations.

- The combined input name, flow name, and folder path of the input (under the Library folder) must add up to no more than 255 characters.

- For localized characters, this limit is even lower. For example, in Japanese, the 255 characters need to be divided by 4, so the number of characters is only 64.

6. (Optional) To make the input mandatory for the step to function, select the **Required** check box on the new row.

7. From the **Type** list, specify how the input gets its value:

- **Single Value**

- **List of Values** This enables you to run an operation with an input that contains multiple values. The values must be provided in a list separated by the delimiter specified in the **Input Delimiter** field.

8. Specify the input source in the Input Inspector:

- To assign the value from a variable with the same name as the input, select the default name (marked by the prefix *<default>*) that appears at the top of the **Assign From** list.

- To assign the value from a different flow variable, select the variable name in the **Assign From** list.

- If you do not want to define a source for this input, select *<not assigned>* in the **Assign From** list.

For a detailed description of how to define different input source types, see "Specifying the Input Source " on page 482.

Note: When you create a new input for an operation, flow, or step, the default value for **Assign From/Assign To** is automatically set to **<not-assigned>** when installing Studio. This functionality can be changed using the properties **dharma.studio.ui.inputinspector.assignfrom.selected** or **dharma.studio.ui.inputinspector.assignto.selected** in the **Studio.properties** file.

This default also applies to the inputs of a new step, and to a step that was dragged from **Projects/Dependencies** pane.

For details on the parameters in the Studio.properties file, see "Configuring Studio Properties " on page 671.

9. In the **Otherwise** list, you select the action to occur if the flow variable that you specified in the **Assign from** box does not exist or has no value stored in it. When you click on an input name, the

right pane is automatically updated with its details. Click ![icon] to open the Otherwise dialog box or fill in the fields in the right pane. The options are:

○ **Use Constant**: Enter the constant value that will be used for the input. For example, an IP address that is always used.

○ **Prompt User**: Set up a prompt for the user to supply the information, either by entering information or making a selection from a list, at the start of the flow.

- **Prompt For: Text** - Prompt the user to enter information. Type the prompt that the user will see in the **User Message:** box. For example, `Enter a value for eventid`.

- **Prompt For: Selection** Prompt the user to select a value. Select the type of list the user will select from:

  Selection List: Select the selection list from the **Named** list. All the selection lists are shown.

  Domain Term: Select the domain term from the **Named** list. All the domain terms are shown.

  Flow Variable: Select the name of a flow variable . Flow variables are defined in the results for flows.Type a delimiter for the flow variables in the **Source Delimiter** field.

  > **Note:** It is not recommended to use the **Prompt User** option for a step in the middle of a flow.

○ **Use Previous Step Result**: Select a previous step result to be used if the **Assign From** variable for the input does not exist or has no value.

> For example, you might have a first step that looks for a piece of information, and which saves this information as a flow variable. Then, you might have a second step that displays that information. The second step uses the flow variable that was created in the first step.

○ **System Account**: Select the system account name and credential type.

○ **Logged-in user credentials**: Select the credential type of the logged-in user.

> **Note:** The **System Account** and **Logged-in user credentials** fields do not appear if you define the input as **List of Values** in the **Type** field.

For more details about different types of input source, select the relevant task in "Specifying the Input Source " on page 482.

The default value that appears in this field depends on the settings in the **studio.properties** file. For details on these settings, see the parameter **dharma.studio.ui.inputinspector.assignfrom.selected** in "Configuring Studio Properties " on page 671.

> **Note:** When you create a new input for an operation, flow or step, the default value for **Otherwise** is automatically set to **Use Constant**.

10. Specify the input target in the Input Editor:

    ○ To assign the value to a variable with the same name as the input, select the default name (marked by the prefix *<default>*) that appears at the top of the **Assign To** list.

    ○ To assign the value to a different flow variable, select the variable name in the **Assign To** list.

    ○ If you do not want to define a target for this input, select *<not assigned>* in the **Assign To** list.

11. By default, Studio creates a flow variable with the same name as the input. This variable can be used in later steps in the flow. It is possible to change this name in the **Assign to Variable** field on the right of the window.

    > For example, if you have a step in which the user needs to enter a password, you might want to name the variable `password` to make it easy to identify.

12. (Optional) To validate that the input is in the correct format, select a system evaluator from the **Validation Format** list.

    For example, if the input is an email address, you can use an evaluator to check that the input is in the correct email format. For information about creating evaluators, see "Evaluating Input Data" on page 494.

    For example, if your flow requires users to enter an email address, you can use an evaluator to check that the input is entered with the correct email format. For information about creating evaluators, see "Evaluating Input Data" on page 494

13. (Optional) To record the value, to make it available for diagnostics or auditing, select **<run history>** from the **Record Under** list.

    ○ **<run history>**

    ○ one of the domain items in the list

Rename an input

1. Open the **Properties** sheet (for a flow or operation) or the Input Inspector (for a step).

2. On the **Inputs** tab, double-click on an input name, or press **F2** and type in a new name.

or:

In the right pane, type in a new input name in the **Name** field.

Remove an input

1. Open the **Properties** sheet (for a flow or operation) or the Input Inspector (for a step).

2. On the **Inputs** tab, select the input that you want to remove, and then click ✖ **Remove Input** from the Input Inspector toolbar. A removed step input appears in italics. You can select multiple inputs using the **Ctrl** and **Shift** keys.

Restore a default input that was removed

If you removed a default input from the **Inputs** tab, you can restore it. Default inputs are those that were created as part of the operation on which the step is based. Default inputs that have been removed appear in the list of inputs grayed out and italicized.

1. On the **Inputs** tab, click **Add Input**.

2. Type the exact name of the input that you want to restore.

3. You can edit any of the input's fields as required.

4. Click **OK**.

   The red/gray color is removed and the input is shown in black.

Reference Material

Input Inspector > Toolbar

GUI item	Description
Add input ➕	Adds a new input row. Type the name of the new input and click OK. The new input's attributes are assigned with default values.
Delete input ✖	Deletes the selected input row.
Move up ⬆	Moves the selected input row higher in the list.
Move down ⬇	Moves the selected input row lower in the list.

Cut	Removes the selected input row from its current location. Use the **Paste** option to place the row in a new location.
	Note: If you try to remove an operation input at the step level, Studio automatically makes it appear in italics and moves it to the bottom of the inputs table. Required inputs appear in red, and optional inputs in gray.
	You can restore the input by changing the **Type**, **Assign To/Assign From** or **Otherwise** fields.
Copy	Copies the selected input row. Use the **Paste** option to place the copy in a new location.
Paste	Pastes the copied/cut input row at the current location. If the input name already exists, a message appears and the input is not pasted. (This is the default behavior.)
	If the input name already exists, Studio automatically names the new input **<input> (Copy 1)**.

Note: You can also cut, copy and paste multiple inputs between steps, flows or operations. Select additional inputs by holding down the **Ctrl** key and clicking on the row.

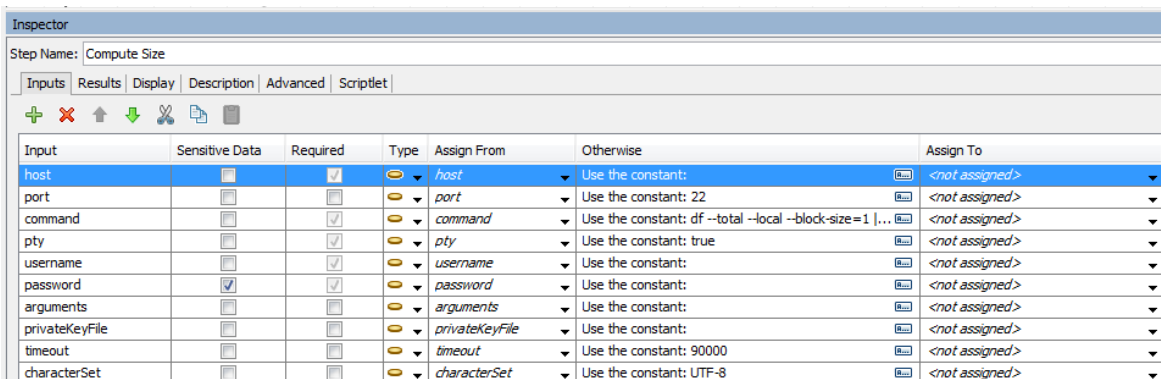## Input Inspector > Inputs tab

The **Inputs** tab in the Step Inspector is where you specify how and when a step in a flow obtains the data that it needs.

GUI item	Description
Input	Displays the name of the input.
Sensitive Data	Makes this input hidden in the Central user interface and in the Studio internal and remote debugger. If this check box is selected, the input value will be encrypted in Central and Studio, will be

	displayed as asterisks and will not be persisted
	This flag is transient, i.e.,assignment from a sensitive variable will also make the assigned variable sensitive. This transitivity is only seen during runtime execution, and is not reflected in the Studio user interface.
	When used in scriptlets, sensitive data is retrieved as encrypted.
	Sensitivity is also retained when using expressions for assignment. For example, ${input1}.
Required	Makes this input mandatory.
Type ⬭ ▾	From the **Type** list, specify how the input gets its value. The choices are: • **Single Value** ⬭ ▾ • **List of Values** 🗄 ▾ to run an operation against multiple targets
Assign From	Specifies where the input gets its value from. From the **Assign From** list, select an input from which to take the value.
Otherwise	Shows what action should occur if the flow variable that you specified in the **Assign From** column does not exist or has no value stored in it. When you click on an input name, the right pane is automatically updated with its details. Click 🔳 to open the **Otherwise** dialog box or fill in the fields in the right pane.
Assign To	Select the flow variable that you want to assign the input's value to.

Note: To edit the **Assign From** and **Assign To** directly, double-click in the field and type in the new value.

Input Inspector > Inputs tab > Otherwise pane/Otherwise dialog box

The **Otherwise** pane appears on the right side of the Input Inspector. In this pane, you define what action should occur if the flow variable that you specified in the Assign From column does not exist or has no value stored in it. Some of these fields also appear in the Otherwise dialog box.

Otherwise Pane
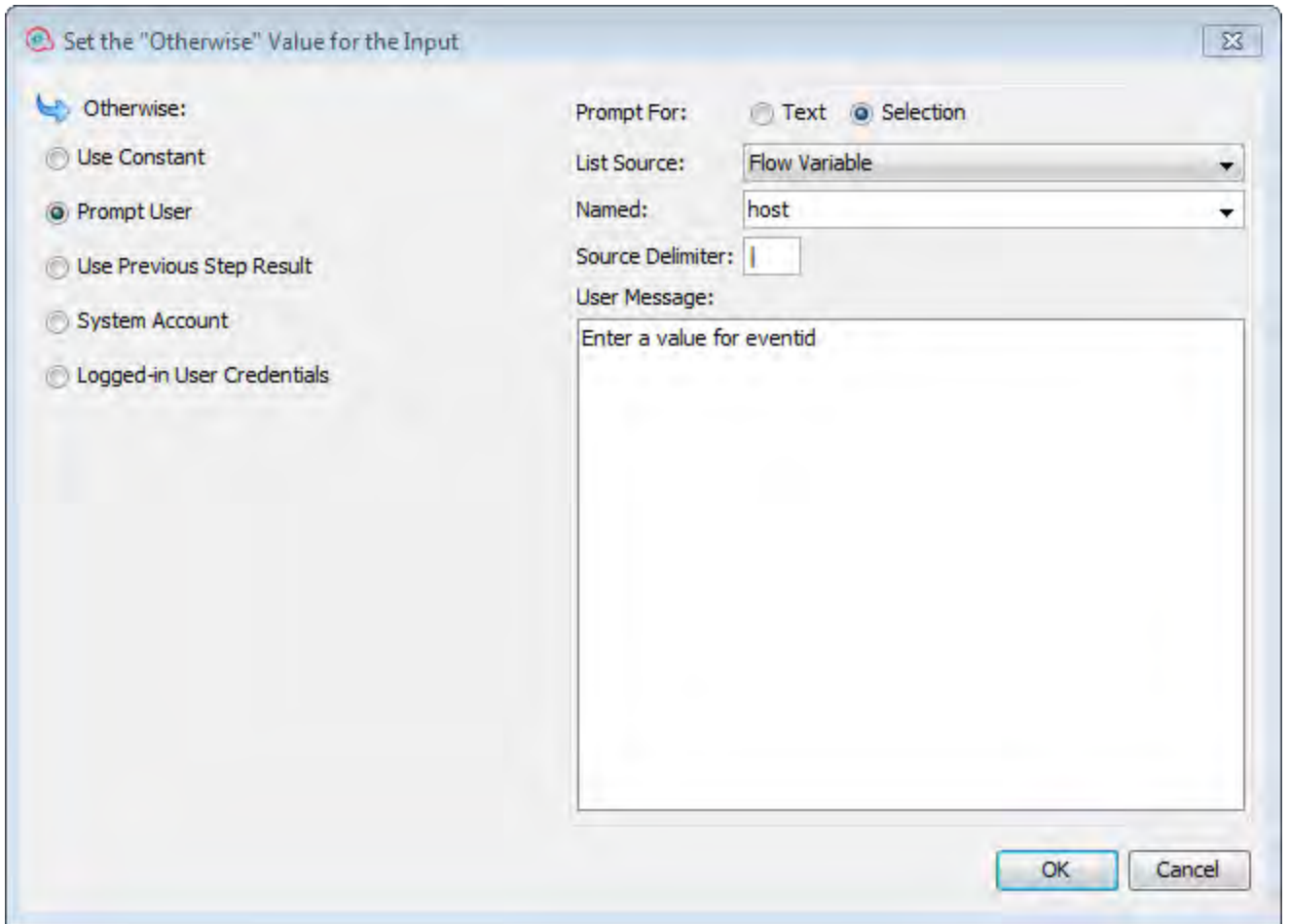
Otherwise Dialog Box

> **Note:** The **System Account** and **Logged-in user credentials** fields do not appear if you define the input as **List of Values** in the **Type** field.

GUI item	Description
Use Constant	Enter the constant value that will be used for the input. For example, an IP address that is always used.
Prompt User	Set up a prompt for the user to supply the information, either by entering information or making a selection from a list, at the start of the flow. • **Prompt For: Text** - Prompt the user to enter information. Type the prompt that the user will see in the **User Message:** box. For example, `Enter a value for eventid`. • **Prompt For: Selection** Prompt the user to select a value.

	Select the type of list the user will select from: **Selection List:** Select the selection list from the **Named** dropdown list. All the selection lists are shown. **Domain Term:** Select the domain term from the **Named** dropdown list. All the domain terms are shown. **Flow Variable:** Select the name of a flow variable . Flow variables are defined in the results for flows.Type a delimiter to the domain terms in the **Source Delimiter** field. **Note:** It is not recommended to use the **Prompt User** option for a step in the middle of a flow.
Use Previous Step Result	Select a previous step result to be used in the case that this input has no value. For example, you might have a first step that looks for a piece of information, and which saves this information as a flow variable. Then, you might have a second step that displays that information. The second step uses the flow variable that was created in the first step.
System Account	Enter the system account name and credential type.
Logged-in user credentials	Enter the user name or password for the logged-in user.
List Source	Select one of the following from the list: • **Selection List** - select from a set of predefined lists. From the **Named** list, select the list you want to present to the user. **Tip:** You can add to the set of predefined lists by creating a list. For information on creating a list, see *Creating selection lists for user prompts*. • **Domain Term** - Domain terms are specialized selection lists. For example, to specify that a flow runs against certain classes of servers, you can add domain terms for the various kinds of servers in your system and create a user prompt in which the user selects the classes of servers that you want to run the flow against. From the **Named** list, select the domain term list you want to present to the user. • **Flow Variable** - create a list that is populated by the contents of a flow variable.

	From the **Named** list, type or select the flow variable that contains the list. You can reference a flow variable from a different flow. For example, to reference a flow under the folder structure **folderA\folderB\flow1**, use the string **${folderA/folder/flow1}**.
	In the **Source Delimiter** box, type the character that separates the elements in the list.
User Message	Type a prompt message to let the user know what kind of data is needed.

Specifying the Input Source

While you are setting up an input in a flow, operation, or step, there are a number of options available for how to specify the input source.

- "Specify the input source as a single constant value "

- "Specify the input source as a single user-entered text "

- "Specify the input source as a single user selection "

- "Specify the input source as a constant list of values"

- "Specify the input source as list of input values obtained from user-typed text "

- "Specify the input source as a single user selection "

- "Specify the input source as the previous step's result "

- "Specify the input source as the system account details "

- "Specify the input source as the user login name or password"

The tasks in this section are optional sub-tasks within the main task of setting up an input. See "Creating Input" on page 472.

What do you want to do?

Specify the input source as a single constant value

Specify the input value source as a static value. For example, a single constant value could be an IP address that is always used in a step.

1. Create an input and set the type to **Single Value**.

2. Click the Otherwise button to open the Otherwise dialog box for that row or fill in the values

directly in the right pane:
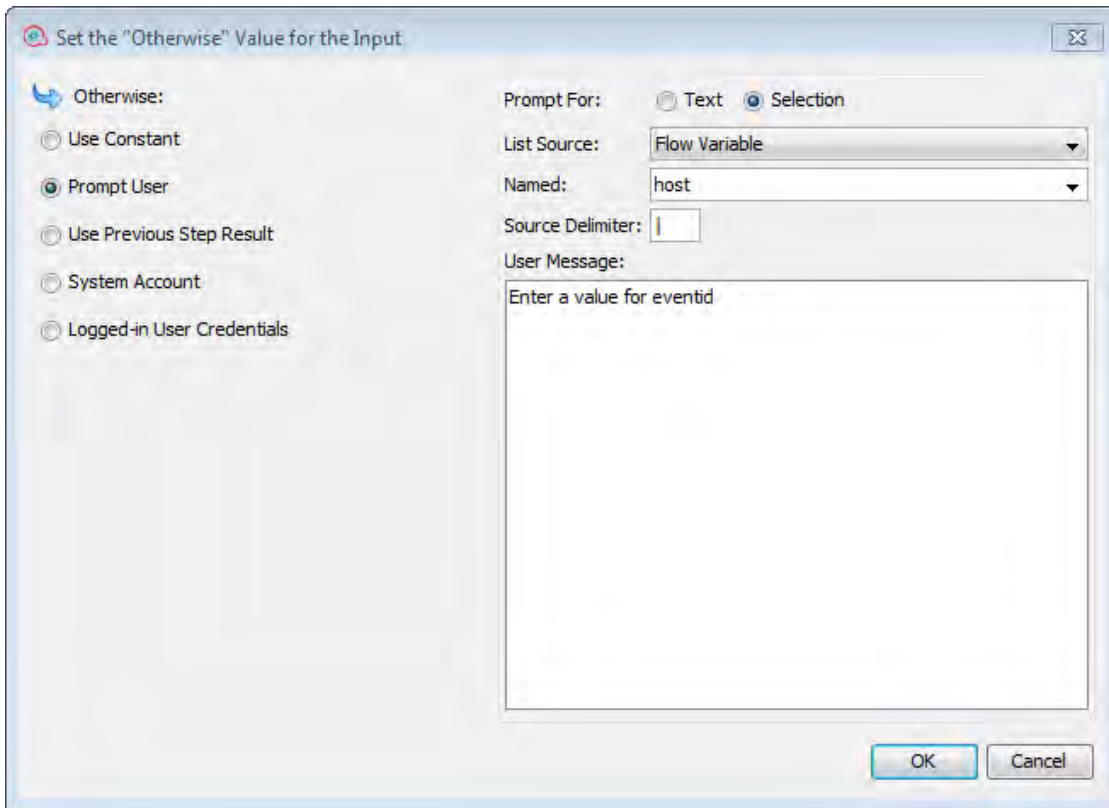
3. By default, *<not assigned>* appears in the **Assign from** field.

   ○ To assign that the input value from the value of a variable of the same name, select *<default> - <name>* from the **Assign from** dropdown menu.

   ○ To assign the input value from the value of a different input, select the input name from the **Assign from** dropdown menu.

   ○ You can also type in a value directly in this field.

4. From the **Otherwise** list, select **Use Constant**. This defines what will occur if the flow variable that you specified in the **Assign from Variable** box does not exist or has no value stored in it. The default value is **Use Constant** when installing Studio, and **Prompt User** when upgrading from a previous version of Studio.

   > **Note:** You can change these defaults in the Editor window under **Configuration > Options**:
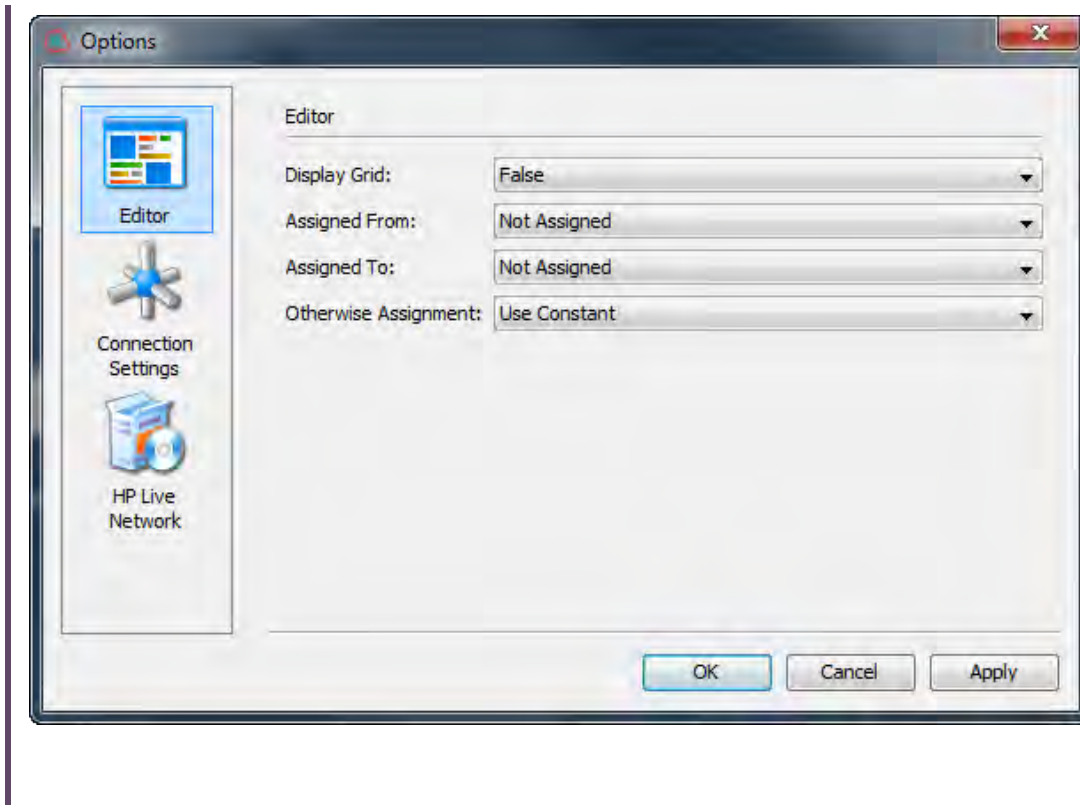
5. In the **'Otherwise: Use Constant' Configuration** section, type the value for the input (for example, `false`). You can also use a combination of text and variable reference, using the following format: `${variablename}`. For example, `Ping of ${targethost" succeeded`. To reference a variable that is stored in a folder use the following format:

`${foldername/variablename}`

Specify the input source as a single user-entered text

Specify the input value source as user-entered text when the user must supply the information necessary for the flow to do its work. For example, you might want the user to specify the IP address of their own server, at the start of the flow.

> **Note:** Do not specify the input source as user-entered text for a step in the middle of a flow.

1. Create an input and select **Single Value** from the **Type** dropdown menu.

2. Click the Otherwise button 🔲 for that row to open the **Otherwise** dialog box and select **Prompt User** or select **Prompt User** in the right pane. The default value is **Use Constant** when installing Studio, and **Prompt User** when upgrading from a previous version of Studio.

   > **Note:** You can change this default by setting the **use.constant.default.value.input** parameter in the **Studio.properties** file.

3. In the **Prompt For:** area, select **Text**.

4. In the **User Message** box, type a prompt message to let the user know what kind of data is needed.

Specify the input source as a single user selection

Another way to get the user to supply the input is to present a list from which the user must make a selection. For example, you might want the user to select from a choice of locations, at the start of the flow.

1. Create an input and select **Single Value** from the **Type** dropdown menu.

2. Click the **Otherwise** button ![icon] for that row to open the **Otherwise** dialog box and select **Prompt User** or select **Prompt User** in the right pane. The default value is **Use Constant** when installing Studio, and **Prompt User** when upgrading from a previous version of Studio.

   > **Note:** You can change this default by setting the **use.constant.default.value.input** parameter in the **Studio.properties** file.

3. In the **Prompt For:** area, select **Selection**.

4. From the **List Source** dropdown menu, select one of the following:

   ○ **Selection List** - select from a set of predefined lists.

   From the **Named** list, select the list you want to present to the user.

   > **Tip:** You can add to the set of predefined lists by creating a list. For information on creating a list, see *Creating selection lists for user prompts*.

   ○ **Domain Term** - Domain terms are specialized selection lists. For example, to specify that a flow runs against certain classes of servers, you can add domain terms for the various kinds of servers in your system and create a user prompt in which the user selects the classes of servers that you want to run the flow against.

   From the **Named** list, select the domain term list you want to present to the user.

   ○ **Flow Variable** - create a list that is populated by the contents of a flow variable.

   From the **Named** list, type or select the flow variable that contains the list. You can reference a flow variable from a different flow. For example, to reference a flow under the folder structure **folderA\folderB\flow1**, use the string **${folderA/folder/flow1}**.

   In the **Source Delimiter** box, type the character that separates the elements in the list.

5. In the **User Message** box, type a prompt message to let the user know what kind of data is needed.

Specify the input source as a constant list of values

This type of input enables you to run a step on multiple targets. For example, to run an operating system health check or install a software update on multiple machines.

1. Create an input and select **List of Values** from the **Type** dropdown menu.

2. Click the **Otherwise** button ⬛ for that row to open the **Otherwise** dialog box and select **Use Constant** or select **Use Constant** in the right pane. The default value is **Use Constant** when installing Studio, and **Prompt User** when upgrading from a previous version of Studio.

   Note: You can change this default by setting the **use.constant.default.value.input** parameter in the Studio.properties file.

3. Un the **Input Delimiter** box, type the character that separates the elements in the list of values.

4. In the **Constant Value** box, do one of the following:

   ○ Type the values for the input, separating between the values with the character that you entered in the **Input Delimiter** box.

   ○ Type one or more flow variable references, using the following format:

      ```
      ${foldername/flowvariablename1}<delimiter>${foldername/flowvariablename2}
      ```

Note: You can include both typed values and variables in the same list. For example, `${folder1/flowvariableA}|${folder1/flowvariableB}|10.2.0.200|18.35.100.7`

In this example, **flowvariableA** contains 220.220.3.9 and **flowvariableB** contains 10.51.110.12 and the delimiter is set to the character "|". If you type the two variable names and type the IP addresses of two others manually into the **Constant Value** box, the operation will run on all four machines: 220.220.3.9, 10.51.110, 1210.2.0.200, and 18.35.100.7.

Specify the input source as list of input values obtained from user-typed text

In this type of input, the user needs to type a list of values, separated by a delimiter. For example, you want the user to enter multiple host addresses for a flow to target, at the start of the flow.

1. Create an input and set the type to **List of Values**.

2. Click the **Otherwise** button ⬛ for that row to open the **Otherwise** dialog box and select **Use Constant** or select **Use Constant** in the right pane. The default value is **Use Constant** when installing Studio, and **Prompt User** when upgrading from a previous version of Studio.

3. In the **Input Delimiter** box, type the character or character sequence that separates the elements in the list.

4. From the **Otherwise** list, select **Prompt User**.

5. In the **Prompt For:** area, select **Text**.

6. In the **User Message** box, type a prompt text that lets the flow user know what kind of data the operation needs.

   > **Note:** Make sure that the prompt text explains to the user how to type a successful list, particularly with attention to the delimiter character (or character sequence) that is required. Note that including a space between list elements when the delimiter character sequence doesn't specify one would cause the operation to fail.

Specify the input source as list of input values obtained from user selections

The user prompt presents the user with a list from which they can select multiple items. For example, the user needs to select a list of machines for the flow to target, at the start of the flow.

1. Create an input and set the type to **List of Values**.

2. Click the **Otherwise** button [ ] for that row to open the **Otherwise** dialog box and select **Prompt User** or select **Prompt User** in the right pane. The default value is **Use Constant** when installing Studio, and **Prompt User** when upgrading from a previous version of Studio.

3. In the **Input Delimiter** box, type the character that separates the elements in the list.

4. In the **Prompt For:** area, select **Selection**.

5. From the **List Source** list, select one of the following:

   ○ **Selection List** - select from a set of predefined lists.

   From the **Named** list, select the list you want to present to the user.

   > **Tip:** You can add to the set of predefined lists by creating a list. For information on creating a list, see *Creating selection lists for user prompts*.

   ○ **Domain Term** - Domain terms are specialized selection lists. For example, to specify that a flow runs against certain classes of servers, you can add domain terms for the various kinds of servers in your system and create a user prompt in which the user selects the classes of servers that you want to run the flow against.

   From the **Named** list, select the domain term list you want to present to the user.

   ○ **Flow Variable** - create a list that is populated by the contents of a flow variable.

   From the **Named** list, type or select the flow variable that contains the list.

   In the **Source Delimiter** box, type the character that separates the elements in the list.

6. In the **User Message** box, type a prompt message to let the user know what kind of data is needed.

Specify the input source as the previous step's result

For example, the previous step might have been to test whether a process works, and the input in the current step is to display the results of that test.

1. Create an input and set the type to either **Single Value** or **List of Values**, depending on whether you want multiple values for the input.

2. Click the **Otherwise** button for that row to open the **Otherwise** dialog box and select **Use Previous Step Result** or select **Use Previous Step Result** directly in the right pane.

3. If the input value comprises more than one value, in the **Input Delimiter** box, type the character that separates the elements in the list.

   Note: If the previous step's result contained multiple items, the input delimiter that you specify must match the delimiter in that result.

Specify the input source as the system account details

If you define the input type as single value, you can use user login credentials as an input source. This lets you enable a flow to perform tasks that require system account credentials.

Specify the input value source as the one of the credentials defined in the user account under which the flow is started (user name or password).

1. Click the **Otherwise** button for that row to open the **Otherwise** dialog box and select **System Account** or select **System Account** directly in the right pane.

2. From the **Account Named** dropdown menu, select the system account to use for the operation's credentials, and choose the property you want to extract from the system account (for example, password/username). This allows the flow to perform tasks that require these account credentials, while protecting the credentials from exposure by keeping them hidden behind the system account name. For more information about system accounts, see "Configuring System Accounts" on page 433.

3. In the Credential Type field, select **Username** or **Password**.

Specify the input source as the user login name or password

You can also use credentials as an input source. This lets you enable a flow to perform tasks that require system account credentials.

Specify the input value source as the one of the credentials defined in the user account under which the flow is started (user name or password).

> **Note:** This input method is for Studio 9.x compatibility only. If you want to define a new input for Studio 10.x , use the **System Account** option described in "Specify the input source as the system account details " on the previous page.

1. Click the **Otherwise** button ![icon] for that row to open the **Otherwise** dialog box.

2. Select **Logged-in User Credentials** or select **Logged-in User Credentials** directly in the right pane.

3. In the Credential Type dropdown, select **User Name** or **Password**.

Specify the input source as the system account details

If you define the input type as single value, you can use user login credentials as an input source. This lets you enable a flow to perform tasks that require system account credentials.

Specify the input value source as the one of the credentials defined in the user account under which the flow is started (user name or password).

1. Click the **Otherwise** button ![icon] for that row to open the **Otherwise** dialog box and select **Use Previous Step Result** or select **System Account** directly in the right pane.

2. From the **Account Named** dropdown menu, select the system account to use for the operation's credentials, and choose the property you want to extract from the system account (for example, password/username). This allows the flow to perform tasks that require these account credentials, while protecting the credentials from exposure by keeping them hidden behind the system account name. For more information about system accounts, see "Configuring System Accounts" on page 433.

3. In the Credential Type field, select **Username** or **Password**.

Reference Material

Input Inspector > Toolbar

GUI item	Description

Add input ➕	Adds a new input row. Type the name of the new input and click OK. The new input's attributes are assigned with default values.
Delete row ✖	Deletes the selected input row.
Move up ⬆	Moves the selected input row higher in the list.
Move down ⬇	Moves the selected input row lower in the list.
Cut ✂ **(or Ctrl + X on the keyboard)**	Removes the selected input row from its current location. Use the **Paste** option to place the row in a new location. **Note:** If you try to remove a required input, Studio automatically marks it as optional and it appears in italics. You can restore the input by marking it as **Required** again.
Copy 🗐 **(or Ctrl + C on the keyboard)**	Copies the selected input row. Use the **Paste** option to place the copy in a new location.
Paste 📋 **(or Ctrl + V on the keyboard)**	Pastes the copied/cut input row at the current location. If the input name already exists, a new input is added with the same name with an addition of `(Copy 1)`. For example, if you copy and past e an input named host, you will have two inputs: `host` and `host (Copy 1)`.

Note: You can also cut, copy and paste multiple inputs between steps, flows or operations. Select additional inputs by holding down the **Ctrl** key and clicking on the row.

Input Inspector > Inputs tab

The **Inputs** tab in the Step Inspector is where you specify how and when a step in a flow obtains the data that it needs.

GUI item	Description
Input	Displays the name of the input.
Sensitive Data	Makes this input hidden in the Central user interface and in the Studio internal and remote debugger. If this check box is selected, the input value will be encrypted in Central and Studio, will be displayed as asterisks and will not be persisted This flag is transient, i.e.,assignment from a sensitive variable will also make the assigned variable sensitive. This transitivity is only seen during runtime execution, and is not reflected in the Studio user interface. When used in scriptlets, sensitive data is retrieved as encrypted. Sensitivity is also retained when using expressions for assignment. For example, ${input1}.
Required	Makes this input mandatory.
Type	From the **Type** list, specify how the input gets its value. The choices are: • **Single Value** • **List of Values** - to run an operation against multiple targets
Assign From	Specifies where the input gets its value from. From the **Assign From** list, select an input from which to take the value or type in a value directly in this field. By default, *<not assigned>* appears in this field. **Note:** In some situations, the flow level input will not appear in Central: • If you have not assigned a value to **Assign From** and you have set **Otherwise** to **Use Constant** • If the value of **Assign From** is different from the flow input name
Otherwise	Shows what action should occur if the flow variable that you specified in the Assign From column does not exist or has no value stored in it. Click to open the Otherwise dialog box or fill in the fields in the right pane.
Assign To	Select the flow variable that you want to assign the input's value to

	or type in a value directly in this field.
	By default, *<not assigned>* appears in this field.

Step Inspector > Inputs tab > Otherwise pane/Otherwise dialog box

The **Otherwise** pane appears on the right side of the Input Inspector. In this pane, you define what action should occur if the flow variable that you specified in the Assign From column does not exist or has no value stored in it. Some of these fields also appear in the Otherwise dialog box. If you make changes in the **Otherwise** pane, the **Otherwise** dialog box is automatically updated accordingly and vice versa.

Otherwise Pane

Otherwise Dialog Box

GUI item	Description
Input column	Displays the name of the input.
Required column	Makes this input mandatory.
Type column	From the **Type** list, specify how the input gets its value. The choices are: • **Single Value** • **List of Values** - to run an operation against multiple targets
Assign From column	Specifies where the input gets its value from. From the **Assign From** list, select an input from which to take the value. **Note:** If you have not assigned a value to **Assign From** and Otherwise is set as Use Constant, the flow level input will not appear in Central. If you have assigned a value to **Assign From** and **Otherwise** is set as **Use Constant**, the flow level input will appear in

	Central.
Otherwise column	Shows what action should occur if the flow variable that you specified in the Assign From column does not exist or has no value stored in it. Click 📖 to open the Otherwise dialog box or fill in the fields in the right pane.
Assign To column	Select the flow variable that you want to assign the input's value to.
Name	Displays the name of the step (read only).
Input Type	Displays the input type. May be modified here.
Assign from Variable	Enter or select the name of the flow variable that will be the source of the input.
Otherwise	Select what will occur if the flow variable specified in the **Assign from Variable** box does not exist or has no value stored in it.
'Otherwise: <action>' Configuration	Configure the details of what happens if the flow variable specified in the **Assign from Variable** box does not exist or has no value stored in it. This section varies, depending on which action is selected in the **Otherwise** list.
Assign to Variable	Select the flow variable that you want to assign the input's value to.
Input Delimiter	Enter the delimiter to be used to separate values in
Required	Makes this input mandatory.
Validation Format	Validates the input's value with a system evaluator. Select a system evaluator from the dropdown list. For example, if you want to prompt the user to provide an IP address, use the IP validation format. Studio will then check that the input provided is in valid IP address format.
Record Under	Makes the value available for diagnostics or auditing. This field is a legacy field from version 9.x. You can select the item under which you want to record the input in the database from the **Domain Terms** list.

Evaluating Input Data

Evaluators are used to validate inputs. For example:

- If the input is an email address, you can use an evaluator to check that the input is in the correct email format.

- If the input must be a numeric value greater than or equal to 1, you can use an evaluator to check that this is the case.

Studio has standard system evaluators for validating the following:

- Alphanumeric values

- Email

- File name

- IP address

- No white space

- Numeric values

- UUID

- Phone number

If there is no standard system evaluators for what you need to validate, you can create additional system evaluators. For information about creating system evaluators, see "Configuring System Evaluators" on page 435.

> **Note:** The default data evaluator for phone numbers supports only the North American telephone number format (1-nnn-nnn-nnnn) for calling from within North America. To validate other regional phone-number formats, you will need to create a system evaluator for this.

Evaluators use the following:

- Simple operators such as `=`, `!=`, `Begins with`, `Contains`, `Match All Words`, and `Match At Least One Word` and so on

- Regular expressions – for more information, see "Using Regular Expressions in a Flow" on page 579

- Scriptlets – for more information, see "Using Scriptlets in a Flow" on page 574

What do you want to do?

Use an evaluator to validate an input

While creating an input in the Input Editor, you can validate the input's value, by selecting an evaluator

from the **Validation Format** list.
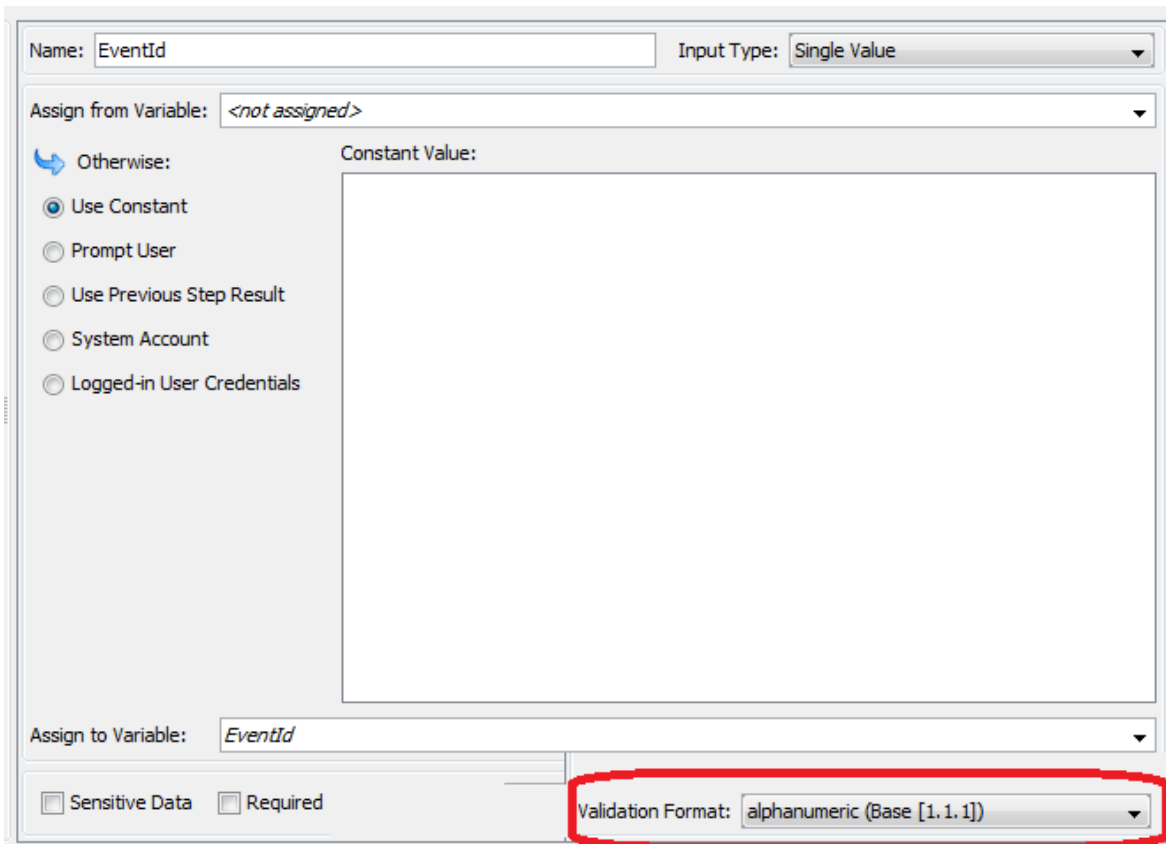
Record Under: Makes the value available for diagnostics or auditing. This functionality is not currently supported.

For more information about creating an input, see "Creating Input" on page 472.

Creating Transitions

You connect any two steps in a flow with a transition. A transition starts from one of a step's responses (represented by a response icon, such as **Success** or **Fail**) and goes to another step. Every response in a flow must have a transition either to a subsequent step or to a return step that returns an outcome for the entire flow and ends the flow.

More than one response can be connected to a given step. For example, several failure responses often are connected to a single failure return step.

For information about setting the responses for an operation, see "Setting Responses" on page 501.

> **Note:** In Operations Orchestration 10.00 and later, transition descriptions are limited to 1000 bytes.

Best Practices

- As much as possible, transition lines should not cross.

- Use straight transitions, if possible. You should only use curved transitions when this is necessary for the flow layout.

- When possible, position steps so that transitions are horizontal, vertical, or at a 45-degree diagonal.

- Collapse multiple transitions from one step to another so that a single line represents all of the transitions.

- Position transition labels so that they do not overlap the step labels or each other.

- Rename transition labels if this will make the flow clearer to another user.

- Place transition labels toward the outside of the flow when possible. For example, if two steps are at the top of the flow canvas, the transition labels should be above the transition lines. If the steps are at the bottom of the canvas the labels should be below the transition lines.

What do you want to do?

Add a transition between steps

1. Open the flow on the authoring pane in Studio.

2. On the step that you want to connect to the next step, click either the response name or the icon that represents one of the responses, and drag to the destination step for that response.

3. Double-click the transition. The Transition Inspector opens.

4. (Optional) To change the transition's name, in the **Name** box, type the new name.

5. In the **Description** box, type a description that explains what happened in the preceding step that caused this transition to be followed. This description will appear in the **Results Summary** area in Operations Orchestration Central, when the flow is run.

> **Note:** A transition's description relates to the step from which the transition originated. For example, the message "localhost successfully pinged" describes what happened in the step "Ping Target System", even though it was written for the transition that follows the step.

Create a description that includes a flow variable

You can use flow variables in the description to store changeable information. For example, to identify a server whose name is stored in the servername flow variable, you could type: "Server ${servername} is available for connection".

1. Create a transition between two steps.

2. Double-click the transition to open the Transition Inspector.

3. In the **Description** box, type a description that includes a flow variable that contains data that came from the step's operation or elsewhere in the flow run. The reference must use the format `${flow variable name}`.

> For example, a step that carries out a ping command can save the host machine's name into a flow variable called `host`. To use this value in the transition description you can reference it with the syntax `${host}`. The description in the transition from the success response might read "Successfully pinged ${host}." When this is run in Central against a host named "server1", the summary description will read "Successfully pinged server1."

Limit who can run the step beyond the transition (gated transition)

Gated transitions let you control who can continue with the flow beyond the transition, by restricting access to the next step to users who belong to a particular role. If someone who is not a member of this role group tries to run the flow, the flow will stop, and the user will have the choice of handing off the flow to another user or canceling the flow.

Gated transitions are colored red.

1. Create a transition between two steps.

2. Double-click the transition to open the Transition Inspector.

3. Select the **Check user's groups before proceeding** check box.

4. From the **Required Role Alias** list, select the role that the user must be assigned in order to continue running the flow.

Require that the run is handed off following the transition

You can set a transition to hand off the flow to another person. This might be necessary if the next step requires information from a different user.

During the flow run, a hand-off transition opens a new email message with the URL of the flow included in the body of the message. The person running the flow can address the email message to the person taking over the flow and then send the message. After the recipient receives the message, they can continue running the flow.

1. Create a transition between two steps.

2. Double-click the transition to open the Transition Inspector.

3. Select the **Hand-off flow after this transition** check box.

Count the completion of the transition in the flow's ROI value

You can associate a value with a transition in the flow. These values represent the return on investment (ROI) value of each transition. When the flow is executed, these values are recorded, based on the transitions actually made. Administrators will be able to view reports, in Central, which display the ROI values of the flow, giving it valuable business statistics.

1. Create a transition between two steps.

2. Double-click the transition to open the Transition Inspector.

3. In the **Transition ROI Value** box, enter a numerical value for the transition.

Add a curve-defining point to create a curved transition

You can add a curve-defining point to reshape a transition from a straight line to a curve. This helps to tidy up your flow or to separate transitions that are stacked.
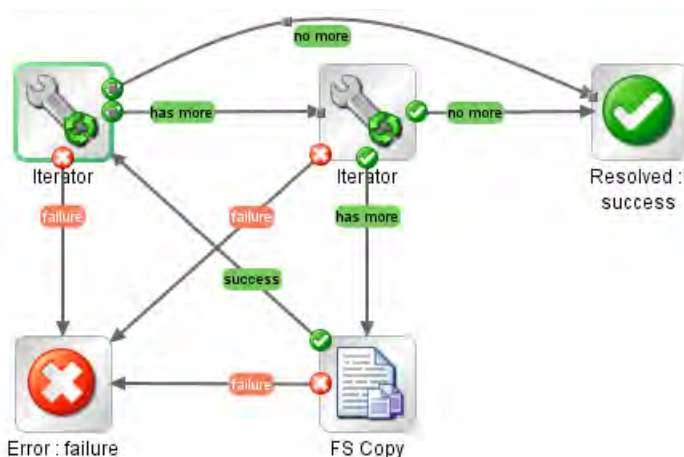
1. Position your mouse over the transition where you want to place the curve-defining point.

2. To create the point, hold down SHIFT and click the mouse.

3. Drag the point until the transition curves the way you want it to.

Remove a curve-defining point

Position the cursor over the curve-defining point, hold down SHIFT, and click the mouse.

Move a transition name

Click the transition name and drag it to another location.

Break a transition between two steps

To break an existing transition between two steps, select the transition and press the Delete key on the keyboard.

Reference Material

Transition Inspector

The Transition Inspector is where you specify the details of a transition.

GUI item	Description
Transition Name	By default, the transition name is the same as the name of the response where it originated (success, failure, and so on), but you can change the transition name.
Required Role Alias	Select the role alias that the user must be assigned to in order to continue running the flow.
Hand-off flow after this transition	Hand-off transitions are not supported in the current version. Select this check box to hand off the flow to another person after the transition.
Transition ROI Value	Enter a value for the transition, so that if the transition is followed during a flow run, its value is added to the value of the flow for that run.
Description	Type a description that explains what happened in the preceding

	step that caused this transition to be followed. The description appears in the **Results Summary** area inOperations Orchestration Central.

Setting Responses

A response is one of a number of possible outcomes of an operation or flow.

The four types of response are:

- **Resolved** – This is the standard response for an operation or flow that runs correctly.

- **Diagnosed** – This response indicates that an operation or flow has determined what a problem is and has opted not to take action on it other than notification.

- **No Action Taken** – This response is used when an operation or flow gathers data but cannot determine any diagnosis or remediation.

  > **Note:** An operation that is intended solely to gather data should return **Resolved** when it is complete, rather than **No Action Taken** .

- **Error** – This response is used if the step or flow fails to run. For example, because of bad input or failure to reach a system.

In some cases, an operation or flow may have multiple responses of the same type. For example, an SQL query operation may have the following outcomes:

- **More items**

- **No more items**

- **Failure**

You can add, delete, and modify the responses in an operation or flow. You cannot modify the responses on a step, with the exception of return steps. For more information about return steps, see "Creating Return Steps" on page 553.

Response Rules

A rule enables you to limit the response so that it only occurs when a particular condition of the operation's result is true. The rule compares a value that you specify with a value in a field of an operation's raw results.

> For example, you can create a rule that will only give a **Success** response if the results include a value that is greater than 1.

If you create more than one rule for a response, then all the rules for that response must evaluate to true for the response to be chosen.

Responses are evaluated in the order in which they are listed on the operation's **Responses** tab. The first response whose rule or rules evaluate to true is the response chosen. So if the port open response's rule evaluates to true, then that is the response chosen, even if the rule for port listening would also evaluate to true. The order of responses can be very important for obtaining the most helpful outcome for your flow.

What do you want to do?

Add a response to an operation or flow

1. Click the **Responses** tab of the operation/flow.

2. Select the row that you want the new response to follow. For example, if you select the first row, the new response will appear on the second row.

3. Click **Add Response** and then type a name for the new response.

4. To make a response the one chosen if an operation fails to execute, select the check box in the **On-Fail** column for that response.

5. To identify a response as the default response, select the check box in the **Default** column. The default response is the one chosen if none of the responses' rules evaluate to true.

6. In the **Type** column, select the type of response:

   ○ **Resolved**: ✅

   ○ **Diagnosed**: ➕

   ○ **No action**: ➖

   ○ **Failure**: ✖

   This determines which response icon will be shown on the operation, when it is used to create a step.
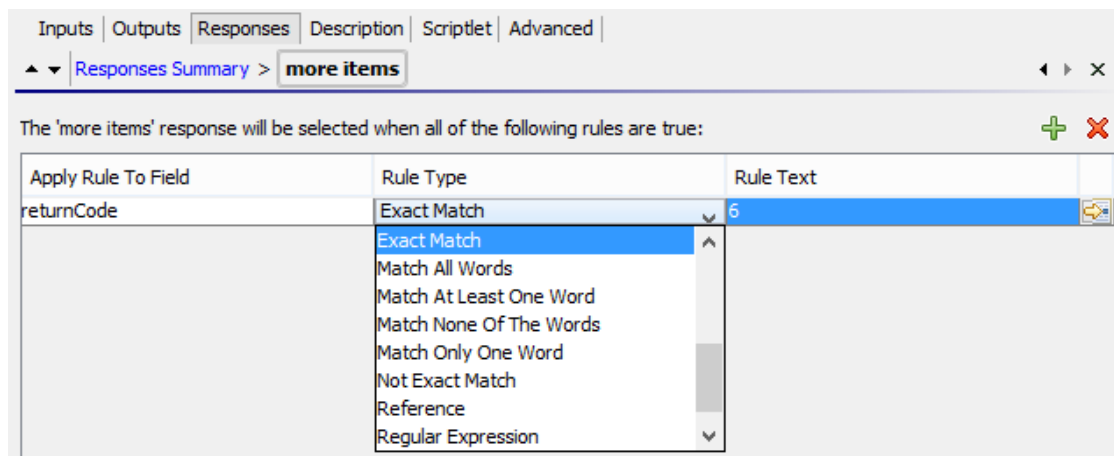
7. To create a rule for the response, at the right end of the response's row, click the right-pointing arrow 📇. For more information, see *Create a rule for the response*, below.

Create a rule for the response

A rule enables you to limit the response so that it only occurs when a particular condition of the result is true.

1. Create a new response in an operation.

2. In the response rule editor, click **Add**.

   The new rule appears.

3. In the **Apply Rule to Field** column, select the results field whose value you want to test a rule against.

   The result fields you can test include the result's exit code, output string, error string, failure message, and timed-out true or false.

   > **Note:** To display more information on these result fields, click the **Description** tab.

> To see what the values for these fields are, test the operation in a flow, using the Studio Debugger. In the Debugger, when you execute a flow, you will find the results for any step in the Step Result Inspector. For more on the Debugger, see "Testing and Debugging a Flow" on page 595.

4. In the **Rule Type** column, select the comparison or match that you want to test the field value with.
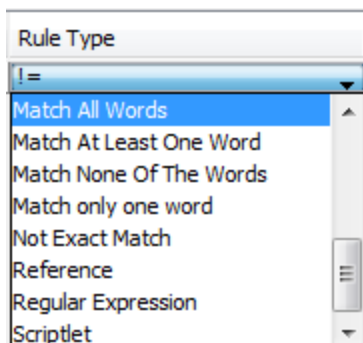
- Select a simple operator such as =, !=, Begins with, Contains, Match All Words, and Match At Least One Word and so on

- Select **Regular Expression** to create a regular expression.

- Select **Scriptlet** to create a scriptlet.

- Select **Reference** to create a reference to a shared rule.
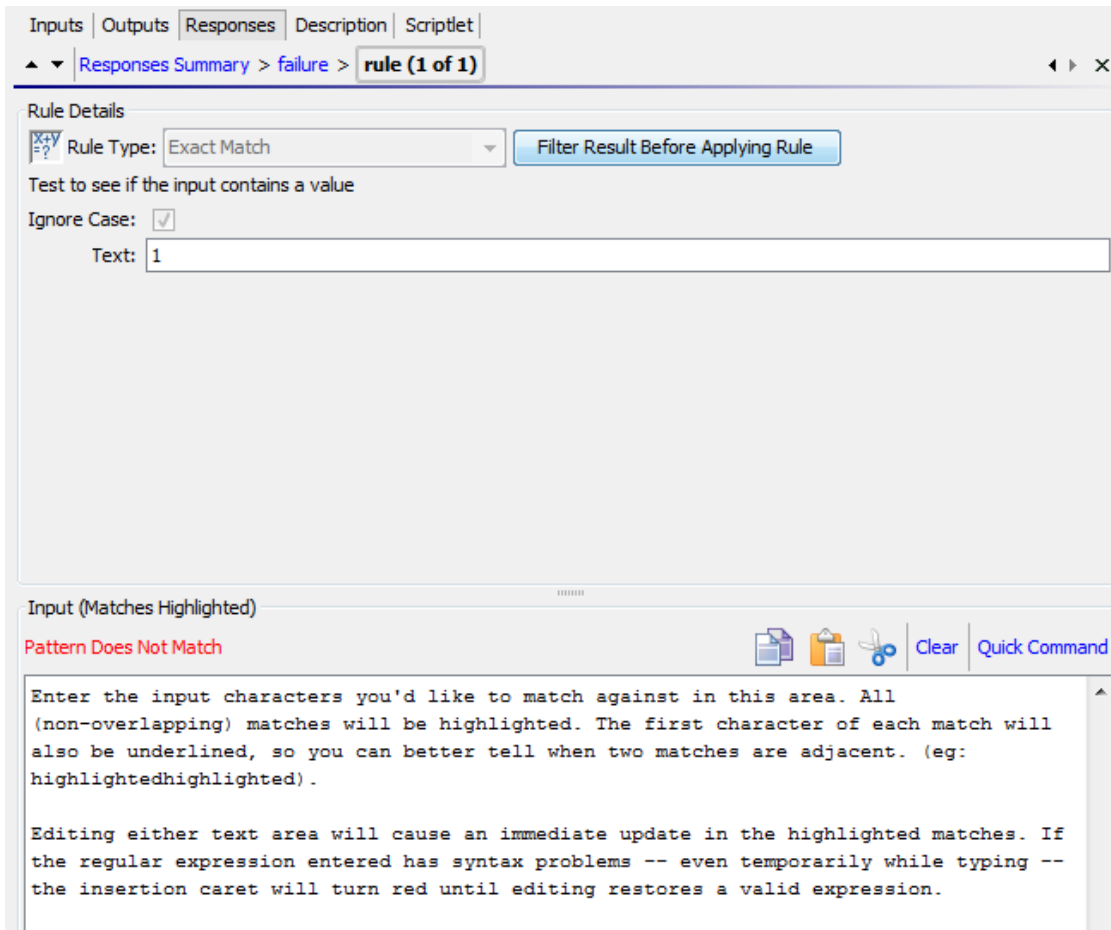
5. In the **Rule Text** column, type the text to use in the test.

Filter and test a response rule

In the Rule Details Editor, you can:

- Specify the rule in greater detail, including the use of rules, filters, regular expressions, or scriptlets.

- Test the rule as you develop it

1. To open the Rule Details Editor, click the right-pointing arrow 🔲 at the right end of the rule's row.

   Note: If you choose **Scriptlet** as the rule type, the Rule Details Editor includes a scriptlet editor. For more information on creating scriptlets, see "Using Scriptlets in a Flow" on page 574.

   If you choose **Regular expression** as the rule type, the Rule Details Editor includes a regular expression editor. For more information on creating regular expressions, see "Using Regular Expressions in a Flow" on page 579.

2. To use a different rule type, select it from the **Rule Type** list.

3. To filter the result before applying the rule, click **Filter Result Before Applying Rule** and, in the Filter Editor, create the filter.

   Creating a filter for a response rule is the same as creating a filter for an output or result. See "Filtering Output and Results" on page 529.

4. For most of the rule types, in the **Text** box, type the text that you want to test comparison with and,

if you want to ignore case, select the **Ignore Case** check box.

For **Regular Expression** rules, specify the regular expression and its application as you do when creating a **Regular Expression** filter for operation results. For information, "Using Regular Expressions in a Flow" on page 579.

The results box displays the results of the test: it displays either **Pattern Matches** or **Pattern Does Not Match** and highlights the matching text.

5. To work on another rule for the operation's response, click the up or down arrow next to **Responses Summary**.

Note: In a rule, when you use a mathematical comparator (such as =, !=, <, or >) in an evaluation of a string that starts with a number, the comparator compares only the numerical portion of the string. For example, if you compare "123" with "123Test" using != (does not equal), the evaluation would be "false", although "123" is clearly not the same as "123Test". You can work around this issue, however, by comparing the strings with the **Not Exact Match** evaluator.

Add a response to a flow

When you create responses for a flow, you make these responses available for return steps in the flow.

For example, if the outcome leading to an **Error** return step is not a failure in an operation but a result that did not meet a required threshold, then you might want to create a new response for the **Error** return step, which reflects this outcome, so that it will appear as **Error: threshold not met**.

1. Open the **Properties** sheet for the flow.

2. Click the **Responses** tab.

3. Click **Add Response** and then, in the text box that appears, type the name of the response. For example, threshold not met.

4. Click **OK**.

When you create an **Error** return step for the flow, you will be able to select **threshold not met** as a response. For more information about return steps, see "Creating Return Steps" on page 553.

Delete a response from an operation or flow

1. Open the **Properties** sheet for the operation or flow.

2. Click the **Responses** tab.

3. Select the response, and then click **Remove Response**.

Reference Material

Flow Properties sheet > Responses tab

The **Responses** tab in a flow's **Properties** sheet is where you can see the possible responses to be available for return steps in a flow.

GUI item	Description
✚ (Add Response)	Adds a new response row.
✖ (Remove Response)	Removes the selected response row.
⬆ ⬇	Click to move the selected response up or down in the list.
Type	The type of response (read-only): • Success / resolved: ✅ • Diagnosed: ➕ • No action: ➖ • Failure: ❌

Operation Properties sheet > Responses tab

The **Responses** tab in an operation's **Properties** sheet is where you specify the possible responses for an operation.

GUI item	Description
Add Response	Adds a new response row.
Remove Response	Removes the selected response row.
⬆ ⬇	Click to move the selected response up or down in the list.
Default	Select to identify a response as the default response. The default response is the one chosen if none of the responses' rules evaluate to true.
On-Fail	Select to make a response the one chosen if an operation fails to execute.
Type	Select the type of response: • Success / resolved: ✅ • Diagnosed: ➕ • No action: ➖ • Failure: ❌
Rules	Displays any rules that have been created for the response.
🔲	Click to display the Rule Editor, to create a rule for the response.

Operation Properties sheet > Responses tab > Rule Editor

The Rule Editor is where you limit the response so that it only occurs when a particular condition of the result is true.

For example, you can create a rule that will only give a **Success** response if the results include a value that is greater than 1.

GUI item	Description
Add Response	Adds a new response row.
Remove Response	Removes the selected response row.
⬆ ⬇	Click to move the selected response up or down in the list.
Apply Rule to Field	Select the results field whose value you want to test a rule against. The result fields you can test include the result's exit code, output string, error string, failure message, and timed-out true or false.
Rule Type	Select the comparison or match that you want to test the field value with.
Rule Text	Type the text to use in the test.
🔲	Click to open the Rule Details Editor, to test and filter the rule.

Operation Properties sheet > Responses tab > Rule Editor > Rule Details Editor

The Rule Details Editor is where you can test and apply filters to a rule.

Rules enable you to you limit a response so that it only occurs when a particular condition of the result is true.

GUI item	Description
Rule Type	Displays the rule type that was selected in the Rule Editor and enables you to select a different rule type.
Filter Result Before Applying Rule	Click to display the Filter Editor, in order to filter the result before applying the rule.
Text	Type the text that you want to test comparison with.
Ignore Case	Select to ignore whether the text is in upper or lower case.
Results box	Displays the results of the test: Displays either **Pattern Matches** or **Pattern Does Not Match** and highlights the text that matches.
Copy	Copy data within the results box.
Paste	Paste data into the results box.

Cut	Cut data within the results box.
Clear	Clear data within the results box.
Quick Command	Type a command that generates the data on which you want to test the filter. The output of the command appears in the results box.A

Flow Properties sheet > Responses tab

The **Responses** tab in an flow's **Properties** sheet is where you specify the possible responses to be available for return steps in a flow. For example, **Error: threshold not met**.

GUI item	Description
Add Response	Adds a new response row.
Remove Response	Removes the selected response row.
⬆ ⬇	Click to move the selected response up or down in the list.

Operation Properties sheet > Responses tab

The **Responses** tab in an operation's **Properties** sheet is where you specify the possible responses for an operation.

GUI item	Description
Add Response	Adds a new response row.

Remove Response	Removes the selected response row.
⬆ ⬇	Click to move the selected response up or down in the list.
Default	Select to identify a response as the default response. The default response is the one chosen if none of the responses' rules evaluate to true.
On-Fail	Select to make a response the one chosen if an operation fails to execute.
Type	Select the type of response: • Success / resolved: ✅ • Diagnosed: ➕ • No action: ⛔ • Failure: ❌
Rules	Displays any rules that have been created for the response.
🔲	Click to display the Rule Editor, to create a rule for the response.

Operation Properties sheet > Responses tab > Rule Editor

The Rule Editor is where you limit the response so that it only occurs when a particular condition of the result is true.

For example, you can create a rule that will only give a **Success** response if the results include a value that is greater than 1.

GUI item	Description

Add Response	Adds a new response row.
Remove Response	Removes the selected response row.
⬆ ⬇	Click to move the selected response up or down in the list.
Apply Rule to Field	Select the results field whose value you want to test a rule against. The result fields you can test include the result's exit code, output string, error string, failure message, and timed-out true or false.
Rule Type	Select the comparison or match that you want to test the field value with.
Rule Text	Type the text to use in the test.
	Click to open the Rule Details Editor, to test and filter the rule.

Operation Properties sheet > Responses tab > Rule Editor > Rule Details Editor

The Rule Details Editor is where you can test and apply filters to a rule.

Rules enable you to you limit a response so that it only occurs when a particular condition of the result is true.

Inputs | Outputs | **Responses** | Description | Scriptlet

▲ ▼ | Responses Summary > failure > **rule (1 of 1)** ◄ ► ✕

Rule Details

[X+Y =?] Rule Type: [Exact Match ▼] [Filter Result Before Applying Rule]

Test to see if the input contains a value

Ignore Case: ☑

Text: [1]

Input (Matches Highlighted)

Pattern Does Not Match

📋 📋 ✂ | Clear | Quick Command

```
Enter the input characters you'd like to match against in this area. All
(non-overlapping) matches will be highlighted. The first character of each match will
also be underlined, so you can better tell when two matches are adjacent. (eg:
highlightedhighlighted).

Editing either text area will cause an immediate update in the highlighted matches. If
the regular expression entered has syntax problems -- even temporarily while typing --
the insertion caret will turn red until editing restores a valid expression.
```

GUI item	Description
Rule Type	Displays the rule type that was selected in the Rule Editor and enables you to select a different rule type.
Filter Result Before Applying Rule	Click to display the Filter Editor, in order to filter the result before applying the rule.
Text	Type the text that you want to test comparison with.
Ignore Case	Select to ignore whether the text is in upper or lower case.
Results box	Displays the results of the test: Displays either **Pattern Matches** or **Pattern Does Not Match** and highlights the text that matches.
Copy	Copy data within the results box.
Paste	Paste data into the results box.

Cut	Cut data within the results box.
Clear	Clear data within the results box.
Quick Command	Type a command that generates the data on which you want to test the filter. The output of the command appears in the results box.

Creating Outputs and Results

One of the ways to capture data for use in a flow is via a step result. There are two ways to assign this data:

- When the output in a result is assigned to a **_flow variable_**, you can pass it as data to other steps in the flow.

- When the output in a result is assigned to a **_flow output field_**, you can pass it as data to a parent flow.

There are a number of steps in this process:

1. Set up the outputs for an operation, including the primary output.

   See "Setting Operation Outputs" below.

2. When you use the operation for a step in a flow, you decide which of the operation outputs you want to use as step results—which ones you want to assign to flow variables or flow output fields.

   See "Setting Step Results" on page 520.

3. (Optional) It is possible to narrow an output or result to a more highly focused selection by creating filters.

   See "Filtering Output and Results" on page 529.

Setting Operation Outputs

The first stage in setting up flow outputs is to set up the outputs in the operation. After this is done, when you (and other flow authors) use this operation in a flow, you will be able to assign outputs to flow variables.

If an operation contains sensitive data that you would like to hide from the end-user, you can define this at the step level. See "Setting Step Results" on page 520.

Types of Operation Output

The different kinds of operation output include:

- **Raw result** is *all* of the operation's return code, data output, and error string.

  The raw output is not directly visible in Studio, except as the raw result of a step that was created from the operation.

- The primary and other outputs are portions of the raw output—for example, success code, output string, error string, or failure message—that you specify as an output.

  - The **primary output** is the output used to populate the step's primary result. The primary output supplies a value to an input whose assignment is **Previous Step's Result**. You can see the primary output in the Outputs Summary above the list of outputs in the Outputs tab.

  - A **secondary output** in an operation is another output, in addition to the primary output.

> **Tip:** You can narrow an output to a more focused selection by creating one or more filters for the output. See "Filtering Output and Results" on page 529.

Examples of Operation Output

Most operations have outputs that are specific to the operation. However, you will frequently encounter the following outputs when working with operations in the Library's **Accelerator Packs**, **Integrations**, and **Operations** folders:

- **returnResult**

  When you see "returns:" with no field named, this is usually the primary output. The primary output is also accessible via **Result** with a capital R (which is universal).

- **response** (or **returnCode**)

  A code or string used to determine the response the operation will take.

- **failureMessage**

  An internal output provided by the infrastructure. If an operation returns a failure, this output provides the exception. Note that many operations do not use this output.

What do you want to do?

Specify the primary output for an operation

When you set up an operation, you can specify its primary output. Once you have created a primary output, you can change its source, but you cannot return to having no primary output.

Note that this can only be done at step level.

1. Right-click the operation in the **Project** pane and select **Properties**.

   > **Note:** To open multiple operations, select them using the SHIFT or CONTROL keys, right-click, and select **Open**.

2. Select the **Output** tab.

3. From the **Extract Primary Output from Field** list, select a source field. For example, **FailureMessage**.

   > **Tip:** For information on the data provided in each output field, click the operation's **Description** tab.

Add a secondary output for an operation

A secondary output in an operation is another output, in addition to the primary output.

1. Right-click the operation in the **Project** pane and select **Properties**.

2. Select the **Output** tab.

3. Click **Add Output**.

4. Type the name of the output.

5. From the **Output Field** list, select the field from which the output gets its data.

6. To create filters for the output data in the secondary output, click the right-pointing arrow at the end of the row.

   For information on creating filters, see "Filtering Output and Results" on page 529.

Change the field from which an output gets its data

1. Open the **Properties** sheet for the operation and select the **Output** tab.

2. To change the field for the primary output, click the downward-pointing arrow to the right of the **Extract Primary Output From Field** box, and then select the desired field from the list.

3. To change the field for a secondary output, click in the **Output Field** column of the output's row, and then select the desired field from the list.

Delete an output from an operation

1. Right-click the operation in the **Project** pane and select **Properties**.

2. Select the **Outputs** tab.

3. Select the output you want to delete and click ..

Cut, copy and paste an output

1. Open the **Properties** sheet for the operation and select the **Outputs** tab.

2. Select an output to cut/copy.

3. From the **Outputs** toolbar, click or .

4. Move to the row above which you want to paste the result.

5. Click .

6. Save the operation.

> **Note:**
>
> - You can copy outputs to other operations.
>
> - You can paste outputs with same name. Studio automatically names the new output **<output> (Copy 1)**.

Reference Material

Properties sheet > Outputs tab

The **Outputs** tab in the **Properties** sheet is where you specify the primary and secondary outputs in an operation.
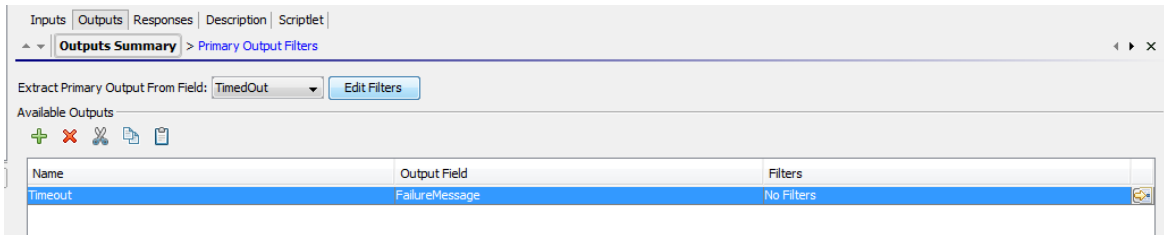
GUI item	Description
Extract Primary Output From Field	Select the field from which the primary output will get its data
Edit Filters	Displays the Filter Editor for the primary output
Add Output	Adds a new output row
Remove Output	Removes the selected output row
Output Field	Select the field from which this secondary output will get its data
	Displays the Filter Editor for the output in the row

Properties sheet > Outputs Toolbar

GUI item	Description
Add output	Adds a new output row. Type the name of the new output and click OK. The new output's attributes are assigned with default values.
Delete output	Deletes the selected output row.
Cut **(or Ctrl + X on the keyboard)**	Removes the selected output row from its current location. Use the option to place the row in a new location.
Copy **(or Ctrl + C on the keyboard)**	Copies the selected output row. Use the option to place the copy in a new location.
Paste **(or Ctrl + V on the keyboard)**	Pastes the copied/cut output row at the current location. If the output name already exists, Studio automatically names the new output **<output> (Copy 1)**.

Setting Step Results

Operations produce a variety of outputs, but outputs are not automatically retained in the flow. If they were, this could affect performance, by slowing down the flow with unnecessary data.

In the **Results** tab of the Step Inspector, you specify the results that you need and you can mark the operation's raw results as sensitive. Results contain output from the operation and additional information that you want to propagate.

You can also mark a result as sensitive data.

You can store results in two ways:

- Create **flow variables**, which are accessible to operations, transitions, and prompts *in the same flow*. For more information, see "Working with Variables" on page 547.

  > **Example**: A step called **LocalPing** determines whether a target host is available and stores the output of the ping operation in a result called **PingOutput**. This creates a flow variable called **PingOutput**, which can be used in later steps.
  >
  > The next step, called **Display**, displays the **PingOutput** variable to the user. The prompt text in this step is set up as `Ping Results: {PingOutput}`.

- Create **output fields**, which are accessible to operations, transitions, and prompts *in the parent flow*, if the flow is used as subflow (a step in another flow). For more information, see "Creating a Subflow Within a Flow" on page 557.

  > **Example**: A parent flow includes a step that contains the **Windows Health Check** flow as a subflow. The results of the **Windows Health Check** flow are stored as value in an output field called **HealthCheckOutput** and are available for the main flow.
  >
  > The main flow includes a **Send Mail** operation, which displays the value of the **HealthCheckOutput** output field in the body of the email.

There are two kinds of step result:

- **Raw result** is *all* the raw data that was returned from an operation executed in the context of a flow. The step's raw and primary results come from the underlying operation's raw output and primary output.

- Other results, which you create on the **Results** tab of the Step Editor. In the Step Inspector, you can create and specify secondary results.

Before setting up the flow results in a step, make sure that the primary output has been set up for the relevant operation. See "Setting Operation Outputs" on page 515.

For security purposes, there may be cases in which you need to hide the results of an operation. For example, an operation may generate a random password, and you want to hide this password from the end user. You can specify this in the Sensitive Data area. You can apply this functionality on an individual step, not only on an entire operation.

> **Tip:** You can narrow a result to a more highly focused selection by creating one or more filters for the output. See "Filtering Output and Results" on page 529.

What do you want to do?

Create a primary result in a step

The primary output is set in the operation. The primary output supplies a value to an input whose assignment is **Previous Step's Result**.

In a step, you can decide to capture this primary output in a flow variable (to be used in other steps in the flow) or a flow output field (to be passed to a parent flow).

1. Double-click a step in the authoring pane.

2. Select the **Results** tab and click **Add Result**.

3. In the **Name** column, enter a name for the result. Press the Return key on your keyboard. This name will be used as the name of the flow variable or flow output field.

   > **Note:** Do not use "Result" for the result name.

4. From the list in the **From** column, select the primary output as the source for the result.

   > For example, you might select **Result Field: returnResult**, which is the primary output for that operation.

   For information about setting the primary output, see "Setting Operation Outputs" on page 515.

Create a secondary result in a step

1. Double-click a step in the authoring pane.

2. Select the **Results** tab and click **Add Result**.

3. In the **Name** column, enter a name for the result. Press the Return key on your keyboard. This name will be used as the name of the flow variable or flow output field.

4. From the **From** list, select the source for the result.

5. From the **Assign To** list, decide where the value will be stored:

    ○ To store the value in a flow variable, select **Flow Variable**

    ○ To make the value available to a parent flow, select **Flow Output Field**

6. From the **Assignment Action** list, select the appropriate action:

    ○ **OVERWRITE** – Replace the current value of the flow variable or flow output field with this value.

    ○ **APPEND** – Place this value at the end of the current value of the flow variable or flow output field.

    ○ **PREPEND** – Place this value in front of the current value of the flow variable or flow output field.

    ○ Use one of the four arithmetic assignment actions, **ADD**, **SUB**, **MULTIPLY**, and **DIVIDE**, to arithmetically modify the current value of the flow variable or flow output field.

    > For example, if the step result is 3.14, and you select **MULTIPLY**, the effect is to multiply the current value of the flow variable or flow output field by 3.14.

7. To create filters for the output data in the secondary result, click the right-pointing arrow at the end of the row.

    For information on creating filters, see "Filtering Output and Results" on page 529.

Hide the operation's raw result

1. In the Sensitive Data area, select the **Hide the operation's results** check box. This will hide the entire raw results of the operation, and make this information displayed as asterisks in the Studio and Central user interface.

    After this check box is selected, the operation's primary output will also be hidden and displayed as asterisks in the Studio and Central user interface.

    When used in scriptlets, sensitive data is retrieved as encrypted.

    For example, if you select the **Hide the operation's results** check box for the step Evaluate Expression, it is displayed in Studio as shown below, and similarly in Central:
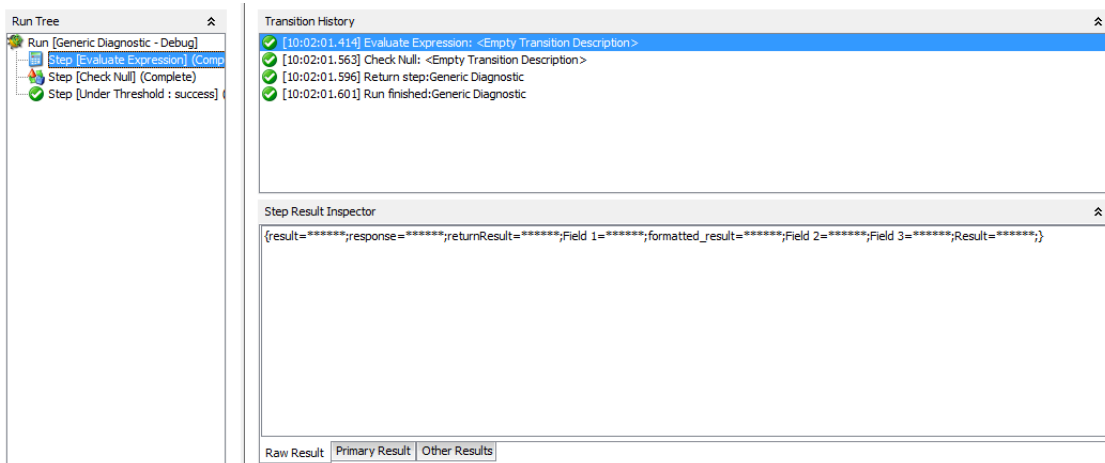
Hide the step result

1. Double-click a step in the authoring pane.

2. Select the **Results** tab.

3. Find the step result that you want to hide from the list of step results.

4. In the Results tab, select the **Sensitive Data** check box.

This makes the result hidden in the Central user interface and in the Studio internal and remote debugger. The result value will be displayed as asterisks and will not be persisted to the Operations Orchestration database.

This flag is transient, i.e.,assignment from a sensitive variable will also make the assigned variable sensitive. This transitivity is only seen during runtime execution, and is not reflected in the Studio user interface.

If a result is assigned from a sensitive input or sensitive result, it will also be sensitive. If a result is assigned from an operation raw result that is hidden, you must mark it as sensitive if you want to hide it in the Studio and Central user interface.

When used in scriptlets, sensitive data is retrieved as encrypted.

Sensitivity is also retained when using expressions for assignment. For example, ${input1}.

Hide the step raw result from the end-user

1. Double-click a step in the authoring pane.

2. Select the **Results** tab.

3. In the Sensitive Data area, select **Hide Operation Raw Results**.

4. Save the step.

> **Note:** If the step contains a sub-flow, there is no **Hide Operation Raw Results** section.

Cut, copy and paste a result

1. Double-click a step in the authoring pane.

2. Select the **Results** tab and select a result to cut/copy.

3. From the **Results** toolbar, click ✂ or ⧉ .

4. Move to the row above which you want to paste the result.

5. Click 📋 .

6. Save the step.

> **Note:**
>
> - You can copy step results to another step results pane only. Flow outputs can be copied to other flows or operations.
>
> - You can paste step results with same name. The new result will be identical to the existing result.

Delete a result from a step

1. Double-click a step in the authoring pane.

2. Select the **Results** tab and from the Results toolbar, click ✖ .

3. Save the step.

Change the field from which a result gets its data

1. Double-click a step in the authoring pane.

2. Select the **Results** tab.

3. Click in the **From** column of the output's row, and then select the desired field from the list.

4. Save the step.

Change the operation linked to a step

You can select a different operation to be linked to a step. After selecting the new operation, you can determine whether to overwrite existing inputs with the inputs of the new operation or to merge them

with the existing inputs.

1. Double-click a step in the authoring pane.

2. In the Step Inspector, select the **Advanced** tab.

   You can see the location and name of the current operation. For example:

3. Click the **Select** button. The Select Operation window opens:
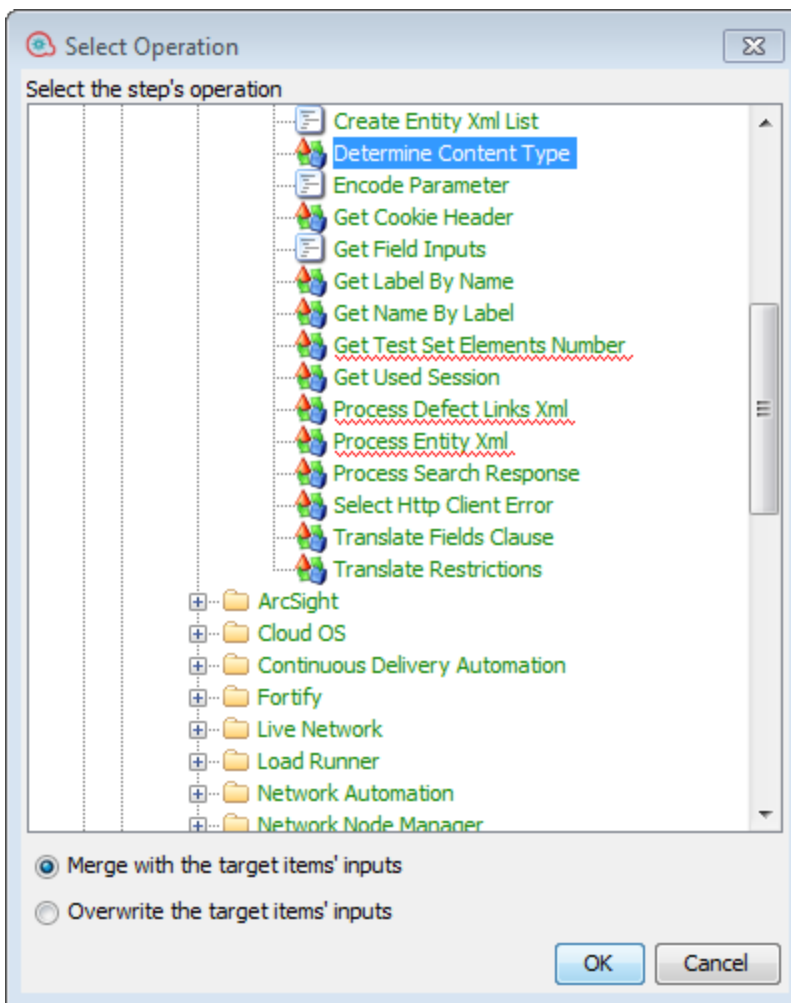
4. Browse to the new operation.

5. To add the new operation's inputs to the current inputs, select **Merge with the target items'**
   inputs.

> **Note:** If an input name already exists, the original input remains and is not overwritten.

6. To delete the current inputs and add only the new inputs, select **Overwrite the target items inputs**.

7. Click OK.

8. Save the step.

Reference Material

Step Inspector > Results tab

The **Results** tab in the Step Inspector is where you specify which outputs will be saved into flow variables or made available for parent flows.
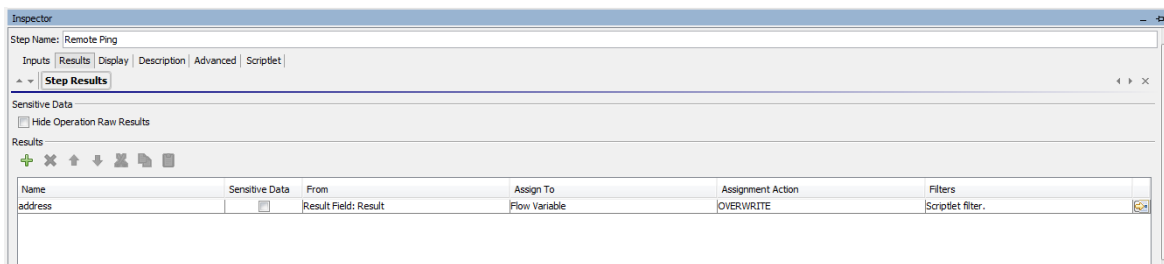
GUI item	Description
Name	Enter a name for the result. This name will be used as the name of the flow variable or flow output field.
Sensitive data	Makes this result hidden in the Central user interface and in the Studio internal and remote debugger. The result will be displayed as asterisks and will not be persisted to the Operations Orchestration database.
Add Result	Adds a new result row.
Remove Result	Removes the selected result row.
From	Select the source for the result.
Assign To	Select where the result value will be stored: • To store the value in a flow variable, select **Flow Variable** • To make the value available to a parent flow, select **Flow Output Field**
Assignment Action	From the **Assignment Action** list, select the appropriate action: • **OVERWRITE** – Replace the current value of the flow variable or flow

	output field with this value.
	• **APPEND** – Place this value at the end of the current value of the flow variable or flow output field.
	• **PREPEND** – Place this value in front of the current value of the flow variable or flow output field.
	• Use the arithmetic assignment actions **ADD**, **SUB**, **MULTIPLY**, and **DIVIDE** to arithmetically modify the current value of the flow variable or flow output field.
	For example, if the step result is 3.14, and you select **MULTIPLY**, the effect is to multiply the current value of the flow variable or flow output field by 3.14.
	Displays the Filter Editor for the result in the row.

Note: After you mark a step result as Sensitive Data, it remains sensitive during the whole run (and therefore encrypted).The step result passes this behavior to every input/step result that is assigned from it.

Step Inspector > Advanced tab

The **Advanced** tab in the Step Inspector is where you specify where you can change the source operation that the step is based on.
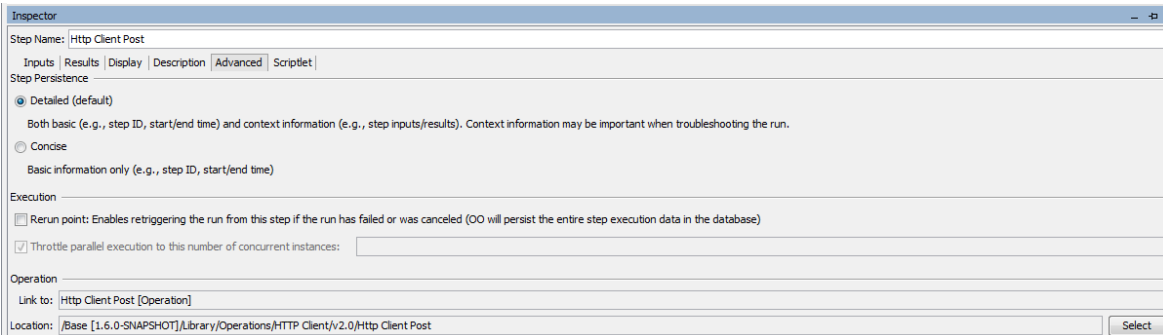
GUI item	Description
Step Persistence	
Detailed	Presents detailed information when running the flow including step ID, start/end time, and inputs and outputs. This information may be useful when troubleshooting the flow.
	This option preserves the behavior that was used in versions 10.60 and earlier.

	See "Testing and Troubleshooting a Flow Run" in the *Operations Orchestration Central Guide* for a sample of how this looks in Central.
Concise	Presents basic information when running the flow including step ID and start/end time. Use this option when you want to reduce the amount of data persisted in the database and you do not need to troubleshoot the specific step. See "Testing and Troubleshooting a Flow Run" in the *Operations Orchestration Central Guide* for a sample of how this looks in Central.
	Note: • The **Concise/Detailed** option is available in Operations Orchestration version 10.71 and above. If you define a step with Concise/Detailed persistency, the flow cannot be deployed on earlier Central versions. • When reverting to HPE OO Studio versions prior to 10.80, flows containing Concise persistency steps can be imported. However, if any change is applied to this content in the earlier Studio versions, the Concise property will be lost. For example, if you set Concise persistency in a flow in 10.80 and then import the flow into Studio 10.60, the flow runs correctly. If you then make changes to the flow in 10.60 (e.g., add a step, change a variable) and save the flow in 10.60, the persistency setting will be lost.
Execution	
Rerun point	Marks the step as a rerun point. If a step is defined as a rerun point, you can rerun the flow from this step in Central without having to rerun the entire flow. This is useful when trying to recover failed flows. The Rerun option is only available in Central if the administrator has marked the **Enable the rerun option for failed/canceled runs** check box in **System Configuration > System Settings.** The Rerun option is only available in Central for flows that have failed or were canceled. **Note:** • It is recommended to consider which steps you want to mark as rerun points. Only mark steps from which it is safe and logical to rerun the flow. • Marking a large number of steps as rerun points means that more data is persisted in the database. As a result, some users may

	experience slower performance during the flow execution.
	• The **Rerun point** option is available in Operations Orchestration version 10.71 and above. If you define a step as a rerun point, the flow cannot be deployed on earlier Central versions.
	• When reverting to Operations Orchestration Studio versions prior to 10.80, flows containing rerun points can be imported. However, if any change is applied to this content in the earlier Studio versions, the rerun points will be lost.
	For example, if you set rerun points in a flow in 10.80 and then import the flow into Studio 10.60, the flow runs correctly.
	If you then make changes to the flow in 10.60 (e.g., add a step, change a variable) and save the flow in 10.60, the rerun points will be lost.
	• The following point types cannot be defined as rerun points: ◦ endpoints ◦ non-blocking steps ◦ steps inside a multi-instance step ◦ steps inside parallel steps
Operation	
Link to	Displays the source operation that the step is based on.
Location	Displays the location of the source operation that the step is based on.

Filtering Output and Results

You can create filters to extract and modify parts of an operation's output or a step's result.

For example, if you only want the maximum, minimum, and average round-trip times for a ping operation to a server, you can isolate and extract all three pieces of information from the operation's raw output by filtering the raw output into three outputs.

What do you want to do?

Create a filter

1. Open the Filter Editor. This step varies, depending on what you are filtering:

- To filter an operation's primary output, open the operation's **Properties** sheet, click the **Outputs** tab, and then click the **Edit Filters** button.

- To create a filter for an operation's secondary output, open the operation's **Properties** sheet, click the **Outputs** tab, and then click the right-pointing arrow ⬆ at the end of the output row.

- To create a filter for a step's secondary result, double-click the step in the authoring pane, click the **Results** tab, and then click the right-pointing arrow ⬆ at the end of the result row.

2. In the Filter Editor, click the **Add** button.

   > **Note:** You can add multiple filters to an output or result.

3. From the **Select Filter** list, select the type of filter.
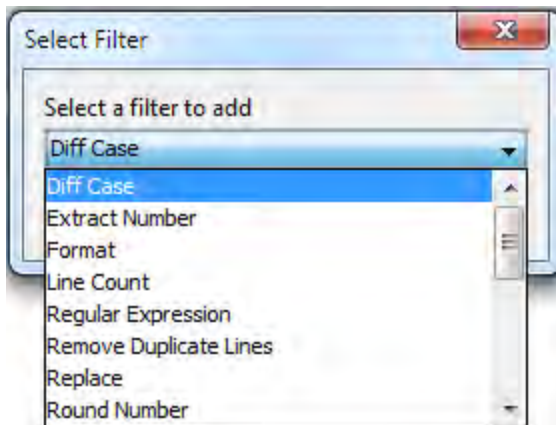
4. In the **Details** area in the upper-right of the Filter Editor, configure the filter. See Filter Options to see the options for different filters.

Test a filter with data from a command line

To test, a filter, paste some data into the **Test Filter Input** box. If this data can be generated by a local command-line command, do the following:

1. Open the Filter Editor for an output or result.

2. Click **Clear** to clear the **Test Filter Input** box.

3. Click **Quick Command**.

4. Type a command that generates the desired data.

5. Click **OK**. The output of the command appears in the **Test Filter Input** box.

6. Do one of the following:

- Click **Test All Filters**

- Select the filters you want to test and click **Test Selected Filters**

The filters are applied to the data in the **Test Filter Input** box (in top-to-bottom order), and the filtered results appear in the **Test Output** box.

Test a filter with data from the debugger

If the data you need is produced by means that you cannot reproduce with a simple command-line command, copy the data from the debugger and paste it into the **Test Filter Input** box:

1. Open the Filter Editor for an output or result.

2. Click **Clear** to clear the **Test Filter Input** box.

3. Run the flow in the debugger.

4. Highlight the relevant step.

5. In the **Step Result Inspector** pane, copy the contents of the **Raw Result** tab.

6. In the Filter Editor, paste the contents into the **Test Filter Input** box.

7. Do one of the following:

   - Click **Test All Filters**

   - Select the filters you want to test and click **Test Selected Filters**

The filters are applied to the data in the **Test Filter Input** box (in top-to-bottom order), and the filtered results appear in the **Test Output** box.

Filter a different output or result

While you have the Filter Editor open, you can click the upward- or downward-pointing arrows beside **Outputs Summary** to create a filter for a different output or result.

Use a system filter in an output or result

1. Open the Filter Editor of the output or result on which you want to use the system filter.

2. In the **Projects** pane, expand the **Configuration** and **System Filters** folders.

3. Drag the filter that you want to use from the **System Filters** folder to the **Filter** list in the Filter Editor.
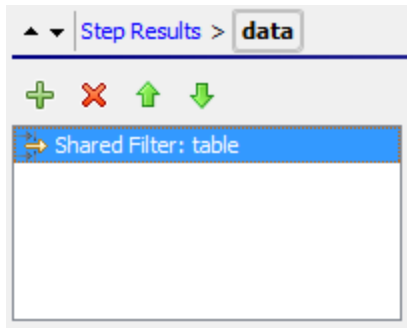
Save a filter for reuse as a system filter

You can take an existing filter in an operation and save it as a system filter. The resulting system filter is independent of the operation for it was created and can be reused in any output or result.

For more information, see "Configuring System Filters" on page 442.

Filter Options

Diff Case

A **Diff Case** filter changes all the characters in the string either to upper case or to lower case. If you leave the **To Upper Case** check box cleared, the filter changes all the characters to lower case.

Extract Number

An **Extract Number** filter extracts the first number found in the result. The filter treats an unbroken series of integers as a single number. For example, the **Extract Number** filter would extract the number "123" from the strings "123Test" or "Test123".

Format

A **Format** filter attaches text to a result or output or replaces the original content of the result or output with the text that you specify.
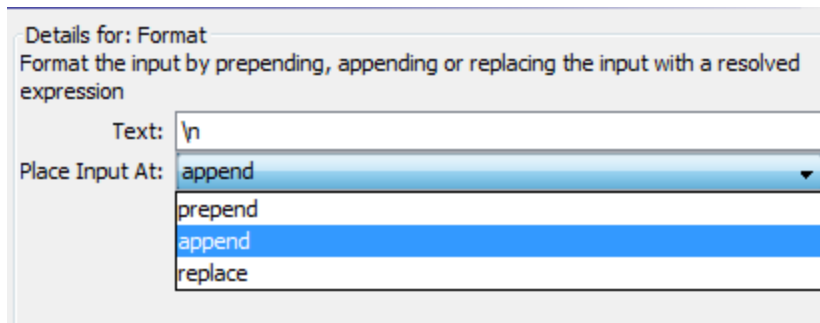
1. In the **Text** box, type the text you want to attach to the result or use to replace the result.

2. From the **Place Input At** list:

   ○ To attach the text to the front of the existing text, select **prepend**

   ○ To add the text to the back of the existing text, select **append**

   ○ To replace the output with the text, select **replace**

Line Count

A **Line Count** filter outputs the total number of lines of the result.

Regular Expression

A **Regular Expression** filter filters the raw results using a regular expression (regex).

1. From the **Expression Type** list, select **Java Style**. Do not use the other styles; they have been deprecated.

2. In the **Expression Value** box, type the regular expression.

3. For **Filter Style**, select **Filter Entire Input** or **Filter Line-by-line**, according to how you want the filter applied to the raw results.

4. To make the regular expression not case-sensitive, select **Ignore Case**.

For more information about working with regular expressions, see "Using Regular Expressions in a Flow" on page 579.

Remove Duplicate Lines

This filter finds lines that are identical and removes all but one of them.

To apply the filter only to duplicate lines that follow each other directly, select **Consecutive**.

Replace

This filter replaces the first or last instance or all instances of one string with another string.

1. In the **Find** box, type the string to search for and replace.

2. From the **Replace**, select **First**, **All**, or **Last,** depending on which instances of the target string you want to replace.

3. In the **With** box, type the string to replace the target string with.

4. To make the search not case-sensitive, select the **Ignore case** check box.

Round Number

This filter rounds numbers up or down.

1. To specify the accuracy of the rounding, in the **Number of Decimal Places** box, type the number of decimal places the number should be rounded to.

2. For **Rounding Type**, specify in which direction the number should be rounded:

   - **Floor** always rounds the number down

   - **Ceiling** always rounds the number up

   - **Round** rounds the number up if the last meaningful place is 5 or more, and down otherwise

Scriptlet

This filters data with a scriptlet that you create.

```
Details for: Scriptlet
Execute a JavaScript scriptlet to filter data

  Scriptlet                                    Insert Template  Check Script

1    scriptletResult = "Timestamp:"+new
2    java.util.Date()+"\n"+scriptletInput+"\n"+scriptletInput.len
3    gth+" characters"
```
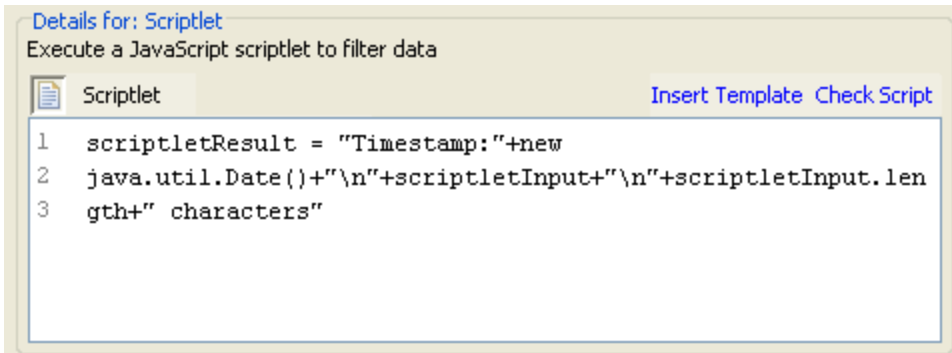
1. To obtain scriptlet lines that you will need for the scriptlet to work as a filter, click **Insert Template**.

   The template that is inserted is specific to the language that you chose and includes the most commonly used commands for accessing flow variables, values of global variables, operation results, and inputs, and for setting and manipulating flow variable values and results.

2. To debug the scriptlet, click **Check Script**.

For more information about scriptlets, see "Using Scriptlets in a Flow" on page 574.

Select Line

This filter defines a line that you want to extract from the raw results.

```
Details for: Select Line
Select a line from the input
Line Select Method:  First Line Starting With          ▼
    Parameter:
Line Ending Type:  ○ Auto
                   ◉ Unix
                   ○ Windows
```
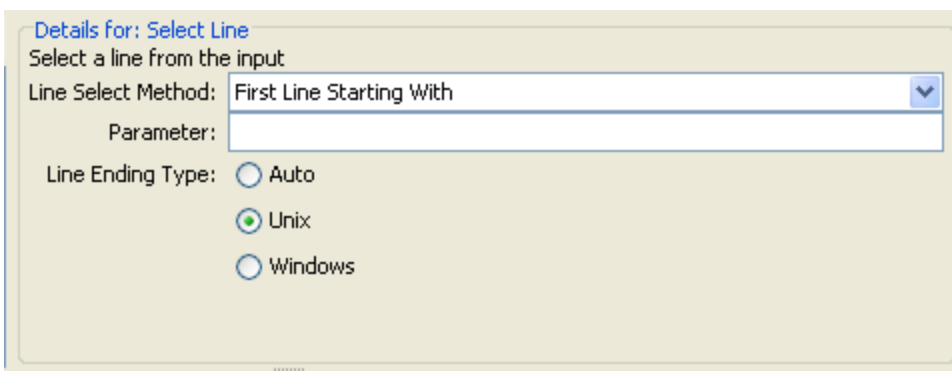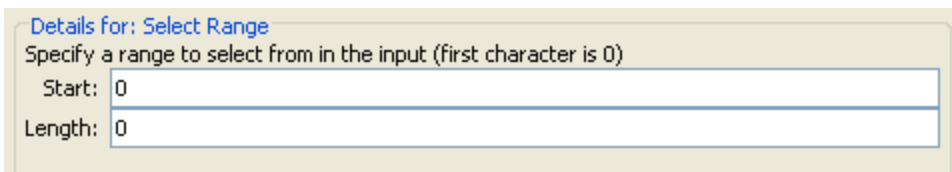
1. From the **Line Select Method** list, select a criterion for the line that you're interested in.

2. In the **Parameter** box, type a string that the line contains.

3. From the **Line Ending Type** group, select one of the following:

   ○ If the text that you're filtering was generated on a Unix operating system (which ends lines with `LF`), select **Unix**.

   ○ If the text that you're filtering was generated on a Windows operating system (which ends lines with `CR/LF`), select **Windows**.

- To enable the filter to accept either type of line ending, select **Auto**.

  Auto is the default selection.

Select Range

This filter defines a string that you want to extract from the input data. The two criteria for defining the string are its length in characters and the position of its first character from the start of the input data.
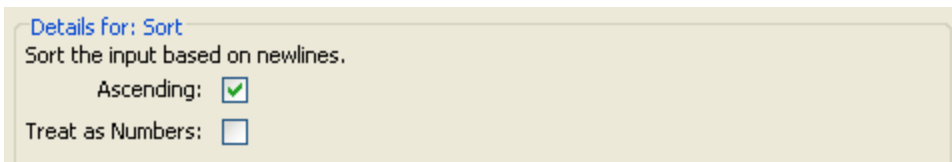
1. In the **Start** box, type the zero-based start position of the string.

2. In the **Length** box, type the number of characters in the string.

   Keep in mind that a new line may count as one or two characters, depending on the operating system from which you obtain the data you're filtering.

Sort

This filter orders the lines of input data by the first character in each line.

1. To specify the direction of the sort:

   - For ascending order, leave the **Ascending** check box selected.

   - For descending order, clear the **Ascending** check box.

2. To order the data by ASCII order, select the **Treat as Numbers** check box.

   Note that ascending ASCII order is roughly as follows, for English characters:

   - White space

   - Symbols

   - Numbers

   - Alphabetic characters

Strip

This filter strips characters from the beginning or end of the raw results.

> **Note:** If this filter follows other filters, the characters are stripped from the beginning or end of the subset of the raw results that was obtained by the processing of preceding filters.

```
Details for: Strip
Remove a matching string from the head or tail of the input
    Strip Method:  All Characters Up To And Including        [v]
Characters to Strip:
```

1. From the **Strip Method** list, select how you want the filter to strip the raw results. You can specify the following options for stripping the string that you specify in the **Characters to Strip** text box:

   ○ **All Characters Up To** the string

   ○ **All Characters Up To And Including** the string

   ○ **All Characters After** the string

   ○ **All Characters After And Including** the string

2. In the **Characters to Strip** text box, type the string to find.

Strip Whitespace

This filter removes all the whitespace characters from the front and the end of the raw results.

Table

A table filter does not convert the raw results into a table, but enables you to manipulate the raw results as if they were a table, including sorting columns and selecting columns, rows, and blocks.

```
Details for: Table
Parses the input as a table and sorts it on a specified column
  Column Delimiter:  Whitespace [v]      Row Delimiter:  NewLine [v]
 First Row is Header:  [ ]          Strip First Row of Result:  [ ]
    Sort On Column:  -1                    Ascending:  [ ]
        Select Row:  0                      Select Col:  0
      Select Width:  1                   Select Height:  1
```

> **Note:** Row numbering is 0-based (starts with 0 [0]), and column numbering is 1-based.

1. In the **Column Delimiter** list, choose the character that will serve to divide the data into columns in a meaningful way.

2. In the **Row Delimiter** list, choose the character that will serve to divide the data into rows in a meaningful way.

   Note: Two or more consecutive whitespaces count as a single whitespace, so a column may be occupied by data that you expected to find in a column to the right. For example, this behavior will appear if you apply this filter to the output of a "dir" command-line command with whitespace specified as the column delimiter.

3. To treat the members of the first row as column headers, select **First Row is Header**.

4. To remove the first row, select **Strip First Row of Result**.

5. To sort on a column, type the (1-based) column number in the **Sort On Column** box.

   Tip: The value **-1** means do not sort on any column.

6. To specify ascending order, select the **Ascending** box.

   By default, the sort order is descending.

7. To select a row you want the filter to extract:

   ○ In the **Select Row** box, type the (0-based) row number.

      Tip: **-1** selects all the rows in the data.

   ○ In the **Select Width** box, type the number of columns in that row that you want extracted.

      Tip: **-1** selects all the remaining columns in the data to the right of the column specified in **Select Col**.

8. To select a column you want the filter to extract:

   ○ In **Select Col,** type the column number.

      Tip: **-1** selects all the columns in the data.

   ○ In the **Select Height** box, type the number of rows in that column that you want extracted.

      Tip: **-1** selects all the remaining rows in the data below the row specified in **Select Row**.

For example, to extract the first 5 rows of the 2nd through 4th columns, you would specify the following. In these settings, the first two settings define the rows selected, and the second two settings define the rows selected.

- In **Select Row**: 0

- In **Select Height**: 5

- In **Select Col**: 2

- In **Select Width**: 3

XML filters

XML filters enable you to parse XML within a step, the XML being obtained from the step's input or result, rather than having to create a flow and pass the XML to one of the XML-processing operations in Operations Orchestration default content.

Using XML filters in an operation and using the XML-processing operations in default content differs in several respects.

- There is the difference between accomplishing the task within an operation and using the infrastructure of a flow to accomplish the task

- Filters within an operation have some limits that the XML-processing operations do not have. These limitations are described in the following sections on the particular filters. Whether you choose to filter input XML with a filter or an operation may depend on how you obtain the XML.

Following are the XML filters:

- XML Get Attribute

- XML Get Element

- XML Get Element Value

- XPath Query

For illustration of the XML filters, the examples reference the following XML example:

```
<?xml version="1.0" encoding="utf-8"?>
<tickets>
  <ticket id="1448" severity="3">
    <customer firstName="John" lastName="Doe">
      <volume>30000</volume>
      <company>myOrg</company>
      <position>CIO</position>
      <contactInfo>
        <email>jdoe@myorg.com</email>
        <email>johnsSecondEmail@myorg.com</email>
        <mobile>12065551212</mobile>
        <description internal="1">Private contact info</description>
        <description>Partial contact info</description>
```

```
      </contactInfo>
      <description>Our best customer</description>
    </customer>
    <details>
      <description>A simple Test xml</description>
      <comment user="john"> Initially raising ticket</comment>
      <comment user="frank"> Problem diagnosed, not a real issue</comment>
      <comment user="albert">ok, I'm going to close it.</comment>
      <state>Closed</state>
    </details>
  </ticket>
  <ticket id="1886" severity="5">
    <customer firstName="Elaine" lastName="Benson">
      <volume>50000</volume>
      <company>herCompany</company>
      <position>CEO</position>
      <contactInfo>
        <email>ebenson@herco.com</email>
        <mobile>011445551212</mobile>
        <description internal="1">Private contact info</description>
        <description>Partial contact info</description>
      </contactInfo>
      <description>Our other best customer</description>
    </customer>
    <details>
      <description>datastream bug</description>
      <comment user="jack">Customer found bug.</comment>
      <comment user="elsbeth">It is a third-party supplier bug.</comment>
      <state>Closed</state>
    </details>
  </ticket>
</tickets>
```

XML Get Attribute

The **XML Get Attribute** filter extracts the value for each of one or more instances of the attribute that you specify. In the Filter Editor, you can control which instance of the attribute the filter is applied to by specifying an element path to the attribute.

You can obtain the value for a single instance of the attribute or for multiple instances, returned in a table. In such a table, the columns are comma-delimited and the rows are new line-delimited.

Details for: XML Get Attribute
Filters an xml document for the requested attribute value. For advanced XML
filtering use the XPath Filter.

Element Path:

Include sub-elements: ☐

Attribute Name:

Result: ⦿ Single Match

○ As Table

1. In the **Element Path** box, specify the path of the element that contains the attribute whose value you want to extract. Use forward slashes to separate the parts of the path to the element.

   To control which instance of the element the filter gets the attribute's value from, add a specification such as [2] or [3]. The numbering of elements is 1-based (starts with [1]). Thus to specify the second instance of an element, you would use [2].

2. To search child elements of the element you've specified, select the **Include sub-elements** check box.

3. In the **Attribute Name** box, type the name of the attribute whose value you want.

4. For **Result**, select one of the following:

   ○ To restrict the result extracted to the value of a single instance of an attribute, select **Single Match**.

   ○ To extract the value of all the instances of the attribute you have named, select **As Table**.

Example: To find the name of a user for one of the comments (using the example XML from the topic *XML Filters*):

In the **Element Path** box, type **/ticket/details/comment**.

Example: To get the name of the user for a particular comment (in this example, the second one):

1. In the **Element Path** box, type **/ticket/details/comment[2]**.

2. In the **Attribute Name** box, type **user**.

3. Beside **Result,** select **Single Match**.

The output will be john.

Example: To find the name of the user for each comment:

1. In the **Element Path** box, type **/ticket/details/comment**.

2. In the **Attribute Name** box, type **user**.

3. Beside **Result,** select **As Table**.

The output will be:

```
Path,user

/ticket/details/comment[1],john

/ticket/details/comment[2],frank

/ticket/details/comment[3],albert
```

XML Get Element

The **XML Get Element** filter enables you to extract an element in its entirety (including child elements, values, and attributes) by describing it in any of the following ways:

- By a relative or absolute path.

- By a child element of the element you want to extract. You can also search by a specific value of the child element.

- By an attribute of the element you want to extract. You can also search by a specific value of the attribute.

```
Details for: XML Get Element
Filters an xml document for elements given a path.  For advanced XML filtering use the
XPath Filter.
    Element Path:  [                                                    ]
    Child Named:   [                          ]  Value:  [                          ]
 Attribute Named:  [                          ]  Value:  [                          ]
```

In the following procedure, you can enter specifications in any one or combination of the text boxes.

1. In the **Element Path** box, type an absolute path to the element.

   Within the path, a relative path indicator indicates location relative to the element that precedes the relative path indicator.

   ○ **../** specifies the parent of the last-named element.

   ○ **./** specifies the last-named element.

   Example: in the XML example, `<volume>` and `<company>` are sibling elements, both children

of the `<customer>` element. You could specify the `<company>` element with the following relative path:

```
/tickets/ticket/customer/volume/../company
```

If there is more than one instance of the identified element, simply specifying the path, as in the preceding example returns all the instances of the element.

You can specify a particular instance of any element in the path with an integer inside square brackets.

Example:

`/tickets/ticket/details/comment` specifies all the comments in the details for all the tickets.

`/tickets/ticket/details/comment[2]` specifies the second comment for each ticket.

`/tickets/ticket[2]/details/comment` specifies all the comments for the second ticket.

2. In the **Child Named** box, type the name of an element that is a child of the element (or elements) that you want to extract. If the child element has a value, you can narrow the results by typing that value in the **Value** box.

    ○ The **Child Named** box works for only one level of child elements. The filter only returns the direct parent of the child element that you specify.

    ○ The **Value** box is intended for brief values. The value that you type there must be an exact match of the value of the child element of the element that you want to extract.

3. In the **Attribute Named** box, type the name of an attribute that is unique to the element you want to extract. To further narrow the results, you can type a value of the attribute in the **Value** box.

Example: In the example XML, there are several ways to extract the customer element and all of its contents:

- In the **Element Path** box, type **/ticket/customer**

- In the **Child Named** box, type any of the child elements of customer:

  company

  position

  contactInfo

  If you type **company** in the **Child** box, then in the accompanying **Value** box, you could also type **myOrg**

- In the **Attribute named** box, type one of the following:

  firstname

  lastname

  In the accompanying **Value** box, you could type the respective value for those attributes:

  John

  Doe

For each of these filters, the output is the `customer` element, as follows:

```
<customer firstName="John" lastName="Doe">
            <company>myOrg</company>
            <position>CIO</position>
            <contactInfo>
              <email>jdoe@myorg.com</email>
              <email>johnsSecondEmail@myorg.com</email>
              <mobile>12065551212</mobile>
              <description internal="1">Private contact info</description>
              <description>Partial contact info</description>
            </contactInfo>
            <description>Our best customer</description>
      </customer>
```

XML Get Element Value

The **XML Get Element Value** filter enables you to get the value of a specific element.

In the **Element Path** box, type the path to the element whose value you're interested in.

As with the other filters, if there are multiple instances of an element, the filter returns the first one, unless you specify a different instance.

Example: Using the sample XML

To get the value for the email element, type **/tickets/ticket/customer/contactInfo/email**

The output will be one of the two emails given:

jdoe@myorg.com

johnsSecondEmail@myorg.com

To specify a particular instance of the email element, type **/ticket/customer/contactInfo/email[2]**

The output will be: `johnsSecondEmail@myorg.com`

XPath Query

The **XPath Query** filter enables you to extract data from the result with queries that use the standard XPath syntax, which you type in the **XPath Query** box.

```
Details for: XPath Query
Filters an xml document based on an xpath query and returns the results of the
query.
XPath Query: [                                            ]
```

In the **XPath Query** box, type the query, using XPath syntax.

- The path that precedes the square brackets identifies the scope of the query with which you are narrowing the results.

- Square brackets contain the filtering portion of the query. There can be more than one set of filters in a query.

Example: Using the sample XML

You can extract a customer who has a volume of more than 40,000 of some units, with either of the following queries:

- This XPath query finds all the companies whose customer's volume is more than 40,000.

`/tickets/ticket/customer/company[../volume>40000]`

The `<volume>` element is a sibling of the `<company>` tag, so to locate the element `<volume>`, you use the following sequence inside the square brackets to articulate the path relative to `<company>`:

`../`

- This XPath query finds all the customers whose volume is greater 40,000 units.

`/tickets/ticket/customer[volume>40000]`

Because `<volume>` is a child of `<company>`, you do not need to specify its relative path.

Reference Material

Filter Editor

The **Filter** list in the upper-left displays a list of the filters as you create them.
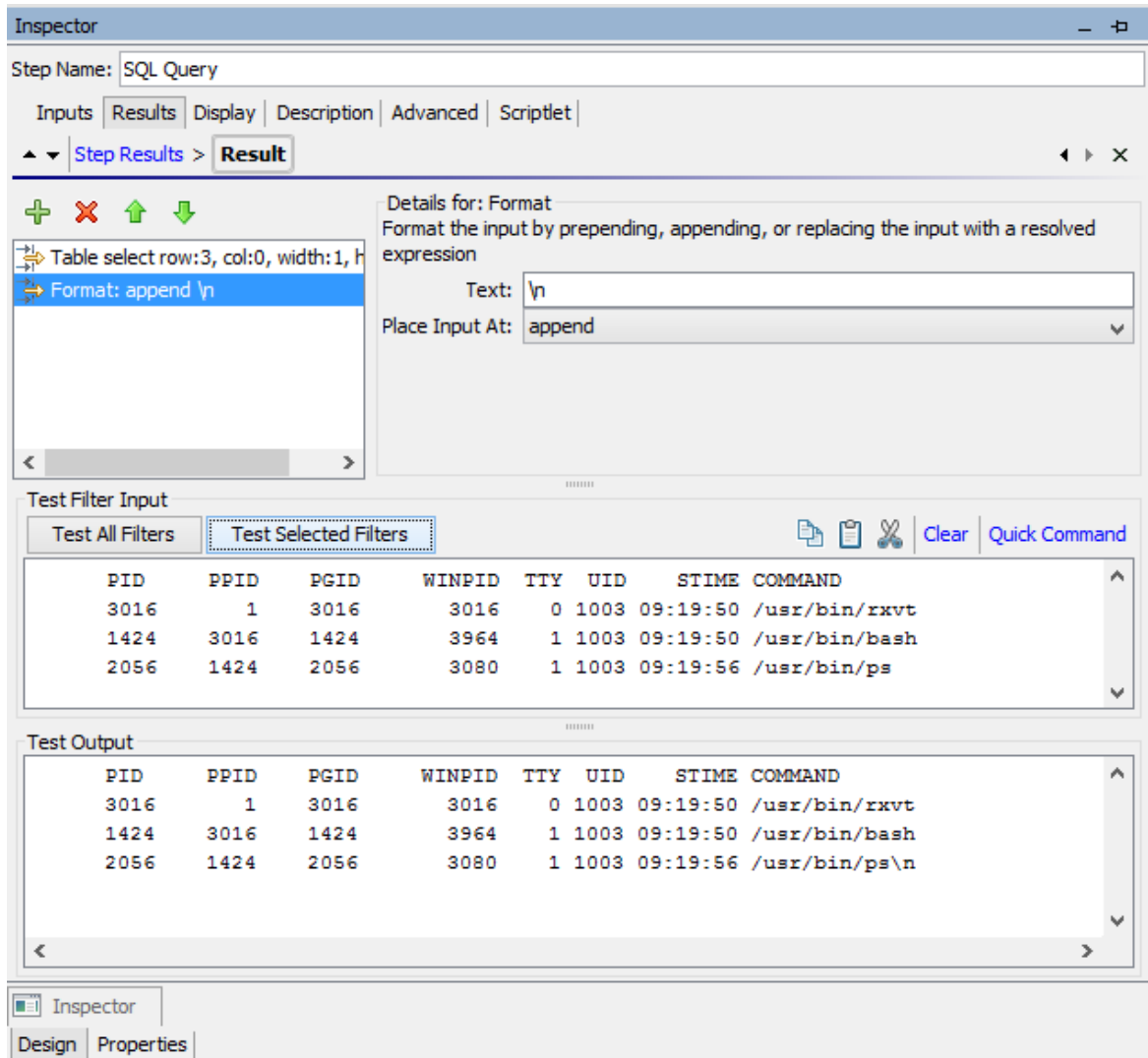
When you create a filter and have selected a filter type, the **Details for:** section in the upper-right changes to show controls for modifying filters, depending on the kind of filter you select.

GUI item	Description
Add	Click to add a new filters.
Remove	Click to delete the selected filter or filters.

⬆ ⬇	Click to move the selected filter up or down in the list. Filters are processed in the order that they appear in the list.
Test Filter Input	This is where you place data in order to test that the filter works as expected.
Test All Filters	Applies the test to all the filters on the output or result.
Test Selected Filters	Applies the test to the selected filters.
Copy	Copy data within the **Test Filter Input** box.
Paste	Paste data into the **Test Filter Input** box.
Cut	Cut data within the **Test Filter Input** box.
Clear	Clear data within the **Test Filter Input** box.
Quick Command	Type a command that generates the data on which you want to test the filter. The output of the command appears in the **Test Filter Input** box.
Test Output	After filters are applied to the test data in the **Test Filter Input** box, the filtered results appear in the **Test Output** box.

Working with Variables

You can use variables to move data within and between flows.

For example, if you have several steps that act on a server, you can have the first step get the IP address of a server and assign that value to a flow variable. Then, any subsequent step that has an input of that name will automatically use that server name.

Flow Variables

Flow variables are available only for the flow within which they are defined.

Assigning Value to Flow Variables

You can assign a value to a flow variable from:

- **A step's result** - for example, a step with an operation to count hits will store the result in a flow variable

- **An input value** - for example, a step that gets an IP address as an input value will store the address as a flow variable

- **A scriptlet** - for example, a scriptlet that evaluates data that is returned from a step's operation will store the data in a flow variable

Using Flow Variables

You can reference a flow variable and the data that it stores in any of the following places:

- **In a different step** in the same flow

- **Within a lane in a parallel split step** – a lane step can use the value of a flow variable, if that value was written to the flow variable by an earlier step in the same lane or prior to the parallel split step. However, a step in one lane cannot use a flow variable value, if this value was written to the variable by a step in a different lane.

- **In an operation input**

- **In flow, step, and transition descriptions** – for example, the **Ping Latency** operation filters out the average duration of the ping. A step associated with this operation could save the average duration as the flow variable latency, then the transition that follows this step could report that value to the user

- **As part of data you are testing with a response rule** – for example, to see whether an output string or error string contains a value that you have stored in a flow variable.

- **In scriptlets** – to make a scriptlet result available outside of the step, the scriptlet must create a flow variable (if the desired one doesn't already exist), and assign the result to it.

- **In operation parameters** – If an operation parameter takes a value, you can access that value by referencing a flow variable that contains it.

The **Flow Variables** pane helps you to keep track of the flow variables that you have created.

Global Variables

Global variables are keyname-and-value pairs that are part of the global context, and so are always available for use or reference in any flow run.

If a flow variable and a global variable have the same name, a reference to that variable name accesses the (local) flow variable of that name, not the global variable. This is the case either for assigning a value to the variable or for getting its value.

When you specify that an input gets its value from a global variable, a flow variable is created with the value of that global variable, and the value is supplied from the flow variable to the input.

What do you want to do?

Assign a value to a flow variable from an input

By default, the value of the input is assigned to a flow variable with the same name as the input.

1. Open the **Properties** sheet (for an operation) or the Step Inspector (for a step).

2. On the **Inputs** tab, select an input or create a new one.

3. From the **Assign to** dropdown, select the variable to which you want the value to be assigned.

4. Save.

For more information about creating an input, see "Creating Input" on page 472.

Assign a value to a flow variable from a result

1. Open the **Properties** sheet (for an operation) or the Step Inspector (for a step).

2. On the **Results** tab, select the row of the relevant result.

3. From the **Assign To** list, select **Flow Variable**.

4. Under **Name**, specify the name of the flow variable.

5. Under **From**, specify the source of the value.

For more information about creating a result, see "Creating Outputs and Results" on page 515.

If necessary, to obtain the precise results you want, create one or more filters for the result. See "Filtering Output and Results" on page 529.

Assign a value to a flow variable from a scriptlet

You can also create and assign a value to a flow variable from a scriptlet.

In the scriptlet, include a command with the following syntax:

```
scriptletContext.putLocal("<localflowvariablename>", <value>);
```

where `<value>` can be a variable or an object created within the scriptlet.

View information in the Flow Variables pane

The **Flow Variables** pane helps you track the storage of data in flow variables:

- How the flow uses flow variables to make data available where it's needed

- Where the flow variables obtain their data

The **Flow Variables** pane presents this information in a tree structure. The **Flow Variables** pane shows all the flow variables used in the current flow, listing each flow variable's creation and/or usages. Any changes you make to a flow variable in the flow are automatically reflected in the **Flow Variables** pane.

Click the **Flow Variables** tab in the upper-right of the Studio window to open the **Flow Variables** pane.

Filter information in the Flow Variables pane

To zero in on the flow variable uses that you're most interested in, you can select which flow variable uses you want to display in the pane.

1. Display the filter buttons by clicking the **Filter** button in the **Flow Variables** toolbar.

2. In the row of buttons that appears, click the buttons to toggle each filter type on or off. As you toggle each type of data source, that type is displayed or removed from the display:

   - Flow input

   - User-prompt step input

- Step input that does not have a user prompt

- Result

- Scriptlet

Locate the input that the flow variable listing references

To view a particular use and open the editor that defines that use, select the use instance in the **Flow Variables** pane.

- If the use is for a flow input, the flow's **Properties** sheet opens on the **Inputs** tab, with the Input Editor for the relevant input open.

- If the use is for a step input or result, the flow diagram opens with the step selected. Beneath the flow diagram the editor for the input or result is opened.

> **Tip:** Use the **Previous** and **Next** buttons to move up or down in the list of uses.

View a global variable

To view all the global variables in a flow, debug the flow. The global variables (as well as the flow variables) and their current values are listed in the Context Inspector.

For more information on the Context Inspector, see "Validating Content" on page 592.

Change a global variable

> **Important!** Before you change the value of a global variable, remember that global variables are available in any run of any flow. Changing the value of a global variable will affect other flows and operations that use that global variable.

To change the value of a global variable, use the **Set System Property** operation available in the HPE Solutions Content Pack.

Reference Material

Flow Variables pane

When you have a flow open in the authoring pane, the **Flow Variables** pane lists each flow variable in alphabetical order, and describes each use of the flow—each place in the flow where the flow variable may be used.

GUI item	Description
	Click to move up in the list of uses.
	Click to move down in the list of uses.
	Click to display the filter buttons, in order to filter the information visible in the in the **Flow Variables** pane.
#	Displays the number of times the flow variable is used in the flow.
	Is checked when a particular use of the flow variable occurs in a flow input.
	Is checked when a particular use of the flow variable gets its value from user input.

Filter buttons in the Flow Variables pane

Display the filter buttons by clicking the **Filter** button in the **Flow Variables** toolbar.

Click the filter buttons to toggle each filter type on or off. As you toggle each type of data source, that type is displayed or removed from the display.

Filter button	Description

	Flow input—the flow variable is referenced in an input
	Step inputs with user prompts
	Step inputs without user prompts
	Results—the flow variable is associated with a step result
	Scriptlets—the flow variable is referenced in a scriptlet

Creating Return Steps

A flow needs one or more return steps to end the flow.

The four types of return step are:

- **Resolved** – This is the standard return step for a flow that runs correctly.

- **Diagnosed** – This return step indicates that a flow has determined what a problem is and has opted not to take action on it other than notification.

- **No Action Taken** – This return step is used when a remediation flow gathers data but cannot determine any diagnosis or remediation.

> **Note:** A flow that is intended solely to gather data should return **Resolved** ✅ when it is complete, rather than **No Action Taken** 🟡.

- **Error** 🛑 – This return step is used if the flow fails to run all the way to the end. For example, because of bad input, failure to reach a system, or a problem with the flow.

In each return step name, the response of the return step is shown after the colon; for example, **Error: failure**. You can modify this response. For example, if the outcome that led to an **Error: failure** return step was not a failure in an operation but a result that did not meet a required threshold, then you might want to create a new response for the **Error: failure** step that reflects this outcome, such as **Error: threshold not met**.

Best Practices

- If you have multiple end steps of the same type in a flow (for example, multiple error end steps), rename the end steps to include the cause of the failure.

- Avoid confusing a failed operation with a negative result. For example, if an operation asks a question for which the answer may be TRUE or FALSE, a FALSE answer is not the same as a failure. In such a case, you need two **Error** return results, one for a FALSE result and one for a

failure of the operation.
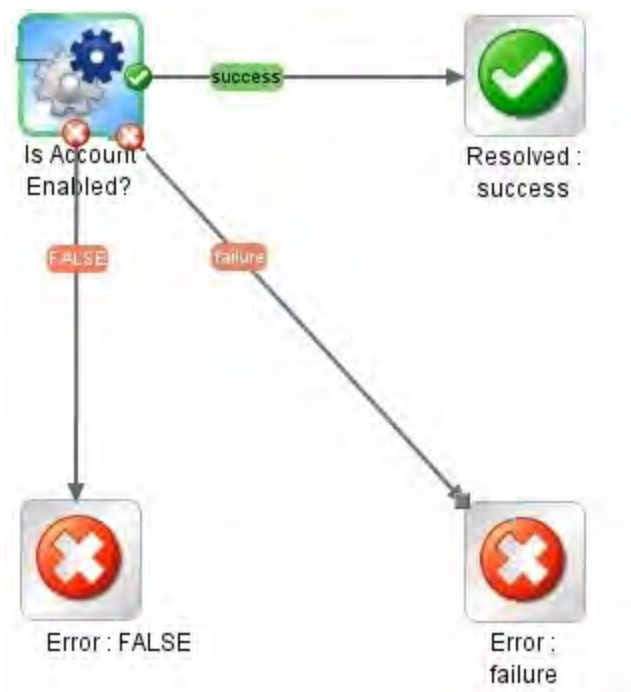
What do you want to do?

Add a return step to a flow

1. On the authoring pane toolbar, click the **Step Palette** button [icon] to display the **Step** palette.

2. From the **Step** palette, drag the appropriate return step icon to the authoring canvas.

3. Create transitions from the flow steps to the return step.

Change a return step's response

You can change the response of a return step to more accurately reflect the outcome that led to the return step. For example, if your flow has multiple error responses (**Error: failure** and **Error: threshold not met**), when you drag the **Error** [icon] icon to the authoring canvas, the error return step may not contain the response that you want.

1. Right-click the return step on the authoring pane and select **Select Response**.

2. Select the response that you want for the return step. For example, **Error: threshold not met**.

Create and assign a new response to the return step

If the list of available responses does not include the response that you need, you can create a custom response.

1. Right-click the return step on the authoring pane and select **Select Response**.

2. Select **Add New Response**.

3. In the dialog box, enter a name for the new response and click **OK**.

Reference Material

Step palette

The **Step** palette contains buttons for dragging return steps, parallel split steps, multi-instance steps, and callouts onto the flow. Display the **Step** palette by clicking the **Step Palette** button from the authoring pane toolbar.

Button	Description
Success	Lets you drag a **Success** return step to the flow.
Diagnosed	Lets you drag a **Diagnosed** return step to the flow.
No Action Taken	Lets you drag a **No Action Taken** return step to the flow.
Failure	Lets you drag a **Failure** return step to the flow.
Parallel Split Step	Lets you drag a parallel split step to the flow.
Multi-instance Step	Lets you drag a multi-instance step to the flow.
Callout	Lets you drag a callout to the flow, providing information to users.
Docking bar	Click to dock and undock the palette.

Advanced Authoring

This chapter covers creating more complex flows. For information about creating simple flows, see "Authoring a Flow – Basics" on page 451.

When creating flows, make sure not to create flows that create unlimited growth in memory. For example, do not create a flow that runs an infinite loop, in which the flow sleeps, performs some tasks, then goes back to sleep. In this case, the Run History grows until the system runs out of memory.

Creating a Subflow Within a Flow

You can simplify a flow by creating steps from subflows. This way, you can:

- Separate the programming tasks into smaller, more manageable pieces
- Test parts of the flow individually
- Reuse the pieces that you create

For example, in the flow below, the **Windows Health Check** step is a subflow.

A subflow is treated as a single step even though it may contain many operations.

Subflows often generate data that steps in the parent flow need to access. Flow variables that you create within a flow cannot be referenced outside that flow. However, you can pass values from a subflow to a parent flow, by saving the subflow results as a ***flow output field***.

Best Practices

- A flow should fit on the canvas on a 1024 x 768 screen with Studio maximized and a 1:1 view magnification. Larger flows are not strictly prohibited, but if a flow is larger, examine it carefully to see whether you can break down some of its sequences of steps into subflows.

- Supply a description and name for all transitions on a top-level parent flow. These transition descriptions should describe what happened in the step that preceded the transition. You need not add descriptions of transitions in a subflow unless the data is critical to see during a run.

What do you want to do?

Create a flow with a subflow

1. Create a flow first and save it.

2. Create a new flow, to act as parent flow.

3. Drag the subflow from the **Projects** pane onto the parent flow, to create a step from the subflow.

Pass data from a subflow to a parent flow

1. Open the subflow on the authoring canvas, and open the Step Inspector for the step whose data you want to make available to the parent flow.

2. Click the **Results** tab and add a result (for more information, see "Setting Step Results" on page 520).

3. Configure the result so that result data is stored in a ***flow output field***. This makes the data available outside of the subflow.

Inputs	Results	Display	Description	Advanced	Scriptlet			

Step Results > Result

Name	From	Assign To	Assignment Action	Filters	
Result	Result Field: returnRe...	Flow Variable	OVERWRITE	2 Filters	
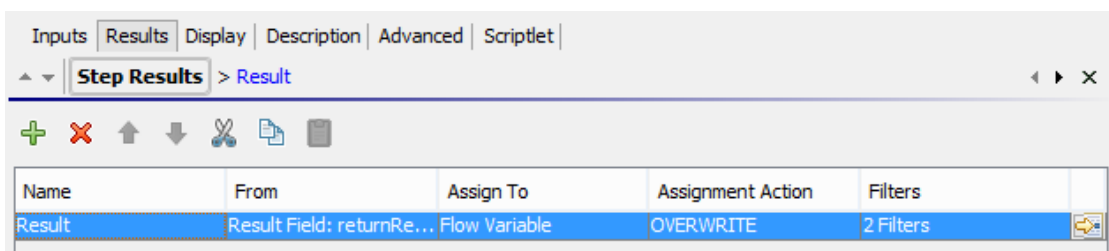
   a. Under **Name**, type a name for the flow output field.

   b. Under **From**, select **Result Field: Result**.

   c. Under **Assign To**, select **Flow Output Field**.

    d.   If required, create a filter to filter the result (for more information, see "Filtering Output and Results" on page 529).

4.   In the parent flow's authoring canvas, open the Step Inspector for the step that was created from the subflow.

5.   Click the **Results** tab and create a step result. By default, this new result:

    ○   Obtains its value from the result field that has the name of the subflow's flow output field

    ○   Has the same name as that of the subflow's flow output field

    ○   Is assigned to a flow variable, which by default has the same name as the result, and which is now available for use in transitions and steps that follow this step

Example

1.   Copy a command operation, and have it execute "dir C:\". Name it **dir**.

2.   Create a flow called **flowdir**.

3.   In the **flowdir** flow, create a step using the operation **dir**.

4.   On the **dir** step, add a result that comes from the operation's output string.

5.   Assign the result to a flow output field, and name the result **foo**. Now the flow has a flow output field also named **foo**.

6.   Create another flow called **parentflow**.

7.   In **parentflow**, create a step from **flowdir**.

8.   Add a result to the **flowdir** step.

    By default, the new result is named **foo**. It obtains its value from the **Result Field: foo**, and the value is assigned to a flow variable also named **foo**. The result **foo** of the subflow's step **dir** is now available to transitions and steps that come after the **flowdir** step in the parent flow.

9.   To test this, after the flowdir step, add a step created from the **Basic Notify** operation.

10.   In this new step:

    a.   Define the `notifyData` input as being a single value, that uses a constant value, and specify that the constant value is `${foo}`.

    b.   Define the `notifyMethod` input as being a single value, which uses a constant value, and specify that the constant value is `Display`.

    c.   Define the subject input as being a single value, which uses a constant value, and specify that the constant value is something like: If this worked, the flow output field says: "contents of

outputString, aka foo".

d. Debug the flow.

Creating a Flow with Parallel Split Steps

A parallel split step is a set of step sequences that are carried out simultaneously. Each series of steps is represented visually in the flow diagram as a lane. The steps contained in each lane are called "lane steps". When you run the flow, the lanes start simultaneously.

Parallel split steps are best used for doing dissimilar things simultaneously and independently of each other. Note the contrast with multi-instance steps, whose instances do the same thing with multiple variations of a single input.

For example, you might use a parallel split step for writing and validating an XML file, and, at the same time, sending email about this to the appropriate person:

- One lane contains the steps for writing and validating the file.
- The second lane sends the email.
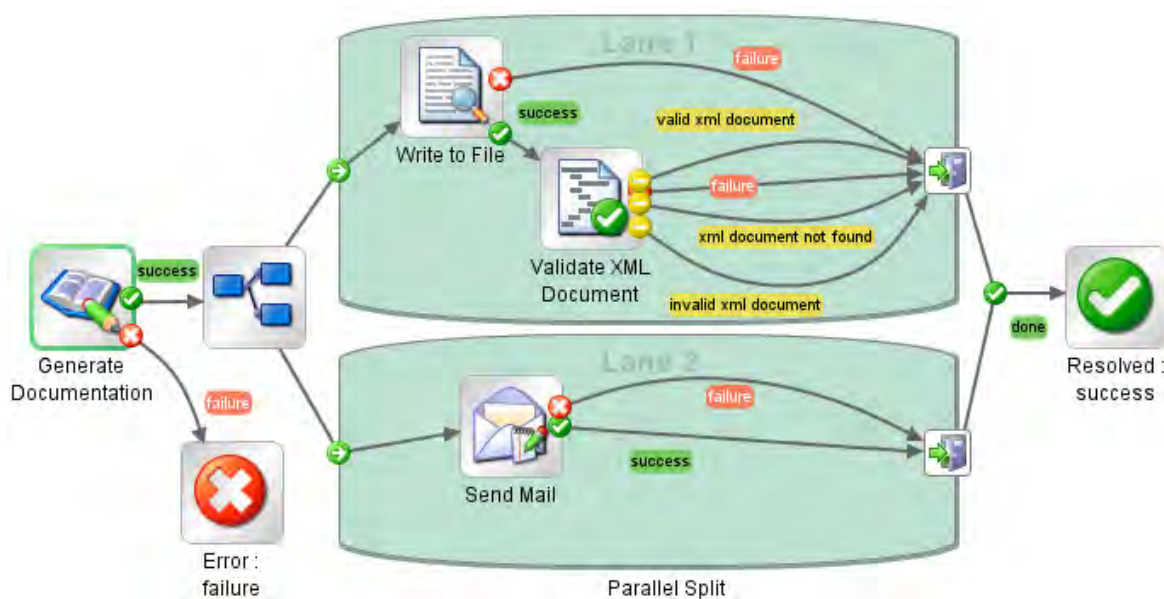
Note: Is not possible for parallel split steps to be non-blocking.

A parallel lane cannot include a response step.

What do you want to do?

Create a parallel split step

1. On the authoring pane toolbar, click the **Step Palette** button to display the **Step** palette.

2. From the **Step** palette, drag the **Parallel Split Step** icon to the authoring canvas. By default, the step has two lanes.

3. Create the step sequence you want within each lane.

   a. Add steps (flows or operations) in the lane.

      > **Note:** You cannot add a response step in a parallel lane.

   b. Connect the steps within each lane.

   c. Connect the last step in the lane to the **Lane-end** icon .

4. Connect the parallel split step to the rest of the flow:

   a. If the parallel split step is not the start step, connect the step that precedes it to the **Parallel Split Step** icon.

   b. Connect the parallel split step's **done** response to the next step in the flow.

Change the visual order of lanes

You can change the visual order of the lanes in the flow diagram, but note that when the flow is run, all the lanes begin at the same time. Their graphical order in the flow diagram does not affect the order in which their processing occurs.

1. Right-click the lane that you want to move.

2. From the drop-down menu, select **Move Lane Up** or **Move Lane Down**.

Move a parallel split step or its components

- To move a parallel split step, click the **Parallel Split Step** icon in the flow diagram, and drag.

- To move an individual lane step, select the step and drag, either within the lane or to another lane.

Copy a parallel split step

1. Right-click the **Parallel Split Step** icon in the flow diagram, and select **Copy**.

2. Right-click on the authoring canvas and select **Paste**.

Copy components of a parallel split step

To copy components of the parallel split step, use any of the following tools:

- The **Edit** > **Copy** and **Edit** > **Paste** menu commands

- The right-click menu

- Keyboard combinations **Ctrl+C**, **Ctrl+V**

- The **Copy** and **Paste** buttons on the authoring pane toolbar

> **Note:** If you are copying a lane, keep the cursor inside the lane when you carry out the **Paste** command.

Add a new lane

1. Right-click an existing lane.

2. From the drop-down menu, select **Add Lane**.

A new, empty lane is added below the currently selected lane.

Duplicate a lane

1. Right-click an existing lane.

2. From the drop-down menu, select **Duplicate Lane**.

A new lane with the same title as the one you copied appears directly below it.

Delete a lane

To delete a lane, use any of the following tools:

- The **Edit**> **Remove Lane** menu command

- The right-click menu

- Keyboard combination **Ctrl+X**

- The **Remove** button ❌ on the authoring pane toolbar

Resize a lane

1. Select a lane by clicking in a blank part of it. Handles appear at the sides and corners.

2. Drag the side or corner handles.

Rename a lane

By default, lanes are called **Lane 1**, **Lane 2**, and so on.

1. Right-click the lane and select **Rename**.

2. In the text box that appears, type the new name of the lane.

Change the start step of a lane

Note that the start step of a lane does not have the green outline that the start step of a flow does.

Drag the **Lane-start** icon 🔄 connector from the lane step that is its current target to the step that you want to be the lane's start step.
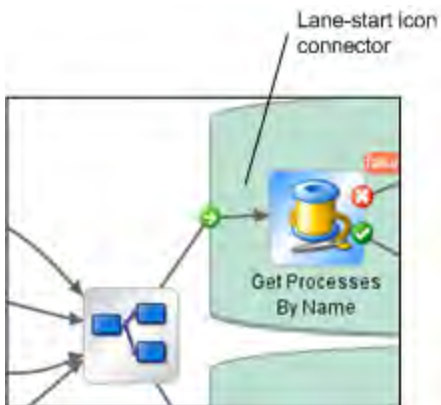
Move data into and out of a parallel split step

When a parallel split step starts, each of its lanes obtains copies of the global context flow variables, local context variables, and the inputs of the parallel split step itself. Each lane can use these variables and can create, modify, or delete these variables according to normal flow rules, independently of any other lane.

A step within one lane cannot pass values to a step in another lane. The steps in each lane only have the values that were available when the parallel split step (all its lanes) started.

As the lanes finish execution, the flow variables in each of the contexts are merged back into the context of the calling flow (the flow that the parallel split step is part of). The order of merging is the order in which the lanes terminate. As a result, if two lanes write to the same flow variable, the last one to finish provides the final value of the variable.

Steps within a parallel split step's lanes can obtain data from the local and global contexts and save data to local context. Lane steps can only write to the global context by means of a scriptlet that uses the `scriptletcontext.putGlobal()` method. For the syntax for using `scriptletcontext.putGlobal()`, on the **Scriptlet** tab of an operation or step, insert the **JavaScript** template.

Debug a parallel split step

In an actual run, the lanes will start and run simultaneously when the flow is run, but when you test them in the debugger, they are executed as a series. You cannot control the order in which lanes are run in the debugger, but by giving them unique names, you can see the order in which they ran.

This is one way in which the debugger does not precisely reproduce the behavior of a flow in a production environment. On the other hand, serial execution in the debugger of parallel split steps enables you to perform controlled tests for various conditions. For more information, see .

Creating a Flow with Multi-Instance Steps

A multi-instance step is a step that executes simultaneously on multiple targets. For example, if you want to run the Windows Diagnostic flow on 100 servers, you can create a multi-instance step that runs the flow on all 100 servers at the same time.

The targets of the operation (in our example, the 100 servers) are defined in an input list in the multi-instance step.

Inside a multi-instance step, you can include one or more operations or subflows. The operations and/or subflows in the multi-instance step run once for each target—these runs are known as *instances*.

Each instance gets, at its beginning, a duplication of the global and local contexts. As it runs, each step in the instance can change the global variables, flow variables, and flow output fields within the multi-instance step.

> **Note:** When an exception is thrown in one of the instances, that instance is stopped. The others continue to run, because they are running in parallel.

If running the multi-instance step against too many targets simultaneously would slow down your infrastructure, you can throttle a multi-instance step, by limiting how many values it can process at once. The throttling size can be a constant (integer) or take its value from a ${variable}.

> **Note:** Is not possible for multi-instance steps to be non-blocking.
>
> A multi-instance lane cannot include a response step.

Differences Between a Multi-Instance Step and a Parallel Split Step

In a multi-instance step, each instance performs the same task on a different target, while in a parallel split step, each parallel step can be set to do something different.

In a multi-instance step, the number of instances can change during runtime, while in a parallel split step, the number of parallel steps is constant.

Saving Flow Data

Flow variables, global variables, and flow output fields that are created in an instance of a multi-instance step are local to the instance in which they are created and populated. These variables and flow output field variables will disappear at the end of the lane, unless you use one of the following ways to make this data available to the rest of the flow:

- Bind the data to results on the multi-instance step
- Create a scriptlet in the multi-instance step to save the data

Saving Data via Results

In order to make the data from flow variables available after the multi-instance step has ended, you can define step results in the multi-instance step, which take their value from flow variables created in the instances. In the **Results** tab of the Step Inspector for a multi-instance step, select a flow variable created in the instances, by selecting **Result <result>** in the **From** column.

You can also save the data from flow output fields created in a subflow in the instances, in a similar way. In the **Results** tab of the Step Inspector for a multi-instance step, select a flow output field created in the instances, by selecting **Result Field <result>** in the **From** column.

You can set the **Assignment Action** field to do different things with the values that are collected. For example, you could append the results of the different instances, or add them together, or get later instances to overwrite previous ones.

In the example below, there are five variables set up for the results of a multi-instance step. The first three take their value from flow variables and the last two from flow output fields.

| Inputs | Results | Display | Description | Advanced | Scriptlet |

Step Results

Name	From	Assign To	Assignment Action	Filters	
var1	Result Field: Result	Flow Variable	OVERWRITE	No Filters	
var2	Result Field: Result	Flow Variable	PREPEND	No Filters	
var3	Result Field: returnRe...	Flow Variable	OVERWRITE	No Filters	
var4	Result Field: TimedOut	Flow Variable	OVERWRITE	No Filters	
var5	Result Field: returnRe...	Flow Variable	OVERWRITE	No Filters	

Assuming that there are two instances, **Instance1** and **Instance2**, that the main flow has empty contexts, and that **Instance2** ends after **Instance1**, the instances provide the following variables:

- **Instance1**:
  - Flow variables:

    ```
    var1 = x

    var2 = y

    var3 = w
    ```
    - Flow output fields:

      ```
      var1 = z
      ```
- **Instance2**:

- Flow variables:

    ```
    var2 = t
    var3 = v
    ```

- Flow output fields:

    ```
    var5 = u
    ```

When the multi-instance step finishes, the values of the variables will be:

`var1` = `NULL` (because in **Instance2**, there is no value for this variable, and the action is to overwrite)

`var2` = `t` (the value in **Instance2** overwrites the value from **Instance1**)

`var3` = `wv` (the value in **Instance2** was appended to the value from **Instance1**)
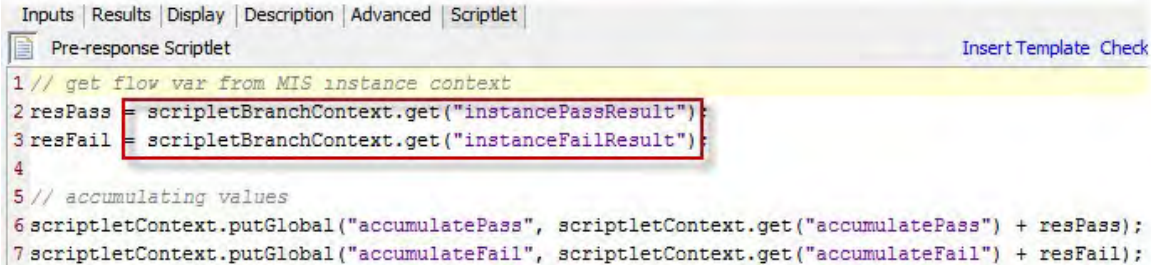
`var4` = `NULL` (because in **Instance2**, there is no value for this variable, and the action is to overwrite)

`var5` = `u`

Saving Data via a Scriptlet

Another way to make data generated within the step available to the rest of the flow is by creating a scriptlet that collects the data and saves it as a variable that will continue to exist after the instance's run completes.

In the example shown, the scriptlet tracks whether each instance run passes or fails, accumulates this data, and saves it as a variable that is available in the global context.

This scriptlet will be executed multiple times, once for each instance. Each time, it will be able to access the `ScriptletContext` of the current instance (called `scriptletBranchContext`), and can modify the parent flow context (by accessing the `scriptletContext`).

`scriptletBranchContext` has same method access as the `scriptletContext`.

For more information about scriptlets, see "Using Scriptlets in a Flow" on page 574.

Merging after Upgrade

After upgrading from a previous version of OO, if a flow contains multi-instance steps that were created using the **Toggle Multi-instance** option, the global variables created in the step are updated, with later instances overwriting earlier ones.

What do you want to do?

Create a multi-instance step

1. On the authoring pane toolbar, click the **Step Palette** button to display the **Step** palette.

2. From the **Step** palette, drag the **Multi-instance** icon to the authoring canvas.

3. From the **Projects** pane, drag the flow or operation into the multi-instance lane.

   Note: You can add multiple flows and operations to the multi-instance lane.

   You cannot add a response step in a multi-instance lane.

4. Set up the list of targets for the multi-instance step, by creating an input that is a list of multiple values. For example, the list of servers that the flow will run against:

   a. Open the Step Inspector for the multi-instance step by double-clicking the **Multi-step** icon at the start of the step.

   b. Create an input. In our example, this could be named **servers**.

   c. Select the **Required** checkbox and set the type to **List of Values**.

Inputs	Results	Display	Description	Advanced	Scriptlet

Input	Required	Type	Assign From	Otherwise	
servers	☑	🗄 ▾	*servers* ▾	Use the constant: server1,server2,server3	🗛

   d. Click the right-pointing arrow at the end of the row to open the Input Editor for that row.

   e. In the **Input Delimiter** box, type a delimiter (a character(s) that separates the elements in the list).

      Note:

- To define a delimiter that contains several characters, use quotation marks. For example, `"$%^"`.

- To use a quotation mark as part of the delimiter, type it in twice. For example, to define the delimiter `"$%`, type `""$%`.

f. Specify the way that the list of values will be input. For example, if you want the multi-instance step to run against a number of servers, you can select **Use Constant**, and specify the server names in the **Constant Value** box. Other ways to populate the list of values are to use the results of a previous step or via integration with another program.
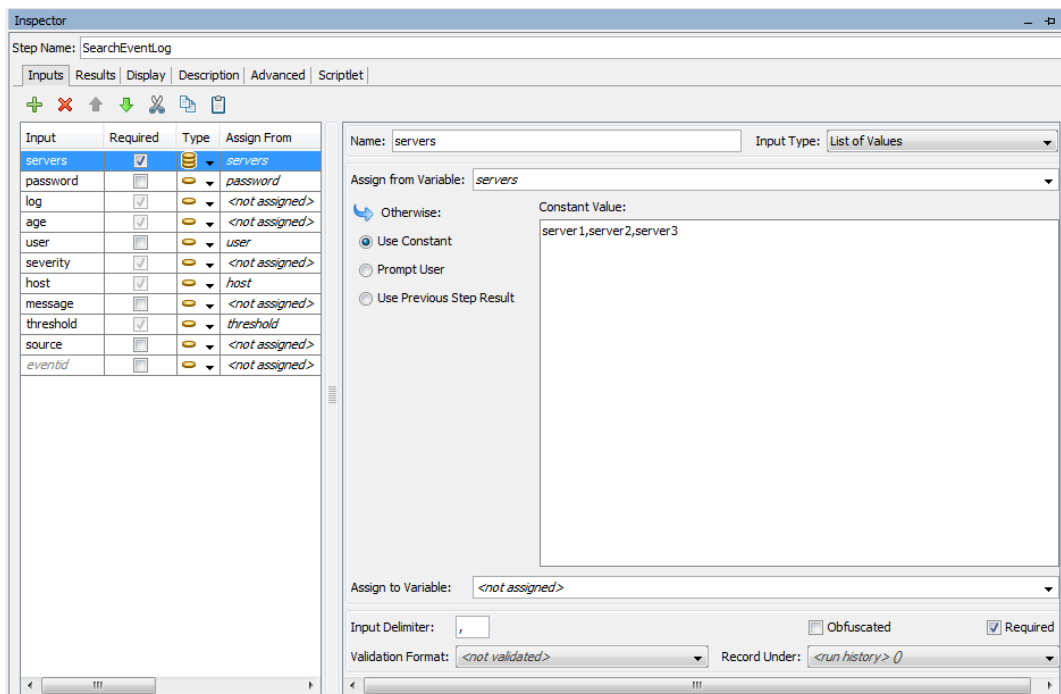
For more information about the options for creating a list of values for input, see "Specifying the Input Source " on page 482.

5. Connect the different parts of the multi-instance step:

a. Connect the **Lane-start** icon to the first step in the multi-instance lane.

b. If there are multiple steps in the multi-instance step, connect the steps.

c. Drag all the response lines from the last step in the lane to the **Lane-end** icon .

6. Apply the list of targets to each internal step:

   a. For each of the internal steps inside the multi-instance lane, open the Step Inspector and add an input.

   b. Open the Input Editor, and in the **Assign From** list, select the variable that you created to hold the list of targets. In our example, this is **servers**.

7. If you want to save the data collected by the different instances of the multi-instance step, create a flow variable to store the result:
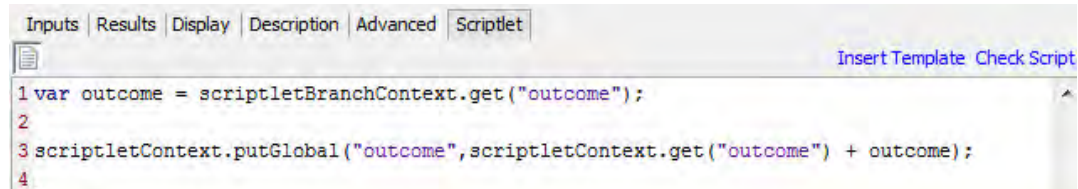
   a. Open the Step Inspector for the multi-instance step by double-clicking the **Multi-step** icon.

   b. Click the **Results** tab and add a result.

   c. In the **Assign To** column, assign the result to a flow variable.

   d. Give a name to the flow variable that will hold the data, for example, **outcome**.

   e. Decide on how you want the data to be stored. In our example, we want to store the results for each server, so the assignment action is **APPEND**. For more details, see *Save output from a multi-instance step*, below.

8. If you want to save the data collected by the different instances of the multi-instance step, to be used in the global context, write a scriptlet to store the result:

a. In the Step Inspector for the multi-instance step, click the **Scriptlet** tab.

b. Write a scriptlet that will collect the data from the `scriptletBranchContext`), and will make it available to the `scriptletContext`.

In the example below, the scriptlet tells the flow to accumulate the all values of the **outcome** variable. This is similar to the **APPEND** action that was selected in the previous step.

```
Inputs | Results | Display | Description | Advanced | Scriptlet
                                                    Insert Template  Check Script
1 var outcome = scriptletBranchContext.get("outcome");
2
3 scriptletContext.putGlobal("outcome",scriptletContext.get("outcome") + outcome);
4
```
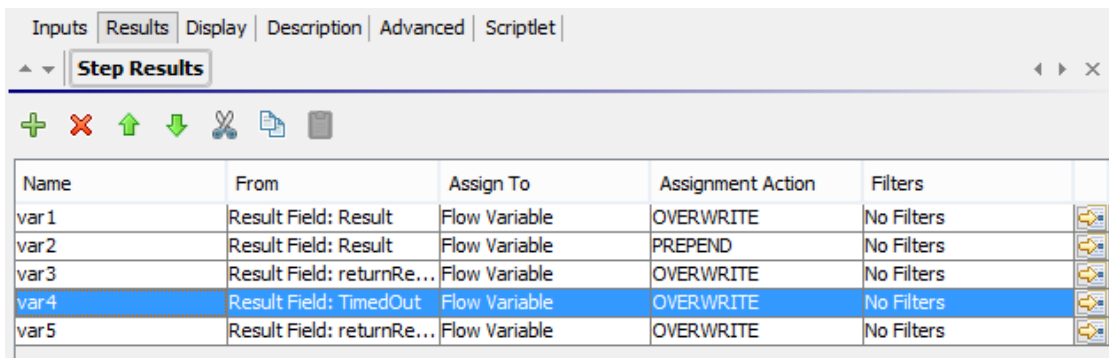
9. Connect the multi-instance step to the rest of the flow:

a. If the multi-instance step is not the start step, connect the step that precedes it to the **Multi-instance** icon.

b. Connect the multi-instance step's **done** response to the next step in the flow.

Save output from a multi-instance step

The data in flow variables and flow output fields in instances are gone after the multi-instance step is completed. To save this data, you can bind it to results in the multi-instance step.

1. Create a multi-instance step, as described above.

2. In the Step Inspector, click the **Results** tab.

3. Add a result line for each flow variable that you want to save.

```
Inputs | Results | Display | Description | Advanced | Scriptlet
     Step Results                                          ◄ ► ✕
  ✚ ✖ ⬆ ⬇ ✂ 📋 🗑
```

Name	From	Assign To	Assignment Action	Filters	
var1	Result Field: Result	Flow Variable	OVERWRITE	No Filters	
var2	Result Field: Result	Flow Variable	PREPEND	No Filters	
var3	Result Field: returnRe...	Flow Variable	OVERWRITE	No Filters	
var4	Result Field: TimedOut	Flow Variable	OVERWRITE	No Filters	
var5	Result Field: returnRe...	Flow Variable	OVERWRITE	No Filters	

4. In the **Name** column, enter a name for the flow variable that you will be saving the data to.

5. In the **From** column, select the flow variable or output field that is the source of the data that you want to save.

- To select a flow variable created in the instances, select **Result <result>** in the **From** column.

- To select a flow output field created in the instances, select **Result Field <result>** in the **From** column.

6. In the **Assignment Action** column, select the action that describes how you want to collect the data.

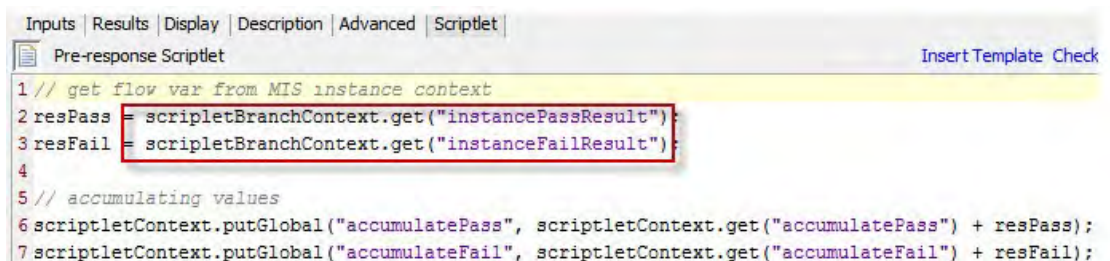   For example, if you wanted to calculate how long it took to run all the instances, you would select **Add**. If you wanted to collect a list of all the servers that were checked in the multi-instance step, you would select **Append**.

7. Save the step. The flow variables that you created will be available for the rest of the flow, after the multi-instance step has finished running.

Save output from a multi-instance step as a global variable

To save the output from a multi-instance step, so that it can be used outside the flow, you can create a scriptlet to save this output as a global variable.

1. Open the Step Inspector for the multi-instance step by double-clicking the **Multi-step** icon at the start of the step.

2. Click the **Scriptlet** tab.

3. Write a scriptlet that will collect the data from the `scriptletBranchContext`), and will make it available to the `scriptletContext`. For example:

```
Inputs | Results | Display | Description | Advanced | Scriptlet |

Pre-response Scriptlet                                          Insert Template  Check
1 // get flow var from MIS instance context
2 resPass = scripletBranchContext.get("instancePassResult");
3 resFail = scripletBranchContext.get("instanceFailResult");
4
5 // accumulating values
6 scriptletContext.putGlobal("accumulatePass", scriptletContext.get("accumulatePass") + resPass);
7 scriptletContext.putGlobal("accumulateFail", scriptletContext.get("accumulateFail") + resFail);
```

Limit (throttle) the number of instances a multi-instance step can run on simultaneously

If running a multi-instance step on many instances simultaneously may slow down your system's performance, you can set a throttle level for the step, by limiting the number of targets the multi-instance step can simultaneously start the flow for.

The performance is related to the target system(s), and whether or not one flow can use all of OOs worker threads.

If throttle is not set, the number of instances is the same as the number of inputs.

If throttle is set, the number of instances is the minimum between throttle size and remaining inputs.

1. Open the Step Inspector for the step, and then click the **Advanced** tab.

2. Under **Execution**, select the **Throttle parallel execution** check box.

3. In the box, type the maximum number of instances of the step that should run at once.

Note for upgraders from OO 9.x:

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

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

- 10.x starts 10 of the 100, waits until all 10 are finished, and then starts another 10.

Please take this into account when you schedule your patching.

Move a multi-instance step

1. Select the **Multi-instance** icon at the start of the lane, which represents the entire step.

2. Drag the step across the authoring canvas.

Resize a multi-instance step

1. Select the lane by clicking in a blank part of it. Handles appear at the sides and corners.

2. Drag the side or corner handles to resize the lane.

Rename a multi-instance step

1. Select the **Multi-instance** icon at the start of the lane.

2. Right-click and select **Rename**.

3. Type a new name in the text box.

Debug a multi-instance step

In an actual run, the multiple instances run concurrently, but when you test them in the debugger, they are executed serially. While this means that you are not testing under actual conditions, it does allow

you to examine how long it takes each instance to finish.

For more information, see "Debugging Complex Flows" on page 609.

Using Scriptlets in a Flow

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

You can use scriptlets to test, format, manipulate, or isolate a particular piece of the results.

You can use scriptlets to:

- Filter the results of a operation, flow, or step

- Determine the response of an operation

- Manipulate data in a subflow before passing the data to the parent flow

Note: When referencing a system property in a scriptlet, you must use the complete path. For example, if there is a system property under the following folder structure `folderA\folderB\my_ci`, use the string `${folderA/folder/my_ci}` to reference it.

Resources to help you write scriptlets

- Scriptlet templates (in Nashorn or Rhino JavaScript) are available in the Scriptlet Editor.

- Default scriptlets are available in the **Configuration\Scriptlets** folder.

- Copy existing scriptlets in default content.

you can get information from Global Context parameters:

${execution_userid} - The value is the user name of the user who is logged in and running the flow

${run_id} - The ID value of runned flow

${execution_thread_id} - The internal thread ID value

What do you want to do?

Create a scriptlet from a template

1. Open the **Properties** sheet or Step Inspector:

   ○ To add a scriptlet to an operation, right-click the operation in the **Project** pane and select **Properties**.

   ○ To add a scriptlet to a flow, right-click the flow in the **Project** pane and select **Properties**.

   ○ To add a scriptlet to a step, double-click the step in the authoring pane.

2. Select the **Scriptlet** tab.

3. From the **Type** list, select the JavaScript language in which you will write your script: Nashorn (recommended) or Rhino.
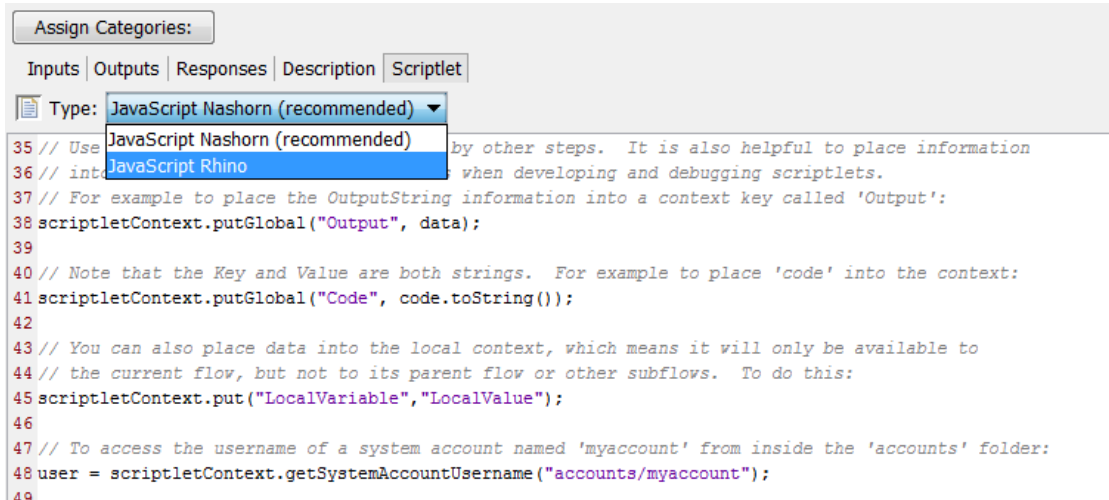
4. Click **Insert Template**.

5. Follow the guidelines in the template to write your script.

6. Click **Check Script** to check for errors.

7. Save.

Use an existing scriptlet

1. Open the **Properties** sheet or Step Inspector:

   ○ To add a scriptlet to an operation, right-click the operation in the **Project** pane and select **Properties**.

- To add a scriptlet to a flow, right-click the flow in the **Project** pane and select **Properties**.

- To add a scriptlet to a step, double-click the step in the authoring pane.

2. Select the **Scriptlet** tab.

3. Open an existing scriptlet in a separate window:

- Double-click a scriptlet from the **Configuration\Scriptlets** folder.

- Open an operation that contains scriptlets (for example, the operations in the **Operations\Operating Systems\Linux\Red Hat** folder).

4. Copy the scriptlet text and paste it into the **Scriptlet** text box for your operation, flow, or step.

5. Modify the scriptlet if required.

6. Click **Check Script** to check for errors.

7. Save.

Filter step or flow results with a scriptlet

You can filter step or flow results using a scriptlet.

1. Double-click a step in the authoring pane.

2. Select the **Results** tab and select the result that you want to filter.

3. Click the right-pointing arrow ⬙ at the end of the result row to open the Filter Editor.

4. In the Filter Editor, click the **Add** button.

5. From the **Select Filter** list, select **Scriptlet**.

6. Create a scriptlet to filter the data, in one of the following ways:

- Click **Insert Template** to use the scriptlet template as a basis.

- Copy and paste the text from an existing scriptlet in another operation or from the **Configuration\Scriptlets** folder.

   For more information about creating filters, see "Filtering Output and Results" on page 529.

7. Click **Check Script** to check for errors.

   If an error is found in a scriptlet, the error is marked with a red underline, as is the flow that contains the scriptlet.

   > **Note:** Flows containing invalid scriptlets are not included when you create content packs from a project. See "Exporting a Content Pack" on page 619.

8. Test the filter and save your work.

> **Tip:** When creating a scriptlet operation, in the scriptlet, specify the scriptlet response as success. Then, on the **Responses** tab of the operation, select failure as the default response.

Create a scriptlet rule for an operation response

You can use a scriptlet to control the response in an operation.

1. Open the **Responses** tab of the operation and select a response.

2. Click the right-pointing arrow ![icon] at the right end of the response's row, to open the Rule Editor.

3. From the **Rule Type** list, select **Scriptlet**.

4. Click the right-pointing arrow ![icon] at the right end of the rule's row, to open the Rule Details Editor.

```
Assign Categories:

Inputs | Outputs | Responses | Description | Scriptlet

Type: JavaScript Nashorn (recommended) ▼
        JavaScript Nashorn (recommended)
35 // Use                              by other steps.  It is also helpful to place information
        JavaScript Rhino
36 // int                              when developing and debugging scriptlets.
37 // For example to place the OutputString information into a context key called 'Output':
38 scriptletContext.putGlobal("Output", data);
39
40 // Note that the Key and Value are both strings.  For example to place 'code' into the context:
41 scriptletContext.putGlobal("Code", code.toString());
42
43 // You can also place data into the local context, which means it will only be available to
44 // the current flow, but not to its parent flow or other subflows.  To do this:
45 scriptletContext.put("LocalVariable","LocalValue");
46
47 // To access the username of a system account named 'myaccount' from inside the 'accounts' folder:
48 user = scriptletContext.getSystemAccountUsername("accounts/myaccount");
49
```

5. Create the scriptlet in one of the following ways:

   - Click **Insert Template** to use the scriptlet template as a basis.

   - Copy and paste the text from an existing scriptlet in another operation or from the **Configuration\Scriptlets** folder.

6. Create the scriptlet and click **Check Script** to check for errors.

7. Save your work.

Use a system scriptlet in an operation, flow, or step

1. Open the **Scriptlet** tab of the **Properties** sheet or Step Inspector for the operation, flow, or step on which you want to use the system scriptlet.

2. In the **Projects** pane, expand the **Configuration** and **Scriptlets** folders.

3. Drag the scriptlet from the **Scriptlets** folder to the **Scriptlet** ![icon] icon in the **Scriptlet** tab of the

Properties sheet or Step Inspector.

The Scriptlet tab shows that there is now a reference to a shared scriptlet.
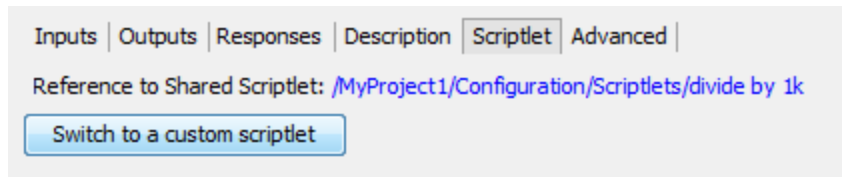
Save a scriptlet to the Configuration\Scriptlets folder

1. Under the **Scriptlet** tab in the **Properties** sheet or Step Inspector, open the scriptlet that you want to save.

2. Drag the **Scriptlet** icon to the **Configuration\Scriptlets** folder in the **Projects** pane.

3. Enter a name for the scriptlet.

Reference Material

Scriptlet Editor

The scriptlet editor has the same appearance, whether you get to it via the **Scriptlet** tab in the **Properties** sheet or Step Inspector or by double-clicking a scriptlet from the **Configuration\Scriptlets** folder.

GUI item	Description
Scriptlet icon	Drag this icon to the **Configuration\Scriptlets** folder, to save a scriptlet

	there for reuse.
Insert Template	Click **Insert Template** to see guidelines to help you write your scriptlet.
Check Script	Click **Check Script** to check the scriptlet for errors.

Using Regular Expressions in a Flow

A regular expression (also known as a regex) allows you to search not only for exact text but also for classes of characters. For example, to match any digit, you can use the wildcard \d.

You can use regular expressions to:

- Create result/output filters that extract key pieces of data for:

  - Saving in variables for use in later operations

  - Testing to determine a step's response

Wildcards and Modifiers for Regular Expressions

The key wildcards for regular expressions are:

Wildcard	Uses
^	Matches the beginning of a string
$	Matches the end of a string
.	Any character except new line
\b	Word boundary
\B	Any except a word boundary
\d	Any digit 0-9
\D	Any non-digit
\n	New line
\r	Carriage return
\s	Any white space character
\S	Any non-white space character
\t	Tab
\w	Any letter, number, or underscore
\W	Anything except a letter, number, or underscore

The modifiers for regular expressions are:

Modifier	Effect
*	Match zero or more
+	Match one or more
?	Match zero or one
{n}	Match exactly n occurrences
{n,}	Match n or more occurrences
{n,m}	Match between n and m occurrences
[abc]	Match either a, b, or c
[^abc]	Match anything except a, b or c

[a-c]	Match anything between a and c
a|b	Match a or b
\	Escape a special character (for example \. Means '.' not match anything

What do you want to do?

Use a regular expression to filter test output

1. Open the Filter Editor for an output or result, and create a new filter. For more information, see "Filtering Output and Results" on page 529.

2. From the **Select Filter** list, select **Regular Expression** as the filter type. The **Details for:** section in the upper-right shows controls for creating a regular expression.

3. From the **Expression Type** list, select **Java Style**. Do not use the other styles; they have been deprecated.

4. In the **Expression Value** box, type a regular expression.

> **Example**: To extract the number of packets lost, you can use the regular expression `Lost = \d`.
>
> This expression tells OO to search for the string "`Lost =`" followed by any number.
>
> The wildcard `\d` tells OO to match any digit.

5. For **Filter Style**, select **Filter Entire Input** or **Filter Line-by-line**, according to how you want the filter applied to the raw results.

6. To make the regular expression not case-sensitive, select **Ignore Case**.

7. Click **Test Selected Filters** to test the filter.

8. Save the filter.

Combine multiple regexes to isolate a value

You can combine multiple regular expressions to isolate the value in a filter.

For example, in the output of the Unix ps command, extracting the time for ps requires two regular expressions: one to filter the output down to the line for ps and the second to extract the time.

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	512	21604	21603	0	75	0	-	1096	wait	pts/1	00:00:00	Bash
0	R	512	2659	21604	0	76	0	-	1110	-	pts/1	00:00:00	Ps

1. Open the Filter Editor for an output or result.

2. Add a new regular expression filter.

3. In the **Expression Value** box, type the first regular expression.

   In our example, type **.*ps**. This extracts any characters ending with "ps".

   > **Note:** Make sure not to omit the leading period [.]

4. Select the **Filter line by Line** check box.

5. Click **Test Selected Filters**.

   In the **Test Output** box, the only output is the line containing "ps".

6. Add a second regular expression filter.

7. In the **Expression Value** box, type **\d*:\d*:\d***

   This represents three sets of digits separated by colons. In our example, this will extract the time from the line.

8. Click **Test Selected Filters**.

9. Save.

The test output now shows only the time from the ps line. Now you can assign this value to a variable.

Reference Material

Filter Editor > Details for: Regular Expression

When you select **Regular Expression** as the filter type, the **Details for:** section in the upper-right shows controls for creating and modifying a regular expression.

Details for: Regular Expression
Filter the input string by extracting the parts that match a regular expression. See the help documentation for a detailed description of regular expressions.

Expression Type: Java Style ▼
Expression Value: .*
Filter Style: ◉ Filter Entire Input
○ Filter Line-by-line
Ignore Case: ☐

GUI item	Description
Expression Type	Select **Java Style** as the type of regular expression to filter the data with. Do not use the other styles; they have been deprecated.
Expression Value	Type the regular expression.
Filter Style > Filter Entire Input	Select to apply the filter to the entire raw result.
Filter Style > Filter Line-by-line	Select to apply the filter to each line separately.
Ignore Case	Select to make the regular expression not case-sensitive.

Customizing Flow Transitions

In this section, you will learn how to customize the transitions between the flow's steps in order to be able to obtain flows with a neat look and feel.

Rounded Transitions

To make a transition round in the same way as Studio automatically does:

1. Make sure you *are not* working with the grid enabled.

2. Press **Shift** and with your mouse click place two points on the line.

3. Place your mouse over one of the points and drag it to a desired position.

4. Repeat with the other point until you reach a final good result.

You can add as many points as you want. If you do not like the result and want to restart, just right click on the transition and select **Layout Edges**.
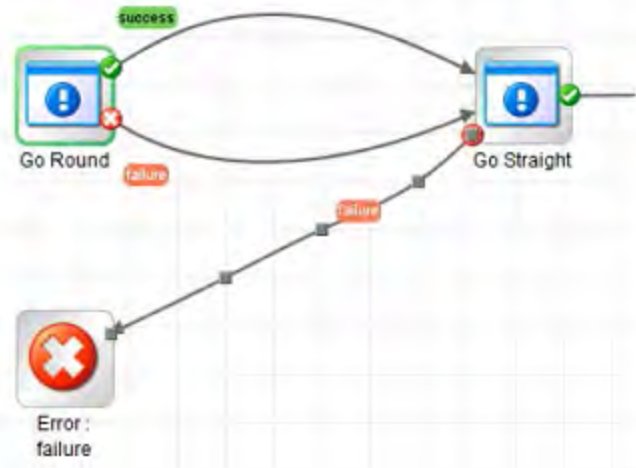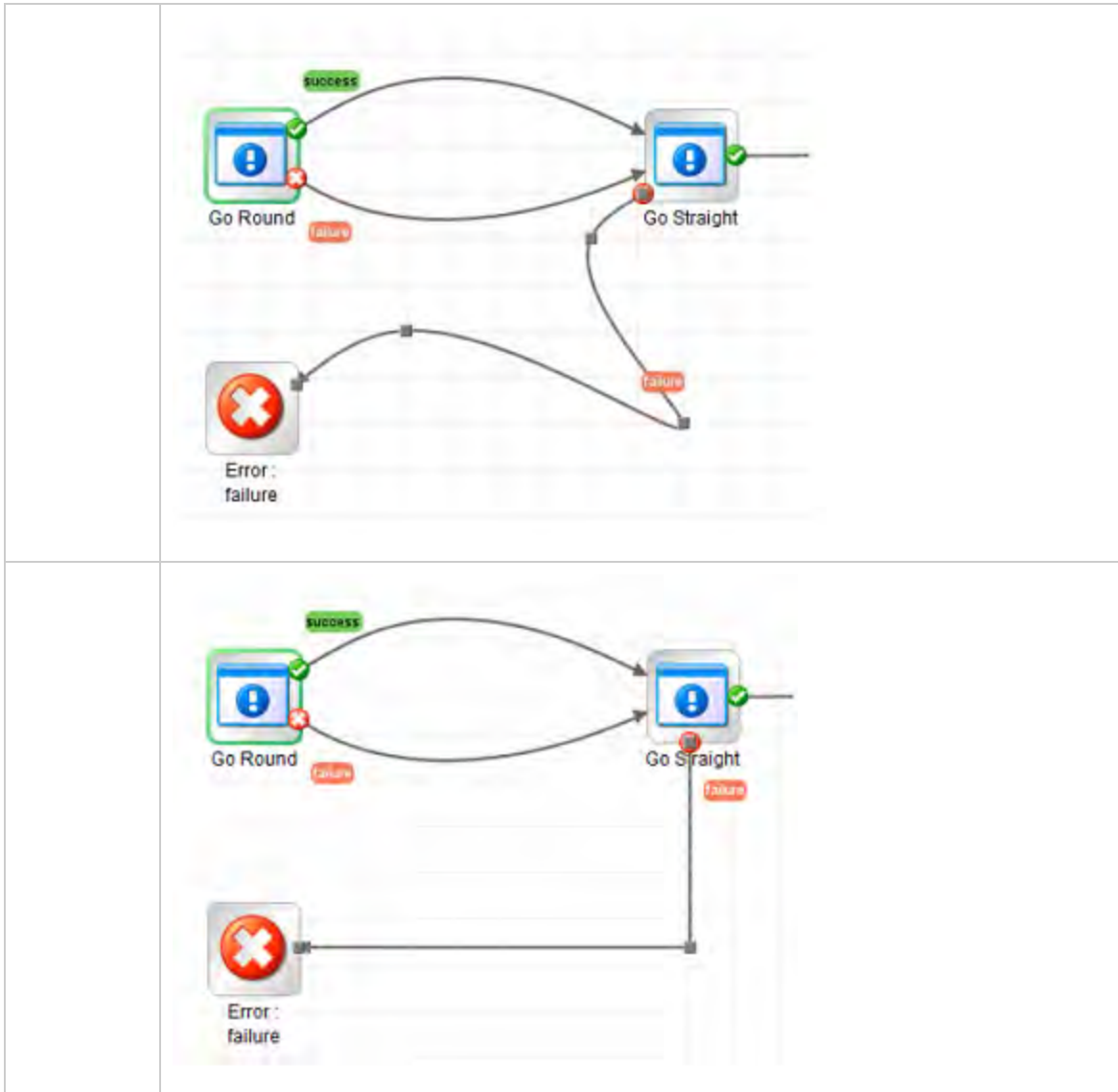
Straight Transitions

1. Make sure you are working with the grid enabled.

2. Press **Shift** and with your mouse click place three points on the line.

3. Place your mouse over the middle point and drag it to what should be a final position.

4. Drag the other two points over it.

5. Select the transition and use the Up/Down/Left/Right arrows to control it and reach the desired result.

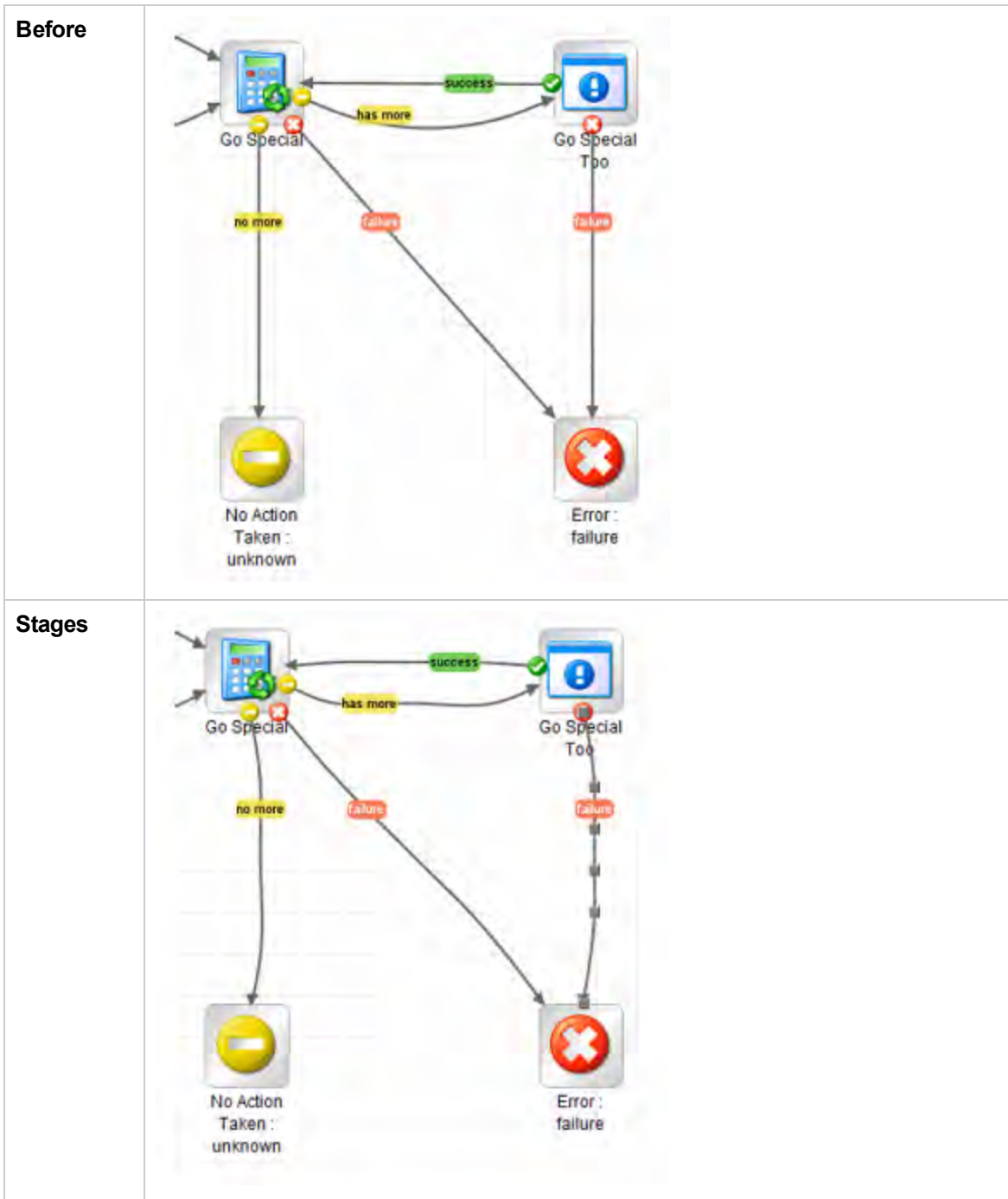|	
Before	
Stages	
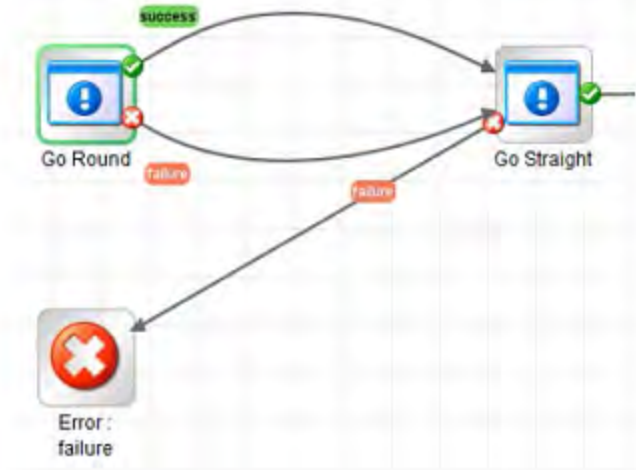
Note: If you do not like the result and want to restart, just right click on the transition and select **Layout Edges**.

Special Transitions

Sometimes there may be situations in a flow when a special transition is needed. For example, when items are too crowded on the canvas, transitions are going back and forth, or text is overlapping. In these cases, follow the procedure below.

1. Make sure you are working with the grid enabled.

2. Press **Shift** and with your mouse click place **four points** on the line.

3. Place your mouse over one of the endpoints and drag it to the final position.

4. Place your mouse over the other endpoints and drag it to the final position.

5. Drag each of the remaining points over the nearest endpoint.

6. Select the transition and use the **Up**/**Down**/**Left**/**Right** arrows to control it and reach the desired result.

Before	
Stages	
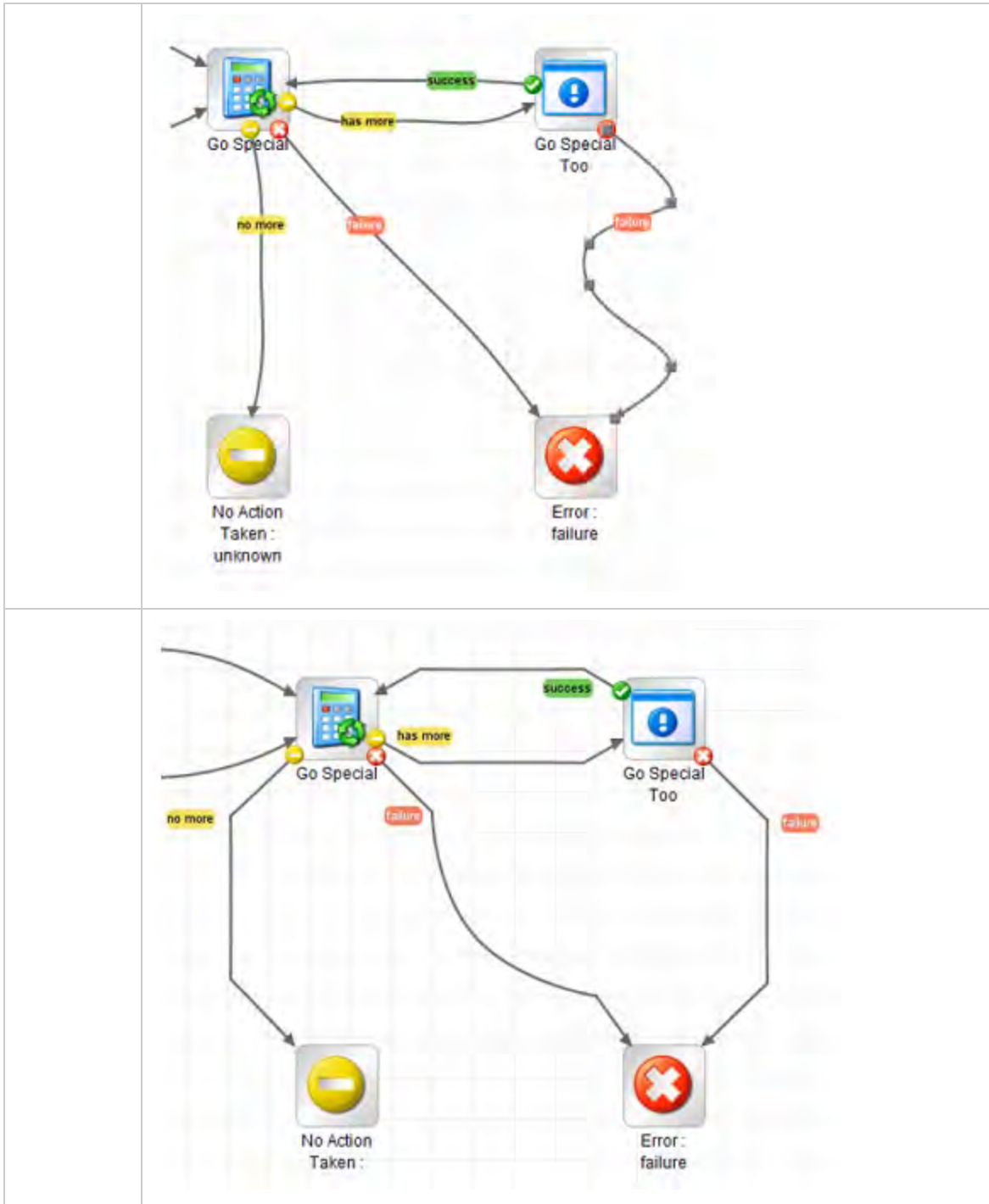
Note: If you do not like the result and want to restart, just right click on the transition and select **Layout Edges**.

Searching Content on ITOM Marketplace from Studio

A Studio author can search for relevant information and content on ITOM Marketplace based on their profile access permissions directly from Studio.

Search Types

The following search types are available:

- **Search by type**: You can search for a flow, operation, system account, system property, selection list, domain term, system evaluator, system filter, scriptlet, group alias, category, and role alias.

- **Search by name**: Searches by name. You can use wild card characters * for incomplete words. The wild card * matches anything. If no name is specified, then any name is a match. This search is case insensitive.

- **Search by description**: Searches the descriptions. You can use wild card characters * for incomplete words. The wild card * matches anything. If no name is specified, then any description is a match. This search is case insensitive.

- **Search by input name**: Searches the input name. You can use wild card characters * for incomplete words. The wild card * matches anything. It will return all items except for configuration items inside all content packs with specific meta-data of each that have inputs matching the search string. This search cannot be used for configuration items.

- **Search by UUID**: Enter the UUID as an exact string. The search is matched based on = operator. This search will return the flow, operation or configuration item with that UUID.

- **Combined search**: Allows you to search with multiple criteria. This includes two or more criteria of type, name, description, input using the AND operation.

> **Note:** A search type with any and all other fields empty is not allowed. In this case, the search button is disabled.

hpln-index-generator Tool

The hpln-index-generator tool, located in **<oo_install_folder>/studio/tools/lib/hpln-index-generator.jar**, generates an index file (.json) describing the flows, operations and configuration items and with the corresponding metadata inside a content pack.

> **Note:** In order to use the ITOM Marketplace search functionality in Studio, the .json file must be

> published to ITOM Marketplace along with the content pack .jar file. See "Run the Index Generator Tool" on the next page for details.

What do you want to do?

## Set up the ITOM Marketplace connection from Studio

1. From the **Configuration** menu, select **Options** > **Live Network**.

2. In the **ITOM Marketplace** dialog box, enter your ITOM Marketplace user ID and password.

> **Note:** The user ID and password that you enter in this dialog box will determine the access permissions and will affect the search results.
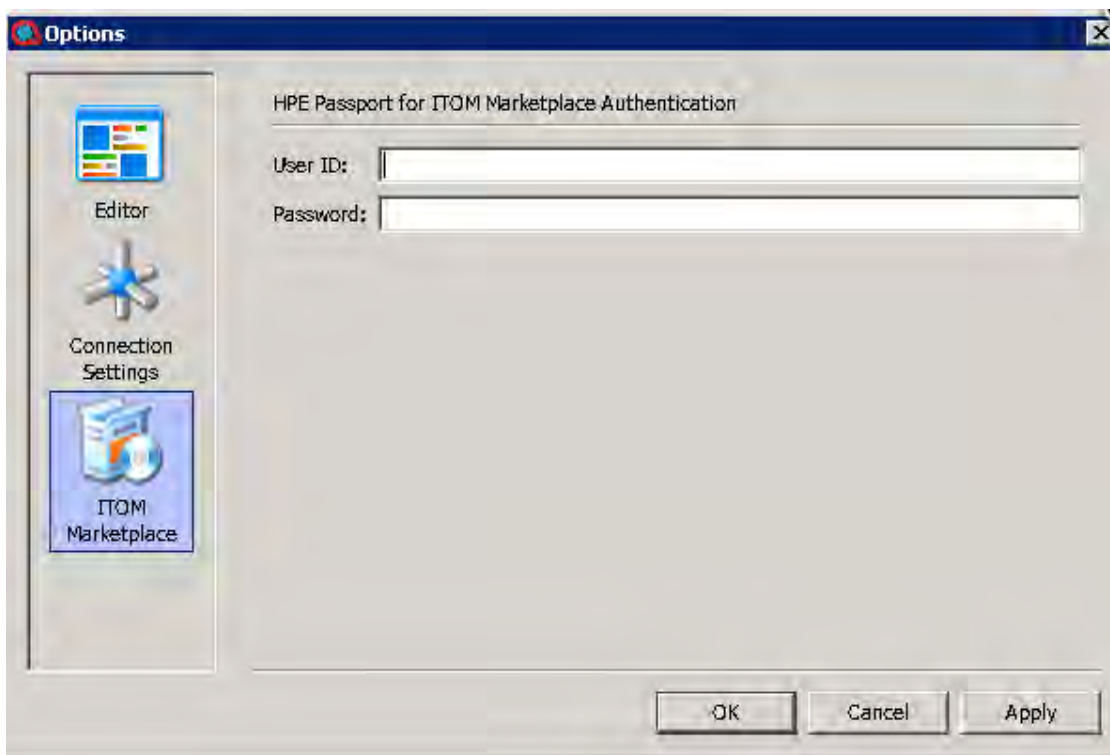
3. Click **Apply** and then **OK**.

Search ITOM Marketplace from Studio

1. From the **Configuration** menu, select **Options** > **Live Network**.

2. In the **ITOM Marketplace** dialog box, enter your ITOM Marketplace user ID and password.

> **Note:** The user ID and password that you enter in this dialog box will determine the access permissions and will affect the search results.

3. Click **Apply** and then **OK**.

   The ITOM Marketplace Search pane is located at the bottom of the Studio Workspace and is open by default.

4. Enter a search criterion and then press Enter or click the Search button.
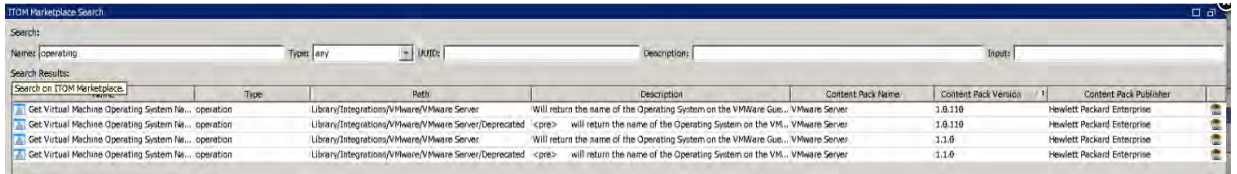
   ITOM Marketplace searches return a maximum of 100 results.

> **Note:** `\, (, ), &` characters are not supported inside the name, description, input, UUID for ITOM Marketplace searches.

Run the Index Generator Tool

1. From the Windows **Start** menu, type `cmd` in the prompt field to open a command line window.

2. Navigate to **studio\tools\lib** under the OO installation folder.

3. At the DOS prompt, type in the following command:

   `hpln-index-generator-<version>.jar --content-pack <arg>`

   where `<version>` is the version number of Index Generator tool, and `<arg>` is the content pack jar file.

   At the end of the process, the tool informs that it has created a JSON file with the content pack name and a .json extension. For example, `oo10-virtualization-cp-1.7.0.json`.
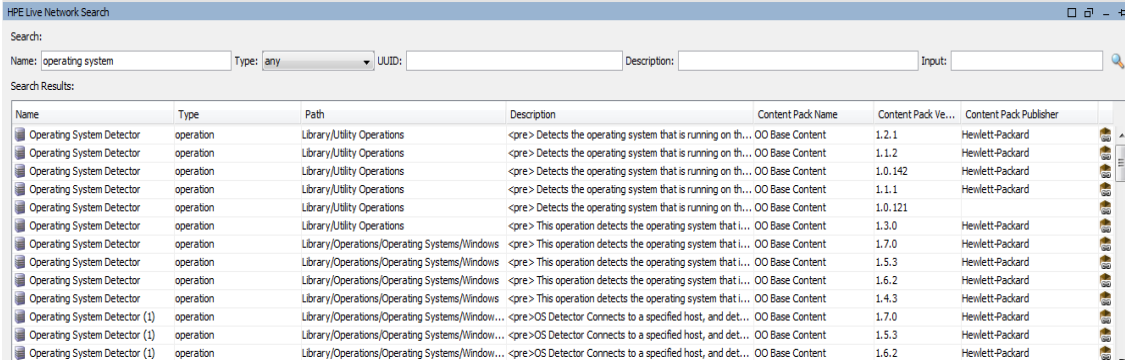
Reference Material

ITOM Marketplace Search Results

The results are displayed in a table which you can sort according to the column name. In addition you can filter the results using multiple values.

When you right-click on a row, you can view the full description of the search result.

You can also jump directly to the location of the content on ITOM Marketplace by clicking the link button ![link button] at the end of a row.

Note: If you want to find a specific item from the latest released content pack, you can filter by the content pack name and then sort descending by the content pack version. The first items in the list are from the latest released version.

Validating Content

Before releasing your content, it is important to test and validate the flows in your project. Studio provides the following tools to help you do this:

- The **Problems** pane displays a list of any problems, with their locations and descriptions, to guide you in repairing these problems

- The debugger helps you track down the causes of errors and unexpected behaviors in flows.

Validating Flows in the Problems Pane

For a flow to run, the flow itself, its operations, configuration items, and any system accounts used in the flow must be valid.

Using the **Problems** pane, you can check an individual flow or operation for problems, or you can validate an entire project. This validates all the flows, operations, configuration items and system accounts in the project.

	Source Type ▲	Name	Description	Location
	Step	Get Row Index by Condition	User Input: hasHeader references missing selection list	/My_Project/Content/Library/Get Cell by...
	Step	Is rowsCount Greater Than 0	Operation cannot be found	/My_Project/Content/Library/Get Cell by...
	Step	SQL Query	User Input: host references missing selection list	/My_Project/Content/Library/Health Ch...
	Transition	success	Transition source step has no operation linked to it	/My_Project/Content/Library/Get Cell by...
	Transition	failure	Transition source step has no operation linked to it	/My_Project/Content/Library/Get Cell by...

Different types of problems may be listed. The following is a partial list:

- The flow has no start/return step
- Operation cannot be found
- Multiple objects have the same UUID
- <variable> references missing selection list
- Transition source step has no operation linked to it

What a flow needs to be valid

To be valid, a flow must have the following:

- At least one step
- One of the steps designated as the start step
- For each step, from each response, a transition connecting the step to a subsequent step
- A way for each step in the flow to be reached in one run or another
- A return step to return a value and end the flow
- Assignment of how each input gets its value

What do you want to do?

Validate a flow or operation

1. Select a flow or operation in the **Projects** pane.

2. Click the **Problems** tab to display the **Problems** pane.

3. Double-click a row in the **Problems** pane to open the item for editing.

   > **Note:** To open multiple items, select them using the SHIFT or CONTROL keys, right-click, and select **Open**.

Validate all items in a project and project dependencies

1. Open the project that you want to validate.

2. From the **Tools** menu, select **Validate**.

A list of any problems appears, with their locations and descriptions, to guide you in repairing the problems.

Resolve duplicate UUIDs

1. Click the **Problems** tab to display the **Problems** pane.

2. Double-click a row containing a duplicate UUID error.

   The item with the duplicated UUID opens. You can then delete the item and proceed to use the original item.

   If you want to keep both items, use the **Copy/Paste** or **Duplicate** commands to save a copy of one of the items with a different UUID, and then delete the item that you copied from.

Reference Material

Problems pane

The **Problems** pane, which you open with the **Problems** tab on the lower edge of the Studio window, lets you check whether a selected flow or operation is valid.

GUI item	Description
Source Type	Displays the type of the element in which there is a problem.
Name	Displays the name of the element in which there is a problem.
Description	Describes the problem, to guide you in how to repair it.
Location	Displays the location of the element with the problem.

Testing and Debugging a Flow

The debugger helps you track down the causes of errors and unexpected behaviors in flows, by displaying the following information:

- A tree showing the steps executed

- Step results and operation outputs generated for each step

- Flow variable values in the various contexts current to each step

- The transition description for each transition followed
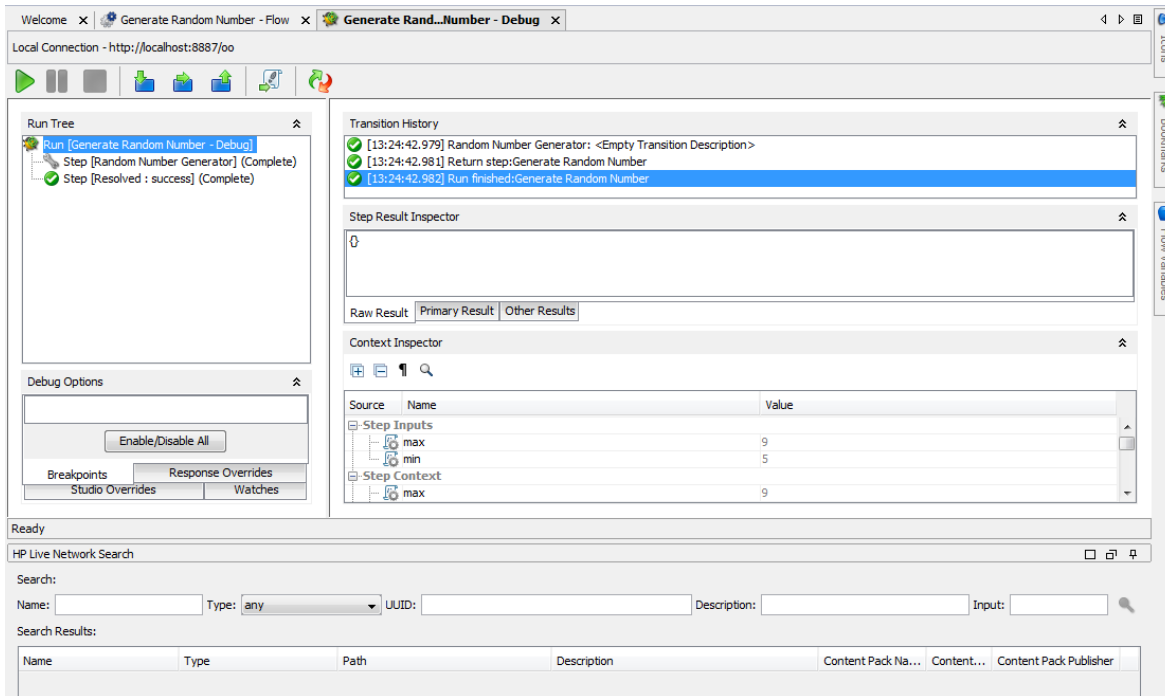
You can also set breakpoints for the debugger and force response choices in order to zero in on the behavior you want to test.

From each step in the debugger, you can jump directly to the relevant step in the flow that opens in the Flow Editor.

Note:

- Values resulting from encrypted or hidden inputs are retrieved with ******characters. You must edit these values in the Context Inspector before triggering the flow, otherwise the value of the applied input will be ****** during the new run.

- In the Context Inspector, you can see the names of hidden outputs, but their values are shown as obfuscated ******.

Debugging Flows that Require User Authentication

If authentication is enabled for a flow in Central (in the **Enable Authentication** check box), when you select to debug the flow, you are prompted to enter user credentials in the Central Authentication dialog box. After they have been entered once, the **Username** and **Password** credentials are remembered for the current session, and you do not need to reenter them when debugging additional flows.
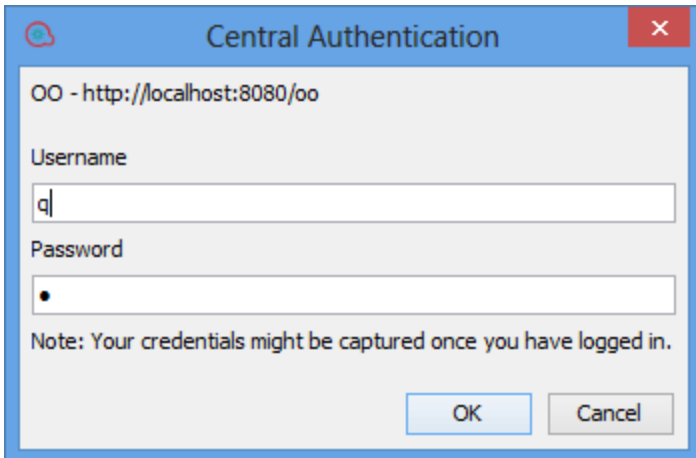
For details on the authentication settings, see "Setting up Security Settings" in the Central Guide.

Note: If authentication has been enabled in Central, you cannot log in with a user that has a colon as part of the user name.

Best Practices

It is recommended to debug subflows before debugging their parent flows.

What do you want to do?

Debug a flow

1. Right-click the flow in the **Projects** pane, and then click **Debug**.

   Note: Alternatively, you can open the flow in the authoring pane and click the **Debug** button.

2. To run the flow to its end, click the **Play** button in the debugger toolbar.

3. To see the information for a completed step, click the step in the **Run Tree** pane.
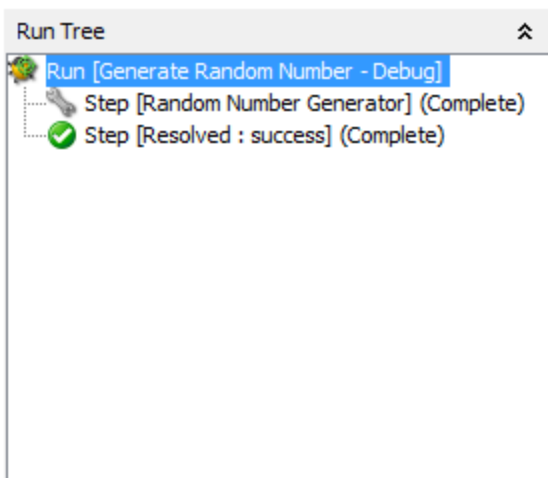
4. In the **Step Result Inspector** pane, you can view the step's raw results, primary result, or other filtered results.
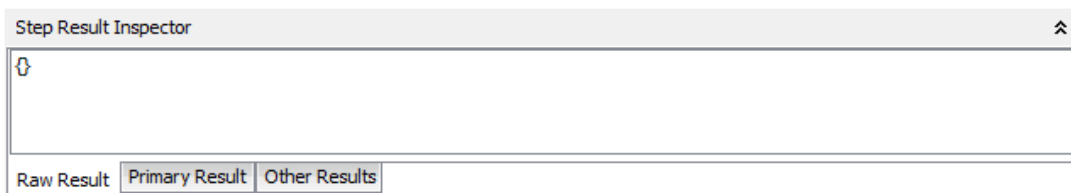
5. To see global variables, flow variables, and their values for the step's inputs and the step and global context, navigate to the appropriate section of the **Context Inspector** pane.
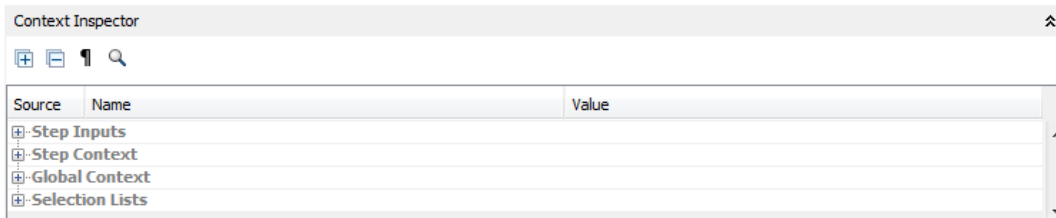
Debug a flow - step by step

1. Open the flow in the debugger.

2. To run the flow step by step, click the **Step Over** button.

Step into and out of a subflow

These actions allow a flow developer to step in and out of the running of a subflow, while debugging a flow. Both actions are available when the debugged flow is paused, waiting for the user's action.

> **Note:** It is recommended to debug subflows before debugging their parent flows.

- To step into a step's subflow, click the **Step Into** button. The debugger will start running the subflow and will pause on the first step of the subflow.

  Clicking **Step Into** at the beginning of the parent flow tells the debugger to pause on the first step of the parent flow.

- To step out of the subflow, click the **Step Out** button. The debugger will run the rest of the steps in the current subflow invocation and will pause on the first step following the subflow (in the parent flow). If the current step is in the parent flow, the action will behave like a resume action.

Collapse/restore panes in the debugger

You may want to collapse some of the panes in the debugger, in order to make more room for another of the panes.

- To collapse a pane, click the upward-facing double chevron ⏫ at the top right of the pane.
- To restore a collapsed pane, click the downward-facing double chevron ⏬ .

Reset and restart a flow in the debugger

When you reset and restart a flow, the values of its flow variables are reset to the values that they had when you opened the debugger.

1. In the debugger toolbar, click the **Reset** button.

2. Click the **Play** button.

Change values of flow variables within the debugger

To see how a flow behaves with different values for its flow variables, you can change the value of a flow variable before running a step.

1. Open the flow in the debugger.

2. Click **Step-Over** until the step in which you're interested is pending.

   > **Note:** If you have set a breakpoint before the step, you can click **Play** to run the flow until it pauses at the step.

The **Context Inspector** pane shows the current values of **Step Inputs** and **Step Context** as of the point at which the step is pending.

You can search, sort, and filter by columns.
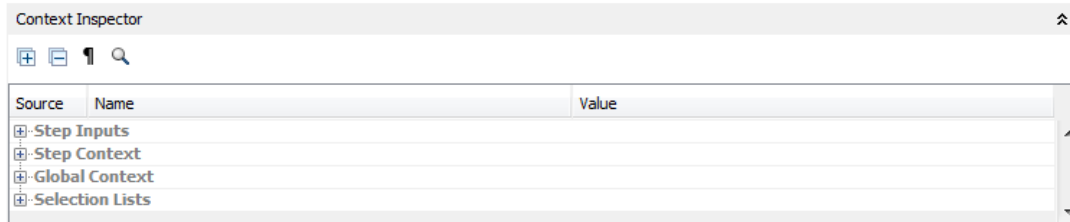
- ○ The values in the **Step Input** section are the values that were assigned to the input before the step started.

- ○ The values in the **Step Context** section are the values that were updated after the step began.

3. To change the value of a flow variable used in this step, under **Step Inputs**, find the listing for the flow variable, highlight its value, and type a new value to replace it. In the above example, the step is a multi-instance step. You could add another IP address to the list in the host flow variable.

4. To change the value of a flow variable that is accessible in this step but is used in a later step, change the value for the flow variable's listing under **Step Context**.

5. Continue to play or step through the flow.

6. To reset any flow variable values that you have changed to the values that were set the last time you saved the flow, click the **Reset** button.

Set a breakpoint in a flow

Breakpoints provide automatic pauses in the running of a flow in the debugger. This can come in handy when you want to, for example:

- Examine the value of a flow variable

- Change the value a flow variable to see its effect on the flow in the rest of the run

You set breakpoints in the flow's diagram, but you can enable or disable any breakpoints that you have set from inside the debugger.

1. Open the flow open in the authoring pane, and right-click the step where you want to set the breakpoint.

2. Select **Debugging** > **Set Breakpoint**.

   In the flow diagram, the breakpoint is indicated by a yellow-and-black border surrounding the step.

List Appender

3. Open the flow in the debugger.

   In the debugger's **Debug Options** pane, the **Breakpoints** tab shows the existing breakpoints.
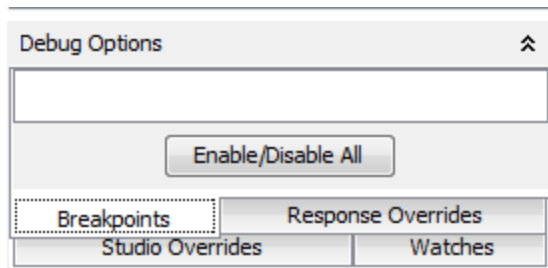
4. Do one of the following:

   ○ To enable a single breakpoint, select the breakpoint's check box.

   ○ To disable a single breakpoint, clear the breakpoint's check box.

   ○ To enable or disable all the breakpoints, click **Enable/Disable All**.

   ○ To clear all the breakpoints, from the **Tools** menu, select **Remove All Breakpoints**.

Override a response in a debug run for a single step

Response overrides force the response that you selected, even if the operation fails.

By overriding a response, you can test a particular path of the flow without having to exit the debugger and change input values.

For example, if you have a step in a flow for which you don't have the necessary information, you might want to test the rest of the flow, regardless of the certain failure of that step. You can force the run to follow the response and transition that you want, rather than the failure response that would come about without your intervention.

1. Open the flow open in the authoring pane, and right-click the step whose response you want to override.

2. Select **Debugging** > **Override Response**, and then click the response you want to force the step to have:

- **None**

- **Success**

- **Failure**

- **Prompt**

After you have created a response override, you can enable or disable the override in the debugger, or choose a different response for it.

3. Open the flow in the debugger.

   In the debugger **Debug Options** pane, the **Response Overrides** tab shows the existing response overrides.
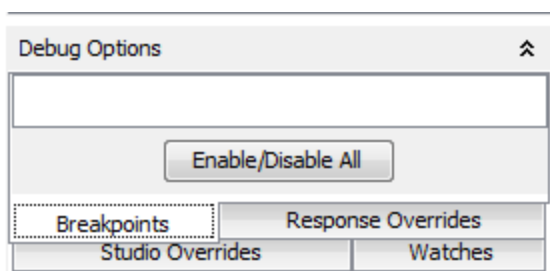
4. Scroll up or down to the response override of interest.

5. Do one of the following:

   - To choose a different response for an override, click the down arrow and select the response.

   - To enable a single response override, select its check box.

   - To disable a single response override, clear its check box.

   - To enable or disable all the response overrides, click **Enable/Disable All**.

   - To clear all the response overrides, from the **Tools** menu, select **Remove All Response Overrides**.

   - To override the response on every step, select the **Override All Responses** checkbox.

     When you run the flow in the debugger after overriding all responses, you are prompted at each

step to manually select a response for the step.
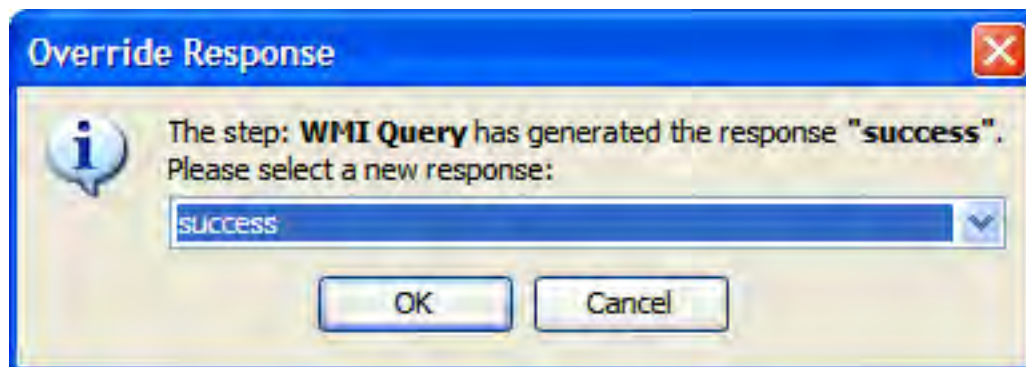
Jump to a flow step from the Run Tree

- Right-click on a step in the Run Tree and select **Go to step**

  or:

  Click the **Go to step** button in the toolbar .

  The Flow Editor opens showing the current flow with the step selected.

Setting Overrides in Studio

In Studio you can define overrides per Central connection, for system properties and system accounts. Creating overrides allows you to change configuration items, system properties, and system accounts (from content packs), before triggering and without changing values in the context inspector. For example you can use these overrides to adjust a read-only value that was imported from a content pack.

> **Note:** You can enable or disable individual overrides. Overrides are unique by name and type, and take precedence over Central overrides.

To access the **Studio Overrides**:

1. From the **Configuration** menu, select **Studio Overrides**.
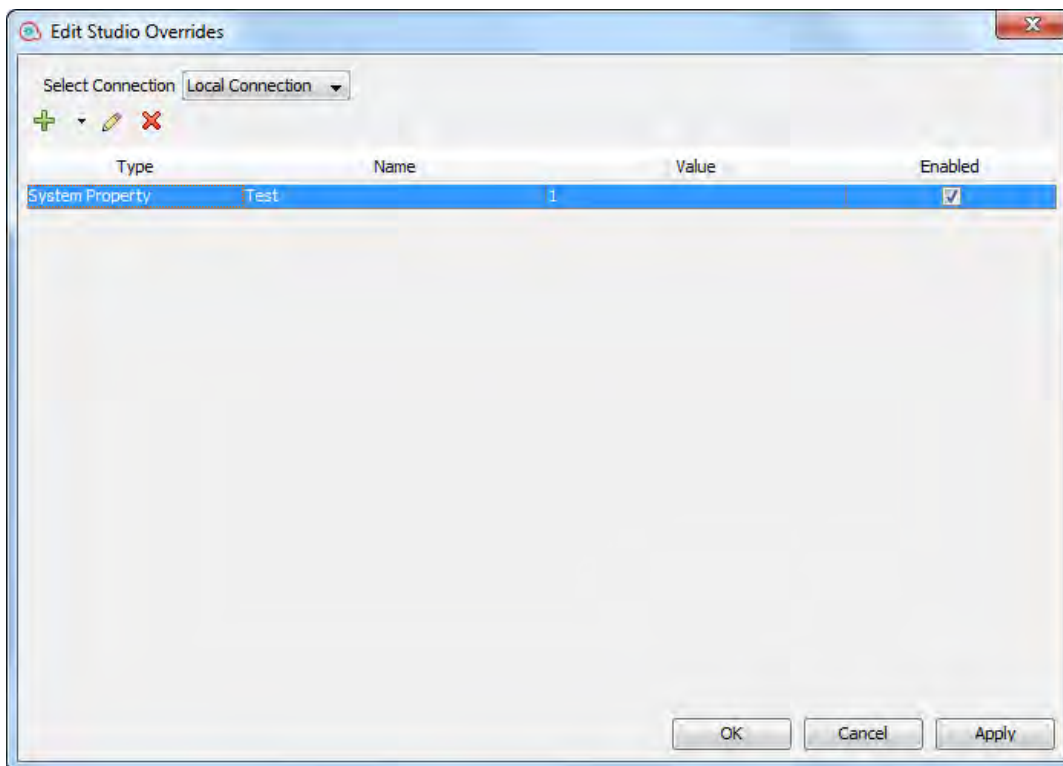
2. Select the Central connection from the **Select Central** drop-down list. When a Central is selected, the overrides of that Central are displayed.

3. You can perform the following:

   ○ ✚ Add a system property or system account override.

   ○ 🖉 Edit the selected system property.

   ○ ✖ Delete the selected system property.

4. Click **Apply**, to apply the changes and **OK** when you are done.

> **Note:** If you override a system property with a blank value, the default value from the **System Properties** definition is used and not the blank value.

> **Note:** When referencing a system property, you must use the complete path. For example, if there is a system property under the following folder structure **folderA\folderB\my_ci**, use the string **${folderA/folder/my_ci}** to reference it.

Reference Material

Run Tree pane
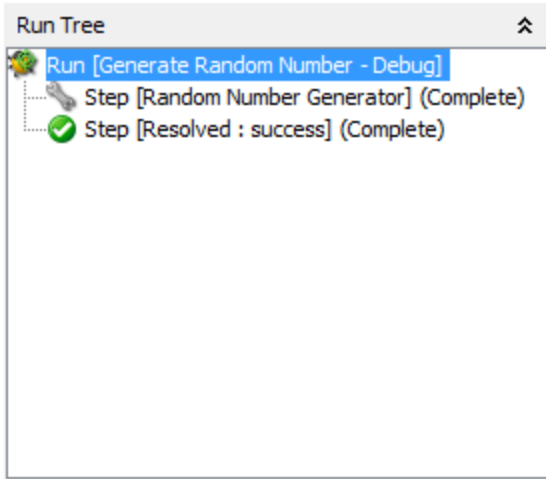
The **Run Tree** pane shows each step that runs, including steps in subflows of the flow.

Steps that will run simultaneously in an actual execution are run in a serial sequence in the debugger.

Debug Options pane
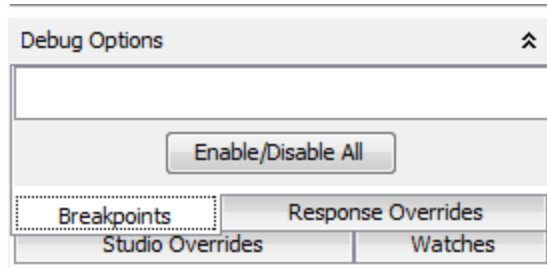
- **Breakpoints** are flags that enable you to automatically pause a run at a certain step in order to examine the results, the path of the run, or the values in the flow variables at a that point.

- **Response Overrides** force the response that you select, regardless of the result of the operation.

- **Studio Overrides** displays the enabled Studio overrides. In this view, you can enable and disable them as required.
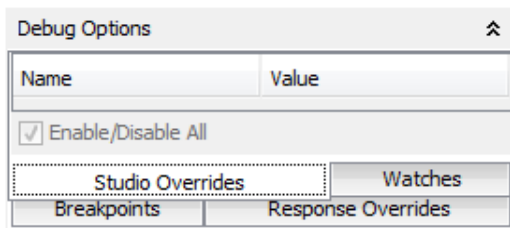
- **Watches** enables the user to create a watch list of variables during the debug. These variables are entries of the various contexts: Step Inputs, Step Context, Global Context, and the Configuration Items: Selection Lists, System Properties, and System Accounts. In the Watch list You can add a variable by name, and all entries with that name from the various context and from the lists of Configuration Items will be shown in the watches.

The **Debug Options** pane displays breakpoints and response overrides, and enables you to remove them or enable or disable them for this run.

Transition History pane
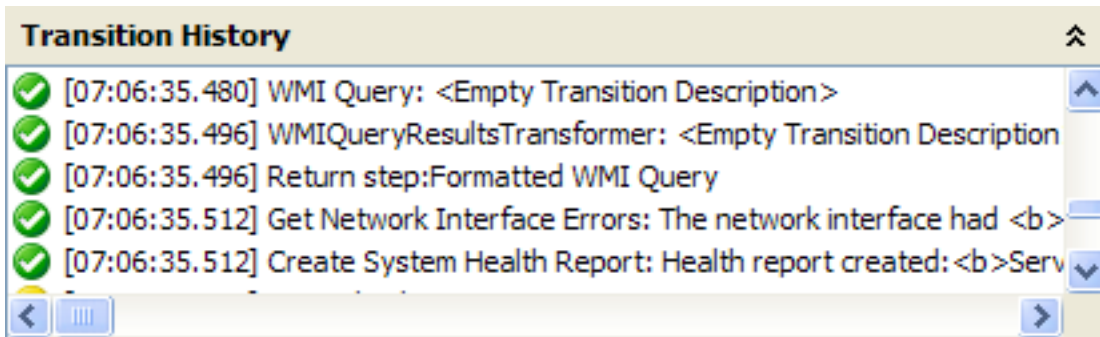
The **Transition History** pane lists the transitions that have been followed in the run and displays their descriptions.

Step Result Inspector pane

The **Step Result Inspector** pane displays the results of the selected step.

- Click the **Raw Result** tab to see the raw results (the results of the step's operation).

- Click the **Primary Result** tab to see the primary result of the step.

- Click the **Other Results** tab to see other results that you may have created.

Context Inspector pane
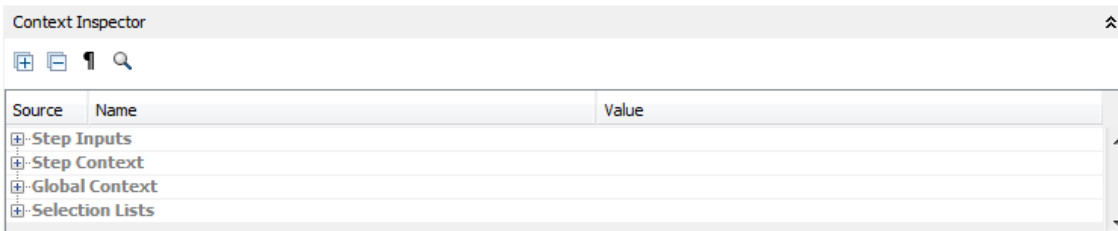
The **Context Inspector** pane displays the current values of flow variables (global as well as local) for each step.

Navigate to the appropriate section of the **Context Inspector** pane, to see global variables, flow variables, and their values for the step's inputs and the step and global context.

- The values in the **Step Input** section are the values that were assigned to the input before the step started. The text boxes that contain the values for flow variables are color coded.
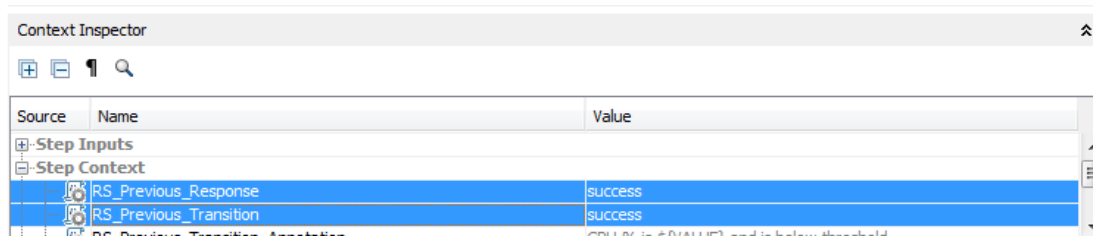
- The values in the **Step Context** section are the values that were updated after the step began.

  A step's context is the collection of flow variables and their value assignments in the local contexts of the step's flow and any parent flows. (If a flow is a step in another flow, the relationship between the two flows is subflow to parent flow.)

- The values in the **Global Context** section are system properties and any global variables that have been created.
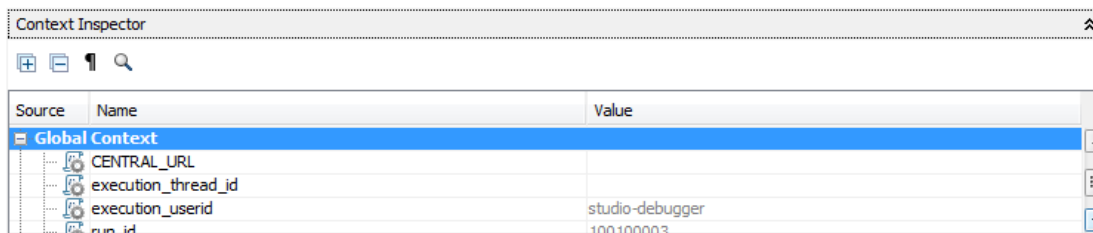
Debugger toolbar
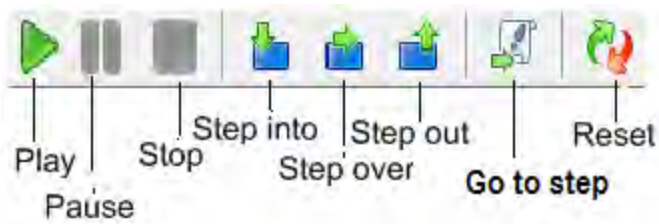
GUI item	Description	Keyboard shortcut
Settings	Set parameters for the run. This opens the Run Settings dialog box.	
Play	Run the flow to its end.	F11
Pause	Pause a flow that is running in the debugger. You can click the **Play** button to start it running again from the point at which it was paused.	ALT + P
Stop	Step a flow that is running in the debugger.	ALT + C
Step Over	Run the flow step by step.	F5
Step Into	Step into a step's subflow.	F6
Step Out	Step out of a step's subflow.	F7
Go to step	Jump to a flow step from the run tree	
Reset	Reset the flow variable values to the values that they had when you opened the debugger.	F12

Debugging Complex Flows

Debugging Flows with Parallel Processing Steps

Studio debugs steps, multi-instance, or parallel split step, with parallel processing. To learn how a flow with steps that use parallel processing will behave in execution, there is no substitute for running the flow in a staging environment after testing the flow in the Studio debugger.

You debug a flow containing a parallel split or multi-instance step the same way you debug a flow without such steps, but you should take into account that they run differently in the debugger.

What do you want to do?

Debug a parallel split step in a flow

In a flow run, the debugger starts the flows at the start time, and the order in which they finish depends on variable factors that cannot be predicted in Studio. Thus the debugger cannot predict considerations such as in the case of conflicting writes to the same flow variable, which lane writes to the flow variable last.

On the other hand, in Studio, you can manipulate the order in which the lanes will finish in the debugger in order to test various scenarios in a controlled fashion.

For more information about parallel split steps, see "Creating a Flow with Parallel Split Steps" on page 560.

Debug a multi-instance step in a flow

In a flow run, the multiple instances run concurrently, and the flow continues with the steps that follow one instance's response while the other instances are processed.

While this means that you are not testing under actual conditions, it does allow you to examine how long it takes each instance to finish.

For more information about multi-instance steps, see "Creating a Flow with Multi-Instance Steps" on page 564.

Debugging a Remote Central with Studio

Studio Remote Debugging allows an OO user to troubleshoot and debug flow runs in a remote Central. This allows customers to use Studio when investigating issues in their Central environments without

having to manually deploy fixes and modify the Central Flow Library.

Before debugging a flow, the user defines the URL to connect to Central.

You can debug the existing flows, locally changed flows or new flows. Studio takes the flow being debugged, along with all the subflows and operations (existing, changed or new) and sends them to Central for execution.

All the debugging or execution of the flow occurs on the selected Central using a merge of the Studio configuration items, Central configuration items and Context Inspector values.

Configuration items can be overwritten either from Studio or from Central. Upon triggering a flow (just before the flow starts), OO determines the initial value of the configuration items as following:

- If the user has changed the configuration item in the Context Inspector, this new value is used regardless of any overrides.

- If a configuration item is not overwritten (neither Central nor Studio) and was not changed in the Context Inspector, the value is used as is.

- If a configuration item is overridden in Central, but not in Studio overrides, the value is the one from Central.

- If a configuration item is overridden in Studio overrides, the value is the one from Studio overrides.

During the debug, when you pause, resume, step over, and so on, the user can change values with the Context Inspector as needed.

> **Note:** The only configuration Items that can be overridden in Central are system properties and system accounts. Override settings for configuration items in new system accounts are used.

Changes of the flows and configuration items do not affect or modify the Central library. These items are volatile, meaning that they are only visible for the debug session.

The traces left by a Remote Debugging session are visible in the Central Run Explorer (drill-down), and Dashboard, as any other execution. The source for such traces can be eventually identified as originating from a debugger execution and can be filtered out. The drill-down reflects the flow sent from the Studio (with the eventual changes).

During the debug session, the user can change the context of the execution through the Context Inspector.

> **Note:** All runs started from the Remote Debugger have the Extended persistence level.
>
> For details on run persistence, see "Persistence Level for the Run Log " in the *Central User Guide*.

> **Note:** In some cases, Studio may start up faster than the time it takes to initialize the local

connection. In these cases, if you start debugging using the local connection right after Studio starts up (the **Local Connection** button shows a progress icon), you will see the following message:
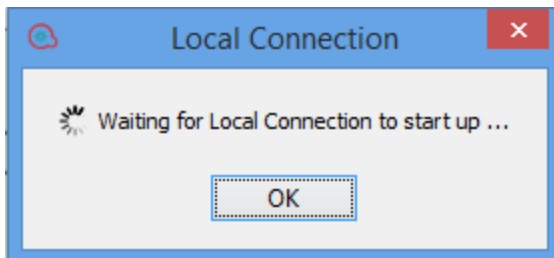
Prerequisites

- **Align Content Packs**

  To start the Remote Debugging session, you need to import into the Studio workspace the Central relevant content packs, with the appropriate version. Therefore, any content packs used by a debugged flow must have been deployed to Central before starting the debug session.

- **Group aliases**

  If you created a new group alias in Studio (and used them with certain operations), you will need to manually configure the groups in Central. The debugger cannot decide to which worker group the group alias should map. If the assigned group of an operation is not found (during the Remote Debugging execution), it will behave as if it was triggered from Central, pause and prompt the user about the problem.

- **Authorized Users**

  Only users that have been assigned the **Remote Debugging** permission are allowed to trigger the **Debugger** on a remote Central. See the *Central User Guide* for more information.

  Note: A user who has remote debugging permission is exposed to all flows, but not to system accounts. The system accounts are only available to users with appropriate permissions.

  Note: This restriction does not apply when debugging using the local connection (the default connection that Studio uses if you do not configure a remote Central).

Typical Workflow

1. Assign to a user a role containing the **Remote Debugging** permission.

2. Set up the Studio workspace to match the Central Library (content packs, flows to be debugged).

3. Configure, if necessary, the Central connection.

4. (Optional) Change the flow before debug.

5. Select a connection from the debug option. When you click the debug option, you are prompted for credentials. These credentials are saved and remembered in Studio.

   Note: If the **Enable Capture of Logged-in User Credentials** check box is selected in the **System Configuration > Security > Settings** option in Central, the following warning message opens in the Studio remote debugger:
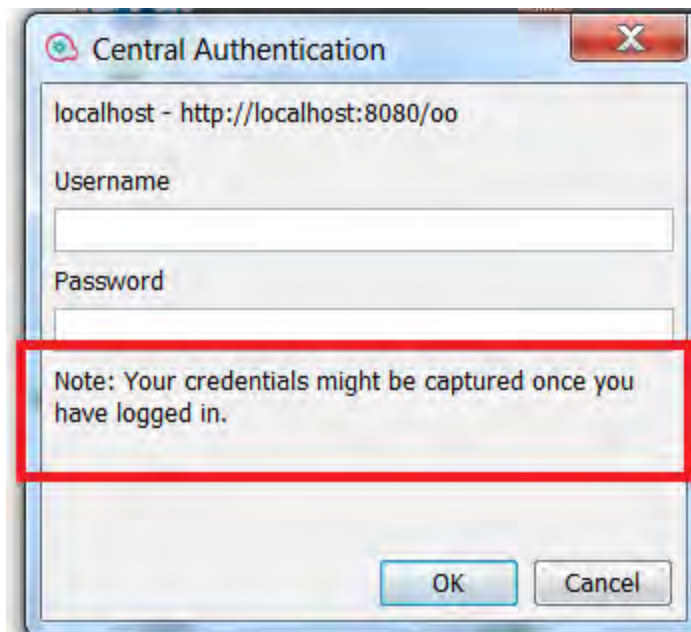
   Note: If authentication has been enabled in Central, you cannot log in with a user that has a colon as part of the user name, when logging into Central for remote debugging.

6. Execution runs with a merge of the **Studio Configuration Items**, **Central Configuration Items** and **Context Inspector**. For more information on the Context Inspector, see "Validating Content" on page 592.

7. (Optional) Set breakpoints, and change variables and configuration items by the context inspector during debug.

8. Execution completes.

What do you want to do?

Add or Edit Central Connections

1. In Studio, select **Configuration** > **Remote Debugger Connections**.

2. Click the debug icon , and from the list select a connection to a remote Central. **Local Connection** is the default connection.
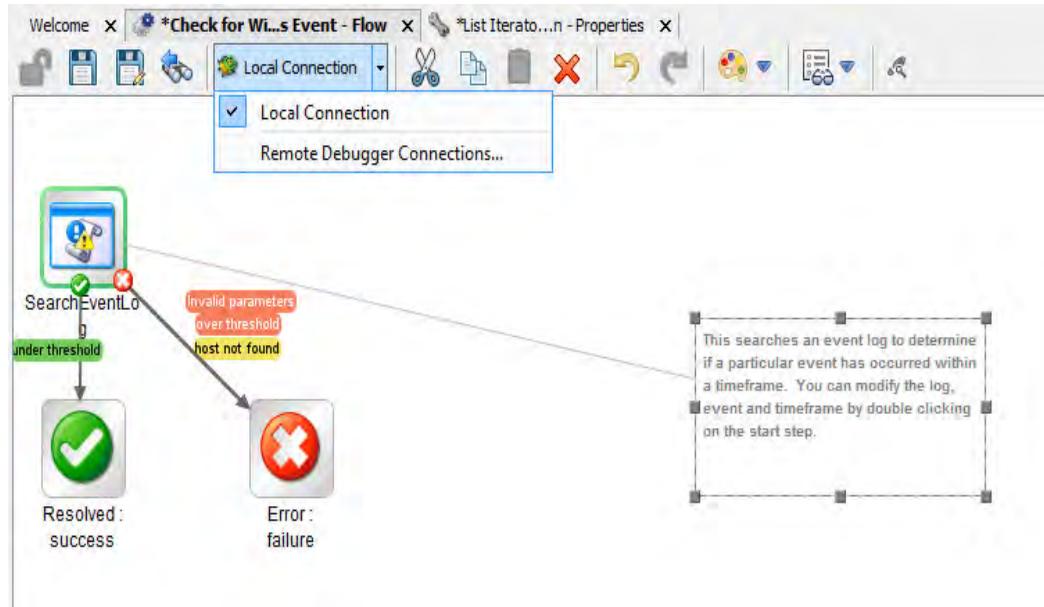
3. To add, edit or delete a connection, select **Remote Debugger Connections**. The **Remote Debugger Connections** dialog box contains a list of available connections. You can edit these connections, delete existing connections and add new connections.
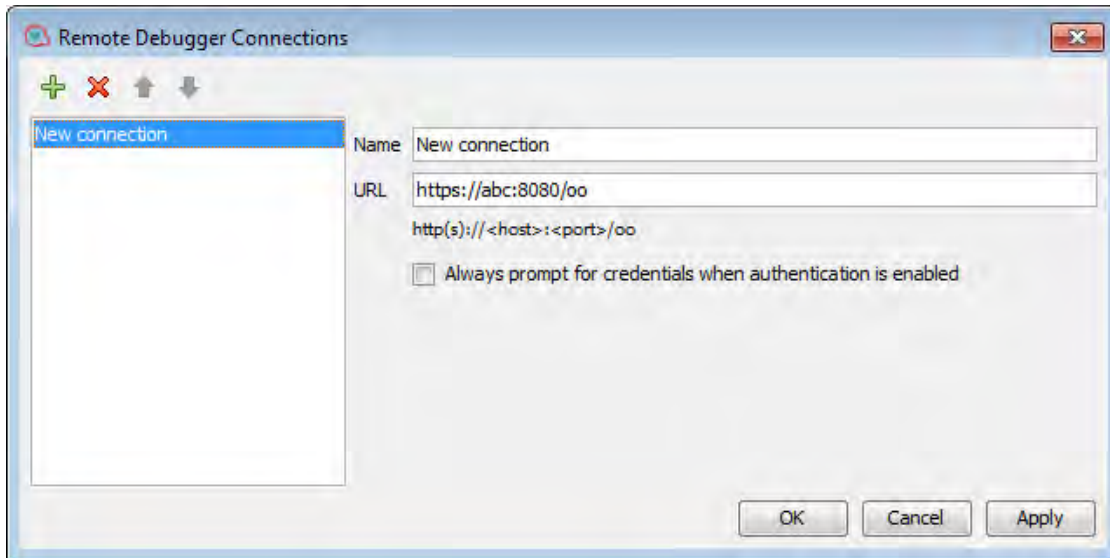
- **Delete**: Select the connection that you want to delete and either press the delete key or click the minus (x) red button.

- **Add**: To add a new connection, click the green (+) button. Enter a name for the new connection and the URL of the remote Central.

  Note: There is no validation with the connection name and URL. The user is required to check these settings. You must assign a unique connection name.

  Always prompt for credentials when authentication is enabled: When you select this option and the selected Central has authentication enabled, it causes Studio to always prompt the user for credentials; if it has already been authenticated, the authentication form will have the credentials already filled in.

  Note: If the user needs to reset the user name or password without restarting Studio, in the **Remote Debugger Connections**, check **Always prompt for credentials when authentication is enabled**.

  Configuring LDAP user and domain: When you create a new connection to Central and connect using a LDAP user, you are prompted to select the domain, username and password.

  Note: You cannot configure the Remote Connections in the **studio.properties** file.

Set the Proxy for Debugging on a Remote Central

When configuring the connection to a Central, you can set the HTTP proxy information: host, port, user name and password. The proxy supports the basic authentication scheme.

To setup the proxy:

1. In Studio from the **Configuration** menu, select **Options** > **Connection Settings**. The **Proxy Configuration** dialog box appears.

2. Enter the proxy information. You must enter a host and port number. When you are done, click **Save**. The proxy settings are validated on the host and port number.

Import Certificates Automatically with a Remote Debugger Connection

1. When you select a remote Central connection using HTTPS, you are prompted with the SSL Certificate message.

   The SSL Certificate message displays the certificate's issuer distinguished name, which identifies the entity that signed the certificate and the certificate's subject distinguished name on separate lines.

   If the certificate is not trusted, the dialog asks whether you want to trust this certificate.

   Note: Studio supports both self-signed and CA signed certificates.

   ○ If you want to trust the certificate, click **Connect**. The Studio trustStore is updated with the certificate(s), and you will be able to connect to Central having a successful SSL handshake.

   ○ If you don't want to trust the certificate, click **Cancel**. The Studio trustStore will not be affected and the connection to Central will fail.

2. If the URL of the server and the certificate do not match, the Hostname Validation dialog box is shown:

○ If you click **Don't Trust**, the connection is not permitted.

○ If you click **Always Trust**, the hostname validation failure is ignored and the connection is allowed.

However, when you restart Studio, the hostname validation message will be displayed again, because you are expected to correct this issue. You should correct the URL to match the hostname in the certificate, or change the certificate to be according the connection URL.

Note: The SSL Certificate dialog box also appears when you connect to a Git server via HTTPS.

For SVN, the certificate chain details are presented in the studio.log.

Note: In the **studio.l4j.ini** file under the Studio installation folder, the default values for **self.signed** is false, and the default value for **verify.hostname** is true.

If the hostname verification fails, an error message opens. You can choose to continue or abort the operation.
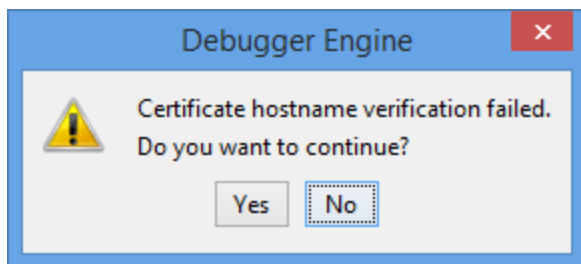
Debugging a Flow on a Remote Central

1. Build a flow using operations and flows from an imported content pack or use an existing one.

2. Open the flow.

3. Run the Debug and select a remote connection that you setup earlier.

   Note: Performing a debug on a Central that is part of a production system could be problematic as it can affect the production data and operation. The Debugger view displays a banner with the name of the remote connection.

What do you want to do?

Debug a Flow on a Remote Central

To debug a flow:

1. Build a flow using operations and flows from an imported content pack or use an existing one.

2. Open the flow.

3. Run **Debug** and select a remote connection that you setup earlier.

   Note: Performing a debug on a Central that is part of a production system could be problematic as it can affect the production data and operation. The Debugger view displays a banner with the name of the remote connection.

Reuse Existing Flow Inputs

You can use the inputs of a run as input for a new run. This allows you to troubleshoot failed flows, without having to re-enter or prepare the inputs.

To load or specify a run ID from the current Central for re-running a flow:

1. In Studio, open a flow for debugging.

2. Select a remote Central.

3. Enter the inputs of an execution.

4. Run the flow.

5. Click **Load Execution Inputs**.

   A dialog box appears with a text field that displays the execution id. Studio loads the inputs of the specified execution id (they are loaded from the currently selected Central). If there are no inputs, then the execution did not include any inputs or the execution was not found. In this case a message appears stating that the current execution has not changed.

   After the inputs have loaded, they are matched (based on name) against the inputs of the current flow. The existing inputs of the flow are updated (no new inputs are added) only if the corresponding loaded value is not empty. If no input of the current execution is changed, a message is displayed.

   The context inspector highlights the modified inputs and displays the new values.

6. You can now trigger the run with the inputs of the specified execution id.

   Inputs are loaded as string values regardless of their type (single value, list of values).

Rerun a Flow

You can rerun a triggered flow using the same inputs as the previous run. In the debug view, the **Rerun** option is disabled initially and enabled after the first run of this flow. When you click **Rerun**, the toolbar changes to the view ready for debug . When you click **Play**, you are no longer prompted for eventual prompt inputs of the flow. The new run will use the flow inputs of the previous run.

Enable Rerunning a Flow in Central from a Specific Step

To enable the rerunning of a flow in Central from a specific step(s) in the flow, you must define these steps as "rerun points":

1. Select the step you want to be defined as a rerun point.

2. In the Step Inspector, select the **Advanced** tab.

3. Select the **Rerun point** check box.

> **Caution:** This setting has an effect on the database. OO will persist the entire step context in the database.

> **Note:** If a step has been marked as "rerunnable", the step's state is saved in the database during the flow execution. Then, at a later stage, you can retrigger the flow from that step. You can rerun a flow several times.
>
> This functionality is available in OO version 10.71 and above. If you use the new Rerun functionality in a flow, you cannot deploy the flow on earlier Central versions.

Reference Material

Remote Debugger Connections

The **Remote Debugger Connections** dialog box contains a list of available connections.
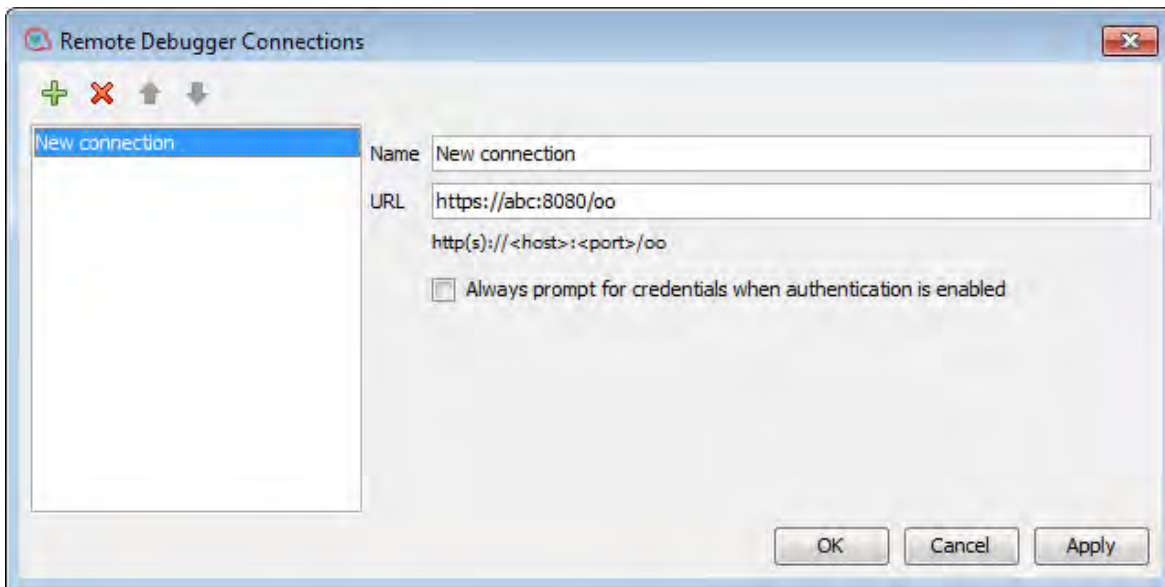
GUI item	Description
Add connection ➕	Adds a new connection. Type the name and URL of the new connection and click **OK**.
Delete connection ❌	Deletes the selected connection.
Move up ⬆	Moves the selected connection higher in the list.
Move down ⬇	Moves the selected connection lower in the list.

Exporting a Content Pack

After you have finished validating your flow, you're ready to release it into a content pack, so that it can be deployed and run.

A content pack is the outcome of a project. It contains entities in the project and reference IDs. A content pack includes not only flows and operations but also actions and configuration items.

The content pack is the element that you release for deployment in Central.

> **Note:** Invalid flows or operations, for example, those containing duplicate UUIDs, will not be included in the content pack.

When you create a content pack from a project, by default, Studio names it with a unique version number. Then, every time you create another content pack from the same project, Studio assigns it with the next minor version number. The naming mechanism is dependent on various conditions and is described in detail in "Content Pack Versioning Lifecycle " below.

> **Note:** Your content pack can include folders inside the configuration items.
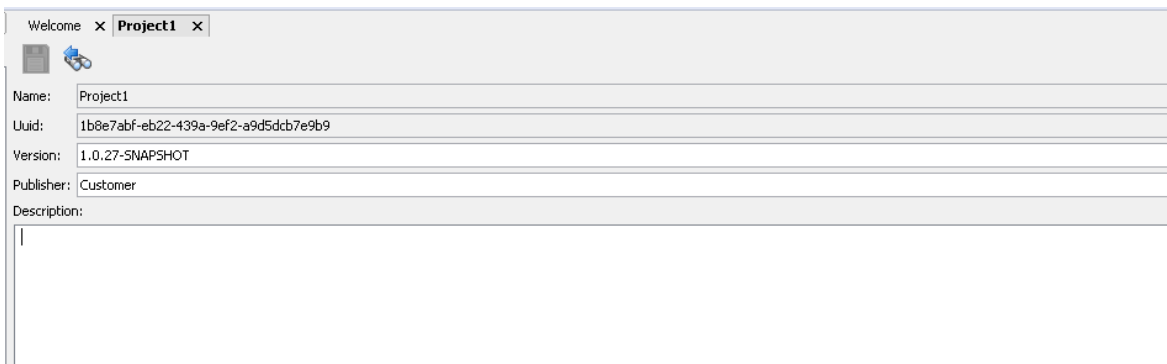
> **Note:** Content packs created with a Studio version later than 10.20 from projects that contain folders inside a configuration section (for example, System Properties, System Accounts) cannot be imported into versions of Studio earlier than 10.20.
>
> However, content packs created with a Studio version later than 10.20 from projects that have no folders in any configuration section, can be imported into versions of Studio earlier than 10.20.

Content Pack Versioning Lifecycle

By default, Studio handles the versioning of content packs automatically. After a content pack is created, the last digit of the version number (generally the minor-minor section) is incremented by 1 and a –SNAPSHOT suffix is added to the current project version. This means that when you work on the project, you are adapting an in-development version.

For example:

You can also change the content pack version manually by typing a new version number in the Version field.

> **Note:**

- Studio increments only the last minor version automatically. If you want to increment a major version (e.g., **1**.0.1 to **2**.0.1), you have to do this manually.

- If you enter a version number that is not in the standard major.minor.minor.minor.minor... (1.x.x.x.x) format, the version number will not be automatically incremented when you create another content pack.

- The number of minors in a version number is variable. So, version numbers such as 1, 1.2, 1.2.3.4.5.06 and 1.1.1.1.1.1.1.2 are also automatically incremented.

Content Pack Versioning Without Revision Control

When working in Studio without version control, Studio provides an automatic method of incrementing the project version when creating a new content pack. In this way, you can easily create multiple content packs with consecutive version numbers, e.g., 1.0.1.1, 1.0.1.2, 1.0.1.3.

When opening the Create Content Pack wizard, Studio reads the version in the content pack properties, removes the –SNAPSHOT suffix, and presents the version number in the Content Pack Properties step.

After clicking **Create CP** in the Dependencies Management step, the content pack is created, and the last digit of the version number is automatically incremented by 1.

The –SNAPSHOT suffix is added to the Project version which means that when you work on the project, you are adapting an in-development version.

Note:

- Studio increments only the last minor version automatically. If you want to increment a major version (e.g., **1**.0.1 to **2**.0.1), you have to do this manually.

- If you enter a version number that is not in the standard major.minor.minor.minor.minor... (1.x.x.x.x) format, the version number will not be automatically incremented when you create another content pack.

- The number of minors in a version number is variable. So, version numbers such as 1, 1.2, 1.2.3.4.5.06 and 1.1.1.1.1.1.1.2 are also automatically incremented.

Content Pack Versioning With Revision Control

When working with revision control, every time you create a new content pack, Studio adds an SCM (SVN or Git) tag that represents a snapshot of the current state of the repository (for Git) or project (for SVN). This tag incorporates the content pack name and version number.

Using this tag, you can easily reset to/update to a previous project version and reset your current environment to the same sources that were used at the time a content pack was created.

In this way, you can easily create a patch for a specific content pack.

> **Note:** Before creating a content pack release, you must ensure that you have no local uncommitted changes on the project and that the project revision is up-to-date. The content pack creation will fail if these two conditions are not met.
>
> However, if the conditions are not met, you can create a temporary "SNAPSHOT" content pack. For this content pack, the version number is not incremented, and you will not be able to run a **Revert to an older project version** operation.

What do you want to do?

Create a content pack

1. In the **Projects** pane, select the project from which you want to create a content pack.

2. Select **File** > **Create New Content Pack**.

   > **Note:** Alternatively, you can select the **Create Content Pack** button in the **Projects** pane or right-click on the project and select **Create Content Pack**.

   If there are unsaved editors open, the **Saving** dialog box gives you the option to save the changes. Click **Yes to All** to save all changes in the open editors, or click **Yes** or **No** for each one individually.
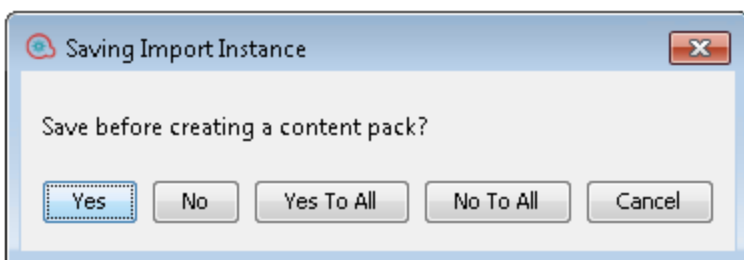
3. In the Content Pack Properties step of the **Create Content Pack** wizard, enter the content pack details:

Note:

- ○ The **Name** field is taken directly from the project name and you cannot change it.

- ○ The new project version is the Development version set in the Advanced tab of the Content Pack Properties step of the Content Pack Creation wizard.

4. In the **Location** field, enter or browse to the location where you want to save the content pack. By default, the path to the project workspace is selected.

   Note: By default, the path shown here is the last location in which you created a content pack.

5. In the **Publisher** field, enter the publisher of the content pack. This information will appear in the content pack's **Properties** page.

6. To edit the **Description** field, right-click on the project in the Projects tree and select **Properties**. Then, type a description in Description area.

7. Select the **Release** check box to create a final version of the content pack for publishing. After checking this box, the -SNAPSHOT suffix is removed from the version number in Version field.

> **Note:**
>
> ○ If the Release check box is disabled, this means you are not connected to any source control management system.
>
> ○ If you select the **Release** check box, the SCM Synchronization window is added as Step 3 in the Content Pack wizard. See "Content Pack Creation Wizard - Step 3" on page 631 for details.

8. In the **Version** field, the next minor version number is shown. The last digit of the version number is automatically incremented when you create a new content pack. However, you can change this number manually by typing a different version number. For details, see "Content Pack Versioning Without Revision Control " on page 621.

   > **Note:** When creating a release version of the content pack, Studio commits the project properties and adds an SCM tag to the content pack.

9. **For source control only:** Select the **Release** check box if you want to increment the revision number when you commit the changes to SVN/Git.

10. Select **Advanced** to show the following fields:

    a. **Next development version:** The next version with the extension -SNAPSHOT automatically appears in this field. You can change it as required.

    b. **For source control only:SCM Tag:** By default, the SCM tag name that will be used when releasing the content pack is [<projectname>-<version>]. However, you can change this as required, and type a different name that helps you to identify the version in the source control.

11. Select the **Include the system accounts passwords?** check box if you want the content pack to include passwords for the system accounts. When the content is deployed on Central, the user names and passwords will be deployed.

    > **Note:** The passwords will be obfuscated inside the content pack. However, this is NOT a safe option and is not recommended.Alternatively, it is recommended to use the option of configuring the system accounts in Central. For details, see "Setting Up Configuration Items for a Content Pack" in the *Central User Guide*.

12. Click **Next**.

    If the dependencies are not up-to-date, the Dependency Management window opens:
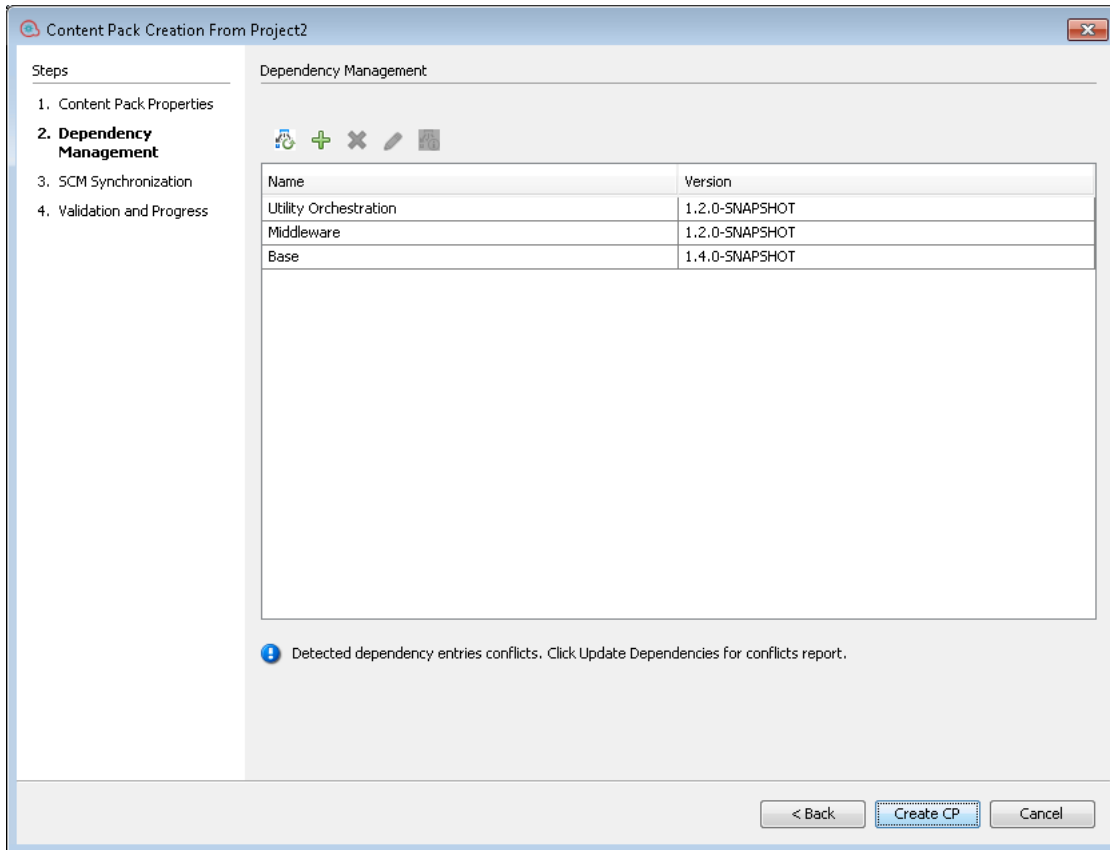
13. From this window, you can manage dependencies - add, delete, edit and update dependencies as in the Dependencies Editor. See "Managing Content Packs and Dependencies in a Project" on page 394 for full details.

14. Click **Next**.

    If you selected the **Release** check box in the first step, the SCM Synchronization window opens.

    > **Note:** The SCM Synchronization page displays only in the following cases:
    >
    > ○ If the project is not synchronized
    >
    > ○ Changes were made to the project dependencies in the **Dependency Management** window
    >
    > ○ If there are uncommitted changes in the workspace.
    >
    > Otherwise, you will see the Validation and Progress window after clicking **Next**.

    If there are any SCM synchronization issues between the local version and the main repository, Studio displays them in the SCM Synchronization window. You must solve these issues before

continuing. You can use the SCM toolbar to commit, push or revert changes as described in "Working with Git Source Management System " on page 346.
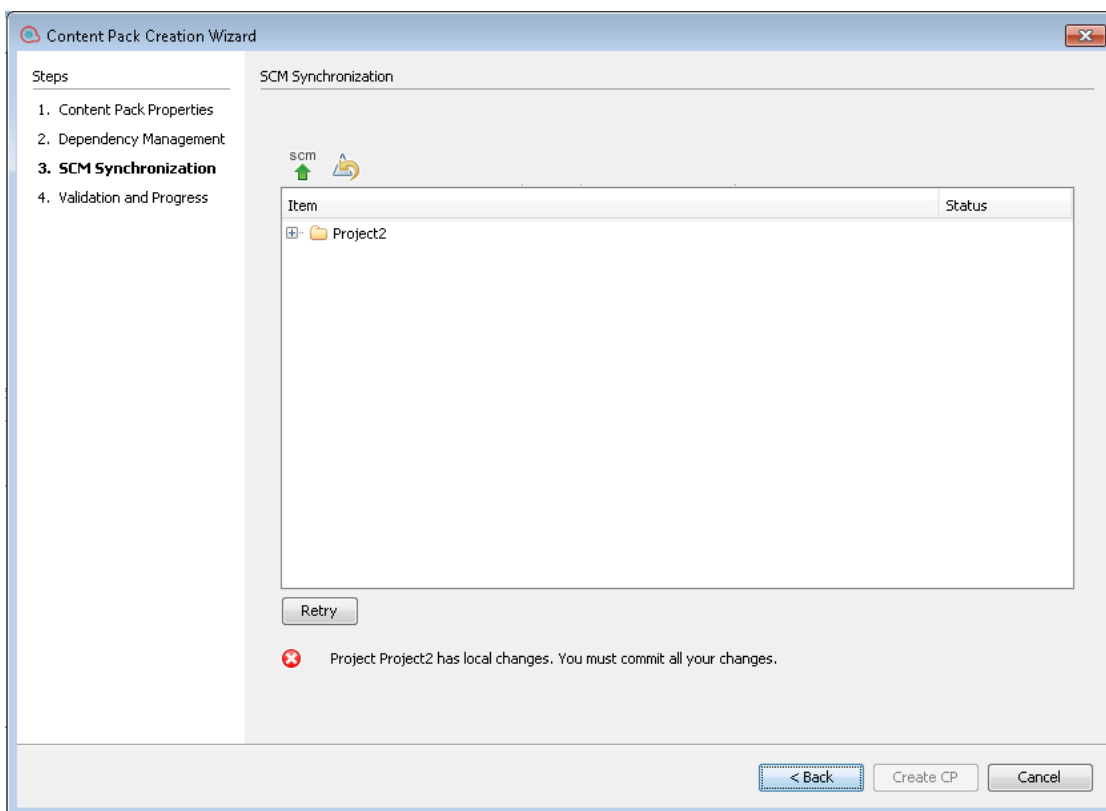
> **Note:** If you modified the dependencies in the Dependencies management step, they will also appear as changes in the SCM Synchronization step.

After solving the SCM issues, click **Retry**. If there are no more issues, you will see the following message:

```
You can now continue creating the content pack.
```

15. Click **Create CP**.

    All the commit operations that were performed for the current branch are automatically pushed to the main Git repository and the content pack is created.

    If there are any invalid flows or dependency warnings in the project, Studio presents them. Click **Details** to see the warnings.

    Invalid flows are not packaged into the content pack.
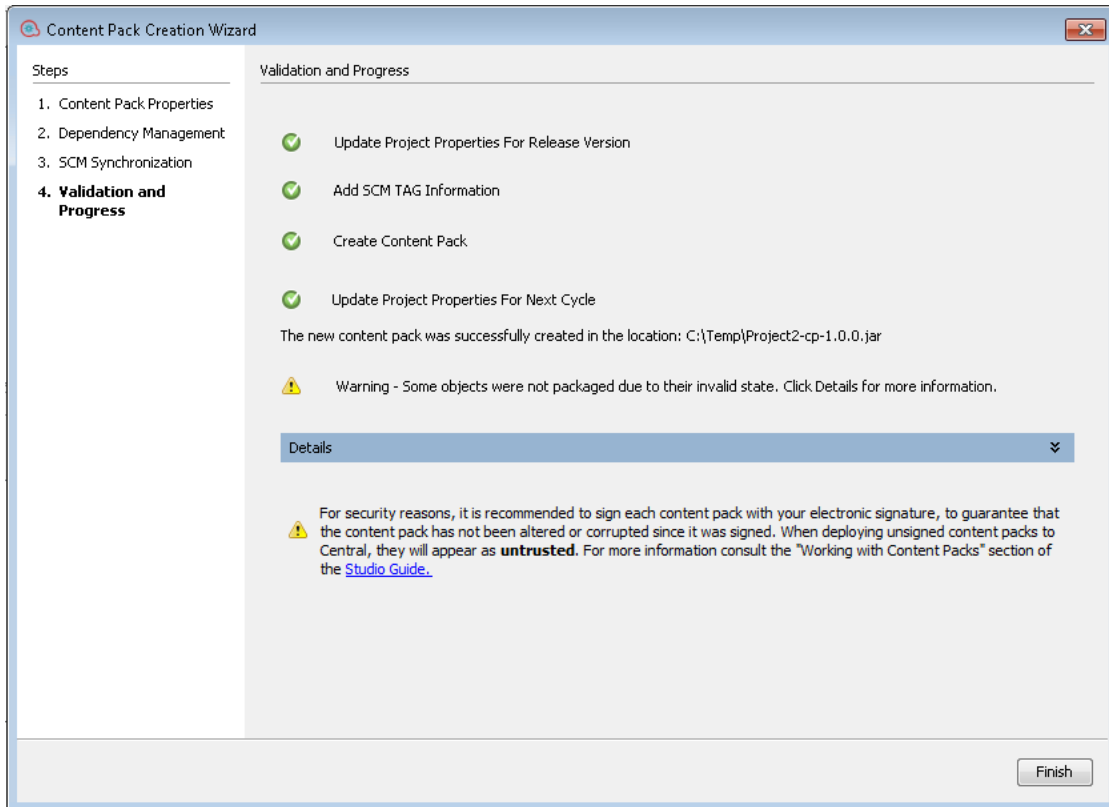
The content pack is created in the specified location.

> **Note:** Central provides a signature validation mechanism for content packs. For details, see "Content Pack Validation - Digital Signatures" in the *"Using Central" on page 42*.

16. After the new content pack is created, a message appears, with a link to the location in which the content pack was created. Click on the link to access the content pack.

17. Click **Finish** to close the wizard window.

The new content pack can now be deployed and run, or imported into another project.

> **Note:** After the content pack is created, the **contentpack.properties** file contains the version number as follows:
>
> - **Content Pack: contentpack.properties** contains the published version. For example, 1.8.3.
>
> - **Project**: **contentpack.properties** contains the next snapshot version. For example, 1.8.4-SNAPSHOT.

> - The version number is saved in the pom.xml file. The pom.xml file also includes project dependencies information.

View/edit the project version

1. In the **Projects** pane, select the project whose version you want to view/edit.

2. Select the **Properties** option in the **Projects** pane.

   or:

   Right-click the project and select **Properties**.

Reference Material

Content Pack Creation Wizard - Step 1

GUI item	Description
Name	The name of the content pack is taken from the project name. This field is read-only.
Location	Enter or browse to the location where you want to store the content

	pack. By default, the path to the project workspace is selected.
Publisher	Enter the publisher of the content pack. This information will appear in the content pack's **Properties** page.
Description	To edit the **Description** field, right-click on the project in the Projects tree and select **Properties**. Then, type a description in Description area.
Release	Select this check box if you want to increment the revision number when you commit the changes to SVN/Git. This also creates a tag with the current state of the project (SVN) or repository (GIT). If you select this check box, Studio adds the SCM Synchronization step to the Create Content Pack wizard.
Version	Enter the version of the content pack. This information will appear in the content pack's **Properties** page.
Advanced	Select the arrows to show the SCM fields.
Development version	The next version with the extension -SNAPSHOT automatically appears in this field. You can change it as required.
SCM Tag	Type the text you want to use as the SCM tag name. You can use this field to identify the version in the source control.
Include the system accounts passwords?	Select the check box if you want the content pack to include passwords for the system accounts. These passwords will be obfuscated. If this check box is not selected, the passwords will not be included in the content pack. **Note:** For security reasons, it is recommended to configure the system account passwords in Central.

Content Pack Creation Wizard - Step 2
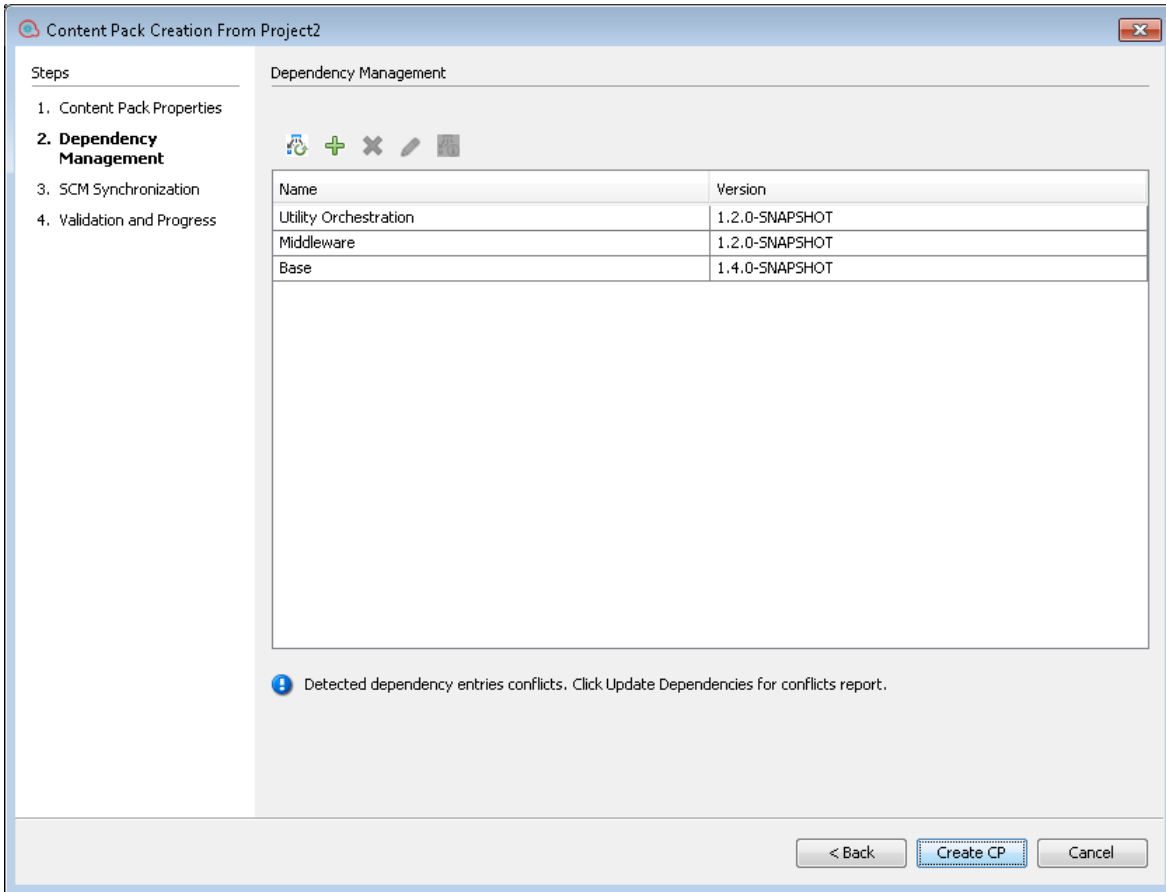
GUI item	Description
	Add a new dependency.
	Delete a dependency from the list.
	Edit a dependency's properties.
	Update dependencies.

Content Pack Creation Wizard - Step 3
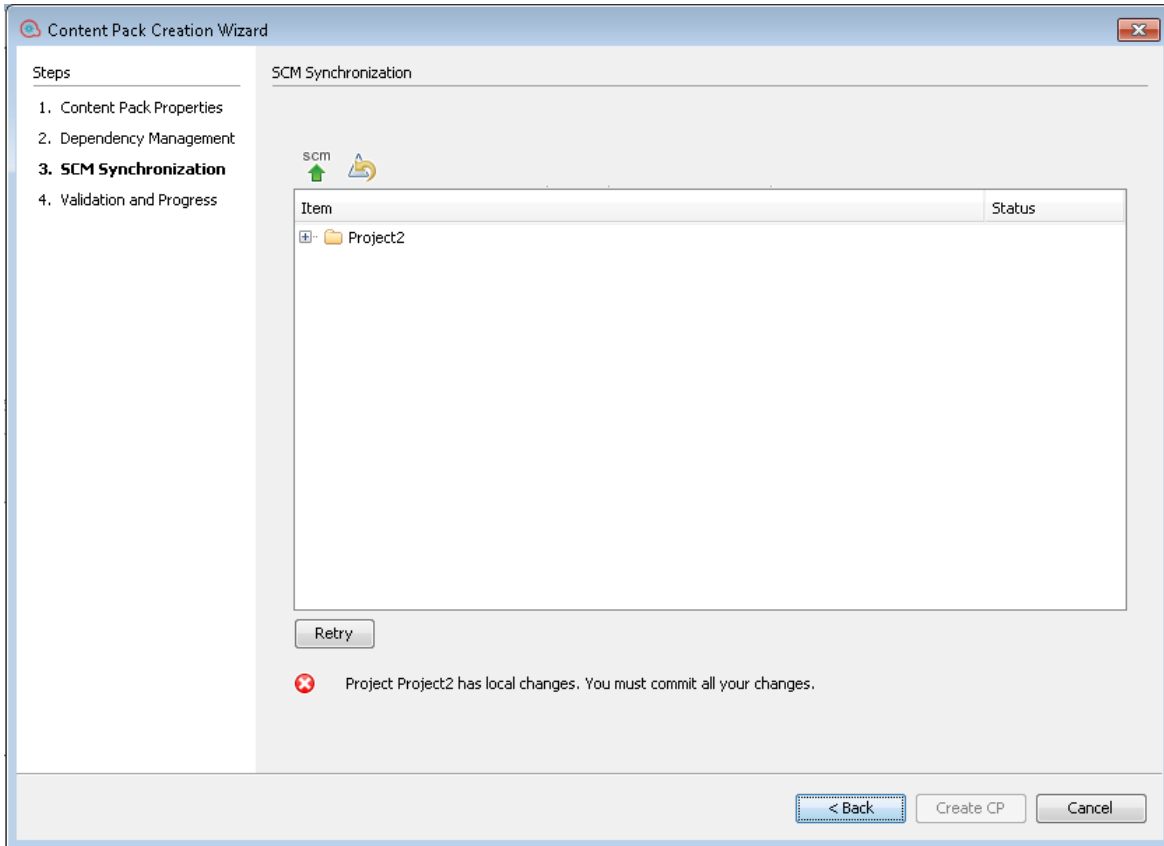
GUI item	Description
scm	**Commit all changes:** Commits all the changes in the **SCM Changes** pane. Only available when there are changes.
	Revert all changes: Reverts all changes that show in the **SCM Changes** pane.

Content Pack Creation Wizard - Step 4
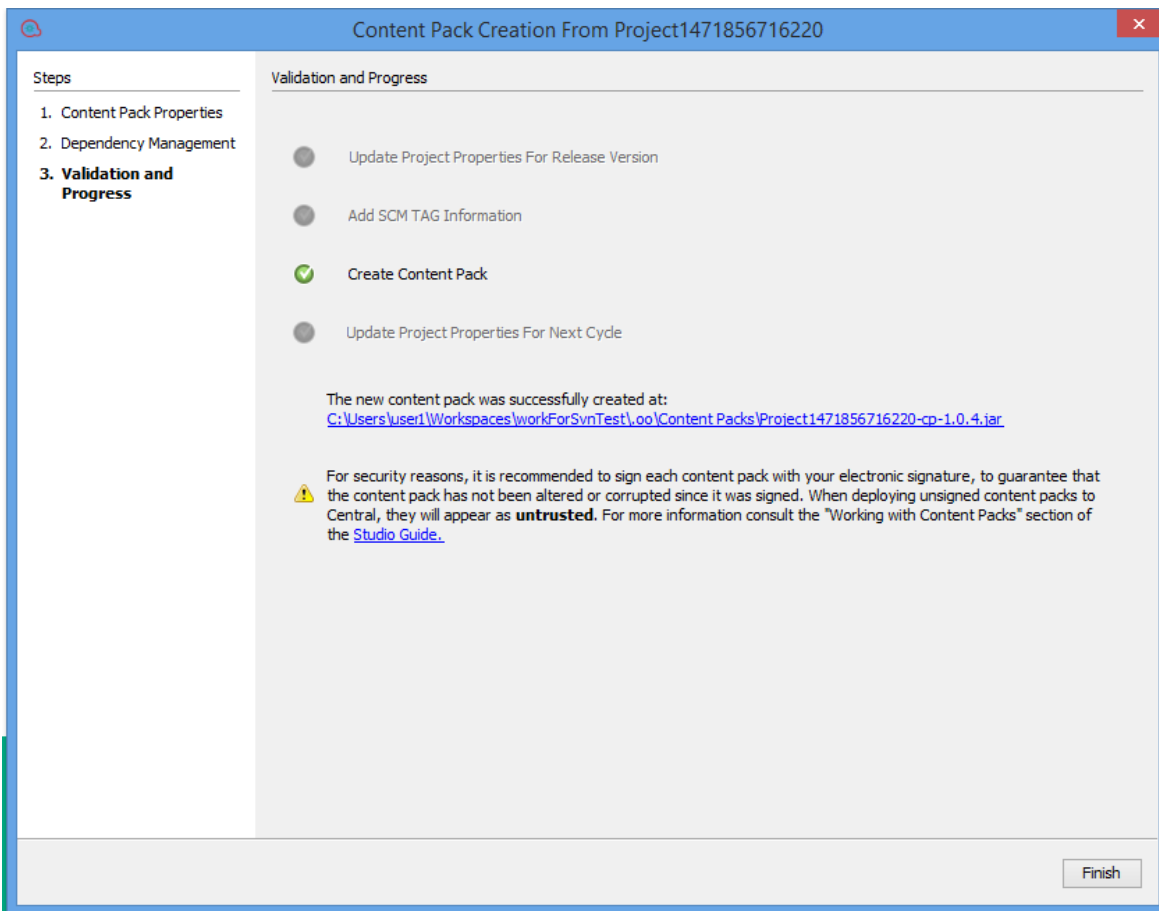
GUI item	Description
<New content pack location>	Studio shows the location of the newly created content pack. Click on the link to go directly to the folder containing the content pack.

Managing Flows and Operations

Your project library may contain a large number of flows and operations. This chapter discusses how to manage this library—how to locate, copy, and bookmark items, how to see how they are used, and how to create new operations.

Creating Operations

There are three ways to create operations in Studio:

- Copying and modifying existing operations.

- Importing an operation from an already existing plugin.

- Creating an action plugin in Java and importing that action plugin to Studio.

Creating Operations from Action plugins

An action plugin is a jar file containing IActions or @Actions. You can import an action plugin to Studio, in order to create an operation from one of the actions in it.

An action plugin may include multiple actions, and you can create an operation from each one of these actions.

For information about developing action plugins, see the *Action Developers Guide.*

Copying Operations that were Created from Action plugins

When you copy an operation that was created by importing an action plugin, 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. This is known as a "soft copy".

The source operation, which this operation was copied from, is displayed in the **Advanced** tab.

Note that the link to the action plugin is the only item that is automatically updated in the copied operations. Changes that you make to the original operation's input, output, variables, scriptlets, and so on, are not updated in the copies.

Valid Operations

A valid operation requires:

- There are operations that require no inputs such as UUID Generator.

- At least one response that is mapped to valid expressions describing outcomes of the operation.

If the new operation is invalid or incomplete, its name is displayed in the **Projects** pane in red zig-zag underscore. Moving the cursor over the name displays a tool tip that specifies how the operation is incomplete.

Best Practice

To help authors who will create flows using the operations you create, add the following information to the operation's **Description** tab:

- A description of what the operation does.

- Inputs that the operation requires, including where authors can find the data that the inputs require and the required format for the data.

- Responses, including the meaning of each response.

- Result fields, including a description of the data supplied in each result field.

- Any additional implementation notes, such as:

  - Supported platforms or applications, including version information

  - Application or Web service APIs that the flow interacts with

  - Other environmental or usage requirements

What do you want to do?

Copy and modify an operation in Studio

Important! If you need to delete, create, or rename an operation inside your project, make sure to do this from within Studio, and not by deleting, creating, or renaming the item in the file system.

1. In the **Dependencies** pane or **Projects** pane, select the operation that you want to copy.

2. Select **Edit** > **Copy**.

3. Select the location in the project tree where you want to paste the copy, and select **Edit** > **Paste**. The operation is treated as a new object.

4. To assign the operation to a category for search purposes, click **Assign Categories** and then select a category from the list.

5. To create an input, click the **Inputs** tab and then click **Add Input**.

6. In the dialog that appears, type the input name and then click **OK**.

   - For information on what inputs are and how to use them, see "Creating Input" on page 472.

   - For information on defining an input data source, see "Specifying the Input Source " on page 482.

7. Add and define any output data.

For information on adding and working with output data, see "Setting Operation Outputs" on page 515.

8. Create any responses needed and map results to the responses.

   For information on defining rules that govern which responses are chosen for the operation, see "Setting Responses" on page 501.

9. Click the **Description** tab and write the description in the text box.

10. Click **OK**.

Change the source operation that the operation is based on

If you created an operation by copying an existing operation, you can change the source operation. Your copied operation will become a copy of the operation that you select as the source.

1. Open a copied operation, and select **Properties**.

2. In the Properties sheet, click the **Advanced** tab.

3. Under **Source Operation**, click the **Select** button.

4. In the Select Source Operation dialog box, navigate to and select the source operation that you want to base the copy on, and then click **OK**.

5. If required, rename the operation to reflect the change.

6. Review and make any changes necessary to the value assignments for inputs, to reflect any differences between the old operation's inputs and those of the new one.

Create an operation by importing an action plugin

The most direct way to create an operation from an action plugin is to import it and create the operation at the same time.

1. Create and pack an action plugin, so that it includes the actions that were developed. For information about how to develop an action plugin, see the *Action Developers Guide*.

2. In Studio, right-click the folder where you want to create the new operation, and select **Import plugin**.

3. In the Import Plugin dialog box, click the browse button to browse to and select the OO plugin that you want to import.
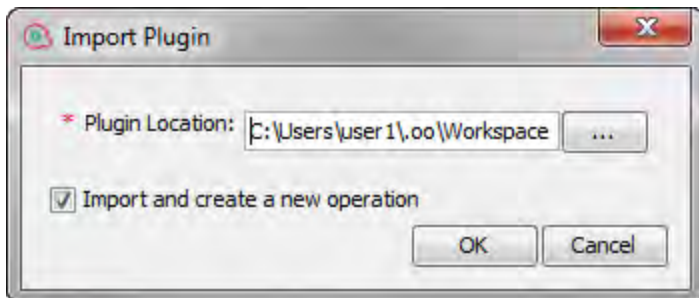
Note: It is possible to import a single plugin (maven artifact), either by the JAR file or by the POM file. The plugin must have both its JAR and POM file at the same location.

If you import a plugin that was already deployed, the new plugin replaces the existing one.

4. Select the **Import and create new operation** check box and click **OK**.

5. In the **Create Operations** dialog box, expand the plugin containing the actions that you need, and select the actions that you want use to create the operation from.
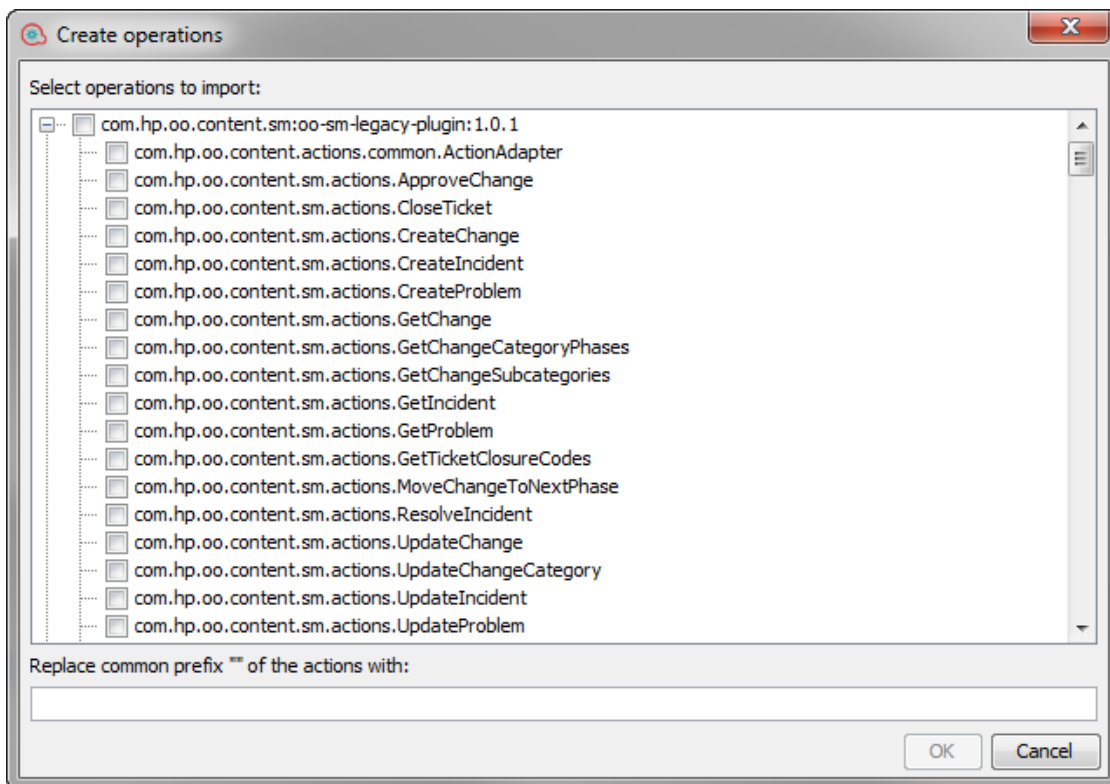
Note: If a plugin contains multiple actions, you can select more than one action, to create multiple operations.

For each action that you selected, a new operation is created in the folder where you right-clicked.

In each operation, the information about the action plugin is displayed in the **Operation Fields** section at the top of the **Inputs** tab, in the operation's Properties window.

6. Save the operation.

Create an operation from an imported action plugin

After action plugins have been imported to Studio's repository, you can create operations from the actions in them.

1. In Studio, right-click the folder where you want to create the new operation, and select **New** > **Operation**.

2. Browse to locate the plugin, from Studio's repository, and click **OK**.

3. In the Create Operations dialog box, select the action that you want use to create the operation from.
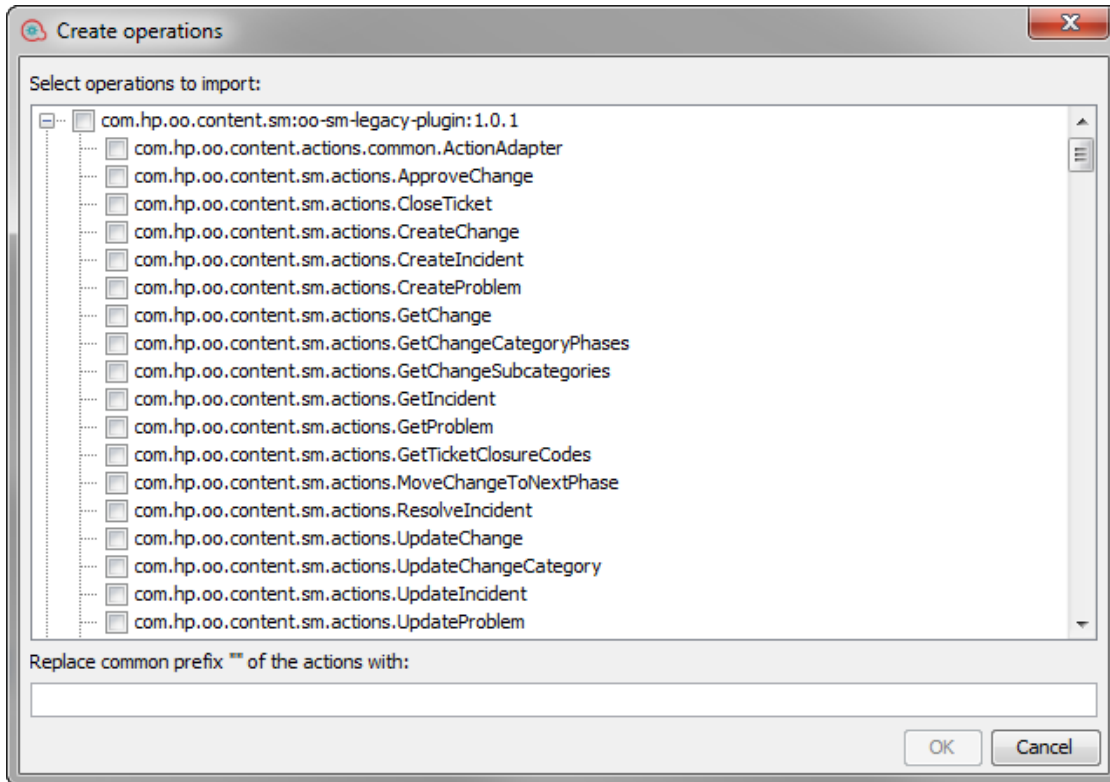
Note: If the plugin contains multiple actions, you can select more than one action, to create multiple operations.

For each action that you selected, a new operation is created in the folder that you right-clicked on.

In each operation, the information about the action plugin is displayed at the top of the **Inputs** tab, in the operation's Properties window.

Import action plugins

It is possible to simply import action plugins to Studio's repository, so that they are available for you or other authors to create operations from, at a later date.

1. In Studio, select **File** > **Import plugin**.

2. In the Import plugin dialog box, browse to and select the OO plugin that you want to import to Studio's local Maven repository.

3. Click **OK**. The plugin is available for authors to create operations from. For more information, see *Create an operation from an imported action plugin*.

Create a manual operation

A manual operation is one that offers a choice of actions. The user will need to select an action at

runtime.

To create a manual operation, you copy the manual operation template from the base content and define the actions that will be made available to the user.

1. In the **Dependencies** pane, select the manual operation template, located in the contact pack, for example, `base-cp/Library/Utility Operations/Manual`.

2. Select **Edit** > **Copy**.

3. Select the location in the project tree where you want to paste the copy, and select **Edit** > **Paste**. This operation is treated as a new object and is detached from the content pack it arrived with.

4. In the operation properties, add the actions that will be available to the user.

Create a display operation

A display operation is one that displays information in a pop-up prompt message, but does not perform any other action. The user will just need to click **Continue** at runtime.

To create a display operation, you copy the display operation template from the base content in `Library/Utility Operations/Display Message` and define the information that will be displayed to the user.

The prompt message can include variables. For example, to tell the user what time the preceding step concluded, you could include a date/time variable (`${dateTime}`) in the message.

1. In the **Dependencies** pane, select the display operation template.

2. Select **Edit** > **Copy**.

3. Select the location in the project tree where you want to paste the copy, and select **Edit** > **Paste**. This operation is treated as a new object and is detached from the content pack it arrived with.

4. In the operation properties sheet, select the **Display** tab.

5. Click the **Display** tab in the Step Inspector.

6. Select the **Always prompt user before executing this step** check box.

7. In the **Prompt Title** box, type the prompt's label (up to 128 characters).

8. In the **Prompt Width** box, type the width of the prompt in pixels.

9. In the **Height** box, the height of the prompt in pixels.

10. In the **Prompt Text** box, type a message to the user.

11. Click **OK**, and save your changes.

Reference Material

Create Operations dialog box
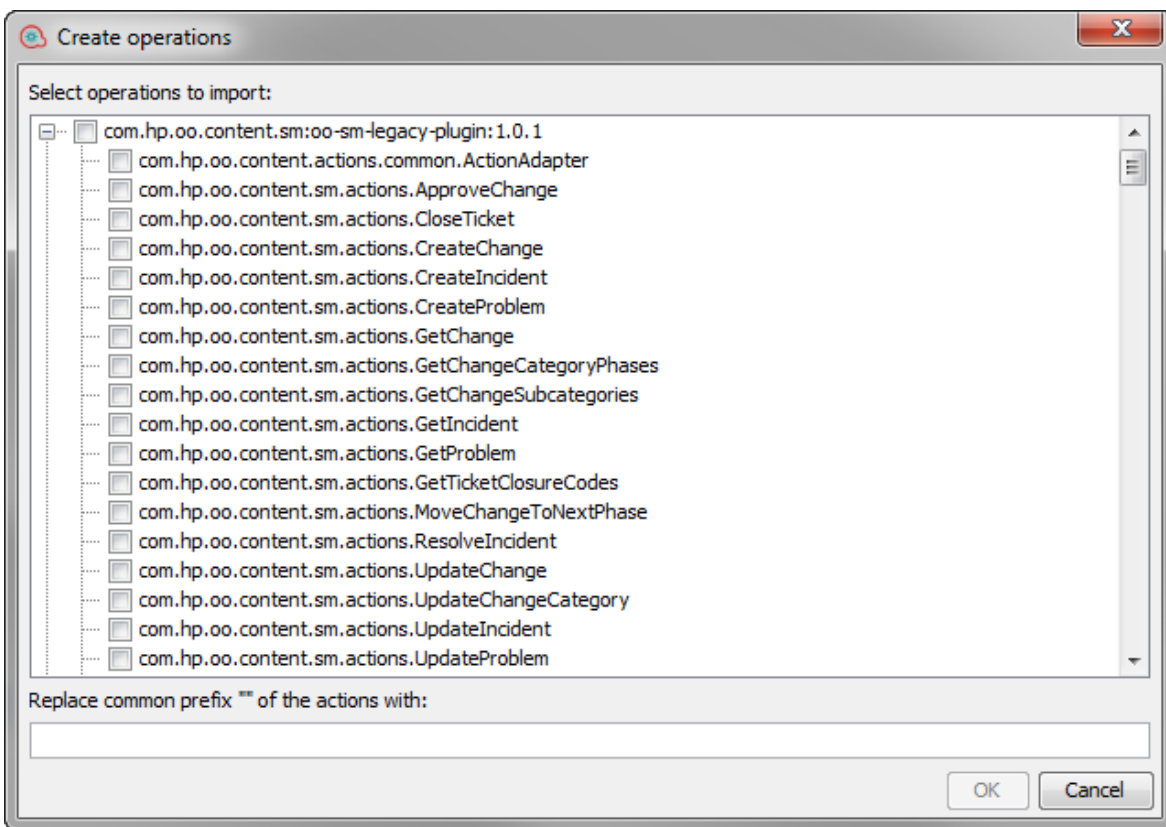
GUI item	Description
Select operations to import	Expand the plugin containing the actions that you need, and select the actions that you want use to create the operation from.

Step Inspector > Display tab

In the **Display** tab of the Operation Properties sheet, you can create a user prompt that is displayed to the user.

GUI item	Description
Always prompt user before executing this step	Select the checkbox if you want the prompt window to appear every time this step is run.
Prompt Title	Type the label that will appear in the title bar of the prompt window (up to 128 characters).
Prompt Width	Type the width of the prompt window in pixels.
Height	Type the height of the prompt window in pixels.
Prompt Text	Type the message that will appear in the body of the prompt window. You can include variables in the message. For example, ${dateTime}.

Import Plugin

GUI item	Description
Plugin location	Browse to and select the OO plugin that you want to import.
Import and create new operation	Select this check box to create a new operation from the imported plugin.

Operation Properties: Advanced tab

GUI item	Description
Link To	Displays the source operation, from which the selected operation was copied.
Location	Displays the location of the source operation.
Select	Lets you select a different source operation
Open	Opens the Properties sheet of the source operation.
Detach	Detaches the operation from the parent plugin.

Finding a Flow or Operation

There are a number of ways to find the flow or operation that you need:

- Browse the folders in the **Projects** pane and **Dependencies** pane

- View the descriptions of the folders, flows, and operations

- Run a search using the **Search** tab

- Run a search using the **Navigate to flow...** option

What do you want to do?

Browse the folders to find a flow or operation

The simplest way to locate a flow or operation is to browse through the folders.

If the folders have been named and structured correctly, this should help you to find what you need.

Use the descriptions to locate a flow or operation

You can view the description in an operation or flow, to see if it is the one you need.

- To view the description of an operation, open the operation in the authoring pane and click the **Description** tab.

- To view the description of a flow, open the flow in the authoring pane, and click **Properties** (at the bottom of the pane), and then click the **Description** tab.

> **Note:** Alternatively, it is possible to right-click the flow or operation in the **Projects** pane or **Dependencies** pane and select **Properties**.

Generate documentation to find a flow or operation

You can also use the Generate Documentation feature to gather this information for many flows and operations into one place. For more information on Generate Documentation, see "Generating Documentation about Flows and Operations" on page 658.

Search for an operation or configuration item using the Search tab

Using the **Search** pane, you can perform a full-text search throughout the Library. You can search for flows, operations, or configuration items by searching on the name or other field properties.

1. Click the **Search** tab at the bottom of the Studio window, to open the **Search** pane.

2. From the **Search** list, select a field to search on. The search criteria are: Name, UUID, Type, Plugins, Description, Categories, Inputs, Domain Terms, <all fields> and <with Lucene query>.

   > **Tip:** Use **UUID** to search for duplicate UUIDs.

   > **Note:** To include all the fields in your search, leave the **Search** list set to **<all fields>**.

3. In the **for** text box, type the text you want to search for.
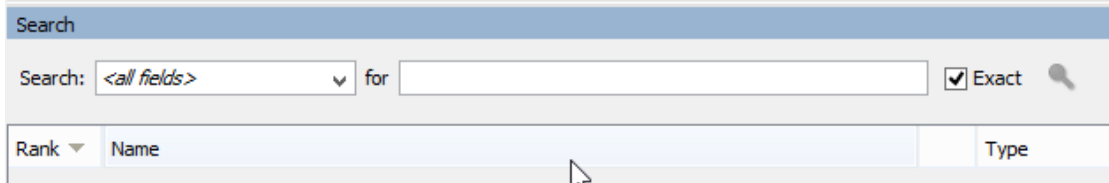
   If the search string contains a space, you can specify either an exact search or a search that includes spaces. An exact search treats the entire string, including any spaces that it contains, as a single search value. If you insert space, the search looks for all of the strings that are separated by spaces.

   > **Note:** The search for the search string you type is not case-sensitive.

4. Define how the search treats spaces:

   - To specify a search that treats spaces as part of a single search string, select the **Exact** check box.

   - To specify a search that treats spaces as separators for alternate search strings, clear the selection from the **Exact** check box.

5. Click the search button 🔍.

6. Check the text in the **Description** tab to identify the operation that you need.

   In search results, the description is taken from the **Description** tab of the operation and includes information that is vital to getting the most from your use of the operation, including:

   - The kind of information the operation's inputs need.

   - The information that the results include.

   - The operation's requirements and assumptions.

   To read the entire description, see the **Description** tab in the operation's **Properties** sheet.

Search using a previous search command

1. In the **Search** pane, click the **Search History** button.

   The Previously Entered Search Strings window opens.
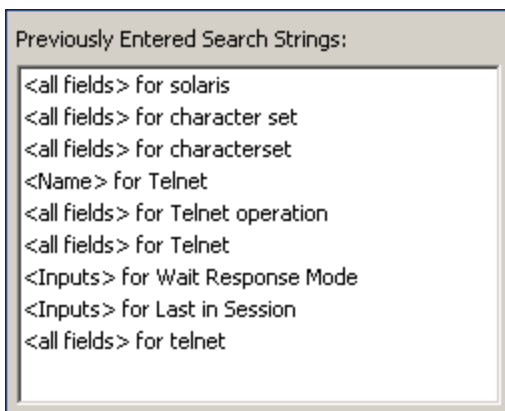
   This window contains a list of up to 25 previous search commands.

   The format of the search commands in the list is "<field name> for search text".

2. Double-click a search command from the list to run it. It is then added to the top of the list.

Sort search results

Click the column header in any of the columns, to sort the search results by that parameter.

Search with Lucene syntax

To target more-specific results, you can construct a search with the Apache Lucene syntax. For more information on the Lucene search syntax, see the Apache Software Foundation Web site.

1. Click the **Search** tab at the bottom of the Studio window, to open the **Search** pane.

2. From the **Search** drop-down list, select **<with Lucene query>**.

3. In the **for** text box, type your query, using Lucene search syntax, and then click the search button
   .

   The simplest Lucene search syntax is:

   `<searchable_field_name>:<string to search for>`

   Tips for searching:

   ○ The search uses a Boolean AND. If you type two words with AND, the search returns only
     operations or flows that contain both words. If you type two words without AND, the search
     finds any results that include either word.

   ○ To obtain results that include only a string that has a space in it, such as in the search
     category:database server, enclose the string in quotation marks: category:"database server"

   You can search for the following field names. Note that this list includes sample search strings.

   ○ Flow or operation name

     Examples:

     ```
     name:Get Temp Dir
     name:Clear Temp Dir
     ```

   ○ Operation type

     Example:

     ```
     type:cmd
     ```

   ○ Category

     Example:

     ```
     category:network
     ```

   ○ Input name

     Example:

```
inputs:server
```

○ Flow or operation UUID

Example:

```
id:1234-3453-3242-32423
```

○ String contained in flow or operation descriptions

Example:

```
description:clear
```

Access an operation from the Search pane

You can work with operations and flows directly from the search results, opening them for editing or adding them to a flow that is open in the authoring pane.

- To open an operation's **Properties** sheet or a flow's diagram, double-click in the row of the operation, in the search results.

- To create a step from an operation in the search results, drag the operation from the **Search** pane onto a flow diagram.

Search for a flow using the Navigate to flow... option

You can also use the **Navigate to flow...** option to easily find and open a flow in your workspace.

1. Select **Tools > Navigate to flow...** .

   The **Search flows by name:** window opens.

2. Type in part of the flow name.

   The window is immediately populated with search results. For example, if you search for a flow whose name contains the string "use", you may see the following results:
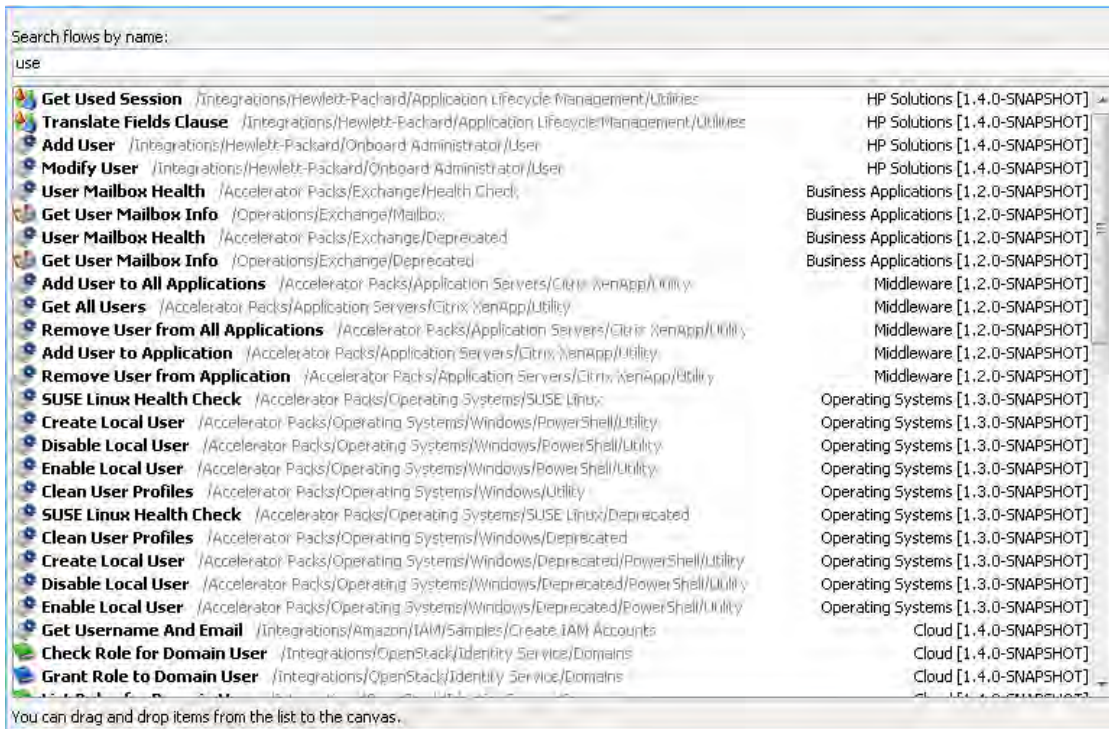
3. Use the Up and Down arrow keys to move in the list, then press Enter to open the selected item. Alternatively, you can open an item with a double-click.

4. You can create steps in the current Flow Editor by dragging and dropping flows from the list.

Reference Material

Search pane

GUI item	Description
Search	• To run the search on one field, select the field on which to run the search. • To include all the fields in your search, select **<all fields>**. • To search using Lucene query, select **<with Lucene query>**.
for	Type the string you want to search for.

Exact	• To specify a search that treats spaces as part of a single search string, select the **Exact** checkbox • To specify a search that treats spaces are separators for alternate search strings, clear the selection from the **Exact** checkbox.
Rank	Displays the ranking of each search result. More stars means a higher rank.
Type	Displays the type of item that was found, for example, a flow.
Path	Displays the location where the item is stored.
Description	Displays the description of the item, taken from the item's **Description** tab.
Search History	Click to display the Previously Entered Search Strings window, and reuse a search command.

Search and Replace References to an Operation or Subflow

Large projects in Studio usually include many flows and operations. Also, generally, the flows are quite complex.

There may be cases in which you want to change the reference to an operation or subflow globally throughout all open projects or in a specific flow.

For example, you may have a reference in your flows to a subflow named Change Status, and now you want to change this reference to a new subflow you created named Update Status. This can be done easily with the **Replace Reference to Operation or Subflow** functionality.

The search and replace can be run on all open projects, the selected folder or the current flow. The search and replace is case insensitive.

> **Note:** You cannot replace references within a content pack.

For example, you may have a reference in your flows to a subflow named **Change Status**, and now you want to change this reference to a new subflow you created named **Update Status**. This can be done easily with the **Replace Reference to Operation or Subflow** functionality.

What do you want to do?

Find and Replace a Reference to an Operation or Subflow

1. Select **Edit > Replace** or press **Ctrl+R**.

   The Replace Flow or Operation Reference dialog opens.

2. In the **Find what** field, type or paste in the full path of the operation or subflow you are looking for. For example, **/My Project/Library/My Ops Flows/Restart Service**.

   or:

   Click [button] and select the subflow or operation from the project tree.

3. Similarly, in the **Replace with** field, type or paste in the full path of the operation or subflow to use

   to replace the original one or click [button] and select the subflow or operation from the project tree.

   > **Note:** In both the **Find what** and **Replace with** fields, when you type in a character(s), the field is immediately populated with search results. For example, if you type in the string "rest" you may see the following results:
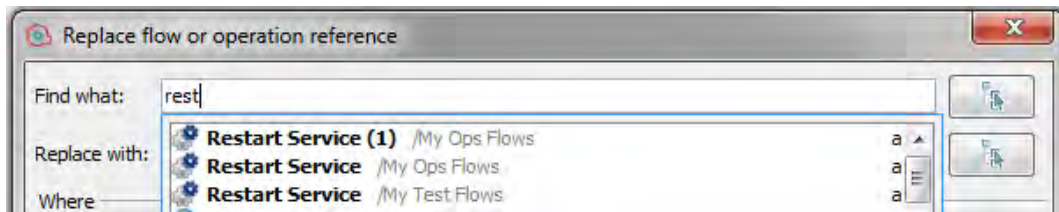
4. In the **Where** area, select the scope in which you want to search:

   - All open projects
   - Selected folder
   - Current flow

5. Select **Include subflows** to include subflows in the search for the source item according to the criteria selected above (i.e., subflows of all open projects, the selected folder or current flow).

6. In the **Inputs** area, select one of the following options:

   - **Merge step inputs** Merge the inputs of the source item with inputs of the target item
   - **Overwrite step inputs** Overwrite the inputs of the target item with the inputs of the source item

7. When you replace a step that points to an operation, note that the results of a step may be mapped

to one of the operation's outputs (as defined in the Results tab of the Step Inspector).

If this output is deleted, there will be unmapped results.

In the **Results** area, select **Remove Unmapped Results** to remove these results.

8. In the **Additional options** area, select the relevant check boxes:

   ○ **Remove step scriplet:** Select this check box to the remove the step scriptlet in the target item if it is no longer needed.

   ○ **Replace step description:** Select this check box to replace the target item's step description with the source item's step description in the **Description** tab:
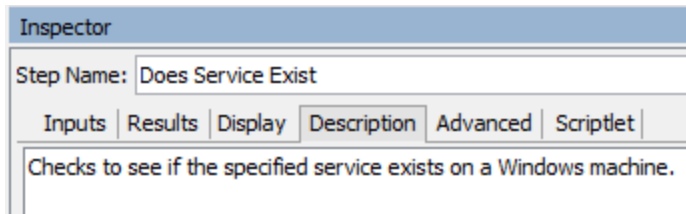
Inspector
Step Name: Does Service Exist
Inputs | Results | Display | Description | Advanced | Scriptlet
Checks to see if the specified service exists on a Windows machine.

   ○ **Replace step name** Select this check box to replace the target item's step name with the source item's step name.

9. To begin the search, click **Find Next**.

   A message appears showing the next occurrence of the item. For example, a search for the step **Does Service Exist?** produces the following results:

   ```
   Found match in flow abc/Library/My Ops Flows/Restart Service
   ```

10. To replace this occurrence of the item, click **Replace**. To replace all occurrences of the item (in the locations specified in the **Where** area), click **Replace All**.

> **Note:** If you open the **Replace flow or operation reference** dialog when a flow is open, the **Find What** field is automatically filled with the current step reference.

Reference Material

Replace Reference
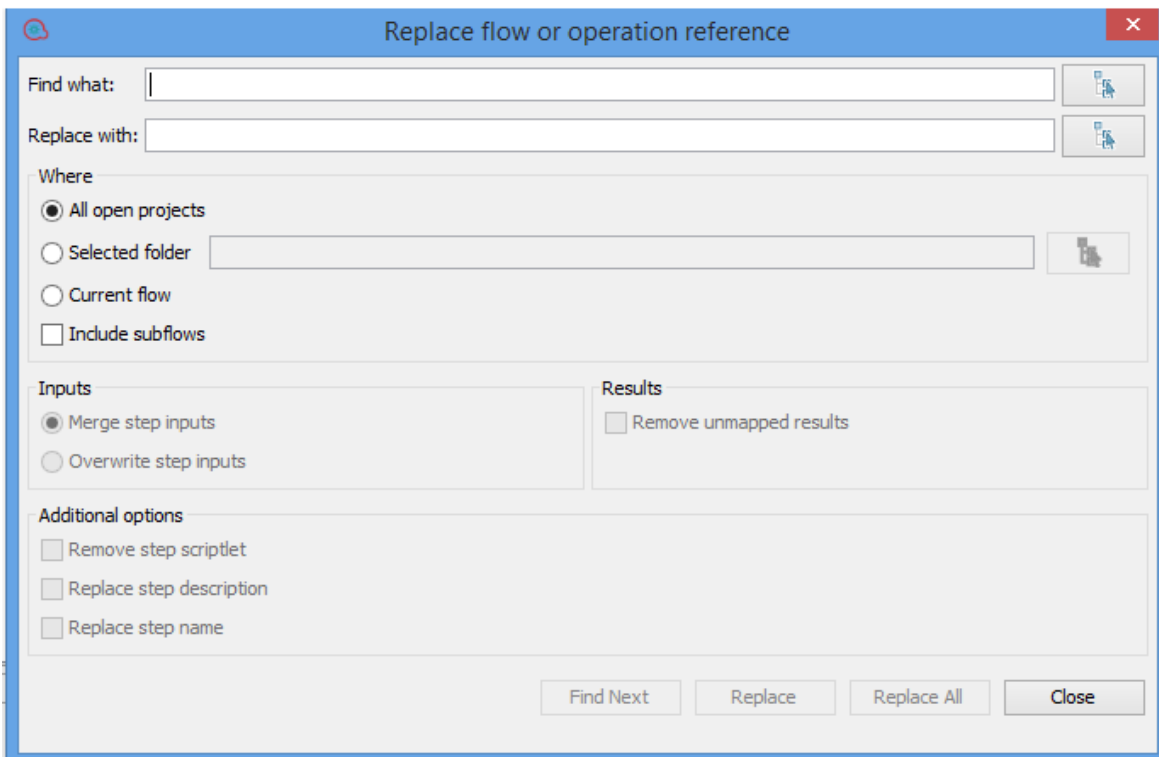
GUI item	Description
Find what	Type or paste in the full path of the operation or subflow you are looking for or click [icon] and select the subflow or operation from the project tree.
Replace with	Type or paste in the full path of the operation or subflow to use to replace the original one or click [icon] and select the subflow or operation from the project tree.
Where	Select where to search for the source item.
All open projects	Search for the source item in all the projects currently open in Studio.
Selected folder	Search for the source item in the selected folder only. Select a folder using the [icon] button.

Current flow	Search for the source item in the current flow only.
Include subflows	Include subflows in the search for the source item according to the criteria selected above (i.e., subflows of all open projects, the selected folder or current flow).
Inputs	
Merge step inputs	Merge the inputs of the source item with inputs of the target item.
Override step inputs	Overwrite the inputs of the target item with the inputs of the source item.
Results	
Remove unmapped results	Remove unmapped results of a step in the target item. See Step 8 in the "Find and Replace a Reference to an Operation or Subflow" on page 649 procedure for details.
Additional options	
Remove step scriptlet	Select this check box to remove the step scriptlet in the target item if it is no longer needed.
Remove step description	Select this check box to replace the documentation property of the step with the documentation of the new flow/operation.
Replace step name	Select this check box to replace the target item's step name with the source item's step name.

Copying Flows and Operations

There are different ways to copy flows and operations:

- If you copy a flow or operation, you can paste it in any folder that is not sealed. If you copy a flow, this copies only the flow and not the operations that comprise the flow.

- If you click **Copy Deep**, this copies not only the flow but also all the operations that comprise the flow. You would do this when you will need to modify the operations in the new flow, and do not want to affect the original operations.

- If you duplicate a flow or operation, the duplicate is automatically placed in the same folder as its original, and is named **Copy of <name>**.

- If you cut a flow or operation, you remove it from its current location, to be pasted somewhere else.

- If you click **Copy Path**, the relative path of the selected items into the operating system clipboard, so you can easily paste them into a text editor.

Copying operations that were created from Action plugins

Soft copies

When you copy an operation that is linked to an action plugin jar file, the copied operation continues to reference the original operation. If the action plugin jar file is upgraded, when you update the original operation to call the new version, the copied operations are all updated automatically. This is known as a **soft copy**.

Note that the link to the action plugin jar file is the only item that is automatically updated in the copied operations. Changes that you make to the original operation's input, output, variables, scriptlets, and so on, are not updated in the copies.

You can copy operations from a content pack, but keep a reference to the parent contact pack. If a future fix is implemented to the parent plugin, then the soft copy will also receive the fix. In certain cases, you may not want to receive fixes to the operation. In this case, you can detach the operation from the parent plugin. However you will need to manually fix the detached operations.

Soft copies have advantages and disadvantages:

- The disadvantage is that if the original operation gets deleted, the copy is parentless and loses its link to the plugin. In this case, a new parent will have to be selected manually.

  Note: In order to fix an orphan soft copy operation that points to a nonexistent parent, use the **Select** button from the **Advanced** tab of the operation to attach a parent operation.

- The advantage is that if the original operation is updated with a different plugin version, the soft copy gets updated.

For more information about creating operations from action plugin jar files, see "Creating Operations" on page 632.

Hard copies

In versions of OO prior to 10.00, when you copied an operation linked to an action plugin, this created a **hard copy**, meaning that the copy 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 OO 10.x, you can create a hard copy by creating a new operation and selecting the relevant plugin. This method creates a new operation according to the IAction getTemplate or the @Action metadata. Is not possible to create a hard-copy of an operation that also duplicates its inputs and outputs.

Hard copies have advantages and disadvantages:

- The advantage is that if the original operation gets deleted, the copy is not affected and doesn't remain parent-less.

- The disadvantage is that if the original operation is updated with a different plugin version, the hard copy does not get updated.

> **Note:** The **Select** button in the **Advanced** tab in the operation editor is not available for hard copies.

Changing From a Soft Copy to a Hard Copy

It is possible to detach a soft copy from its parent and make it a hard copy. In the operation's **Advanced** tab, you can detach the operation from its parent by clicking the **Detach** button. A confirmation message appears and the plugin's GAV parameters are taken from the original parent.

Replacing a Plugin in a Hard Copy

You can search for all the hard copied operations that use a specific plugin, and choose the operation, and replace the plugin GAV parameters.

What do you want to do?

Copy a flow or operation

1. In the **Dependencies** pane or **Projects** pane, right-click the flow or operation that you want to copy.
2. Select **Edit** > **Copy**.
3. Navigate to the folder in which you want to place the copy, right-click and select **Edit** > **Copy**.

Duplicate a flow or operation

When you duplicate a flow or operation, the duplicate is automatically placed in the same folder as its original, and is named **Copy of <name>**.

1. In the **Dependencies** pane or **Projects** pane, right-click the flow or operation that you want to copy.
2. Select **Edit** > **Duplicate**.

Deep copy a flow or operation

Deep copying a flow copies not only the flow but also all the operations that comprise the flow.

1. In the **Dependencies** pane or **Projects** pane, right-click the flow that you want to copy.

2. Select **Edit** > **Copy Deep**.

3. Navigate to the folder in which you want to place the flow and its operations, right-click and select **Edit** > **Paste**.

Copy the relative path of a flow, operation or configuration item

Studio allows you to copy the relative path of the selected items into the operating system clipboard, so you can easily paste them into a text editor.
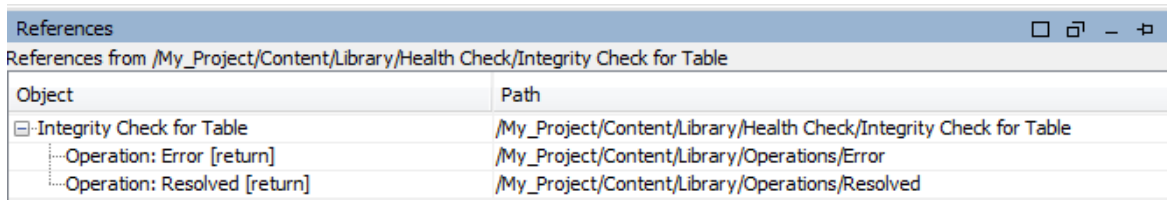
1. In the **Dependencies** pane or **Projects** pane, right-click the flow, operation or configuration item that you want to copy.

2. Select **Edit** > **Copy Path**.

3. Navigate to the folder in which you want to place the flow and its operations, right-click and select **Edit** > **Paste**.

Cut a flow or operation

1. In the **Dependencies** pane or **Projects** pane, right-click the flow or operation that you want to move.

2. Select **Edit** > **Cut**.

3. Navigate to the folder in which you want to place the flow or operation, right-click and select **Edit** > **Cut**.

Finding Out How Flows and Operations are Used

You can learn more about ways to use and implement an operation or flow by looking at how it is used in existing flows. This can be done in the **References** pane.

References	□ ⬚ – ⊟
References from /My_Project/Content/Library/Health Check/Integrity Check for Table	

Object	Path
⊟ Integrity Check for Table	/My_Project/Content/Library/Health Check/Integrity Check for Table
⋯ Operation: Error [return]	/My_Project/Content/Library/Operations/Error
⋯ Operation: Resolved [return]	/My_Project/Content/Library/Operations/Resolved

Studio has two kinds of references:

- References *to* the operation or flow – lists the flows that have a step created from the selected operation or flow

- References *from* the operation or flow – lists the objects (selection lists, permissions assigned to groups, system filters, and so on) that the selected operation or flow makes use of. In the case of flows, these are the operations (including subflows) from which the flow's steps were created.

Best Practices

Changing an operation or flow can break other flows that use it. Before making changes to a flow or operation, use **References** > **What uses this?** to check whether other flows use it.

What do you want to do?

Identify what uses a flow or operation

1. In the **Project** pane, right-click the operation or flow, and select **References** > **What uses this?**.
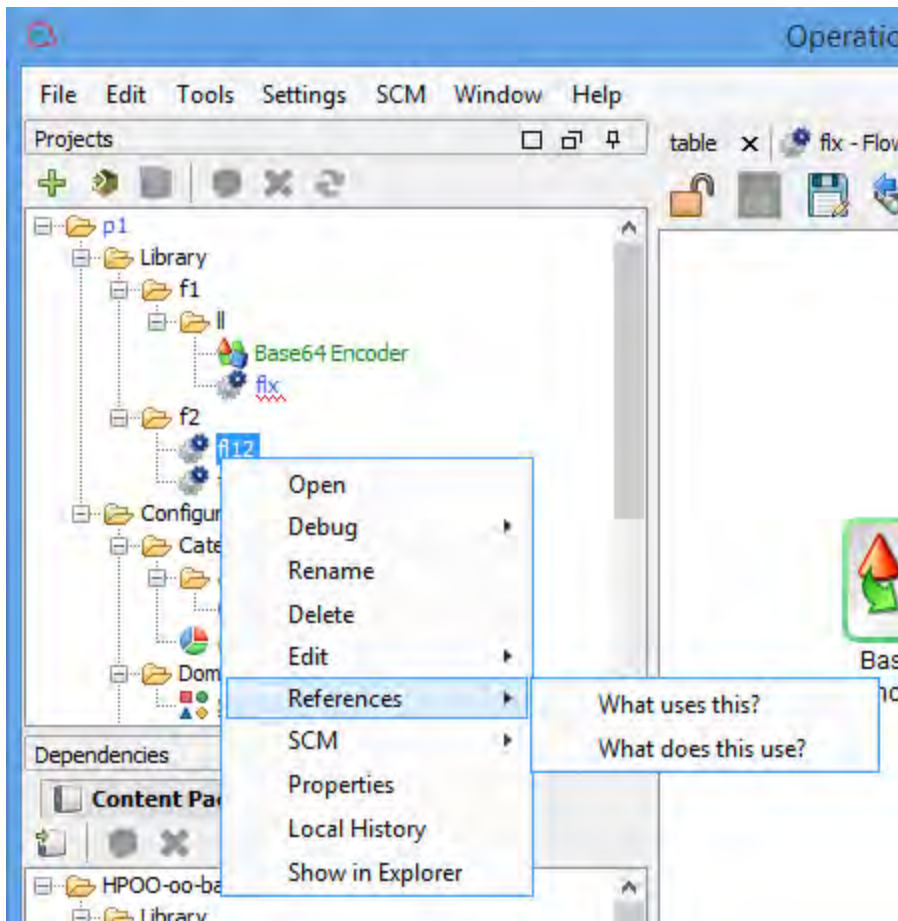
The **References** pane opens, displaying the references to the operation or flow.

2. Double-click a row in the **References** pane to open the item for editing.

   Note: To open multiple items, select them using the SHIFT or CONTROL keys, right-click, and select **Open**.

Identify what a flow or operation uses

1. In the **Project** pane, right-click the operation or flow.

2. Select **References** > **What does this use?**.

   The **References** pane opens, displaying the references from the operation or flow.

   Tip: The referenced flows and operations are valuable as samples that you can copy, paste, and modify.

Reference Material

Reference pane

In the **References** pane, you can see how an operation or flow is used in existing flows.

References	□ ⊡ – ⊣
References from /My_Project/Content/Library/Health Check/Integrity Check for Table	

Object	Path
⊟ Integrity Check for Table	/My_Project/Content/Library/Health Check/Integrity Check for Table
└ Operation: Error [return]	/My_Project/Content/Library/Operations/Error
└ Operation: Resolved [return]	/My_Project/Content/Library/Operations/Resolved

GUI item	Description
Object	Displays the object that is used or uses the selected flow or operation
Path	Displays the location of the object that is used or uses the selected flow or operation

Generating Documentation about Flows and Operations

> **Important**: In the current version, the Create Documentation functionality is not supported at runtime. You can generate documentation from within Studio, but if you create a flow with a Create Documentation step, this step will not work at runtime.

You can document flows and operations, to provide more information about them for other users:

- Export the flow as a PNG image.
- Use the Generate Documentation feature to create an HTML page with information about flows and operations.

Structure of Generated Documentation

When you generate documentation, an HTML page named **index.html** is created. This page contains the following frames:

- **Overview** frame – In the upper-left, the **Overview** frame lists the sub-folders contained in the folder for which you generated documentation. Select a folder to display its contents in the **Folder** frame.

- **Folder** frame – In the lower-left, the **Folder** frame lists the flows and operations in the folder that is selected in the **Overview** frame.

- **Header** frame – In the upper-right, the **Header** frame contains an OO banner.

- **Operation** frame – In the lower-right, the **Operation** frame displays the description of the flow or operation. This is the information that was entered in the **Description** tab of the **Properties** sheet.
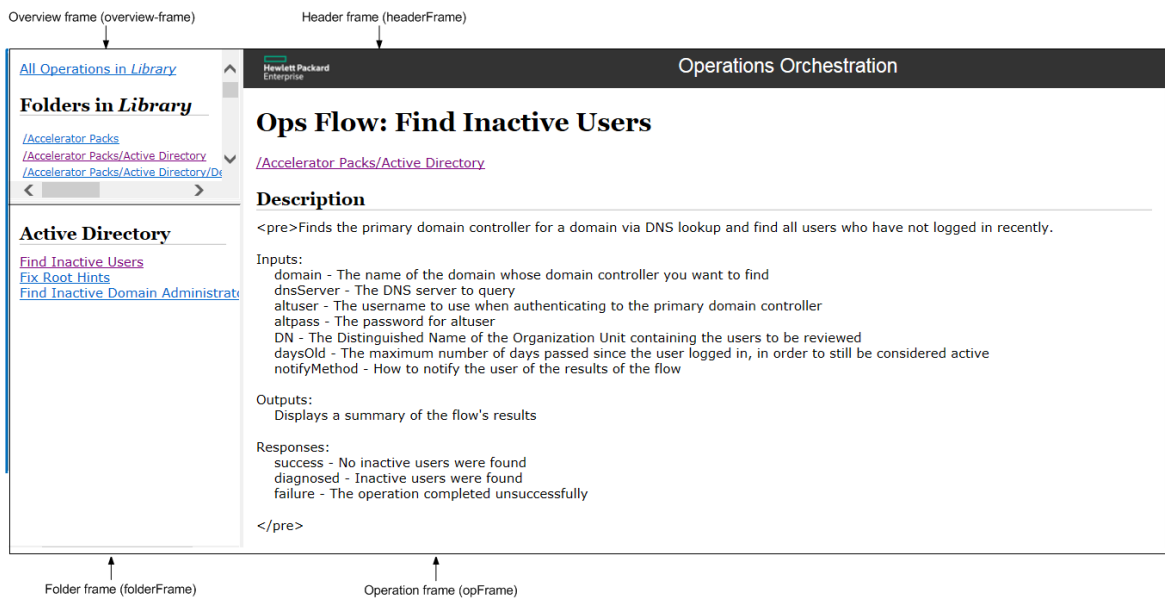
What do you want to do?

Export a flow as a PNG image

1. Open the flow in the authoring pane.

2. Right-click anywhere in the authoring pane and select **Export to PNG**.

3. Browse to the location in which to store the image and click **Save**.

Generate documentation in standard format

You can generate documentation for a folder containing specific flows and/or operations, or for the entire **Library** folder.

1. Right-click the folder for which you want to create documentation.

   Note: You can only generate documentation in the **Content Packs** tab. This option is not available in the **Library** view.

2. Select **Generate Documentation** > **Standard Format**.

3. Browse to the location in which to store the documentation files and click **Save**. The HTML file, **index.html**, opens in a Web browser.

4. If you need to overwrite a previous version of **index.html**, click **Yes To All**.

> **Note:** If you prefer not to overwrite the previous documentation, click **Cancel** and repeat the process, while saving the files in a different location.

Generate documentation in a custom format

1. Right-click the folder for which you want to create documentation.

2. Select **Generate Documentation** > **Custom Format**.

3. In the Select Documentation Generation Template dialog box, select the template to be used when generating the documentation.

4. Browse to the location in which to store the documentation files and click **Save**. The HTML file, **index.html**, opens in a Web browser.

Create a customized Generate Documentation template

Documentation templates are stored in the **Studio\template** folder. They have the suffix `.vm` and can be edited in a text editor. For information about the templates, see the *Reference Material* section below.

Any new `.vm` files that you create in the **Studio\template** folder will appear in the template list in the Select Documentation Generation Template dialog box.
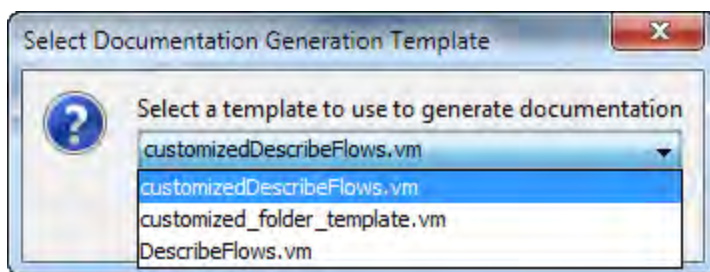
1. Make a copy of the relevant `.vm` template, and rename the copy.

> **Caution:** Do not modify or rename the original `.vm` templates.

2. In a text editor, make your changes to the new template, and save.

3. In Studio, right-click the folder that you want to document and select **Generate Documentation** > **Custom Format**.

4. In the Select Documentation Generation Template dialog box, select the custom template that you created.

Reference Material

.vm Template Files

Folder_template.vm

The root template, which generates a frameset and calls the following to populate it:

- **All_folders_template.vm** - generates a list of the sub-folders of the folder and places it in **overview-frame** (upper-left).

- **All_ops_template.vm** - generates a list of all operations and places it in **folderFrame** (lower left).

- **Header.html** - places the header in **headerFrame** (upper right).

- **Folder_overview_template.vm** - generates information about one or more operations and places it in **opFrame** (lower right).

All_folders_template.vm

Generates a table of contents for the folders.

- **Header.css** – style sheet used for general fonts, colors, and so on.

- **All_ops_template.vm** – generates a list of all operations and creates a link to display it in **folderFrame** (lower left).

- **Folder_contents.vm** – generates a list of the selected folder's contents and creates a link to display it in **folderFrame** (lower left).

All_ops_template.vm

Generates a table of contents for all operations and the documentation for every child operation.

- **Header.css** – style sheet used for general fonts, colors, and so on.

- **Op_template.vm** – generates and creates a link to display it in **opFrame** (lower right).

Folder_overview_template.vm

Generates a tabular summary describing the contents of a folder.

- **Header.css** – style sheet used for general fonts, colors, and so on.
- **Folder_contents.vm** – generates and creates a link to display it in **folderFrame** (lower left).

Op_template.vm

Generates documentation for a single operation.

- **Header.css** – style sheet used for general fonts, colors, and so on.
- **Folder_template.vm** – generates and creates a link to display it in same frame (up to parent folder).
- **Folder_contents.vm** – displays the folder contents in **folderFrame**.

Flow_template.vm

Generates the documentation for a single flow.

- **Header.css** – style sheet used for general fonts, colors, and so on.
- **Flow_template.vm** – generates and creates a link to display it in same frame (up to parent folder).
- **Folder_contents.vm** – generates a list of the folder contents and creates link to display it in **folderFrame** (lower left).
- **Op_template.vm** – generates and creates a link to display it in **opFrame** (lower right).

Folder_contents.vm

Generates a table of contents for a single folder.

- **Header.css** – style sheet used for general fonts, colors, and so on.
- **Op_template.vm** – generates and creates a link to display it in **opFrame** (lower right).

Header.html

The Hewlett Packard Enterprise banner.

Hp_rockwell.css

Style sheet for the Hewlett Packard Enterprise banner.

Hp_steps_307x39.jpg

Graphic for the Hewlett Packard Enterprise banner.

Logo_hp_smallmasthead.gif

Logo for the Hewlett Packard Enterprise banner.

Managing Version History of Flows and Operations
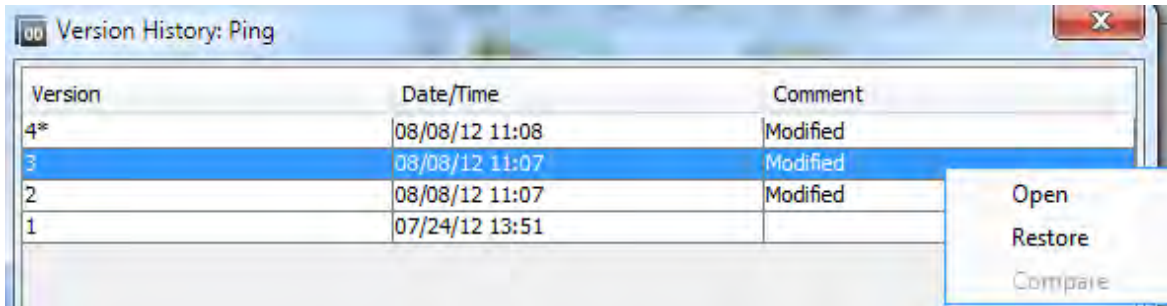
Each time that a configuration item, flow, or operation is saved, a new version of the item is created. The Version History dialog box list these versions, and lets you:

- View an earlier version of an item
- Save the earlier version under a different name
- Restore an item to its earlier version
- View the differences between two versions

> **Note:** Version history functionality only applies to your local authoring work in Studio, and not to versions saved to a source control management repository. Working with version history is something you would do *before* committing your work into the source control repository. If you have committed a flow to a shared repository, it is *not* recommended to use the OO local version history control function to revert to a previous version.
>
> For full details of how to handle version history with a source control repository, see "View a history of Git operations in the repository" on page 371.

What do you want to do?

Open an earlier version of a configuration item, flow, or operation

When you open an earlier version of an item for viewing, you can save the earlier version under a different name.

1. Right-click the configuration item, flow, or operation and select **Show History**.

2. Right-click the version you need, and then click **Open**. The selected version is opened in the authoring pane.

3. To preserve the version that you have opened, click the **Save As** button and give it a unique name. The two versions of the project are saved separately.

4. Click **OK** to close the Version History dialog box.

Restore a configuration item, flow, or operation to a previous version

This procedure restores a configuration item, flow, or operation to an earlier version. To keep both the current and earlier versions, see *Open an earlier version of a configuration item, flow, or operation*.

1. Right-click the configuration item, flow, or operation and select **Show History**.

2. Right-click the version you want to revert to, and then click **Restore**. The version that you are restoring to is opened in the authoring pane.

3. Click **OK** to close the Version History dialog box.

4. Save the project. The opened version is saved over the current version.

Compare versions

The Version History dialog box also enables you to compare versions of a configuration item, flow, or operation. The current version is displayed on one side, and a previous version is displayed on the other.

1. Right-click the configuration item, flow, or operation and select **Show History**.

2. Hold down the **Ctrl** key and select both the current version (on the top line) and an earlier version.

3. Right-click and select **Compare**. The differences between the current state of the item and the earlier version are displayed.

4. Click **OK** to close the Version History dialog box.

Reference Material

Version History dialog box
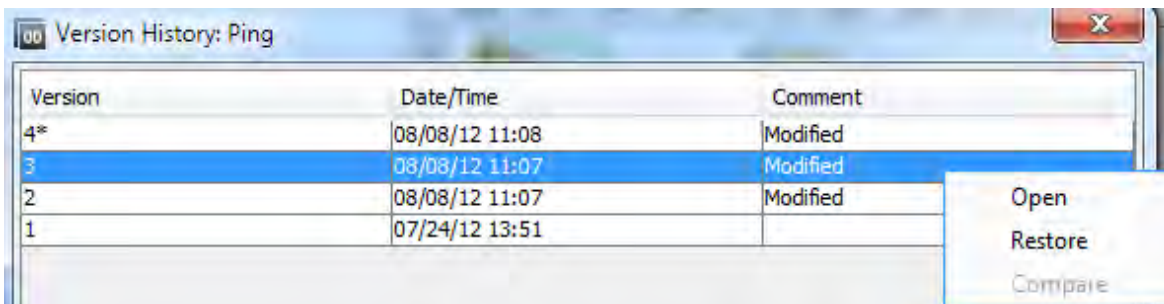
Menu item	Description
Version	The number of the project version, generated automatically.
Date/Time	The date and time that the version was changed.
Comment	The type of change that was made.

Bookmarking Flows and Operations

The **Bookmarks** pane makes it easier to find and use the operations and flows that you use frequently. Bookmarked flows and operations are still available in their normal location in the Library.

Adding a flow or operation to the **Bookmarks** pane makes it available from the right-click menu in the authoring canvas.

You can export and import bookmarks from one installation of Studio to another.

What do you want to do?

Add a bookmark

1. Click the **Bookmarks** tab in the upper-right of the Studio window, to open the **Bookmarks** pane.

2. To keep the pane open, click the **Pin** 📌 icon in the upper-right-hand corner of the pane.

3. Drag a flow or operation from the Library or the **Search** pane results to the appropriate shelf of the **Bookmarks** pane.

Add a shelf to the Bookmarks pane

The **Bookmarks** pane has two default shelves, for flow and operations, but you can add customized shelves to organize your bookmarks.

1. Right-click in the title bar of one of the shelves of the **Bookmarks** pane, and then select **Add**.

2. Enter a name for the shelf and click **OK**.

Rename a shelf

1. Right-click in the title bar of the shelf that you want to rename, and then select **Rename**.

2. Enter a new name for the shelf and click **OK**.

Remove a shelf

1. Right-click in the title bar of the shelf that you want to remove, and then select **Remove**.

2. Click **Yes** in the confirmation window.

Hide/show a shelf

1. Right-click in the title bar of the shelf that you want to hide, and then select **Hide**. The shelf is no longer visible in the **Bookmarks** pane.

2. To show the hidden shelf, right-click in the **Bookmarks** pane and select **Show**, and then select the name of the hidden shelf.

3. To show all the hidden shelves, right-click in the **Bookmarks** pane and select **Show All**.

Move a shelf up or down

Right-click in the title bar of the shelf that you want to move up or down, and then select **Move Up** or **Move Down**.

Collapse/expand a shelf

1. To collapse a shelf, click the double chevrons ⌃ in the shelf's title bar. The shelf title is visible but the bookmarks in the shelf are hidden.

2. To expand the shelf, click the double chevrons again.

Export bookmarks

You can export your bookmarks from one installation of Studio and import them to another.

1. Right-click anywhere inside the **Bookmarks** pane, and then select **Export**.

2. In the Export bookmarks dialog box, select the bookshelves that you want to export.
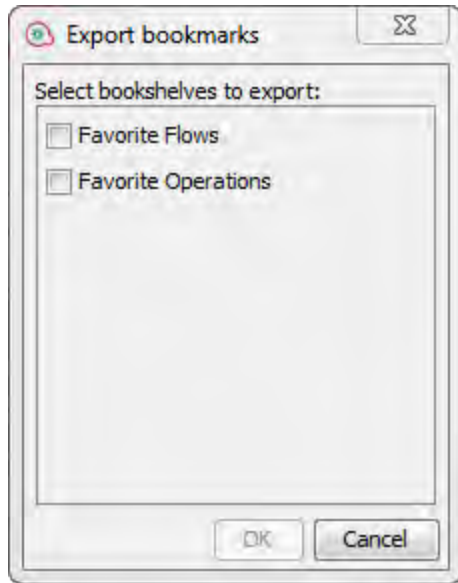
3. Click **OK**. The Select export file dialog box appears.

4. Browse to the location where you want to save the bookmarks, and enter a name for the file.

5. Click **Save**.

Import bookmarks

1. Right-click anywhere inside the **Bookmarks** pane, and then select **Import**.

2. In the Select import file dialog box, locate and select the bookmarks file, then click **Open**.

3. In the Import bookmarks dialog box, select the bookshelves that you want to import, and then click **OK**.
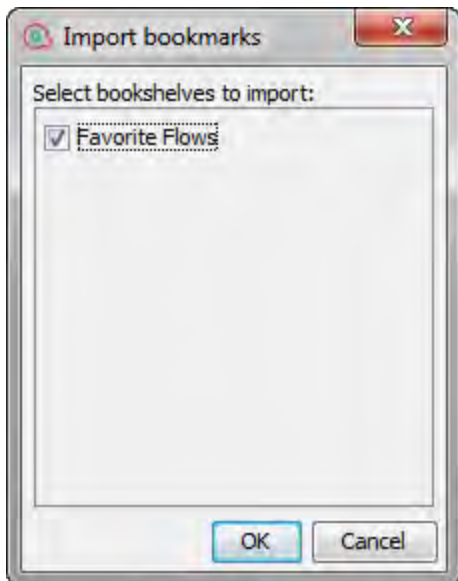
During import, bookmarks in the same bookshelf are merged, based on the UUID. Existing bookmarks remain, no duplicates are created, and new entries are added to the bookshelf.

Copy a flow or operation from the Bookmarks pane to a project

You can also drag flows and operations from the Bookmarks pane to the **Projects** pane, to copy them to a project.

1. Select a flow or operation in the **Bookmarks** pane.

2. Drag the item to a project in the **Projects** pane.

Reference Material

Bookmarks menu

When you right-click in the **Bookmarks** pane, the Bookmarks menu is displayed. The items shown in the menu vary, depending on the item that was selected when you right-clicked.

Menu item	Description
Add	Adds a new shelf to the **Bookmarks** pane.
Remove	Removes the selected shelf from the **Bookmarks** pane.
Rename	Renames the selected shelf in the **Bookmarks** pane.
Move Up	Moves the selected shelf higher in the **Bookmarks** pane.
Move Down	Moves the selected shelf lower in the **Bookmarks** pane.

Hide	Hides the selected shelf in the **Bookmarks** pane.
Show	Lets you select a hidden shelf, in order to show it in the **Bookmarks** pane.
Show All	Shows all hidden shelves in the **Bookmarks** pane.
Collapse ⌃	Collapses the shelf in the **Bookmarks** pane, so that the title is visible but the bookmarks are hidden.
Import	
Export	

Configuring Studio Properties

You can configure some of the properties in Studio by changing the default settings in the **Options** dialog. In addition, you can configure all the different properties in Studio by changing the default settings manually in the **Studio.properties** file. For details of all the settings, see the Reference Material.

Studio supports multiple workspaces. Therefore, each workspace has its own **studio.properties** file under the **.oo** folder.

- If you want the same properties to be used in all new workspaces, edit the **studio.properties.template** file located under **Studio\conf**.

- To configure different settings for each workspace, you can still edit each of the **studio.properties** files individually. These are located under the **.oo** folder inside each workspace.

In the following sections, you can see descriptions of the more frequently used parameters.

Defining the default value of Assign from and Assign to

By default, when you create a new input in an operation or a flow step, no value is assigned to the input source. Your flow may require that the input value to be assigned from a flow variable.

Similarly, by default, Studio does not assign the input value to a flow variable. Again, your flow may require that the input's value be assigned to a flow variable.

You can configure Studio so that its default behavior is to assign a flow variable of the same name as the input either or both of the source of the input's value. You do so in the **Studio.properties** file.

Changing the setting's value from **not-assigned** to **assigned** causes the **Assign from Variable** and/or **Assign to Variable** to be assigned with the flow variable of the input with the same name, when you:

- Add a new input to an operation or flow step

- Create a new step in a flow

- Create a new operation

Changing the default value of Otherwise to Use Constant

In Studio 10.20 and later, when you create a new input in an operation or a flow step, by default the **Otherwise** parameter is set to **Use Constant** with an empty value for the **Constant** value. With this assignment, some inputs may not pass validation.

Similarly, the same rule applies when you create a new step in a flow using drag and drop from the Project or Dependencies pane. For the step inputs that have the **Otherwise**parameter set to **Prompt User** using text before a drag and drop operation, after the drag and drop, these inputs are changed to **Use Constant** with an empty value. The step inputs that were set to **Prompt User**, but are marked as **Required** or are part of a list (selection list, domain term, flow variable) remain unchanged.

You can configure the default behavior in Studio to use **Prompt User** when a new input is created in an operation or a flow step. In this case, when a new step is created in a flow using a drag and drop operation, the **Prompt User** setting will remain unchanged.

Changing the default behavior from **Use Constant** to **Prompt User** causes the selected **Otherwise** option to be **Prompt User - Prompt for Text** when performing one of the following actions:

- Add a new input to an operation or flow step

- Create a new step in a flow

- Create a new operation

Showing special characters in the Context inspector

The ¶ button enables you to toggle on and off hidden formatting marks on variables. For example, white space, carriage returns, special characters, ^Ms, and more. By default, this is disabled.

When you enable this switch, the value elements inside the categories, Step Context, Global Context, Select Lists and System Properties, show these special characters and white space inside the tool tips associated with the value.

- If invalid or empty parameters are used in the studio.properties file, the default parameters are used.

- If the preferred value for a property is not properly displayed inside the tool tip, then you can switch to a different value, or leave that property empty.

- New line characters, <CR><LF>, <CR>, <LF> are followed by a new line in the tool tip along with a visual representation.

  - The vertical tab character \u000B is displayed as ^K.

  - The form feed character \u000C is displayed as ^L.

What do you want to do?

Edit the studio properties

1. In Windows Explorer, go to **oo/studio.properties** and open the file in a text editor.

2. If you know the name of the parameter you want to change, search for it in the text editor.

3. If you do not know the name of the parameter, review the table in "Configuring Studio Properties " on page 671 to find it.

Specify the default values for Assign from, Assign to and Otherwise using the Options dialog

1. In Studio, select **Configuration > Options**.

   The Options dialog opens:
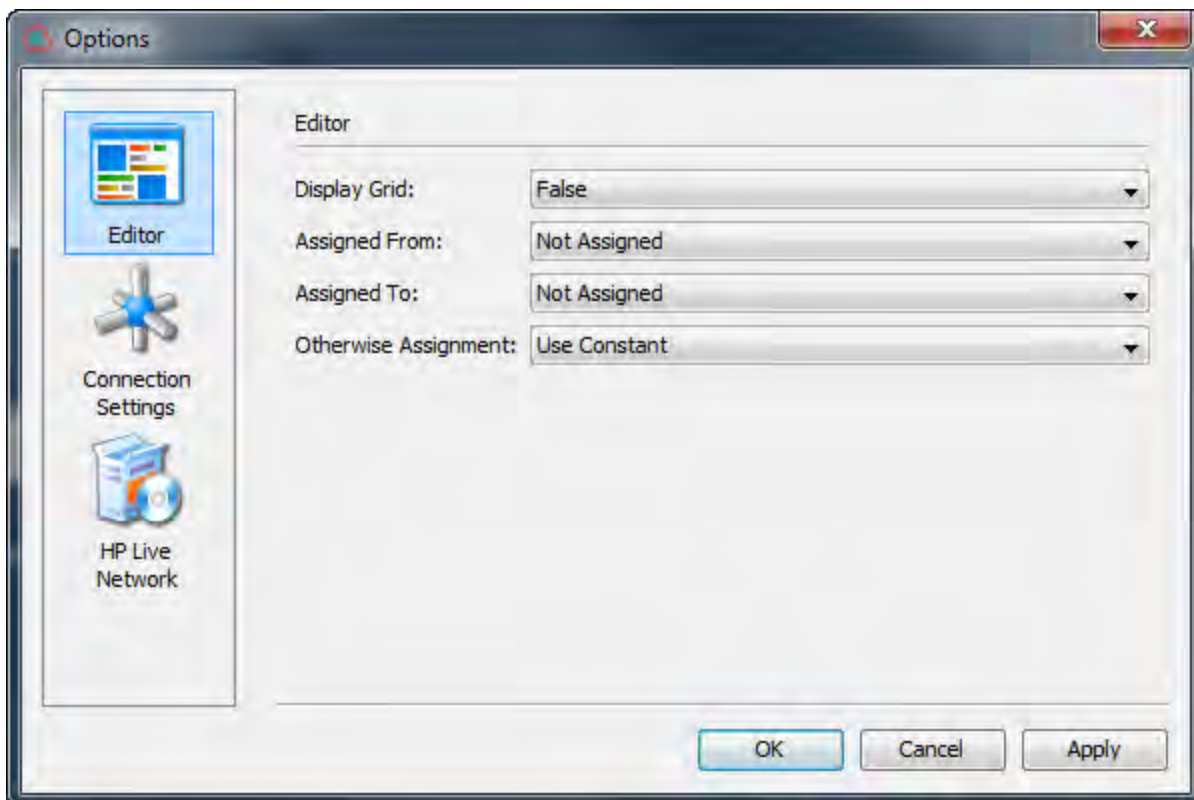
2. For **Assign From** and **Assign To**, select one of the following:

Not Assigned	No value is assigned to or from the input.
Assigned	The value of a flow variable with the same name as the input is assigned to or from the input.

3. For **Otherwise**, select one of the following:

Use Constant	The **Otherwise** field is set to **Use Constant** with an empty value.
Prompt User	The **Otherwise** field is set to **Prompt User**.

Define the default value of Assign from and Assign to manually

1. In Windows Explorer, go to **oo/studio.properties** and open the file in a text editor.

2. Open the file **studio.properties** for editing.

3. Find the following lines:

   dharma.studio.ui.inputinspector.assignfrom.selected=not-assigned

   dharma.studio.ui.inputinspector.assignto.selected=not-assigned

For each parameter, set one of the following values:

not-assigned	No value is assigned to or from the input.
assigned	The value of a flow variable with the same name as the input is assigned to or from the input.

Note: This behavior is true in the case of a clean installation of Studio 10.5x. If you upgrade to version 10.5x, the default values of **Assigned from Variable** and **Assigned to Variable** are automatically set to the values assigned in the previous version.

Change the default value of Otherwise to Use Constant

1. In Windows Explorer, go to **oo/studio.properties** and open the file in a text editor.

2. Open the file **studio.properties** for editing.

3. Find the following line:

   default.behavior.define.input=use_constant

4. For each parameter, set one of the following values:

use_constant	The **Otherwise** field is set to **Use Constant** with an empty value.
prompt_user	The **Otherwise** field is set to **Prompt User**.

Note: This behavior is true in the case of a clean installation of Studio 10.5x. If you upgrade to version 10.5x, the default value of is **default.behavior.define.input** is automatically set to the value assigned in the previous version.

Make the grid appear by default when opening the flow editor

This is a global setting to control the activation of the grid for all flow editors, it makes the flow editor start with the grid activated each time. You can still change the grid using the Show/Hide Grid button for an individual flow editor instance.

By default, the grid is not shown when opening the flow editor.

You can make the grid appear from the Options dialog or by editing the **studio.properties** file.

From the **Options** dialog:

1. In Studio, select **Configuration > Options**.

2. In the **Display Grid** field, select **True** to show the grid or **False** to hide the grid.

From the **studio.properties** file:

1. In Windows Explorer, go to **oo/studio.properties** and open the file in a text editor.

2. Open the file **studio.properties** for editing.

3. Find the following line:

   dharma.studio.ui.activegrid=false

4. Change the value of the setting to true:

   dharma.studio.ui.activegrid=true

Specify the number of characters for each tooltip

1. In Windows Explorer, go to **oo/studio.properties** and open the file in a text editor.

2. Open the file **studio.properties** for editing.

3. Find the following line:

   dharma.studio.ui.tooltip.showwhitespacecharacters.max.chars.per.line=5000

4. Change the value as required.

Show special characters in the context inspector

1. In Windows Explorer, go to **oo/studio.properties** and open the file in a text editor.

2. Find the following lines:

   dharma.studio.ui.tooltip.showwhitespacecharacters.max.chars.per.line=5000

   dharma.studio.ui.tooltip.showwhitespacecharacters.space=·

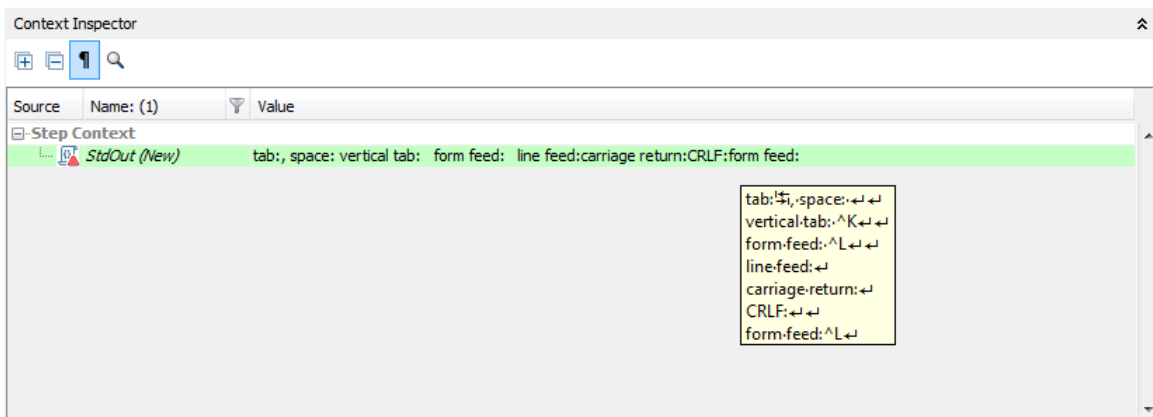   **dharma.studio.ui.tooltip.showwhitespacecharacters.tab=↹**

   dharma.studio.ui.tooltip.showwhitespacecharacters.enter=↵

   dharma.studio.ui.tooltip.showwhitespacecharacters.otherwhitespace=•

3. Change the values as required.

For example, if you define character overrides, you can see them in the Context Inspector as follows:

Disable ITOM Marketplace search

By default, the panel appears in the lower area of Studio. You can disable this panel by changing the value of the **hpln.enabled** parameter.

1. In Windows Explorer, go to **oo/studio.properties** and open the file in a text editor.

2. Find the following line:

   hpln.enabled=true

3. To disable the search, change the value to false:

   hpln.enabled=false

Specify the ITOM Marketplace connection/read timeout

1. In Windows Explorer, go to **oo/studio.properties** and open the file in a text editor.

2. Find the following lines:

   hpln.connection.timeout=180

   hpln.read.timeout=300

3. Set the timeout value as required. For no timeout, set these parameters to 0.

Specify the maximum number of ITOM Marketplace search results

This configuration setting allows you to specify the maximum number of search results to be displayed inside the search results panel.

1. In Windows Explorer, go to **oo/studio.properties** and open the file in a text editor.

2. Find the following lines:

hpln.search.results.maximum.count=100

3. Set the maximum number of search results as required.

Reference Material

Studio Properties List

Parameter	Default Value	Description
dharma.default.repository	location=default/repo	The default Studio local repository.
user.language	en	The locale of Studio. These value do not affect the SCM repository. If they are not set or are invalid, Studio uses the system locales.
user.region	US	
dharma.repo.max_history_length	50	The number of item to be saved in the history for the offline repository.
dharma.studio.ui.lookandfeel	windows	The look and feel of the Studio application.
dharma.studio.ui.input.constant.max.chars	8192	A positive integer representing the maximum number of characters of the constant value. Must be less than 8193.
dharma.studio.ui.tooltip. showwhitespacecharacters. max.chars.per.line	5000	Showing special characters in Context Inspector How many characters per tooltip line
dharma.studio.ui.tooltip. showwhitespacecharacters.space	·	One of the following values: ·•
dharma.studio.ui.tooltip. showwhitespacecharacters.tab	↹	One of the following values: ↹,⇥,→,¬
dharma.studio.ui.tooltip. showwhitespacecharacters.enter	↵	One of the following values: &#8629,↲,¶
dharma.studio.ui.tooltip. showwhitespacecharacters. otherwhitespace	•	One of the following values:•,·,☯
dharma.scripting.template.Rhino dharma.scripting.template.Nashorn	This template shows how to access operation return data, inputs and context data.	

Parameter	Default Value	Description
	Examples:	

To access an input, simply refer to the input name from the Inputs pane. For example: **myData = inputName;**

To access the context instance of the current branch:
myBranchData = scriptletBranchContext.get ("myBranchVarName");

To get value associated with the context key myContextKey:
myContextData = scriptletContext.get("myContextKey");

To get a value associated with the context key myContextKey from the local context:
myContextData = scriptletContext.getLocal("myContextKey");

To access the return code from a command line or script (note that this is a string and must be converted into an integer for numeric processing):
code = parseInt(scriptletRawResult['Code']);

To access the output from the operation (for example, stdout):
data = scriptletRawResult['Output String'];

To access the error string from the operation (for example, stderr):
error = scriptletRawResult['Error String'];

Other operations may have different result variables available. To see the list for a particular operation open the **Output Field** dropdown on the Outputs tab.

To set the response of the operation (must match one of the responses from the Responses tab):
scriptletResponse = "success";

To set the result of the operation:
scriptletResult = "Your Result Here";

Use the context to store data used in other steps. It is also helpful to place information into the context to examine variables when developing and debugging scriptlets.For example to place the OutputString information into a context key called 'Output':
scriptletContext.putGlobal("Output", data);

Note that the key and value are both strings. For example, to place 'code' into the context:
scriptletContext.putGlobal("Code", code.toString());

You can also place data into the local context, which means it will only be available to the current flow, but not to its parent flow or other subflows. To do this, use:
scriptletContext.put("LocalVariable","LocalValue");

Note: If a flow is larger than 64K, when you try to save the flow, you will see an exception.Also, the exception will occur every time you try to access that flow, even if you restart Studio.

You should avoid making scriptlets too long. When the scriptlets

Parameter	Default Value	Description
		are compiled, if they are larger than 64K, you get an exception when trying to save the flow.The exception will occur anytime you try to access that flow, even if you restart Studio.
dharma.studio.ui.inputinspector. assignfrom.selected	*see description*	This property allows setting the initial option in the "Assign from Variable" combobox, in the Input Inspector. Possible values: assigned - default when upgrading from an earlier Studio version not-assigned - default when installing Studio 10.20
dharma.studio.ui.inputinspector. assignto.selected	*see description*	This property allows you to set the initial option to be shown in the "Assign to Variable" combobox, in the Input Inspector. Possible values: assigned - default when upgrading from an earlier Studio version not-assigned - default when installing Studio 10.20
default.behavior.define.input	*see description*	This property allows you to set the default behavior that will be used when defining inputs in a flow. Possible values: prompt_user- default when upgrading from an earlier Studio version use_constant - default when installing Studio 10.20
dharma.studio.ui.activegrid	false	When set to true, the grid is active by default in the Flow Editor.
engine.assigner.trigger.repeatInt erval	10	
community.home.page.link	https://hpln.hpe.com/group/operations-orchestration	Community Home Page
git.history.page.size	100	Specifies the number of

Parameter	Default Value	Description
		items to be shown in a single Git history page
git.history.changes.length	500	Specifies the default number of changes to show in the **Changes under the selected item** and **Other changes** sections in the Git history.
hpln.enabled	true	Specifies whether or not ITOM Marketplace search is enabled
hpln.connection.timeout	180	Specifies the ITOM Marketplace connection timeout in seconds. Use 0 for infinite timeout.
hpln.read.timeout	300	Specifies the ITOM Marketplace read timeout in seconds. Use 0 for infinite timeout.
hpln.connection.url	https://api.hpln.hpe.com/hpln	Specifies the ITOM Marketplace connection URL
hpln.authentication.service.uri	rest/authenticate	Specifies the ITOM Marketplace authentication service uri
hpln.content.packages.service.uri	rest/contentpackages	Specifies the ITOM Marketplace content packages service uri
hpln.content.offerings.service.uri	rest/contentofferings	Specifies the ITOM Marketplace content offerings service uri
hpln.search.service.uri	rest/contentpackages/search	Specifies the ITOM Marketplace search service uri
hpln.search.service.search.param	search	Specifies the ITOM Marketplace search service search parameter
hpln.search.service.size.param	pageSize	Specifies the ITOM Marketplace search service size parameter
hpln.search.service.provider.param	provider	Specifies the ITOM Marketplace search service provider parameter

Parameter	Default Value	Description
hpln.search.service.product.name.param	primaryproduct	Specifies the ITOM Marketplace search service product name parameter
hpln.search.service.product.version.value	10.20	Specifies the ITOM Marketplace search service product version value. This version number is identical to the Studio version. For example, if the Studio version is 10.20, hpln.search.service.product.version.param is 10.20.
hpln.search.results.maximum.count	100	Specifies the maximum number of ITOM Marketplace search results. This value must not be greater than 100.
local.debugger.prompt.for.run.user	true	When set to true, the author is prompted for a run user on the local debugger
online.content.packs.link	https://marketplace.saas.hpe.com/itom/category/all?product=Operations%20Orchestration	Online Content

Note: If a property is missing from the **studio.properties** file, the default value is to prompt the user to enter a value.

Troubleshooting

Troubleshooting Studio for Those Upgrading from OO 9.x

Where's the Studio User Interface Item?

If you are used to working with OO 9.x and cannot locate a user interface item in Studio, use these troubleshooting hints to help you find what you're looking for.
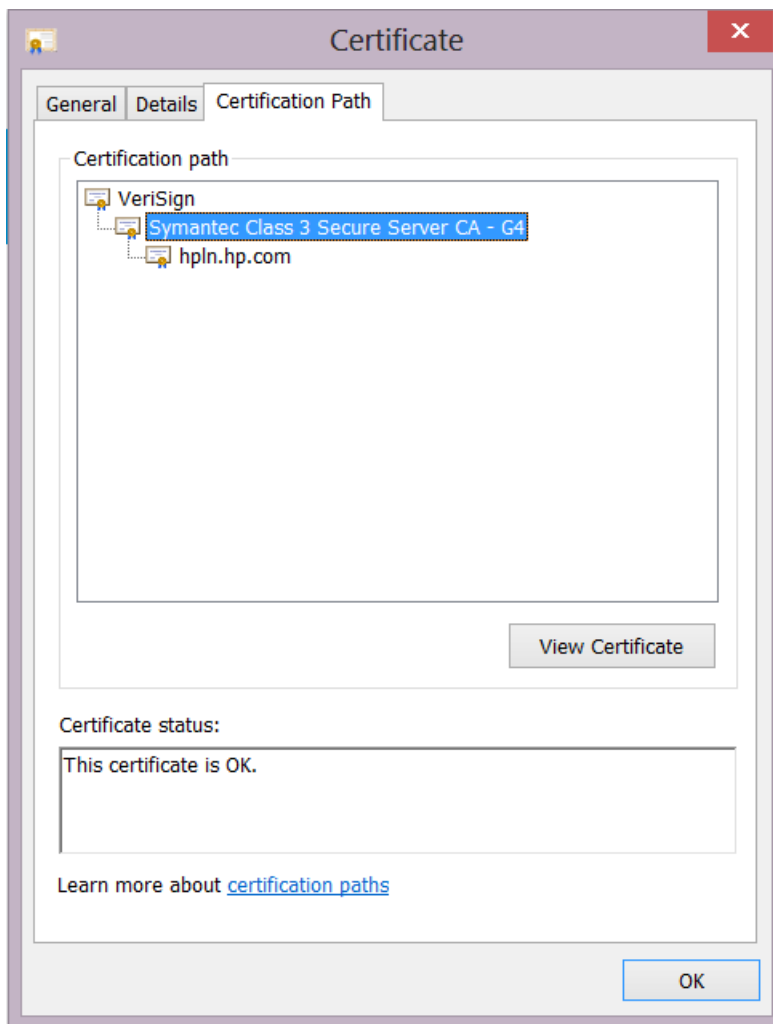
Where's the repository?

OO no longer uses repositories. Files are stored locally on your file system and it is recommended to use a source control application for collaboration.

Where are the Check In Check Out buttons and the My Changes/Checkouts pane?

You can commit and check out from the SCM Changes pane if you are connected to source control. See Working with Source Control.

Why do the Projects and Dependencies panes seem to contain the same items?

The **Projects** pane and the **Dependencies** pane are different:

- The **Projects** pane contains the *editable* flows, operations, and other OO objects that you can use in the project.
- The **Dependencies** pane contains *read-only* flows, operations, and other OO objects. You can use these objects in your project but you cannot edit them. If you want to edit any of these objects, copy them into the **Projects** pane.

Why can't I create operations?

It still exists but now it creates an operation. What we removed is the option to create built-in operations such as HTTP, SSH, Command line etc. Those have to be copied from an existing template operation.

You cannot create built-in operations, such as HTTP, SSH, or Command line. You will need to copy them from an existing template operation, and create a new operation from an action within the plugin. For more information, see "Creating Operations" on page 632.

Why can't I create sleep scriptlets?

Sleep scriptlets have been deprecated. In OO 10.x, scriptlets must be written in Rhino or Nashorn.

Where is the Categories domain term?

There is now a **Configuration\Categories** folder, where you can store categories for classifying flows. This replaces the **Categories** domain term.

Comparison of versions OO 9.x and 10.x

Task	How it was done in OO 9.x	How it is done in OO10.x
Creating operations	Use the **New** > **Operation** menu option, and select the type of	Import actions plugins, or create operations from imported action plugins.

Task	How it was done in OO 9.x	How it is done in OO10.x
	operation.	See "Creating Operations" on page 632.
Checking flows into a shared repository	Use the **Check In** button within Studio.	Save projects locally and commit them to a shared repository, using a source control management tool. See "Working with Source Control in Operations Orchestration Studio" on page 320.
Deploy and run a flow	Open the flow in the Central application, and run the flow.	Release the flow as a content pack and deploy it to the OO server, via API. See the *OO Installation Guide* and the *OOApplication Program Interface (API) Guide*.
Create multi-instance steps	Right-click a step and select the **Toggle Multi-instance** option to turn it into a multi-instance step. Then, create multiple loops for the different targets of the step.	Drag the **Multi-instance** icon on the **Step** palette to the authoring canvas. Add one or more subflows or operations to the multi-instance lane, and set multiple targets for the step via an input list of values. See "Creating a Flow with Multi-Instance Steps" on page 564.
Create actions for operations	Create IAction implementation classes, compile them into a **.dll** or **.jar** file, copy the **.dll** or **.jar** file into your Web service, and import the Web service into Studio.	Create and pack an action plugin, import it into Studio, and create a new operation from it. See "Creating Operations" on page 632.
Create categories to use for classifying flows	In the domain term called **Categories**, in the **Configuration\Domain Terms** folder, add a new row for the new category.	Create a new category in the **Configuration\Categories** folder.

ITOM Marketplace Troubleshooting

What can I do if I get an error when performing an ITOM Marketplace search?

OO no longer uses repositories. Files are stored locally on your file system and it is recommended to use a source control application for collaboration.

You may receive the following error when performing an ITOM Marketplace search:

```
sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to
find valid certification path to requested target
```

To solve this issue:
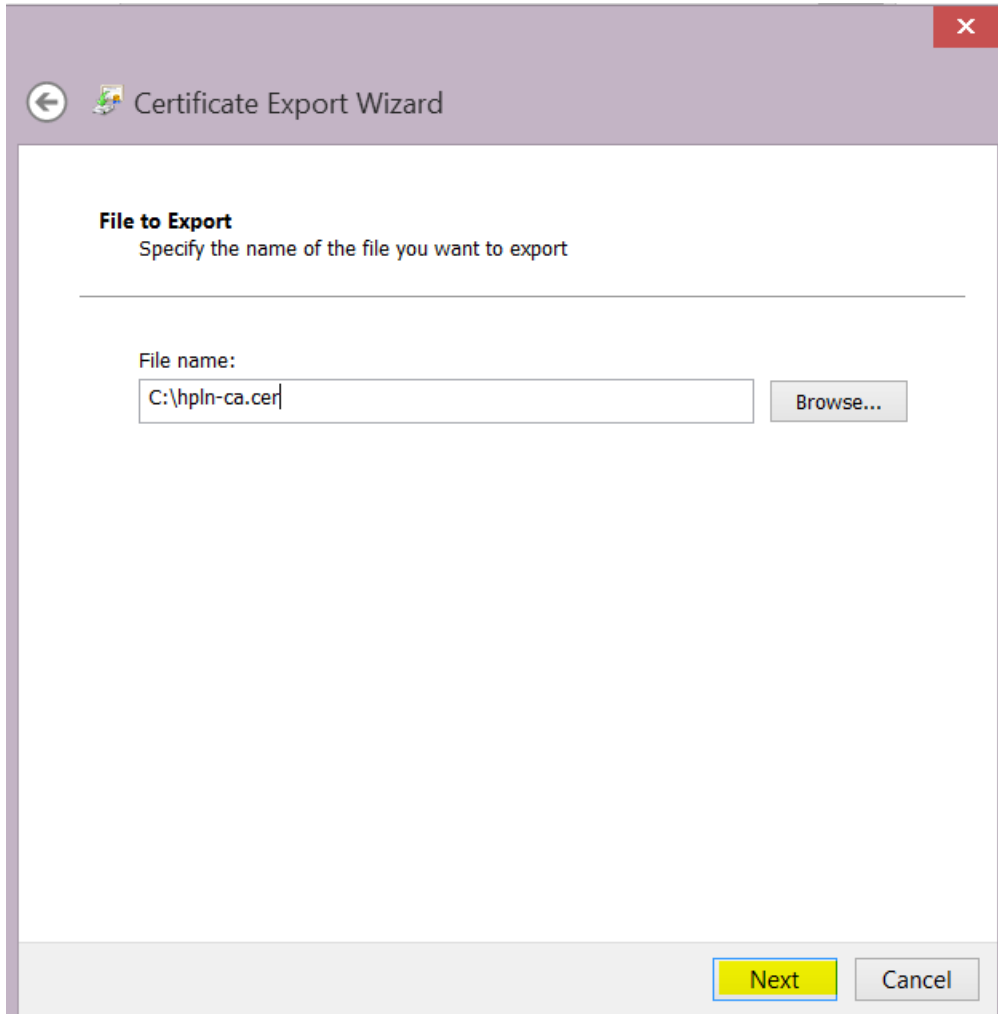
1. Close Studio.

2. Export the HPE Live Network root certificate, (the certificate with the name **Symantec Class 3 Secure Server CA - G4**—the second one in the chain) into a file named **hpln-ca.cer** in DER encoded binary X.509 format.

You can use a browser to navigate to https://api.hpln.HPE.COM/hpln and then export the root certificate as shown below:
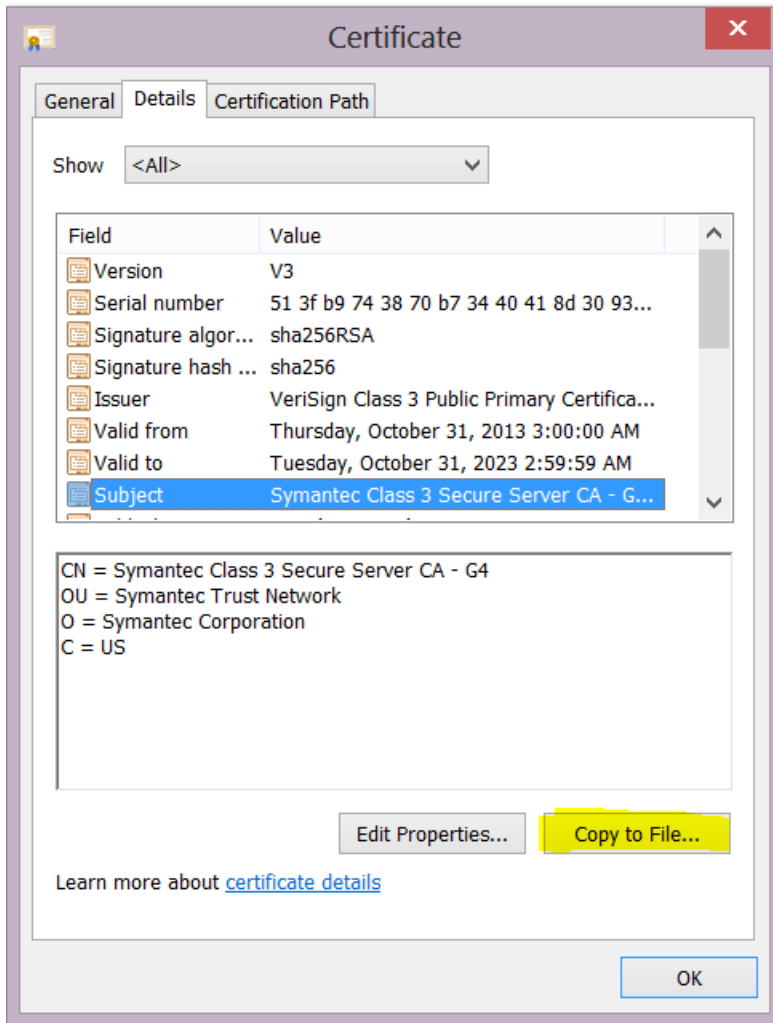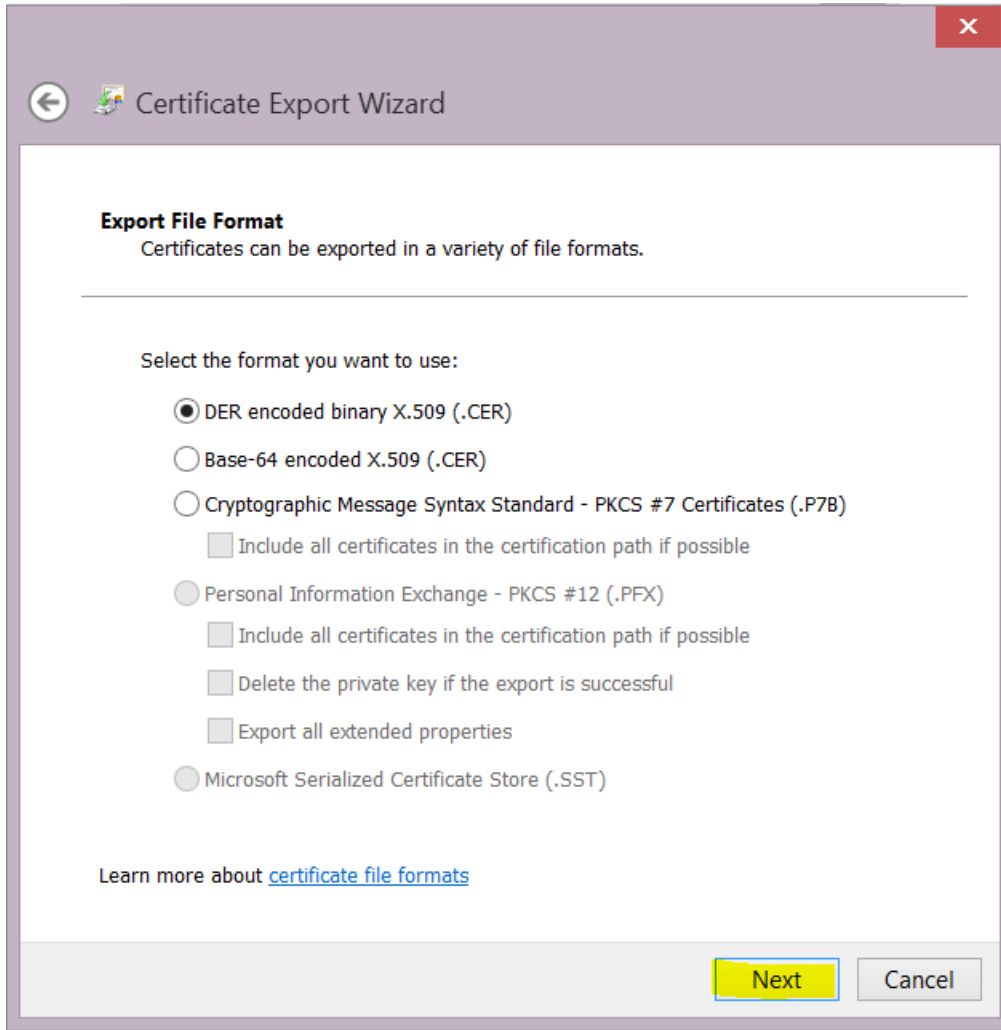
Use

Use

Use
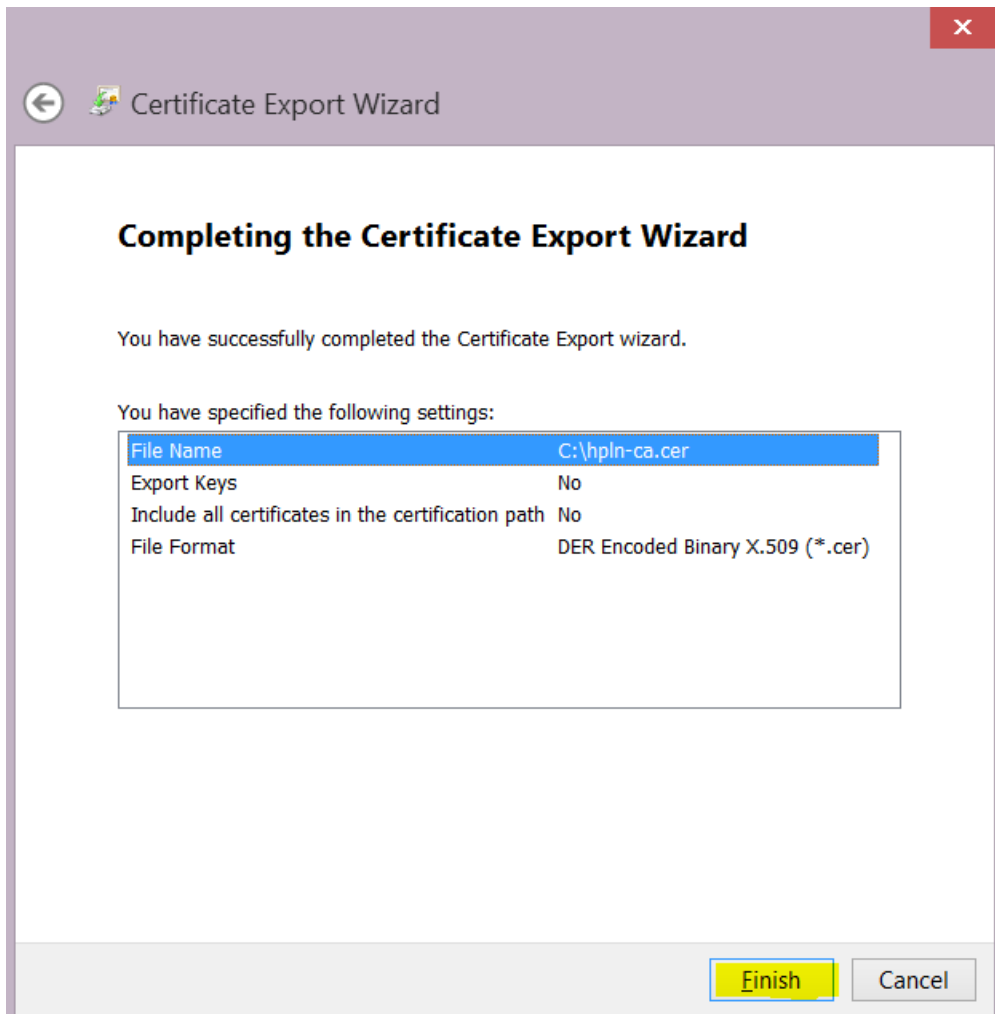
3. Copy the certificate file **hpln-ca.cer** to the **<oo_install_folder>\studio\var\security** folder.

4. Import the certificate into Studio's TrustStore using the following command:

```
<oo_install_folder>\java\bin\keytool -import -alias hplnroot -keystore
client.truststore -file hpln-ca.cer
```

> **Note:**
>
> ○ The default password for the keystore is **changeit**. If you changed the default password,
>   when prompted, use the value of the parameter **Djavax.net.ssl.trustStorePassword** from
>   the **<oo_install_folder>\studio\Studio.l4j.ini** file.
>
> ○ To change the Studio Truststore password, add the property **client.truststore.password**
>   with value the password in obfuscated format to the Studio.properties file from the ".oo"
>   folder.

```
client.truststore.password=={OBFUSCATED}6L9+NqBjKYp5heuvMEzg0g==
```

> If this property is not defined, Studio will fall back to the system property **javax.net.ssl.trustStorePassword** for the truststore password.

5. Restart Studio.

GIT Troubleshooting

What can I do if I get a "System cannot find specified file" error when performing Git operations?

There may be cases in which you may see the following error when performing Git SCM operations:

```
"Cannot run program "C:\<oo_installation_folder>\ studio\Git\bin\git" (in directory
"C:\<user_home_folder> \.oo\Workspace"):
CreateProcess error=2, The system cannot find the file specified "
```

1. Download the Git client from the following URL: https://github.com/git-for-windows/git/releases/download/v2.9.2.windows.1/Git-2.9.2-64-bit.exe.

2. Save the Git client to **<oo_installation_folder>/studio/Git**, so that the **bin** folder is directly under **<oo_installation_folder>/studio/Git**. In the Git installation wizard, use the default options.

Alternatively, if you already have a Git client installation on your local disk, point Studio to use that Git installation by performing the following steps:

1. Close Studio.

2. Go to the user home folder **C:\Users\<user>\.oo** (the Studio workspace location) and locate the **Studio.properties** file.

3. Modify the **Studio.properties** file by adding the following property at the end of the file:

```
studio.git.installation.location=<git_installation_folder>
```

For example:

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

The **bin** folder should be directly under **C:/Program Files/Git**. Note that **/** should be used as a path separator.

4. Save the **Studio.properties** file and start Studio.

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

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

If you have another version of the Git client installed, note that you must use version 2.9.2 of Git with Studio. This is the version that was validated with Studio. While other versions might still work correctly, they are not officially supported.

SVN Troubleshooting

What can I do if I updated my projects and deleted someone else's changes?

- In the following scenario a conflict can occur during update:

  a. User1 edits Flow1, and commits.

  b. User2 edits Flow1, and then updates.

  Since User2 will receive updates for a file that has local changes, a conflict will occur.

  Studio resolves such conflicts with the updating user's copy. This means that in our case, any changes User1 made to Flow1 will be deleted.

  To avoid such cases we recommend always updating and locking any item before editing it.

- **I made changes directly in the file system and something went wrong**

  While it is possible to create directories and rename files directly in the file system it is not recommended. It is preferable to perform all tasks from within Studio.

  If changes were made in the file system and those changes results in problems, it is recommend to revert those changes using an outside SVN tool such as SlickSVN or TortoiseSVN to perform a clean-up of the workspace.

Viewing Studio Errors in the Log Viewer

The **Log Viewer** is a useful tool that allows you to see all the errors that have occurred in the current user session in one central location within Studio. The errors are presented in the same way as they are displayed in the **Studio.log** log file.

The Log Viewer shows two types of errors:

Use

- Fatal errors (labeled FATAL in Studio.log)

- Errors (labeled ERROR in Studio.log)

> **Note:** Warnings (WARN) and Information (INFO) messages are not shown in the Log Viewer but can be seen in the **Studio.log** file.

If the **Studio Log Viewer** option in the Window menu is selected, the Log Viewer tab appears as one of the tabs at the bottom of the Studio window when Studio opens.

What do you want to do?

Enable the Log Viewer tab

1. From the Window menu, select **Studio Log Viewer**.

Using the OO Wizards

OO Studio comes with five wizards:

- The REST wizard enables administrators to create OO flows based on URL to Resource Listing or local API Definition files.

- The Third Party Content Pack wizard enables authors to generate a new third-party content pack with any required third party jars, which are not delivered with the out-of-the-box content packs.

- The Web Services wizard creates OO flows based on the API described in the Web Service Definition Language (WSDL) of the web service that you identify.

- The Shell wizard enables administrators to create OO flows that are integrated with Shell.

- The PowerShell wizard enables authors to generate OO flows from the selected PowerShell cmdlets found in a list of modules/snapins.

Using the REST Wizard

Purpose of the REST Wizard

This integration enables administrators to create OO flows based on URL to Resource Listing or local API Definition files.

The REST Wizard supports the Swagger, RAML and WADL formats for RESTful web service.

Swagger is a specification and a complete framework implementation for describing, producing, consuming, and visualizing RESTful web services. Swagger is JSON-based. The main goal of Swagger is to enable client and documentation systems to update at the same pace as the server. The documentation of methods, parameters, and models are tightly integrated into the server code, allowing APIs to always stay in sync.

The Web Application Description Language (WADL) is a machine-readable XML description of HTTP-based web applications (typically REST web services). WADL models the resources provided by a service and the relationships between them. WADL is intended to simplify the reuse of web services that are based on the existing HTTP architecture of the web. It is platform- and language-independent and aims to promote reuse of applications beyond the basic use in a web browser. It was designed to provide an alternative to the WSDL.

The RESTful API Modeling Language (RAML) is a YAML-based language for describing RESTful APIs. It encourages reuse, enables discovery and pattern-sharing, and aims for merit-based emergence of best practices.

For more information about creating OO flows, see the *OO Studio Authoring Guide*.

This document explains how this integration has been implemented, and how the integration's operations and flows communicate between OO and RESTful web services.

Audience

This guide is intended for system administrators who establish and maintain the implementation of the integration between RESTful web services and OO. This guide assumes that you have administrative access to both systems.

Supported Versions

Content Pack	Operations Orchestration Version	REST Wizard Version
Base Content Pack	10.x	10.20 and later

Getting Started with the REST Wizard

Installing the REST Wizard

The wizard is installed if Studio is selected from the OO installer. The REST Wizard is located under **<OOInstallPath>\studio\tools**.

REST Wizard Logs

The logs are located in the **<OOInstallPath>\studio\tools\logs\rest-wizard.log** file.

Uninstalling the REST Wizard

The wizard is uninstalled when Studio is uninstalled.

REST Wizard System Requirements

The following are the minimum software requirements for systems running REST Wizard for OO:

- The environment must have Java SE Runtime Environment 8 (also known as JRE) installed (for running the wizards).

REST Wizard Steps

To start the REST wizard, run the **rest-wizard.bat** file. This section describes the steps that you have to perform.

Step 1. Welcome Page

When you start the REST Wizard, the **Welcome** page opens.

Select **Swagger**, **WADL**, or **RAML**.

Click **Next** to open the **Destination** page.

Step 2. Destination

In this step, you select the location of an existing Studio project or a folder where the new project will be created.
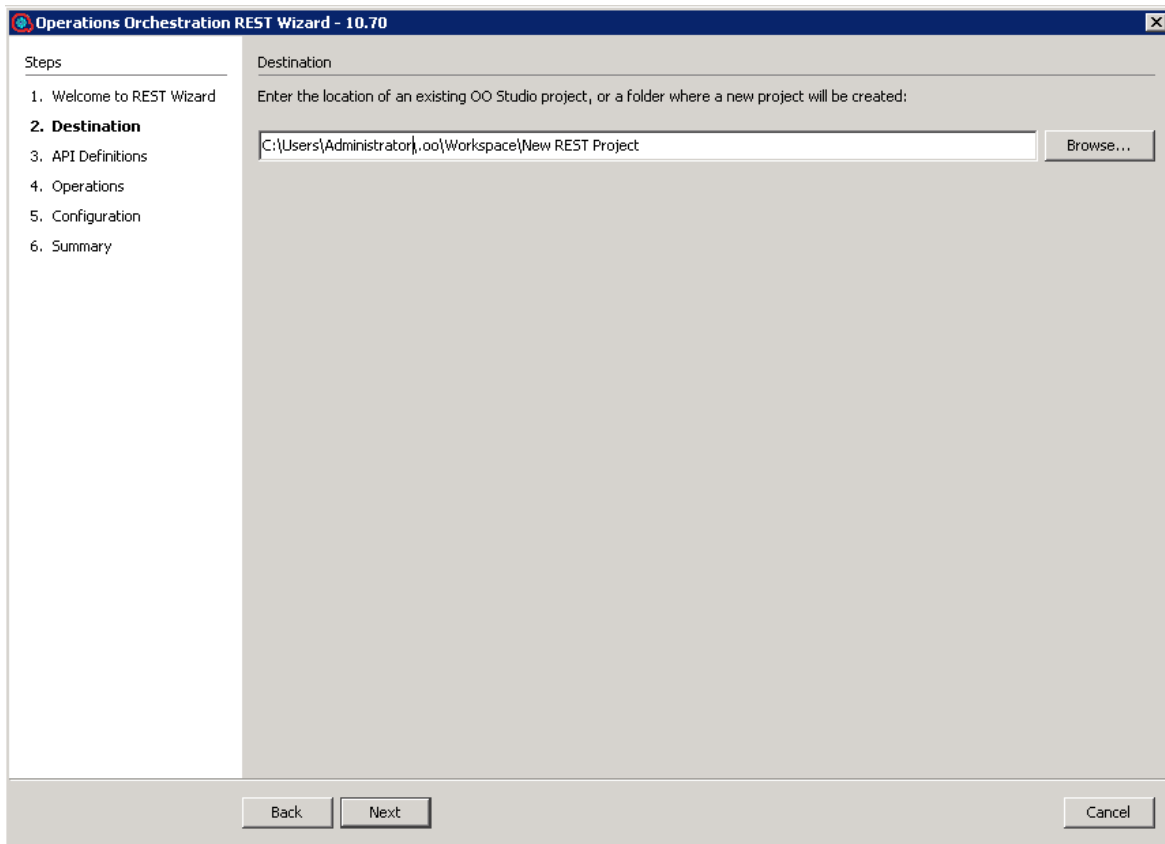
Enter or select a location of a Studio project for the flows you want to create and then click **Next**.

If this is not the first time you have run the wizard, the location is populated with the path to the last project created.

Step 3. API Definitions

The **API Definitions** page will vary, depending on whether you selected Swagger or WADL format.

Enter the base URL to the Swagger or WADL resource listing or click **Browse** to select one or more files that contain REST API Definitions in a Swagger or WADL format. When the base URL is entered, the **Next** button is enabled.

If the base URL is protected by authentication, specify the **Username**, **Password**, and **Authentication** type. The currently supported authentication types are **Basic**, **Digest**, and **Anonymous**.

If the HTTP request to the base URL is done through a proxy, specify the proxy settings. The currently supported proxy authentication types are **Basic**, **Digest**, and **Anonymous**.

(Swagger only): If the **Resolve relative references according to the RFC3986** check box is selected, the relative paths discovered in the Swagger file found at the provided URL will be automatically resolved, and the HTTP requests will be made at the correct resource listing paths. If this check box is not selected, it is possible that for relative paths discovered, the resource listing paths will be incorrect and an exception containing the message "Invalid URL to a Swagger file" will be thrown.

Step 4. Operations

On the **Operations** page, all the resources discovered in the previous page are displayed: the resource path, the HTTP method available for the resource and a brief description of the resource.
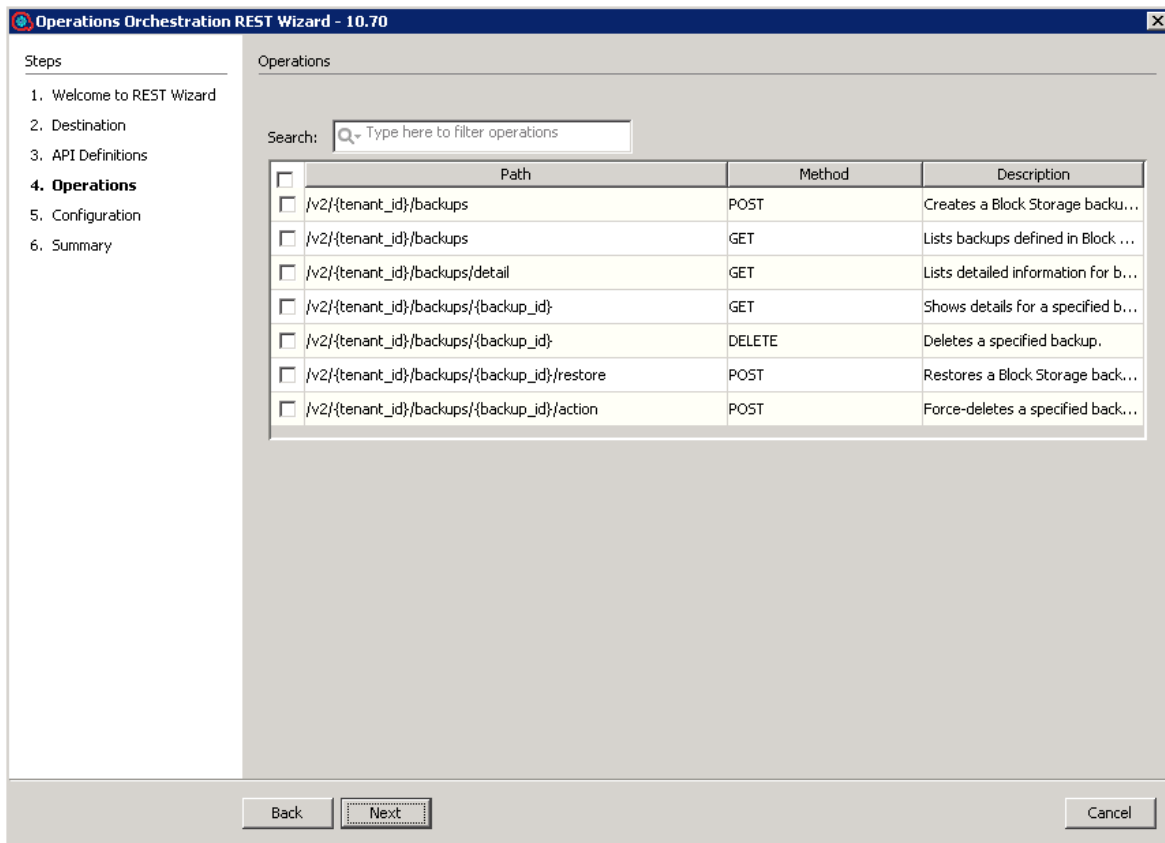
For the selected resources, the OO flows will be generated and will contain an operation that performs an HTTP request based on the HTTP method available for each resource.

Enter text in the top **Search** field, to filter the resources table by this text.

Select one or more resources.

> **Note:** To select multiple operations, hold down the CTRL key.

Click **Next**.

Step 5. Configuration

All the values provided in this step will be used as inputs for the generated flows and operations.

- In the **Authentication** section, if the **Username** and **Password** are provided, a new system account will be generated under the **New REST Project\Configuration\System Accounts** folder and the user name and password inputs of generated flows will reference this system account.

- For the **Proxy user name** and **Proxy password** pair, a new system account will be generated with the **Proxy** suffix, following the same rule as for the **Username** and **Password** from the **Authentication** section.

- The currently supported authentication types are: **Basic**, **Digest**, and **Anonymous**.

  - If the selected authentication type in both the **Authentication** section and the **Proxy** section is **Anonymous**, the **authType** input from the generated flows will be initialized with an empty constant.

  - If the selected authentication type in the **Authentication** section is **Basic** and the authentication type from the **Proxy** section is **Digest**, the **authType** input from the generated flows will be initialized with a **basic,digest** constant.

- In the **Headers** text box, you can add multiple headers in the generated flows, in the following format: headerName1=headerValue1\nheaderName2=headerValue2. Use the new-line character to separate the headers.

- If the **Trust all roots** or **Enable cookie tracking** check boxes are selected, the **trustAllRoots**, and **useCookies** input from the generated HTTP operation will be set to **true** and the **x509HostnameVerifier** input will be set to **allow_all**.

  Otherwise, these inputs will be set to **false** and **x509HostnameVerifier** will be set to **strict**.

- If the **Suppress user prompts for optional inputs** check box is selected, the non-required operation parameters discovered from the API Definitions document will be generated as flow inputs set to use constants with empty values. Otherwise, these flow inputs will be marked as prompts.

- Select the **Overwrite the flow if already exists** check box if the new generated flow should overwrite an existing flow from the same location specified in "Step 2. Destination" on page 694.

- Some Swagger or WADL files do not have the base URL set correctly. In this case, after the project is imported into Studio, check the **<ProjectName>/Configuration/Sytem Properties** folder, to see if the generated **System Properties SP_<ProjectName>_<ResourceName>_Url** value is initialized correctly.

Click **Next** for the **Summary** page.

Step 6. Summary

This step contains information about the successfully generated flows number from the selected number of resources.
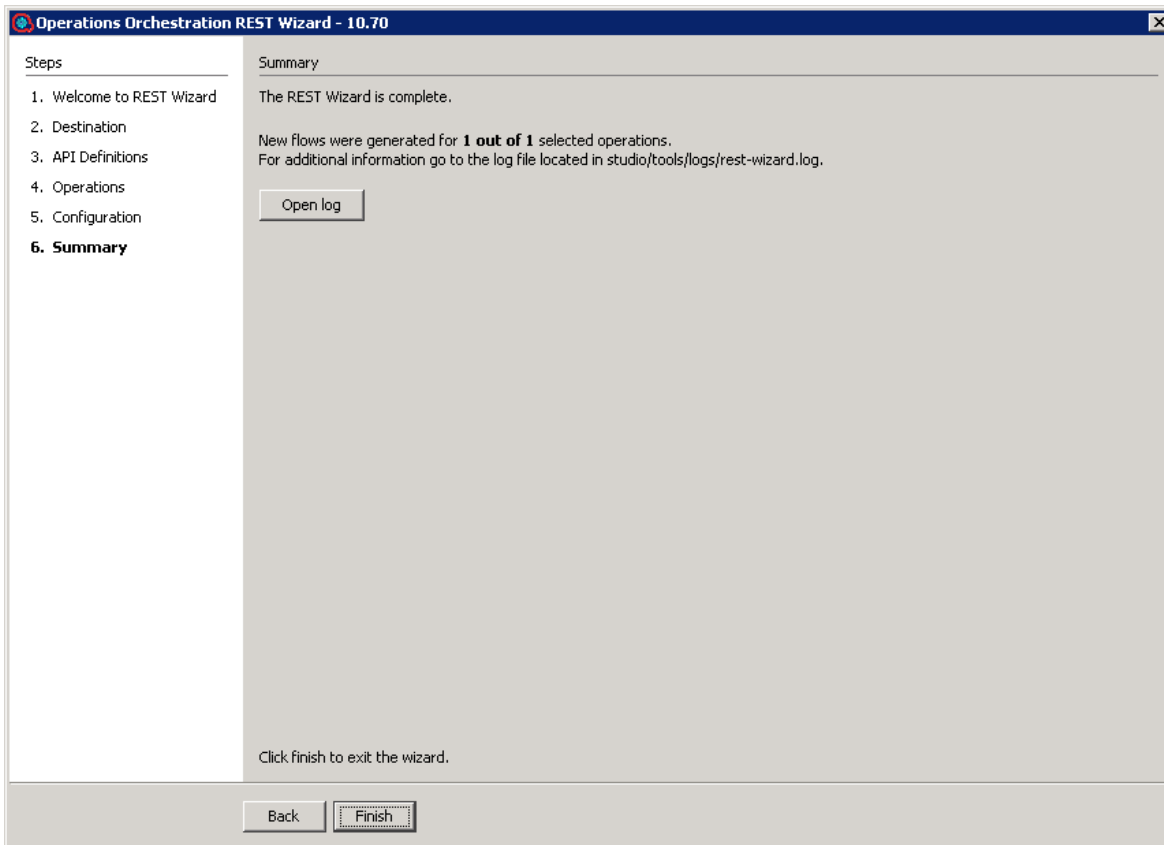
For more information about the flow generation, click **Open log** to view the log files.

Importing the Generated Flows into Studio

Before importing the generated flows into Studio, import the project generated by the wizard in Studio. See the "Managing Projects" section in the *Operations Orchestration Studio Authoring Guide* to see how to import a project.

> **Note:** In order to correctly run flows generated the REST Wizard, clients have to update the **Service_Url** from the System Properties with appropriate request address.

Generated Flows

The REST Wizard generates a flow with the name specified in the Destination step of the wizard. If a project with the same name already exists, the new flow is added to it.
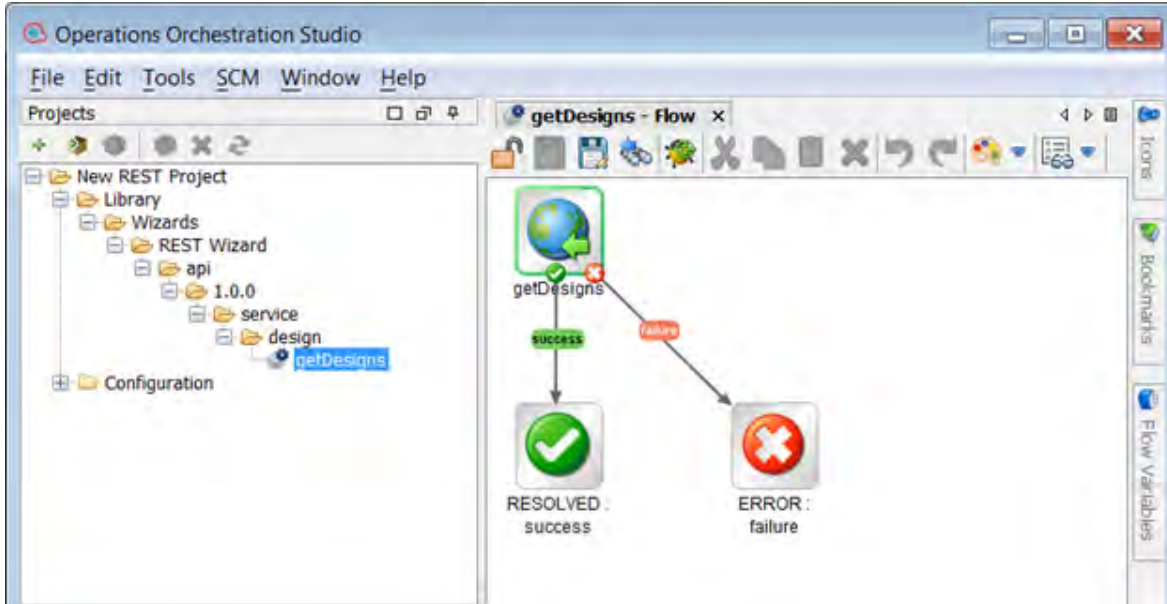
Inputs

Each flow has the following inputs, which are common to the REST operation:
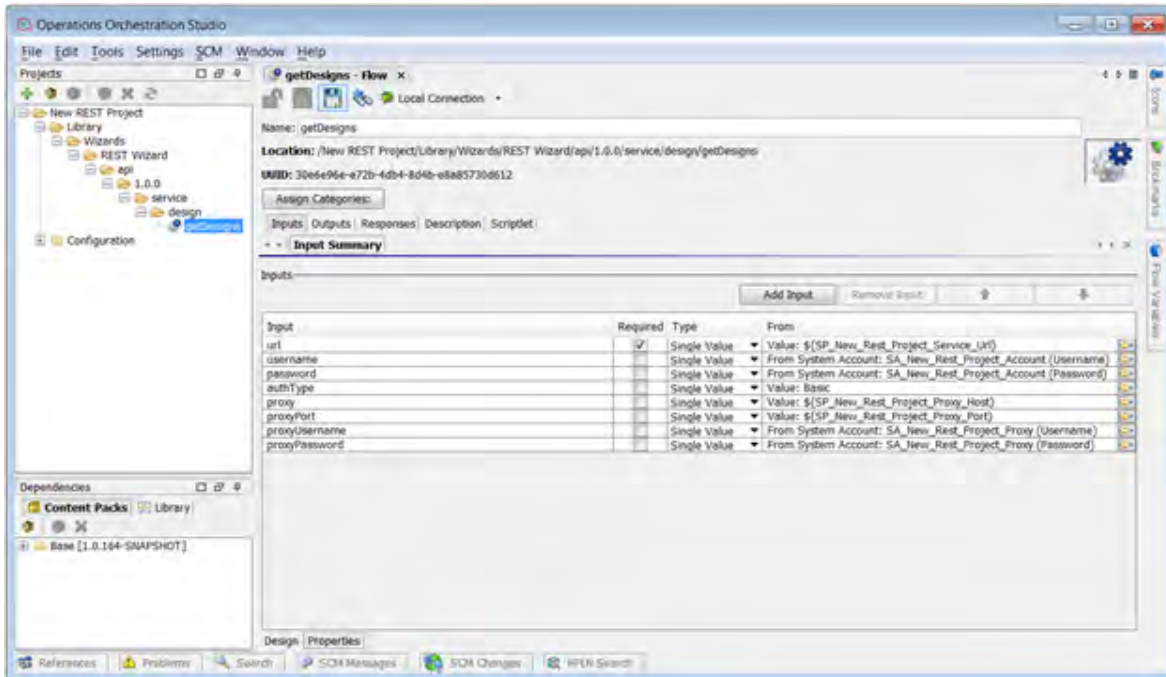
Outputs

On the **Outputs** tab, the flow's primary result is **document**. All Operations Orchestration flows have the **FailureMessage** and **TimedOut** outputs.
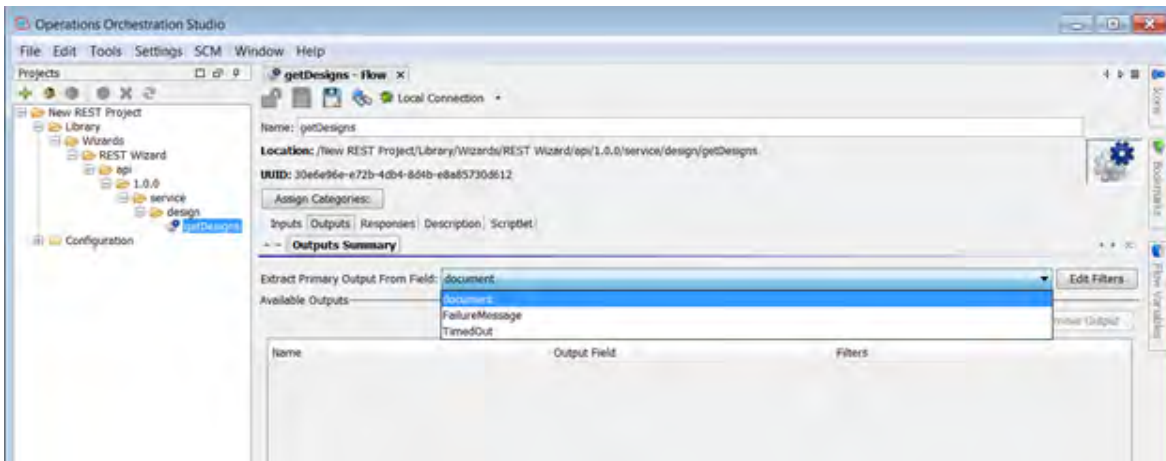
Responses

The success and failure responses of the flows are the same as the REST operations.
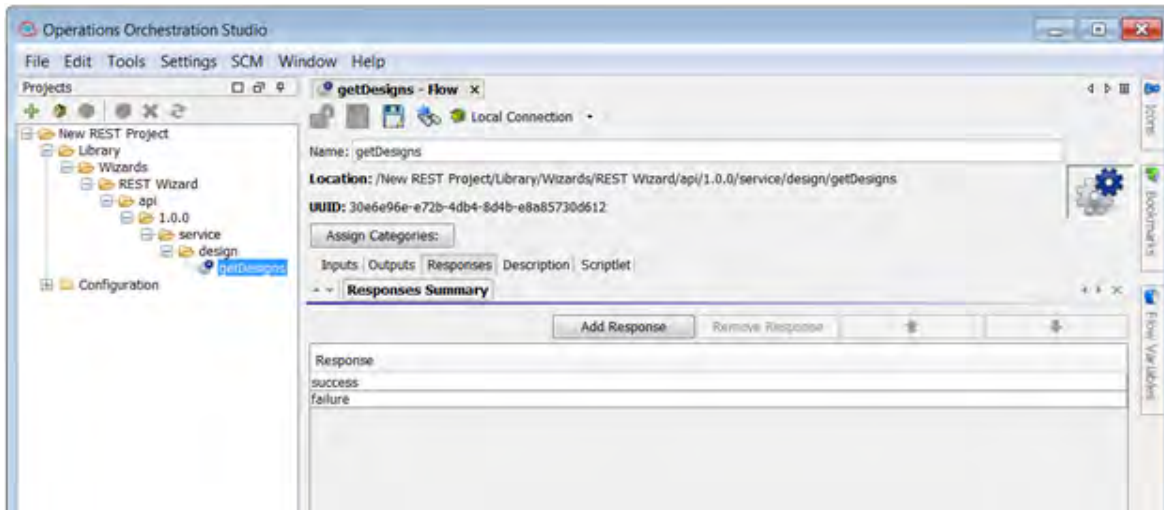
Descriptions

The description of each generated flow contains the following items:

- The description of the flow

- The description of the flow inputs

- The description of the results

- The description of the responses
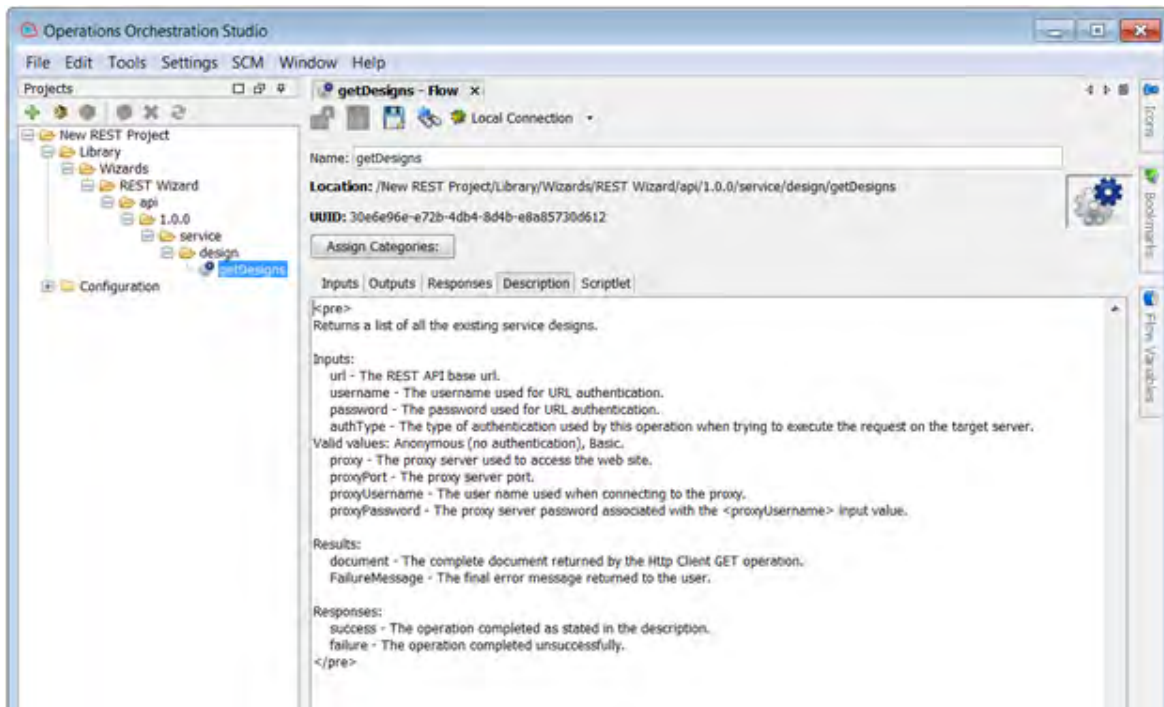
REST Operation

This operation is used to execute a HTTP request on a target host, either local or remote.

Operation Inputs

Operation Results

Troubleshooting

Troubleshooting Overview

This section provides troubleshooting procedures and tools that you can use to solve problems you may encounter while using this integration.

Unable to locate resource

Possible reasons are:

- Invalid URL to Swagger/WADL/RAML Resource Listing or API Definition local files.

- The connection to a Swagger/WADL/RAML Resource Listing URL must be done through a proxy.

- The URL to the resource is invalid based on the provided JSON/XML document.

Using the Third Party Content Pack Wizard

With this wizard, you can generate a new third-party content pack with any required third party jars, which are not delivered with the out-of-the-box content packs.

Purpose of the Third Party Content Pack Wizard

The main advantage of this wizard is automation.

This helps you to avoid repeating the same time-consuming process of creating the **third-party-cp.jar** content pack. You can perform the steps in the wizard as an alternative to using the steps described in **oo10-third-party-cp-release-notes-en.pdf**.

Supported Versions

Operations Orchestration Version	PowerShell Wizard Version
10.2x and later	10.20 and later

Getting Started with the Third Party Content Pack Wizard

Installing the Third Party Content Pack Wizard

The wizard is installed if Studio is selected from the Operations Orchestration installer.

Third Party Content Pack Wizard Logs

The logs are located in the **<OOInstallPath>\studio\tools\logs\third-party-cp-wizard.log** file.

Uninstalling the Third Party Content Pack Wizard

The wizard is uninstalled when Studio is uninstalled.

Third Party Content Pack Wizard Requirements

The following are the minimum software requirements for systems running Third Party Content Pack Wizard for OO:

The environment must have Java SE Runtime Environment 8 (also known as JRE) installed (for running the wizards).

Third Party Content Pack Wizard Steps

To start the wizard, run the **third-party-cp-wizard.bat** file.

Step 1. Welcome Page

When you start the Third Party Content Pack wizard, the **Welcome** page is displayed.

Click **Next** to continue.

Step 2. Libraries Selection Page

In this step, you will import an XML file containing the third party definitions that are needed to generate the third party content pack. This is a standard file that is required for a content pack that has missing third party files, which cannot be delivered with the content pack.

Each entry in the file should have the following format:

```
<dependencies>
…
        <dependency>
            <groupId>weblogic</groupId>
            <artifactId>weblogic</artifactId>
            <version>10.3</version>
        </dependency>
                …
    </dependencies>
```

Based on these entries, Studio and Central will validate a content pack with those missing dependencies.

Currently, there are three XML lists for **oo10-base-cp**, **oo10-middleware-cp**, and **oo10-sap-cp**.

1. Click the **Import XML** button to import one or more third party lists.

2. Click the **+** button to expand the list.

   > **Note:** Alternatively, you can use the **Shift** + **Space** shortcut.

3. After the list is expanded, select an item, click the **Edit** button, and select the path to the selected third party file.

   > **Note:** The **Edit** button is enabled only when a third party item is selected.

   > **Note:** You can also double-click any third party item and provide the path.

4. By default, when the list is loaded, all items are red. When you give an item the path to a third party file on the disk, the item changes color to green. In most cases, you will need to provide all required third party files to the list. However, there are cases where this is not needed, because you are using only some specific third party files from a content pack. In that case, you will just need to provide the required third party files.

   After you have provided a real path for all the third party files that you need, click the **Create CP** button to continue with the generation of the content pack.

   > **Note:** You need to select the root element of the corresponding list to enable the **Create CP** button.

5. In the Create Content Pack dialog box, enter the content pack name, location, and version (the same details as in the corresponding Studio dialog box). It is optional to enter information about the publisher and to add a description of the content pack.

   The result is a content pack that can be imported into Studio or Central. For example:
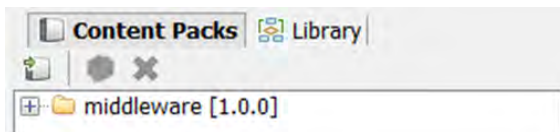
   > **Note:** In Central, you will not see a deployed content pack.

Third Party Wizard Command Line Tool

The wizard lets you use a command line tool to generate a content pack from the command line.

To start the wizard in command line mode:

1. Navigate to the wizard jar location at **<OOInstallPath>\studio\tools\lib** and start a command line here.

2. Run the following java command:

   `java -jar third-party-cp-wizard-jar-with-dependencies.jar -help.`

   After using this command, you will see a list of arguments that can be passed to the wizard in order to generate a content pack.

The most important constraint in command line generation is that all libraries need to respect the format `[artifactId]-[version].[type]` that is present in the XMLinput list. The default type format is jar.

If an entry in the XML file is in the format shown below, you need to provide a library in the `-third-parties-paths` argument using the `artifactId-version.type` format. In this case, you will need to have the **weblogic-10.3.jar** on disk. This helps the wizard to map between the XML list provided in the `-xmlPath` argument and the items provided in the `-third-parties-paths` argument. In the `-third-parties-paths` argument, you will provide the real paths of the jars needed by the wizard.

```
<dependency>
    <groupId>weblogic</groupId>
    <artifactId>weblogic</artifactId>
    <version>10.3</version>
</dependency>
```

Troubleshooting

If a content pack cannot be generated, check whether:

- The third party files on the disk are locked or in use

- The path where you want to save the content pack is locked

- The content pack jar file is already generated and locked

By default, if you try to generate a content pack with the same name as an existing one, the original content pack is overwritten by the wizard.

Using the Web Services Wizard

Overview of the Web Services Wizard

When you run the Web Services Wizard (wswizard.exe), you provide it with the WSDL for a web service. The Web Services Wizard creates OO flows based on the API described in the Web Service Definition Language (WSDL) of the web service that you identify in the wizard. The WSDL string you provide as a pointer can be a file's location and name or a URL.

The Web Services Wizard helps you create OO flows when:

- An OO integration does not exist.

- An OO integration does exist, but the customer has modified the application. For example, a customer using Remedy may have modified a form or added a field. To take advantage of the customer's modifications, the Remedy web service is updated. You can use the Web Services Wizard to create OO flows from the modified web service.

- If a new version of an application with an OO integration is released and the integration content does not support the new version, you can use the Web Services Wizard to create new OO flows.

Example

You have an application named MyAlert that creates a ticket through a web service and API, and you want to tell MyAlert to create a ticket. The Web Services Wizard extracts the application's APIs from the Web service's WSDL for the actions that can be performed with the application, such as creating or changing a ticket. The WSDL defines the web service's methods, the inputs for each method, and the required format for each input.

When you provide the wizard with the WSDL (in our example, for MyAlert) and run the wizard, it generates flows that can run against the web service. All flows created using the Web Services Wizard have a single step that is built from the Invoke Method 2 operation in the **Library/Operations/Wizards/Web Services Wizard** folder from the Base Content Pack. The flows are created in the project location folder specified by the user. Running the flows requires a Remote Action Service (RAS) that has access to the web service. For information on creating and configuring RAS references, see "Configuring Group Aliases" in the *OO Studio Authoring Guide*.

System Requirements

The minimum software requirements for systems running Web Services Wizard for OO:

- The environment must have Java SE Runtime Environment 8 (also known as JRE) installed.

Installing the Web Services Wizard

The Web Services wizard is automatically installed if Studio is selected in the Operations Orchestration installer.

Web Services Wizard Code Dependencies

The wizard does not have any dependencies. All third parties are encapsulated into the executable files.

Configuring Logging Settings

The configure logging settings are no longer supported in the 10.x wizard.

WS Wizard Enhancements from 9.x

- The wizard now appears in the taskbar and can be closed, minimized, or brought to the front.
- The UI is correctly divided. The scroll bar is now not necessary.
- Operation selection has a search functionality which allows you to quickly find an operation by typing letters of the operation name.
- The wizard includes functionality to override an existing flow (or flows).
- The wizard supports the use of credentials when retrieving the WSDL.
- The flows created by the wizard support HTTP headers and outputs.

Downloading OO Releases and Documents from ITOM Marketplace

ITOM Marketplace provides an OO Community page in which you can find and download supported releases of OOand associated documents.

To download OO releases and documents, go to HPE Marketplace.

This site requires that you register for an HPE Passport and sign-in.

To register for an HPE Passport ID, go to http://h20229.www2.HPE.COM/passport-registration.html

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

On the ITOM Marketplace page, click **Operations Orchestration Community**.

The Operations Orchestration Community page contains links to announcements, discussions, downloads, documentation, help, and support.

1. On the left-hand side, click **Operations Orchestration Content Packs**.

2. In the **Operations Orchestration Content Packs** box, click **Content**. The HPE Passport sign-in page appears.

3. Enter your HPE Password User ID and Password and click **Sign-in**.

4. Click **HPE Operations Orchestration 10.x**.

5. Search for **Base Content Pack**.

Wizard Processing Details

How the Web Services Wizard Uses SoapUI

SoapUI is an open-source web service testing tool. It provides web service inspection, invoking, development, and simulation. The Web Services Wizard uses SoapUI to parse the WSDL and create a template SOAP request.

This template is an XML file with placeholder tokens that are replaced with real data in order to make a request to the server. If you run SoapUI manually and create a project referencing a WSDL, you will see it create these request templates in the tree as nodes named **Request 1** for every operation in the WSDL. This is the template that the Web Services Wizard receives from SoapUI, and uses to populate the **xmlTemplate** input.

Similarly, the Web Services Wizard (in OO versions 9.00 and later) retrieves a SOAP response template with tokens that indicate how the response will look. This is a little more difficult to reproduce in the SoapUI GUI, as it requires creating a Mock Response and then using the Open Editor function to look at the XML.

For OO 9.x, support was added to specify a web proxy via the properties **http.proxyHost** and **http.proxyPort** in the **wsw.properties** file in the OO Home folder under **/Studio/tools/conf/**. You only need to enter the configuration information once (the first time you run the wizard against a WSDL outside the firewall). It is then read from the **wsw,properties** file and pre-populated in the wizard GUI. You can change it in the file or in the wizard GUI and the values are saved for the next time you run the wizard.

For OO10.00 and later, specifying a web proxy in the **wsw.properties** file is no longer supported. You can change it only from the wizard GUI.

After retrieving the templates for the request and the response, the WSDL is discarded. No further information is obtained from the WSDL, and all subsequent operations in both the Web Services Wizard and the Invoke Method 2 operation are based entirely on the templates returned from SoapUI.

Processing Templates

The template processing logic parses through a SOAP template (either a request template or a response template) looking for tokens. It is called in different ways for different purposes—for processing the request template and for processing the response template:

- Locating input tokens in the request template to create the input map (in the wizard)

- Locating output tokens in the response template (in the wizard)

- Replacing input tokens with actual values to build the SOAP request (in the **Invoke Method 2** operation)

In all cases, the logic skips any leading XML elements until it finds an element whose namespace prefix is either **soap** or **soapenv** and whose element name is not **envelope**, and then begins with the content of that element; this effectively ends up arriving at the topmost element under the outermost Body element.

Locating Inputs and Creating the inputMap

In the wizard, the request template is processed, and for each token that is found, a pipe-delimited value is returned indicating its path in the template, but with the outermost SOAP envelope information removed. For example, if the template looks like this:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
      <Test>
         <Name>?</Name>
         <Address>?</Address>
      </Test>
   </soapenv:Body>
</soapenv:Envelope>
```

it returns the values **Test|Name** and **Test|Address**. Note that the whole path is needed, as an element (such as **Name**) may appear in more than one place in a template, and there needs to be a unique path to each.

If, during this input processing, the wizard encounters a comment that indicates that it is at the beginning of an array ("x or more repetitions" or "m to n repetitions"), the value zero (0) is inserted at that point. For example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
      <Test>
         <!--1 or more repetitions-->
         <Name>?</Name>
         <!--1 or more repetitions-->
         <Address>?</Address>
      </Test>
   </soapenv:Body>
</soapenv:Envelope>
```

This returns the values **Test|0|Name** and **Test|0|Address**. As arrays may be nested, there may be templates whose values contain more than one zero (0).

The next task is to define a meaningful set of input names to be created. This is done using an input map. An input map permits a user-friendly name to be associated with each value. For example, **Address** is mapped to **Test|Address** and **Name** is mapped to **Test|Name**. The **inputMap** input that is generated in the operation is a list of these mappings between pipe-delimited paths and user-friendly names. In the first example above, the **inputMap** contains:

```
Test|Name=Name
Test|Address=Address
```

The creation of the **inputMap** is a little complex. Use the following tips when determining a name for each path name:

- Use the last part of the path (for example, **Name** or **Address**) if it is unique within the template.

- Avoid using a friendly name that is already one of the input names to the **Invoke Method 2** operation, such as **xmlTemplate**.

- If there are duplicate names, add a prefix for additional levels (with a period separator) onto the user-friendly name to make the name unique. For example, if the template yielded **One|Name** and **Two|Name**, the following input map would be created:

```
One|Name=One.Name
Two|Name=Two.Name
```

This is because both would otherwise map to the same value of **Name**.

- Single zeros in the pipe-delimited path (indicating the beginning of an array) are replaced with wildcards (*). The position of the wildcard in the user-friendly name is moved to the end of the next element. In the above example, for **Test|0|Name** and **Test|0|Address**, the following input map would be created:

```
Test|*|Name=^Name*$

Test|*|Address=^Address*$
```

> **Note:** The purpose of moving the wildcard position is to allow more intuitive input names like **Name0** and **Name1**.
>
> The value on the right side of the equal sign for array types is surrounded by the **^** and **$** symbols as a workaround for resolving the issue of parameters having similar names. These values are used as regex patterns for array types and similarly-named parameters without these symbols corrupting the algorithm.

- The simplification of friendly names (see the first bullet in this list) only applies to the part of an array to the left of the wildcard; all elements to the right will remain. For example, the items Test|0|Extra|Stuff|Name and Test|0|Extra|Stuff|Address results in:

```
Test|*|Extra|Stuff|Name=^Extra*.Stuff.Name$

Test|*|Extra|Stuff|Address=^Extra*.Stuff.Address$
```

This is regardless of the fact that the **Extra** and **Stuff** are otherwise unnecessary.

The wizard then uses the **inputMap** to create step-level and flow-level inputs for each item in the map. Any occurrences of wildcards are replaced with zeros in the input names. If the flow developer wants to provide additional elements (beyond just the 0th), s/he needs to add them both as step level inputs and flow level inputs. Using our previous example:

```
Test|*|Name=^Name*$

Test|*|Address=^Address*$
```

Name0 and **Address0** are created as inputs to the step and the flow.

The Web Services Wizard accepts JSON-formatted arrays for the array types found in the WSDL. So, instead of entering a new input for each element in the array, you can now enter a JSON-formatted array as the input value instead of creating additional inputs.

When you run the Web Services Wizard, you must check the Use JSON arrays for WSDL array type option on the Select operation(s) screen. This will add the input field "usesJSON" with a value of "true" to the created **Invoke Method 2** step. Then for the inputs, use a JSON format array for the

"0" element and the **Invoke Method 2** operation to create the required elements to send in the request.

For example, for an array structure defined by the following in the **xmlTemplate**:

```
<ns:AffectedCI type="Array">
   <!--Zero or more repetitions:-->
  <ns:AffectedCI type="String" mandatory="" readonly=""></ns:AffectedCI>
</ns:AffectedCI>
```

- The **inputMap** entry for this array must use the following wildcard format:

```
CreateChangeTaskOORequest|model|instance|middle|AffectedCI|*|AffectedCI=^Affe
ctedCI*$
```

- The associated Web Services Wizard created AffectedCI0 input field JSON array formatted value should be similar to:

```
["CIvalue1","CIvalue2","CIvalue3"]
```

Locating Outputs and Creating Operation Outputs

Locating outputs in the XML template uses the same logic as finding inputs, but instead of returning a pipe-delimited path, the process returns an XML XPath expression. This is nearly the same thing except with a slash as a delimiter rather than a pipe. There are, however a few differences:

- **/text()** is appended to the XPath in order to correctly extract the text of the simple elements. For example, the following template corresponds to the outputs **/Test/Name/text()** and **/Test/Address/text()**:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
      <Test>
         <Name>?</Name>
         <Address>?</Address>
      </Test>
   </soapenv:Body>
</soapenv:Envelope>
```

- Nothing is appended to the XPath of array elements. This causes the entire portion of the XML document to be returned in a single output, and it is the flow developer's responsibility to use other operations (like XML or JSON ones) to extract the relevant items. This difference is due to the fact that arrays can become arbitrarily nested, and returning such structured data in a simple variable is not an easy task. For example, the following template yields just the single output **/Test:**

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
      <Test>
          <!--1 or more repetitions-->
          <Name>?</Name>
          <!--1 or more repetitions-->
          <Address>?</Address>
      </Test>
    </soapenv:Body>
</soapenv:Envelope>
```

If JSON arrays are being used, an additional output named **jsonStripped** is populated with the SOAP response in a JSON-formatted string.

The wizard then creates step outputs for each output that was located in the template, assigning an XPath filter to each one (whose value was determined above). At this point, the wizard has completed its main lifting. The remainder of the process resumes when the flow is run, calling the **Invoke Method 2** operation.

Populating InvokeMethod2 Default Values for All Operations

The Web Services Wizard allows setting **InvokeMethod2** inputs so that each operation created from the WSDL has the inputs set by default. For example, the timeout input can be the same for all web service operations and setting the value once in the wizard will, in turn, set the timeout input value for all operation(s) selected on the selection page. Setting the default values in the Web Services Wizard is optional.

The Web Services Wizard does not validate the default inputs entered. This validation takes place during the flow run. The Web Services Wizard allows you to specify default values only for the authentication type selected. For example, if the HTTP authentication type is selected, the wizard allows you to enter the default inputs for HTTP authentication only and skips the WS-Security page when you click the **Next** button.

The Invoke Method 2 Operation

Overview of the Invoke Method 2 Operation

The **Invoke Method 2** operation is called when the flow is run. Its basic tasks are to:

- Build a SOAP request based on the **xmlTemplate**, the **inputMap**, and the inputs supplied to the operation (see the next section).

- Perform security functions as indicated by input values, such as signing the outbound request, encrypting it, and setting up SSL for HTTPS.

- Perform an XSLT transformation on the SOAP reply to populate the **documentStripped** and/or **jsonStripped** output, which strips the namespace prefixes from all of the output fields. For example, the reply:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
      <xyz:Test>
         <xmlns:Name>Real Name</xmlns:Name>
         <abc:Address>An address</abc:Address>
      </xyz:Test>
   </soapenv:Body>
</soapenv:Envelope>
```

would become:

```
<Envelope>
    <Body>
      <Test>
         <Name>Real Name</Name>
         <Address>An address</Address>
      </Test>
   </Body>
</Envelope>
```

or in a JSON formatted string:

```
{"Body":{"Test":{"Name":"Real Name1","Address":"An address1"}}}
```

This conversion is necessary as the operation outputs use XPath filters or JSON to extract values, and XPath expressions do not work well with XML that contains namespaces.

Building a SOAP Request

Building a SOAP request includes the following steps:

- Input resolution

  This step uses the **inputMap** together with the operation inputs, to determine the values to be substituted.

  For example, if the **inputMap** contains **Test|Name=Name** and there is an input named **Name** with the value **George Washington**, this step combines them to determine that the element in the request corresponding to **Test|Name=Name** should have the value **George Washington**. This

step also handles wildcards in array references. For example, an **inputMap** containing **Test|*|Name=Name*** and inputs **Name0** and **Name1** should have values corresponding to the elements in the SOAP request corresponding to **Test|0|Name** and **Test|1|Name**.

- Completing values

This step parses through the SOAP template looking for tokens. When it finds one, it attempts to find a value resolved from the previous step, and substitutes it if found. If no input is found with the specified name, the token is removed.

If the processing encounters the beginning of an array (indicated by the special comments in the template ("x or more repetitions" or "m to n repetitions"), the resolved inputs for that array are sorted numerically (so that 10 appears after 9 rather than between 1 and 2), and then substituted into the SOAP request.

> **Note:** Any missing gaps in the input names are ignored. For example, if the inputs are **Name0** and **Name2** (and **Name1** is missing), then only two values are substituted in the template (the values for **Name0** and **Name2**); no empty entries are created for missing values.

Complete Set of Inputs

Input	Description
contentType	Sets the HTTP Content-Type header to the given value. Defaults to **text/xml**.
ICONCLUDE_WSW_VERSION	Must be the constant **2**.
header_*	Any input that begins with **header_** is processed by the **HTTP Client Post Raw** operation, which then creates an HTTP header out of it. For example, if the input named **header_Accept-Encoding** contains the value **gzip**, the request will be altered to add the HTTP header **Accept-Encoding: gzip**.
headers	The list containing the headers to use for the request separated by a new line (CRLF).The header name - value pair will be separated by ":".
inputMap	Described in Locating Inputs and Creating the inputMap.
password	The password sent to the web service.
proxy	The name of the proxy host that is used to make the web service request across a firewall (optional).

Input	Description
proxyUsername	The proxy user name, if necessary, used when making web service requests across a firewall (optional).
proxyPassword	The proxy password used when making web service requests across a firewall (optional).
proxyPort	The port on the proxy host used to make the web service request across a firewall (optional).
returnXMLRequest	If this input is set to **true**, a new output named **rawXMLRequest** is returned by the operation, which contains the text of the SOAP request that was sent. This is useful for troubleshooting purposes.
timeout	The timeout in ms for the HTTP connection. Note that there may be other timeouts that affect the connection, such as the timeout between Central and the RAS.
trimComments	Removes all comments from the outbound SOAP request (hidden input).
trimNullOptionalTypes	By default (true), for every element in **xmlTemplate** that is marked as **Optional** and for which no token has been substituted with a value, the element is removed from the outbound SOAP request (hidden input).
trimNullComplexTypes	By default (true), for every element in xmlTemplate that has sub-elements (including arrays) and for which no token has been substituted with a value, the entire element (and all of its embedded elements) is removed from the outbound SOAP request (hidden input).
trustAllRoots	When set to true, when HTTPS connections are made, it ignores the signing authority of the certificate (permitting self-signed certificates) and ignores discrepancies between the host name on the certificate and the actual server name that is hosting the web service.
url	The URL of the web service, extracted from the WSDL. This generally has variable references to the host and port so that this value does not need to be changed to send a request to a host or a port different from the one hosting the WSDL.

Input	Description
useCookies	Determines whether the HTTP client will use cookies (store them during the connection and send them back for subsequent HTTP requests to the same server).
usesJSON	Use JSON arrays for all inputs of array type.
username	The user name sent to the web service.
xmlTemplate	Described in How the Web Services Wizard Uses SoapUI.
WSSecurityEncryptRequest	A Boolean value (default false) indicating whether or not to encrypt the SOAP request.
WSSecurityKeystore	When encrypting or digitally signing the SOAP request, this indicates the keystore containing the certificate.
WSSecurityKeystorePassword	When encrypting or digitally signing the SOAP request, this indicates the password to the keystore.
WSSecurityKeystoreType	When encrypting or digitally signing the SOAP request, this indicates the keystore type.
WSSecuritySignRequest	A Boolean value (default false) indicating whether or not to digitally sign the SOAP request with an X509 signature.
WSSecurityTimestampRequest	A Boolean value (default false) indicating whether or not to securely timestamp the SOAP request.
wswAuthenticationType	Can be assigned one of the following values: **http**, **ws-security text**, **ws-security digest**, and **none**. **http** is used for normal HTTP authentication, where the user and password are sent as HTTP headers. The two ws-security* options use SOAP WS-Security protocols.
*	All other headers are passed intact to the **HTTP Client Post Raw** operation, which can interpret them.

Using the Web Services Wizard to Create Web Service Flows

Using the Web Services Wizard to Create OO Flows from Selected WSDL Operations

The Web Services Wizard creates OO flows based on the operations available in the WSDL that you specify when you run the wizard. This tool is available by launching the wizard executable file. The Web Services Wizard is a is a simple and intuitive tool that leads the user through the tasks and simplifies the process of flow creation.

To use the Web Services Wizard to create an OO flow from a WSDL

1. Start the Web Services Wizard.

   The Welcome page opens.

2. Click **Next** to continue.

   The **Select Destination** page opens.
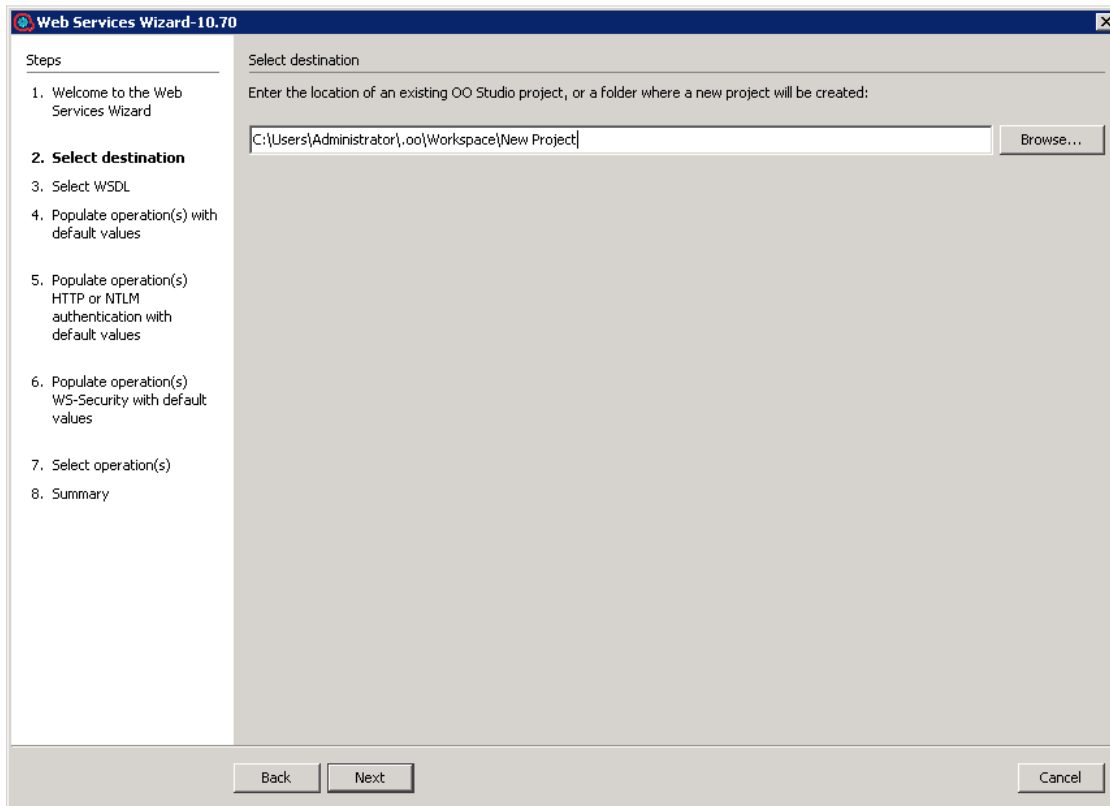
In the **Enter the location** field, enter the required project path or click **Browse** to locate the project location, and then click **Next**.

The wizard generates a 10.x studio project but not a content pack or a repository. The project has a default location: **C:\Users\[username]\.oo\Workspace\New Project**.

If this is not the first time you have run the wizard, the location is populated with the path to the last project created.

3. Enter the URL to the WSDL, or select a local WSDL from a file system.

If proxy information is required to access the WSDL URL, enter it here. If loading the WSDL succeeds, the **Populate operation(s) with default values** page opens. In this page, you can set default values for the flows that the Web Services Wizard generates.

4. (Optional) Enter values for any common inputs (these are the default inputs of Invoke Method 2 operation, so every flow created by the wizard will contain them). If the inputs are common for all flows created, they can be entered on this page. The default values are populated on the page.

> **Note:** If you set the values here, each operation will be assigned with the preset values . To change the value, you need to modify each flow in Studio or rerun the Web Services Wizard and select the **Overwrite the flow if already exists** check box.

5. Click **Next** to continue to either the **Populate operation(s) HTTP authentication with default values** page or the **Populate operation(s) WS-Security with default values** page or the **Select operation(s)** page, depending on the authentication type selected.

For example, if you select an authentication type of **ws-security text**, the next page will be the optional step of populating the WS-Security default input values.

6. (Optional) Enter values for the common HTTP authentication inputs. If the inputs are common for all flows created, they can be entered on this page.

> **Note:** If you set the values here, each operation will be assigned with the preset values . To change the value, you need to modify each flow in Studio or rerun the Web Services Wizard and select the **Overwrite the flow if already exists** check box.

7. Click **Next** to continue.

8. (Optional) Enter values for the common WS-Security inputs. If the inputs are common for all flows created, they can be entered on this page.

> **Note:** If you set the values here, each operation will be assigned with the preset values . To change the value, you need to modify each flow in Studio or rerun the Web Services Wizard and select the **Overwrite the flow if already exists** check box.

9. Click **Next** to continue.

10. Select the operation(s) for which you are interested in creating flows. The available operations are displayed in the list. If you want to use JSON-formatted arrays for all array type inputs in all the generated flows, check the **Use JSON arrays for WSDL array types** box. If you do not check the box, you can still use JSON formatted arrays, but you have to manually set the input usesJSON in Invoke Method 2 to **True** for all the flows that you want to accept JSON data.

Use
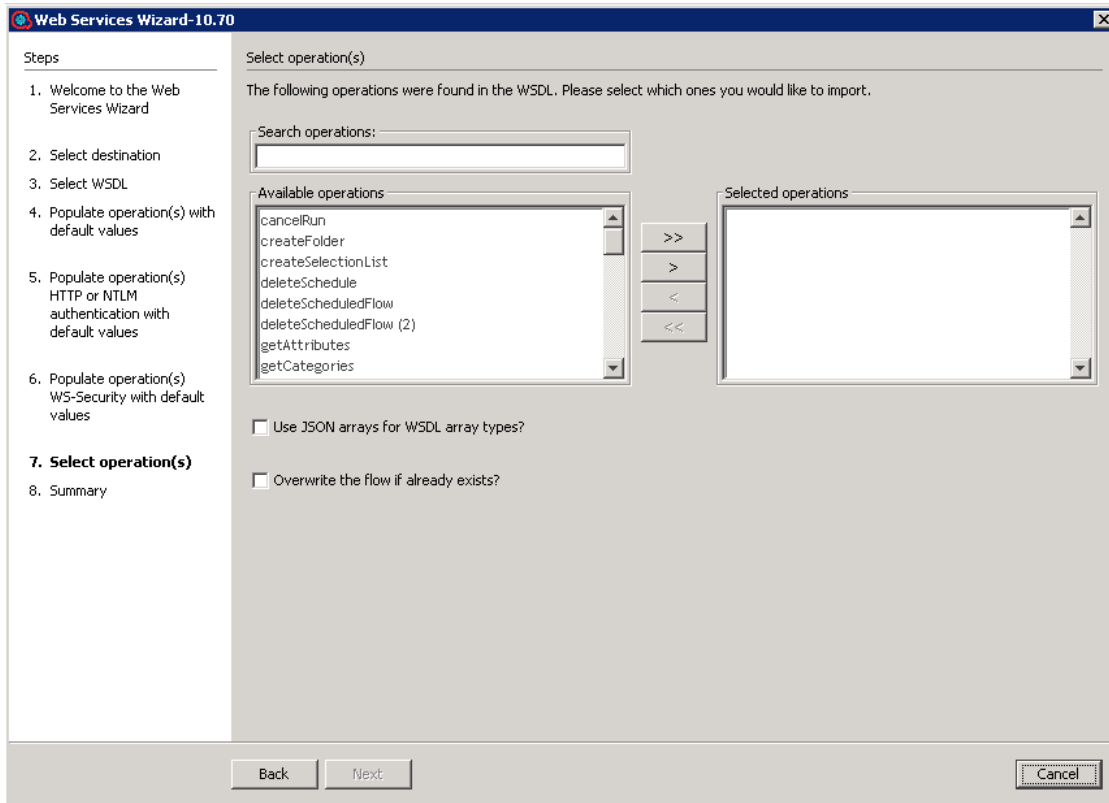
11. Click **Next** to continue.

    You can move the operations from one column to the other. Use the search text box if the list is long, and you cannot find the required cmdlet. The wizard searches the list for the operations with names containing the search text. In addition, the wizard updates the list as you type.

    After the flows are successfully created and saved in the repository, the Web Services Wizard finishes.

After Running the Web Services Wizard

If the Web Services Wizard ran successfully, you will have a new set of flows ready for use. However, may have to make some adjustments before the operations can be used, due to the following issues:

- The source WSDL may have problems or may have changed.

- There may be undocumented headers.

To diagnose and correct these situations, read this section, along with the Troubleshooting section.

> **Note:** In addition to inputs, the parsing obtains results that can be captured as operation outputs (which are expressed as results in steps). Any arrays in the XML are extracted as a single XML

result from which the flow author can extract narrow subsets.

- In the flows that the Web Services Wizard generates, the flow inputs that correspond to web services inputs are optional. Sometimes some of the inputs that the web service definitions indicate as required are not actually required, and mirroring these settings in the flow would force the flow user to enter unused values when running it. So, the Web Services Wizard sets all inputs as optional. When the web service does indicate that a field is optional, it precedes the field with the comment "<!-Optional:>" or "<!zero or more repetitions->". For information on which inputs should be required, see the documentation for the relevant web service.

- If the web service whose WSDL you are accessing resides on the other side of a firewall from your Studio machine, you must specify an HTTP proxy to be used to reach the web service.

Troubleshooting

General Troubleshooting Principles

- If you experience difficulties running the Web Services Wizard, first check any changes you make on one input before trying them on all inputs.

- If you experience difficulties running the Web Services Wizard against a WSDL with a URL that starts with HTTPS, try opening the WSDL in a browser and saving it to the local file system. Make sure to copy all dependencies (such as XSD files) because it is difficult to access them through the wizard. These files can be found under **<xs:schema><xs:import>** tags. Then, run the Web Services Wizard against the WSDL file instead.

- If an unexpected error message is returned after running the OO flows that the Web Services Wizard created, try adding and setting the **trimNullOptionalTypes** and/or **trimNullComplexTypes** to **false** in the **Invoke Method 2** operation of your flow. This results in the outbound SOAP request looking more like the request sent by SoapUI when inputs have null values.

- If the Web Services Wizard fails to import the Exchange WSDL due to a `java.lang.OutOfMemoryError:` Java heap space error, this could be because the attached XSD is invalid or because the WSDL is invalid.

Troubleshooting Steps

If the Web Services Wizard fails to load the operations for selection and returns with a null pointer exception:

- Try removing any white space around the comments section of the WSDL. This is a known issue with the SoapUI utility that the Web Services Wizard uses.

- The Web Services Wizard passes on the SoapUI's "Null-pointer Exception" message followed by the message "Failed to load WSDL" if you attempt to load an invalid WSDL. This is a known issue with the SoapUI utility that the Web Services Wizard uses.

- Validate that the XML request is as you expected. This can be done by setting the **returnXMLRequest** input value to **true** in the **Invoke Method 2** operation in your newly created flow. This will add an output result of the actual XML request that was sent.

- Try the request in SoapUI to verify that the Web service is working correctly. Install SoapUI (http://www.soapui.org/), create a project from the WSDL and a request object for the operation in question. Then, replace its content with the XML request from the output above.

- Some WSDLs have been written in a way that causes the Web Services Wizard to fail to recognize some array types. When one of these OO flows runs, it may return the following exception:

```
<faultcode><soapenv:Server.userException</faultcode><faultstring>org.xml.sax.
SAXException: Found character data inside an array element while
deserializing</faultstring><
```

The original WSDL file, which was correctly processed by the Web Services Wizard, used the **ArrayOf_xsd_String** implementation:

```
<wsdl:message name="createSelectionListRequest">

…

<wsdl:part name="values" type="impl:ArrayOf_xsd_String"/>

</wsdl:message>
```

The modified WSDL file, which is correctly processed by the Web Services Wizard, redefines the type **ArrayOf_xsd_String** to **WSListValues** (this is a specific case for the **createSelectionList** operation from the example). Using the **WSListValues** type definition, you can also define your own array of string types (for example, **ArrayOfStrings**) in place of **ArrayOf_xsd_String**.

```
<wsdl:types>

…

<complexType name="WSListValues">

<sequence>

<!--Zero or more repetitions:-->
```

```
<element maxOccurs="unbounded" minOccurs="0" name="value" type="xsd:string"
/>

</sequence>

</complexType>

…

</wsdl:types>

<wsdl:message name="createSelectionListRequest">

…

<wsdl:part name="values" type="tns1:WSListValues"/>

…

</wsdl:message>
```

- In some cases, the **Invoke Method 2** operation fails to run when used in a flow in Studio. This occurs because the SOAP envelope isincorrect. A single flow UUID is passed within **<flowUuids></flowUuids>** instead of an array. This occurs for all soap requests that have an element wsdl:arrayType that does not have nillable="true".

  In the SOAP UI, the generated request contains an empty **flowUuids** array by default. You should edit it manually and insert any relevant UUIDs of interest. For example, the **xmlTemplate** input of the **Invoke Method 2** step in the generated Get Flow Details can be modified manually as shown below:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsc="http://wscentralservice.services.dharma.iconclude.com"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
   <soapenv:Header/>
   <soapenv:Body>
      <wsc:getFlowDetails
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <flowUuids xsi:type="xsd:string">?</flowUuids>
    </wsc:getFlowDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

Creating an OO Web Service Tool with Proxy

If you cannot launch the OO Web Services wizard to create operations from a WSDLfile (local or remote), create an XMLfile named **soapui-settings.xml** in the following format and save it in your home directory:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<con:soapui-settings xmlns:con="http://eviware.com/soapui/config">
<con:setting id="WsdlSettings@cache-wsdls">true</con:setting>
<con:setting id="WsdlSettings@pretty-print-response-xml">true</con:setting>
<con:setting id="HttpSettings@include_request_in_time_taken">true</con:setting>
<con:setting id="HttpSettings@include_response_in_time_taken">true</con:setting>
<con:setting id="WsdlSettings@name-with-binding">true</con:setting>
<con:setting id="HttpSettings@max_connections_per_host">500</con:setting>
<con:setting id="HttpSettings@max_total_connections">2000</con:setting>
<con:setting id="ProxySettings@host">111.222.111.322</con:setting>
<con:setting id="ProxySettings@port">3128</con:setting>
<con:setting id="ProxySettings@enableProxy">true</con:setting>
</con:soapui-settings>
```

If this fails due to proxy issues, this file will ask for your credentials for the proxy (hard-coded in the file). Modify the proxy host and port in the file, place it in your home directory, and enter the user name and password, when prompted. This procedure will allow the WSDL file to be loaded correctly.

Using the Shell Wizard

Purpose of the Shell Wizard

This integration enables administrators to create OO flows that are integrated with Shell.

The Shell Wizard helps to create a set of predefined commands that represent a workflow or procedure. The wizard includes a recorder technology that creates steps in a flow from the commands that are used in the wizard's Shell window.

To learn how to create OO flows, see the *OO Studio Authoring Guide* in the documentation set for the current OO release.

This document explains how this integration has been implemented, and how the integration's operations and flows communicate between OO and Shell.

Audience

This guide is intended for system administrators who establish and maintain the implementation of the integration between Shell and OO. This guide assumes that you have administrative access to both systems.

Supported Versions

Content Pack	Operations Orchestration Version	Shell Wizard Version
Base Content Pack	10.x	10.x

Getting Started with the Shell Wizard

Installing the Shell Wizard

The wizard is installed if Studio is selected from the Operations Orchestration installer. The Shell Wizard is located under **<OOInstallPath>\studio\tools**.

Configuring Logging Settings

The configure logging settings are no longer supported in the 10.x wizard.

Uninstalling the Shell Wizard
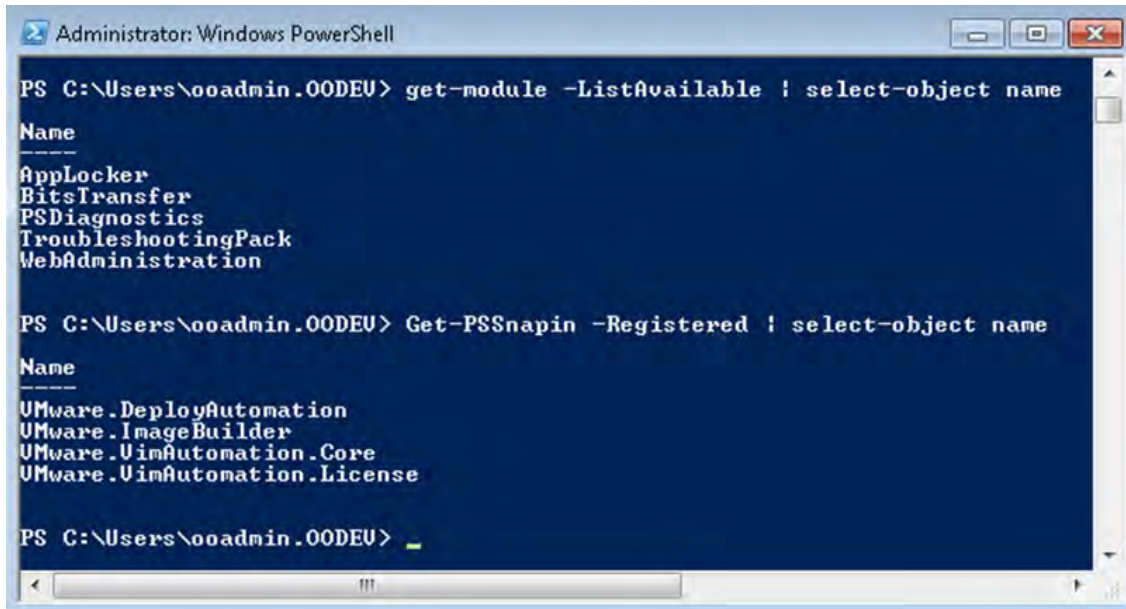
The wizard is uninstalled when Studio is uninstalled.

Shell Wizard Requirements

The following are the minimum software requirements for systems running Shell Wizard for OO:

- The environment must have Java SE Runtime Environment 8 (also known as JRE) installed (for running the wizards).

Target Host

- The machine must have SSH or Telnet protocol enabled.

Shell Wizard Enhancements from 9.x

- The wizard now has the version (10.x) displayed in the title.

- The wizard now appears in the task bar and can be closed, minimized or brought to the front.

Shell Wizard Steps

Step 1. Welcome Page

Start the Shell Wizard to open the **Welcome** page. Click **Next** to open the **Select Destination** page.

Step 2. Selecting the Destination

Here you need to select the location of an existing studio project, or a folder where the new project will be created.

Enter or select a location of a Studio project for the flows you want to create, and then click **Next** to open the **Specify flow information** page.

If this is not the first time you have run the wizard, the location is populated with the path to the last project created.
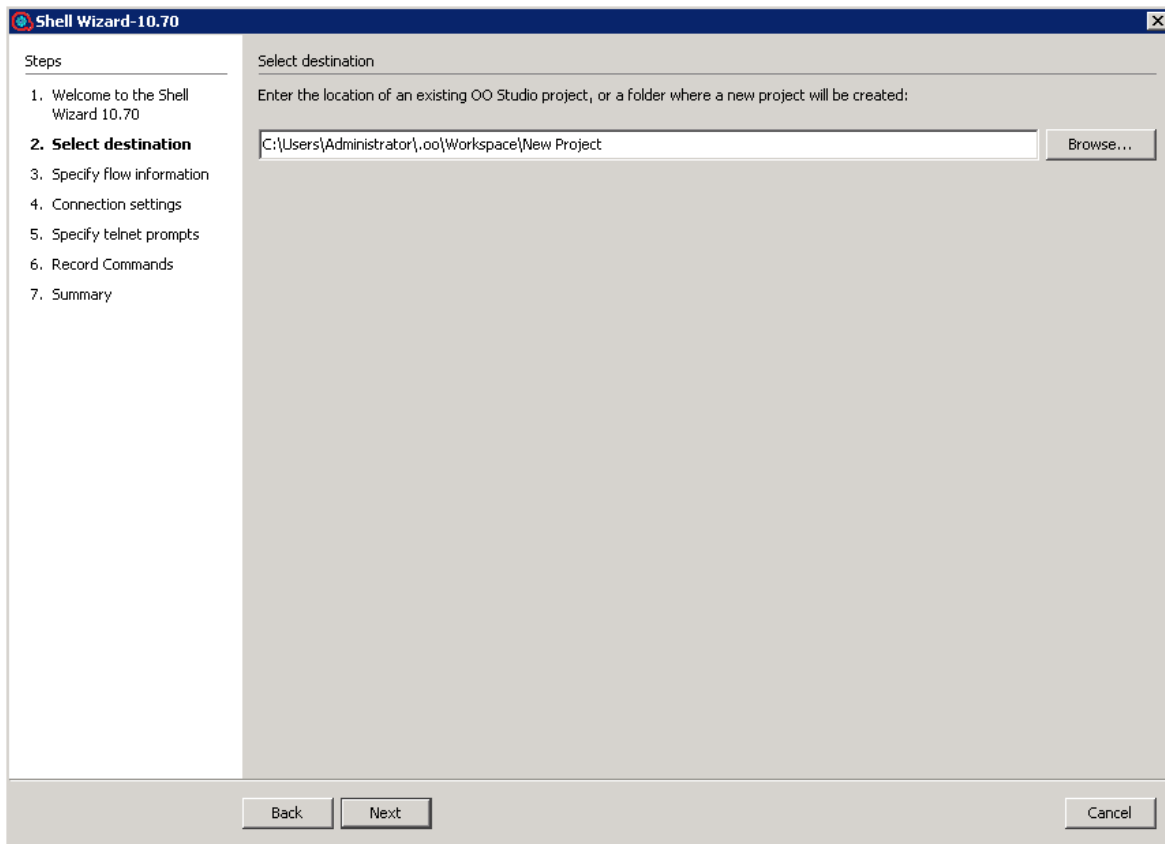
Step 3. Specifying Flow Information

Enter the flow name and flow description. If you want to overwrite the flow, check the option with the same name available in this window.

Click **Next** to open the **Connection settings** page.

Step 4. Entering Connection Settings

Enter the connection settings inputs.

- If the chosen protocol is SSH, the **Record Commands** page opens.

- If the chosen protocol is Telnet, the **Specify telnet prompts** page opens.

- If the chosen protocol is SSH, the Authenticating window opens until the connection is established.

Step 5. Specify Telnet Prompts

If the Telnet protocol was chosen, the **Specify Telnet prompts** page opens.
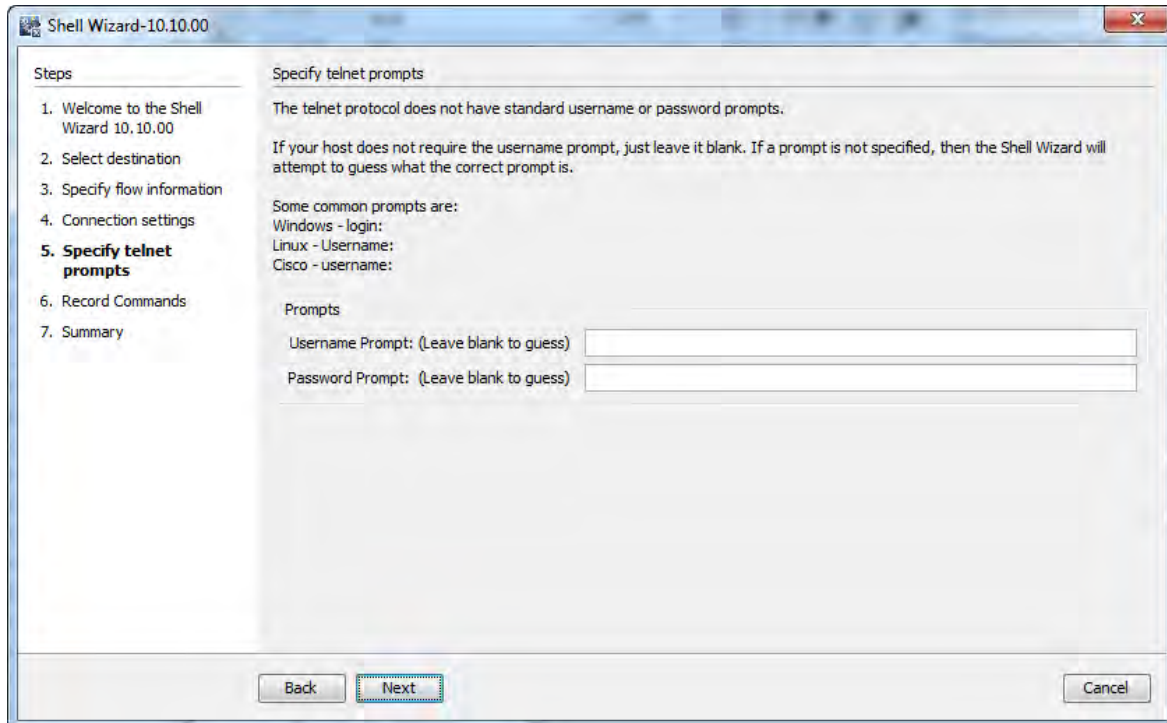
Click **Next** to open the connection. The Authenticating window opens until the connection is established.

Step 6. Recording Commands

If the SSH protocol was chosen, the **Record Commands** page opens.

The commands written in this screen are recorded and executed as a step in the generated flow.
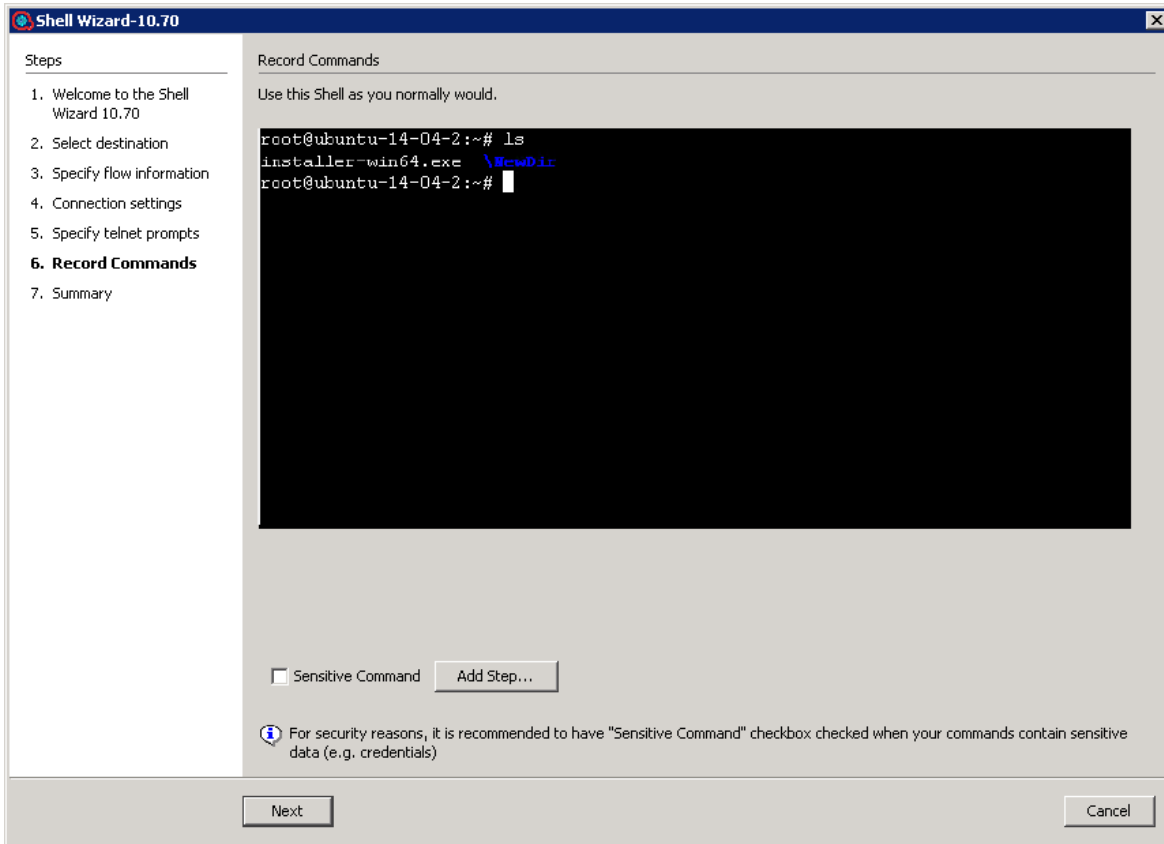
Click **Next** to go to the **Finish** page.

If the **Add Step** button is clicked, the Step Added window is displayed for a new connection to the server as a new step of the flow.
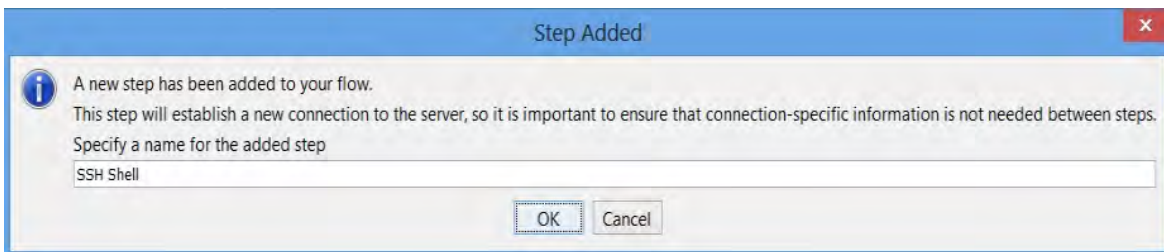
Step 7. Finish

Click **Finish** to exit the wizard.

Importing the Generated Flows into Studio

Use

Import the project generated by the wizard in OO Studio. See the "Managing Projects" section in the *Studio Authoring Guide* to see how to import a project.

Generated Flows

The Shell Wizard generates a flow with the name specified in the **Specify step information** step of the wizard. If a project with the same name already exists, the new flow will be added to.

Inputs

Each flow has the following inputs, which are commons to the Shell operation:
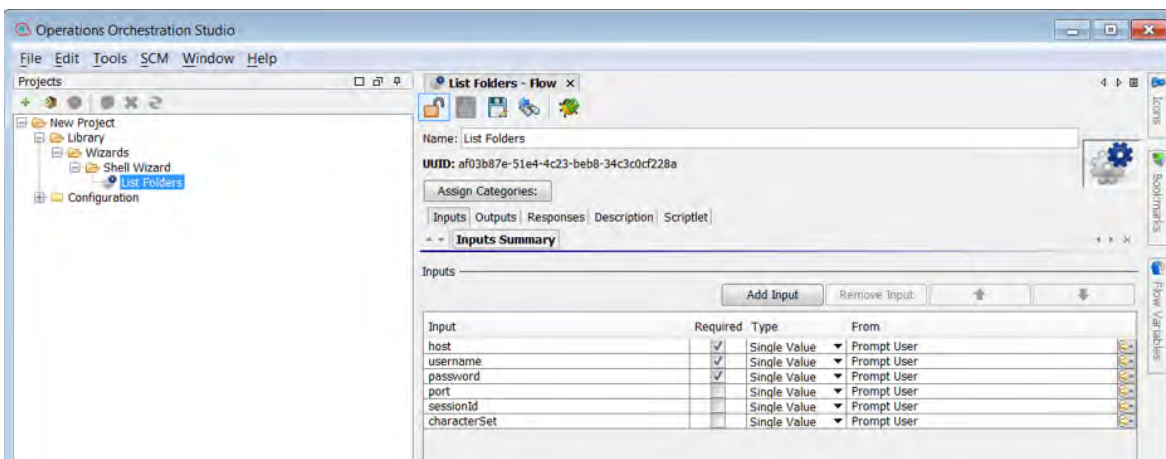
Outputs

On the **Outputs** tab, the flow primary result is **stdout** in addition to the **returnResult** result.

All OO flows have the **FailureMessage** and **TimedOut** outputs.

All generated flows have the **returnResult** output (which is the generic operation's result), and **stdout** result.
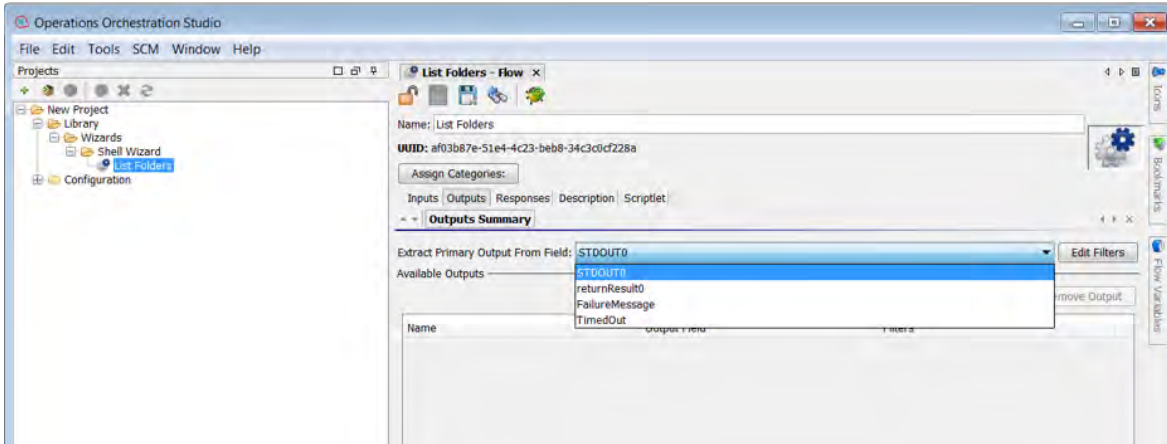
Responses

The success and failure responses of the flows are the same as in the Shell operations.
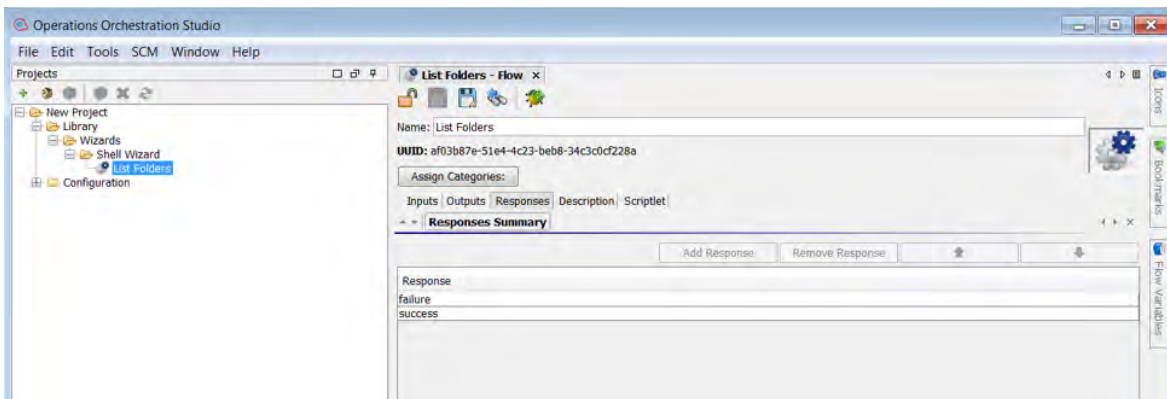
Descriptions

The description of each generated flow contains the following items:

- The description of the flow inputs

- The description of the results

- The description of the responses
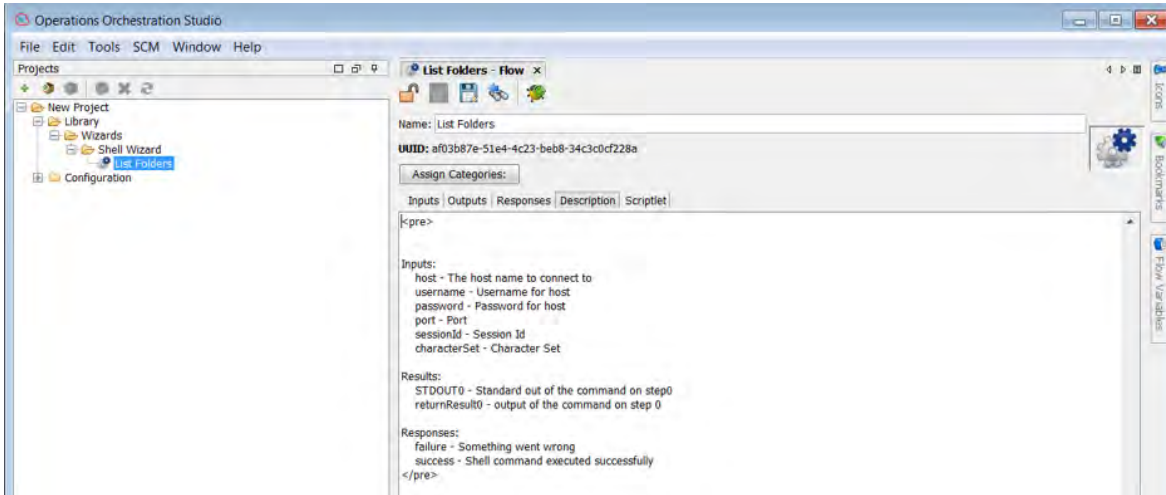
Shell Operation

The operation is used to execute a Shell script on a target host, either local or remote.

Operation Inputs

Operation Results

Troubleshooting

Troubleshooting Overview

This section provides troubleshooting procedures and tools that you can use to solve problems you may encounter while using this integration. It also includes a list of the error messages you may receive while using the integration and offers descriptions and possible fixes for the errors.

Could not connect to the host

The possible reasons are:

- The user credentials are not correct.

- Telnet protocol is not enabled on the machine.

- The user does not have permission to execute Shell scripts on the target host. Make sure the user has admin rights for executing the file.

- Authentication problems (most common). See "Running a PowerShell Script on a Remote Host".

Error Messages

This section lists the error messages you may receive while using this integration. Each error message includes possible causes and fixes for the error.

```
java.net.ConnectException: Connection refused: connect
```

This error occurs when SSH or Telnet protocol are not supported on the machine.

Using the PowerShell Wizard

Purpose of the PowerShell Wizard

With this integration, users can generate Operations Orchestration flows from the selected PowerShell cmdlets found in a list of modules/snapins.

Its main advantages are:

- Automation. Avoid having to repeat the same time-consuming process of creating flows which execute PowerShell cmdlets. Perform the following steps as an alternative to using the PowerShell Wizard (multiply by the number of cmdlets):

  - Create an empty flow

  - Drag and drop the PowerShell Script operation

  - Search for the cmdlet description

  - Set the required input values

  - Set the description of the flow. The step inherits its description from the PowerShell Script operation, but this is not available for the flow.

- Authoring ease. The description of each flow contains the default description of the corresponding cmdlet which it executes. Therefore, the user is not forced to open the cmdlet description in a browser and switch between OO and the Internet.

- Module and cmdlet discovery. The wizard discovers the available modules and cmdlets from a target host.

Supported Versions

Operations Orchestration Version	PowerShell Wizard Version
10.x with Base Content Pack	10.x

Getting Started with the PowerShell Wizard

Starting the PowerShell Wizard

If Studio is selected from the Operation Orchestration installer, the wizard is located under **<installation folder>\studio\tools**.

Double-click the **ps-wizard.bat** file under**<installation folder>\studio\tools**.

Configuring Logging Settings

The configure logging settings are no longer supported in the 10.x wizard.

Uninstalling the PowerShell Wizard

The wizard is uninstalled when Studio is uninstalled.

PowerShell Wizard Requirements

The minimum software requirements for systems running the PowerShell Wizard for Operations Orchestration are:

- Java SE Runtime Environment (also known as JRE) 8

- .NET Framework 2 or a later version

Target Host:

- PowerShell with remoting enabled.

- For the products providing PowerShell cmdlets, the target host must have the modules and snapins available. Run one of the following cmdlets to list the required modules and snapins. Otherwise, it means that the host does not have the cmdlets provided for that product.

Figure 1: How to list the modules and snapins in the PowerShell console

PowerShell Wizard Enhancements from 9.x

- The wizard now has the version (10.x) displayed in the title.

- The wizard now appears in the task bar and can be closed, minimized or brought to the front.

PowerShell Wizard Steps

The PowerShell Wizard contains only a few steps. The Welcome page contains a short summary of the wizard. This section describes the steps that you have to perform.

Step 1. Selecting the Repository

In the **Enter the location** field, type the required project path or click **Browse** to locate the project location.

The wizard generates a 10.x Studio project, but not a content pack or a repository. The project has a default location: **C:\Users\[username]\.oo\Workspace\New Project**.

If this is not the first time you have run the wizard, the location is populated with the path to the last project created.
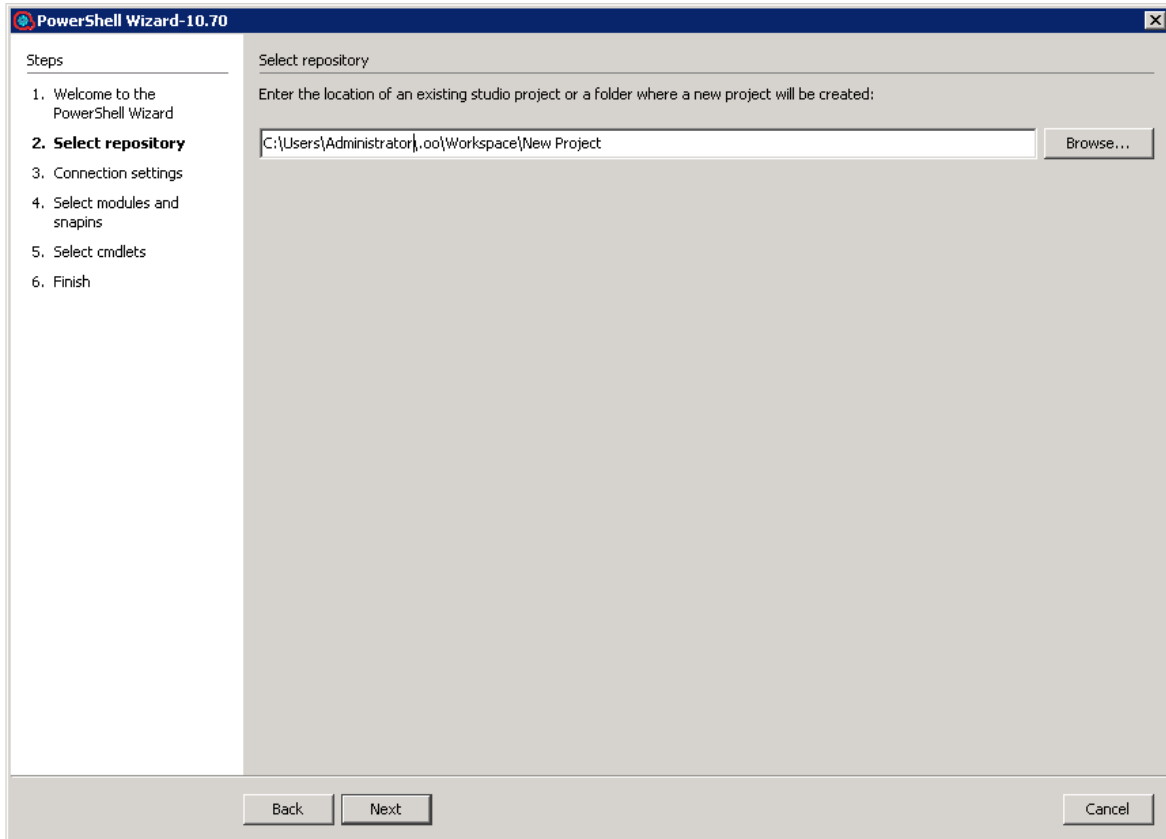
Figure 2: PowerShell Wizard: Select repository page

Step 2. Configuring the PowerShell Connection

Figure 3: PowerShell Wizard: Connection settings page

Host - Type the name of the host that you want to connect to. If you leave the **Host** field empty, the PowerShell Wizard uses localhost as the default.

> **Note:** If **Host** is empty then the authentication type will be **NegotiateWithImplicitCredential**. If the host has been defined, the wizard considers the host definition provided by the user.

Username - Enter the user name to connect to the target host..

Password - Enter the password.

Port - The port values can be in the range of 1- 65535. If you set the port value to 0, the wizard ignores it and uses the default port values. The default port values are: 5985 (HTTP) and 5986 (HTTPS).

Step 3. Selecting the Modules
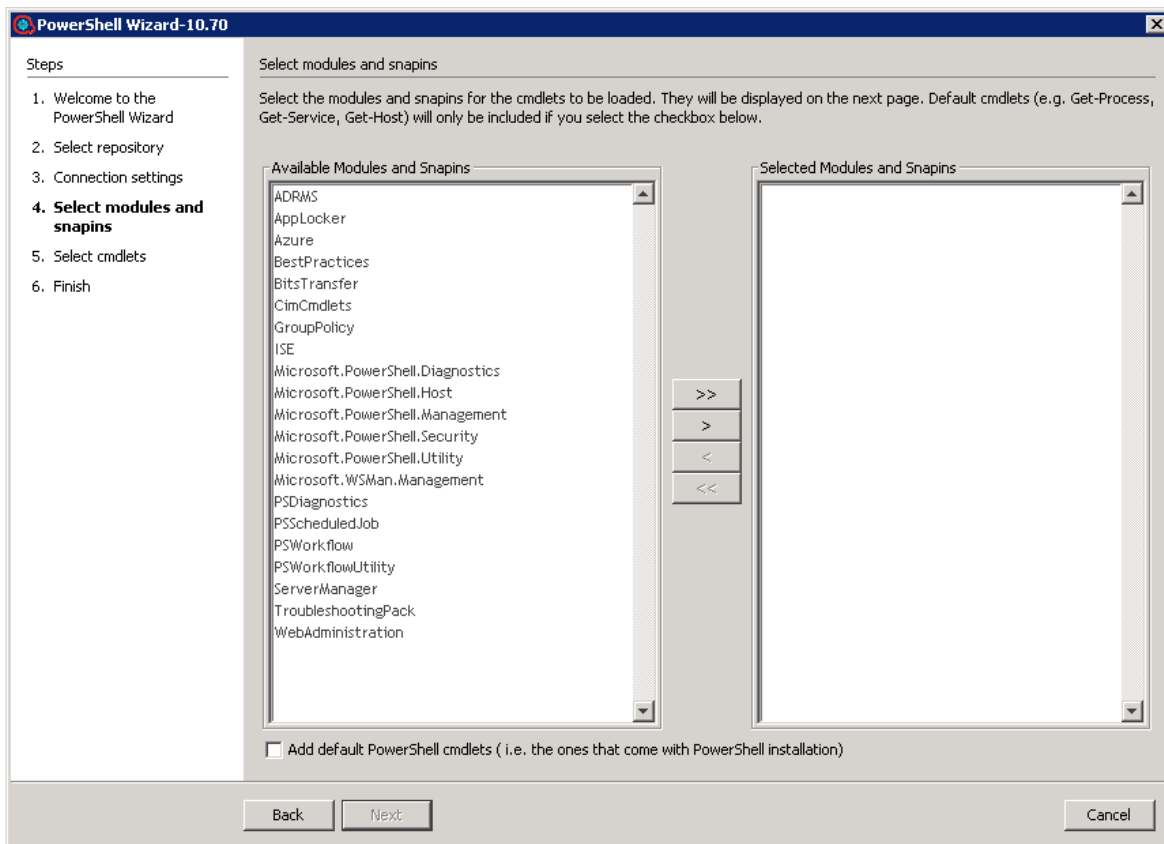
Figure 4: PowerShell Wizard: Select modules and snapins page

The wizard detects all the available modules/snapins on the target host and displays them in a list as shown above. You can select/deselect any module and the wizard retrieves only those cmdlets contained in the selected modules.

Cmdlets such as **Get-Process** and **Get-Service** are not contained in the list of available modules. These are cmdlets which are available by default in PowerShell. To retrieve the list of default cmdlets, select the **Add default PowerShell cmdlets** check box .

> **Note:** If you select the **Add default PowerShell cmdlets** check box, the lists are disabled.

Step 4. Selecting Operations (Cmdlets)
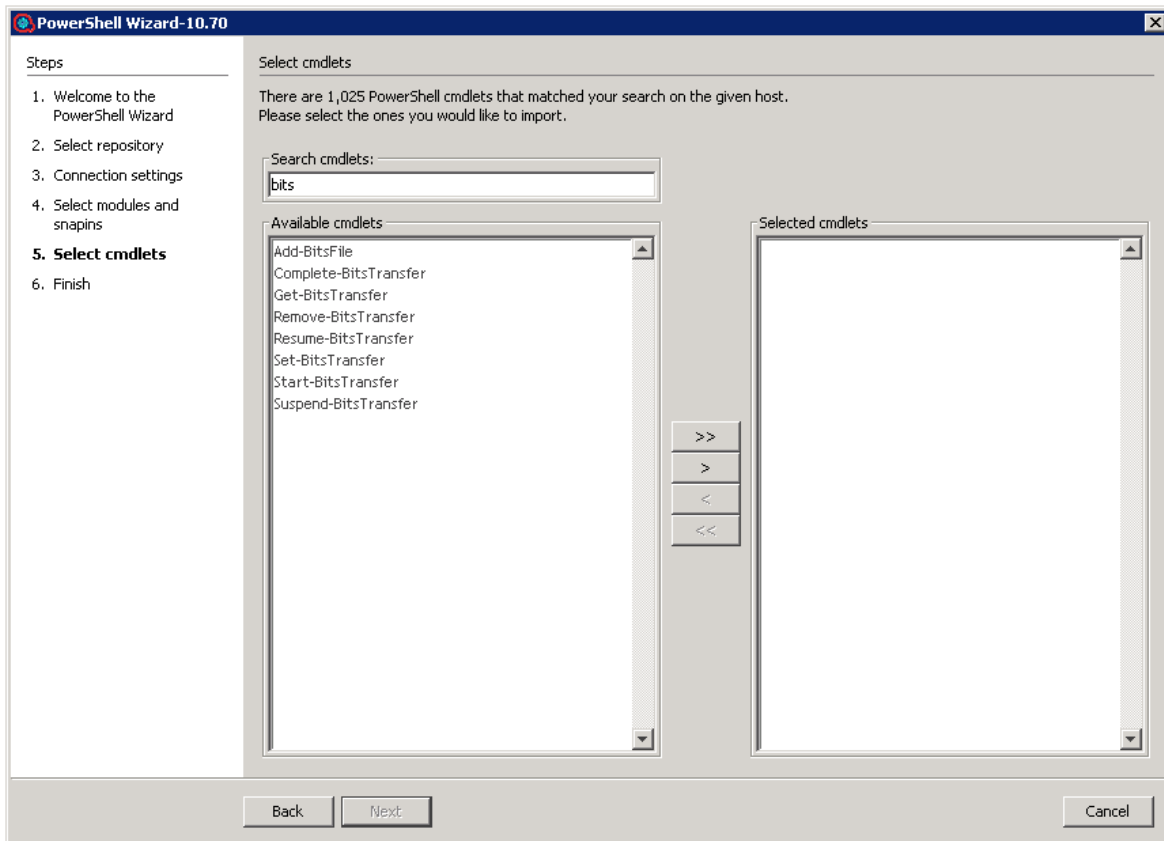
Figure 5: PowerShell Wizard: Select cmdlets page

The selected modules are loaded to the PowerShell runspace, and the wizard retrieves the names of the cmdlets from those modules.

You can move the cmdlets from left to right or right to left. Use the search text box if the list is very large, and you have difficulties finding the required cmdlet. The wizard searches the list for the cmdlets with names containing the search text. In addition, the wizard updates the list while you are typing.

Using the PowerShell Wizard

PowerShell Wizard Operations and Flows

This section describes the operations and flows in the PowerShell wizard.

Generated Flows
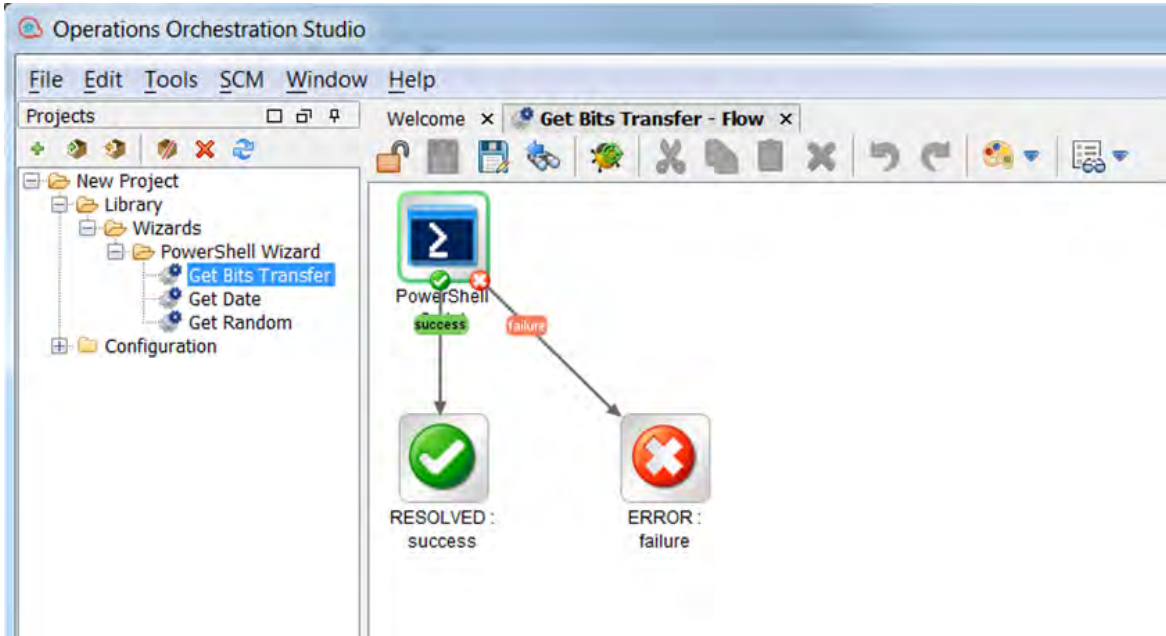
Figure 6: Example of a generated flow

The PowerShell wizard generates one flow for each selected cmdlet unless the project already contains the flow. For example, if the flow was generated in a previous run of the wizard.

The name of the flow is obtained from the name of the cmdlet by applying the following rule:

- Replace "-" with " ".

For example, the name of the flow for the cmdlet **Get-IScsiHbaTarget** is changed to **Get IScsi Hba Target**.

Inputs

Figure 7: Inputs of a generated flow

Each flow has the following inputs:

- Common inputs. All the inputs of the PowerShell Script operation

Descriptions

Figure 8: Description of a generated flow

The description of each generated flow contains the following items:

- A description of the cmdlet as found in its native documentation

- A link where the user can find a detailed description of the cmdlet written by the provider of the module

- The description of the common inputs copied from the PowerShell Script operation's description.

- The description of the common results

Most of the information included in the description can be obtained from the PowerShell console as shown below:

Figure 9: Obtaining the description from the PowerShell console

PowerShell Script Operation

The operation is used to execute a PowerShell script or cmdlet on a target host, either local or remote. If the operation executes a single cmdlet, the parameters of the cmdlet should be passed to the operation inputs.

> **Note:** When interacting with a Microsoft Exchange server, use the generic PowerShell operation available in the **Business Application** content pack, under **/Library/Operations/Exchange/<version specific>** folder, instead of loading the Exchange snap-in to this PowerShell Script operation. This is a Microsoft Exchange design.
>
> See http://www.get-exchange.info/2012/12/30/powershell-scripting-for-exchange-server-some-tips/
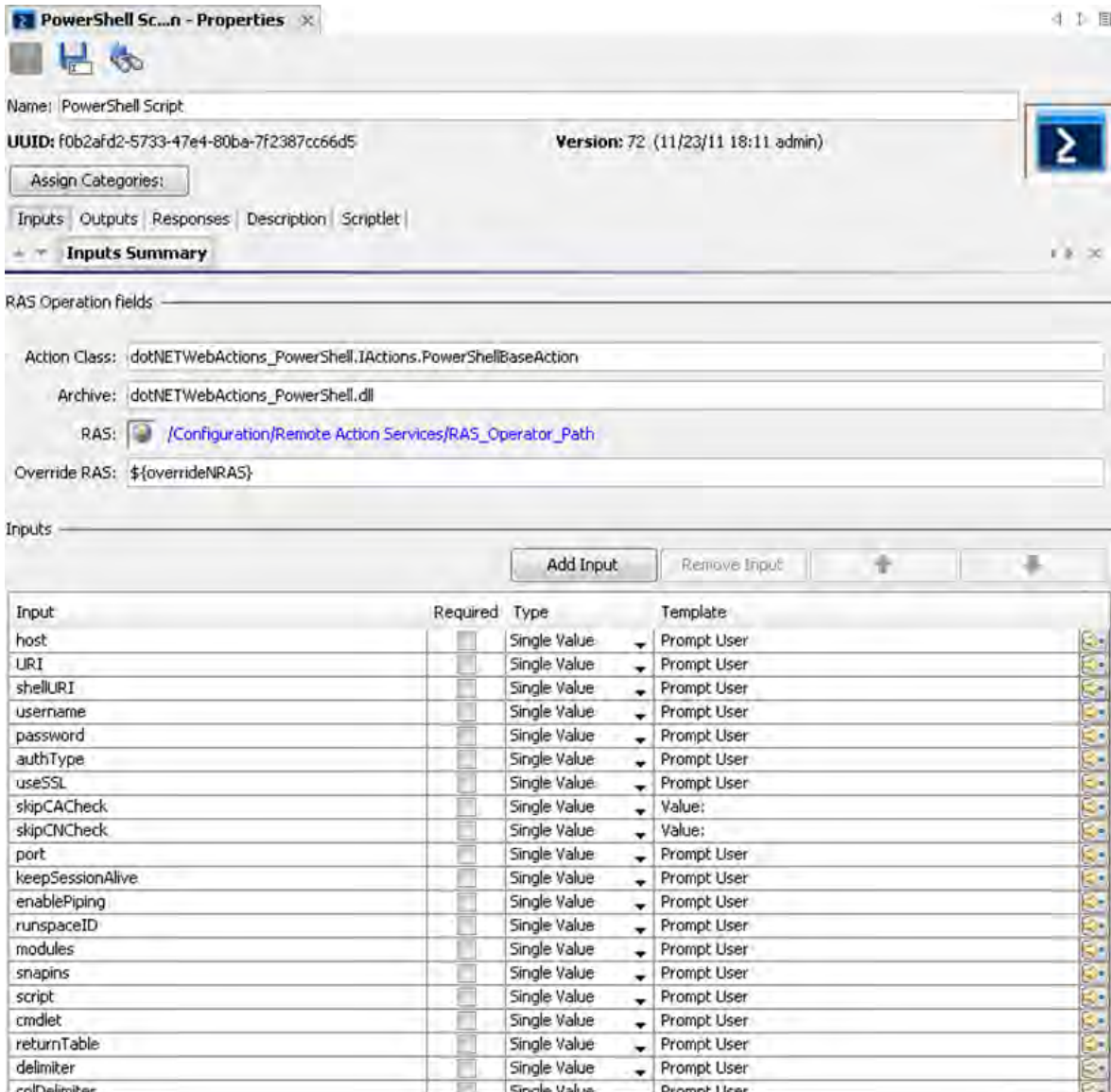
Inputs

Figure 10: Inputs of the PowerShell Script operation

Connection Inputs

host

The hostname or IP address of the PowerShell host. This input is mutual exclusive with URI and shellURI

Default value: localhost

URI

Specifies a Uniform Resource Identifier (URI) that defines the connection endpoint for the interactive session.

Value format: **<Transport>://<ComputerName>:<Port>/<ApplicationName>**

Example: **http://exch2010CAS1/Powershell?serializationLevel=Full**

shellURI

Gets the Uniform Resource Identifier (URI) of the shell that is launched when the connection is made. This input is mutual exclusive with host.

Example: **http://schemas.microsoft.com/powershell/Microsoft.Exchange**

Username

The user name to use when connecting to the server.

Value format: **username@domain or domain\username**

password

The password to use when connecting to the server

authType

Specifies the mechanism that is used to authenticate the user's credentials. Valid values: **Default**, **Basic**, **Credssp**, **Digest**, **Kerberos**, **Negotiate**, **NegotiateWithImplicitCredential** (case-insensitive).

Default value: Default

useSSL

If true, the operation uses the Secure Sockets Layer (SSL) protocol to establish a connection to the remote computer. By default, SSL is not used. WS-Management encrypts all Windows PowerShell content transmitted over the network. UseSSL is an additional protection that sends the data across an HTTPS connection instead of an HTTP connection.

Default port for SSL: 5986.

Default value: false

skipCACheck

Indicates when connecting over HTTPS that the client does not validate that the server certificate is signed by a trusted certificate authority (CA).

Default value: true

skipCNCheck

Indicates whether the certificate common name (CN) of the server and the hostname of the server are not checked for being the same.

Default value: true

port

Specifies the network port on the remote computer that is used for this connection. To connect to a remote computer, the remote computer must be listening on the port that the connection uses. The default ports are 5985 (the WinRM port for HTTP) and 5986 (the WinRM port for HTTPS).

Session and Piping Inputs

keepSessionAlive

If true, the operation will not close the PowerShell runspace (i.e. the PSSession in case of remote connections) created during the execution and returns the runspaceID as the result of the operation. Actually the PowerShell runspace is saved in the OO session and can be used in other operations using the runspaceID result of previous operation.

Default value: false

enablePiping

If true, the operation enables piping the way PowerShell does. This input should be used only when keepSessionAlive input is true. First time the piped objects are null and the scope is to save the objects resulted from the PowerShell script execution. Next time the piped objects can be referred in the script as "$_" objects.

Default value: false

Example: First, run "Get-Service" script with enablePiping=true and keepSessionAlive=true, then run the script "Select-Object -property name,displayname,status|where-object {$_.Name -lt "C"}" with enablePiping=true keepSessionAlive=true. Observe the use of "$_".

runspaceID

If this is not empty, the operation searches the OO session for keys which equal the runspaceID. If the operation finds the runspace specified by the runspaceID it tries to use it and does not create another PowerShell runspace with provided inputs. If the runspace exists but is broken or unavailable the operation uses its authentication parameters and tries to reconnect and recreate the PowerShell runspace.

> **Note:** The operation uses the connection parameters of the runspace identified by runspaceID, not the values provided as user inputs. The operation processes the connection inputs, for example, host, username, password, authType, useSSL that is provided by the user only if the runspaceID does not exist in the OO session.

Additional Modules and Snapins

Modules

A list of PowerShell modules that is loaded after the PowerShell connection is established. Each value from the list specifies the name of the module to import. Enter the name of the module or the name of a file in the module, such as a .psd1, .psm1, .dll, or ps1 file. File paths are optional. Wildcards are not permitted. Specify only the module name whenever possible. When you specify a file name, only the members that are implemented in that file are imported. If the module contains other files, they are not imported, and you might be missing important members of the module. The list of modules should be separated by the comma "," delimiter.

Example: FailoverClusters

Snapins

A list of PowerShell snapins loaded after the PowerShell connection is established. Each value from the list specifies the name of a registered snapin, for example, the Name, not the AssemblyName or ModuleName.

Example: Microsoft.Exchange.Management.PowerShell.E2010

PowerShell Script and cmdlet Inputs

script

The script to execute on the PowerShell host. If you want to execute a script from a file just provide the file path.

Example:

```
C:\PowerShellScripts\GetHost.ps1
```

Cmdlet

The name of the PowerShell cmdlet to invoke. If the cmdlet has additional parameters, please provide them as inputs to the operation. If the parameter has the same name as one of the operation's inputs just prefix it with "_". This input is intended to be used together with the PowerShell wizard so please use the script input whenever possible.

Formatting the Result

returnTable

If true, the operation will return a table containing a row for each PSObject that the script emits. The table's columns represent the properties of these PSObjects, in the

propertyName<delimiter>propertyValue format. If false the operation returns a string representation of the result similar to the output from the PowerShell console.

Default value: false.

delimiter

The delimiter used to separate each property name from the property value in the output table.

Default value: ":".

colDelimiter

The delimiter used to separate columns in the output table.

Default value: ",".

rowDelimiter

The delimiter used to separate rows in the output table.

Default value: newline.

Running a PowerShell Script on a Localhost

The only setting required to execute the PowerShell scripts on the localhost is that the ExecutionPolicy must be RemoteSigned. Use Get-ExecutionPolicy to display the current execution policy and Set-ExecutionPolicy to set the execution policy.

In addition, the required input is the script input.
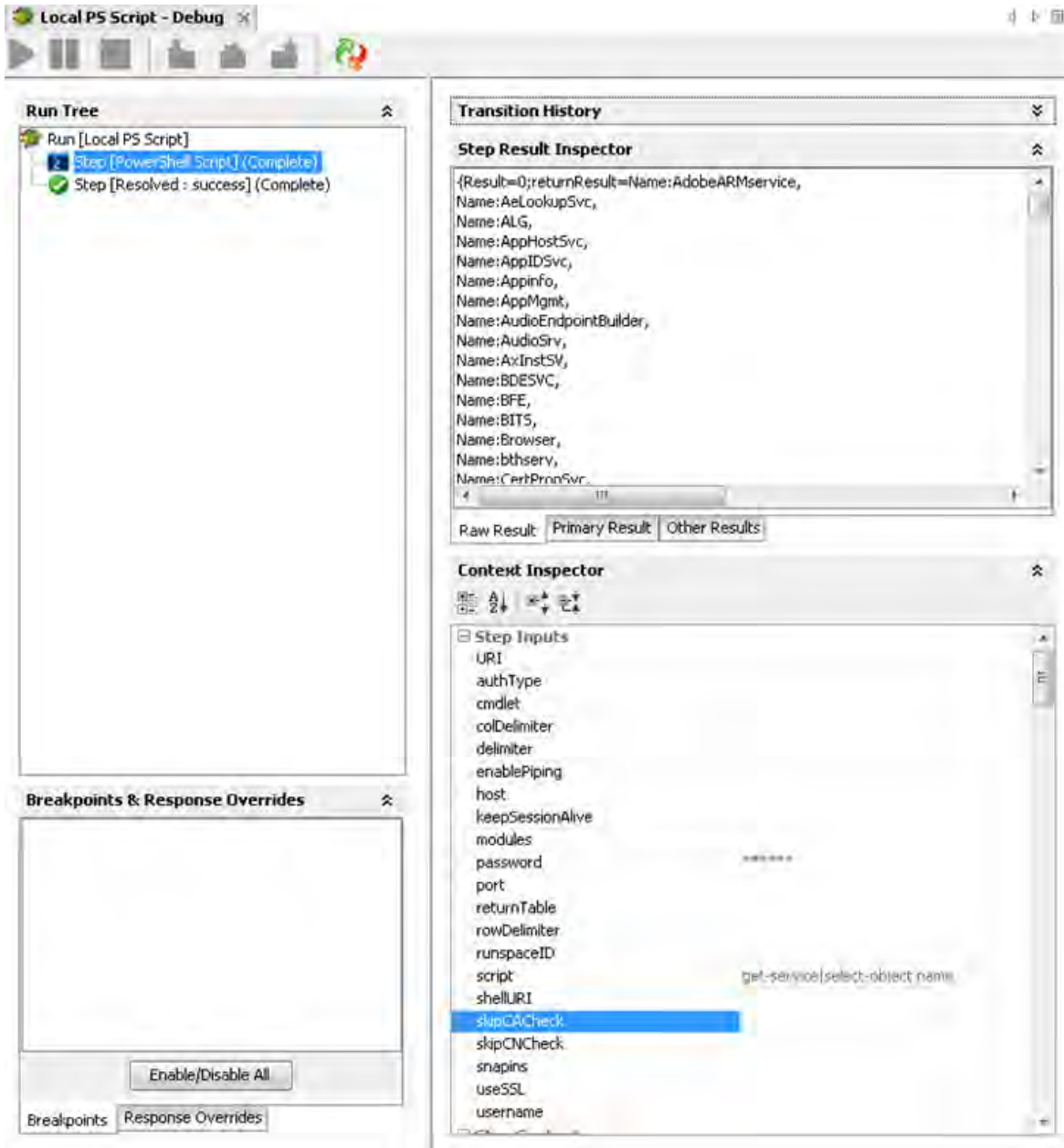
Figure 11: Running the PowerShell script operation on localhost

If one script requires elevated rights, enter a user name and a password.

Running PowerShell Scripts from a File
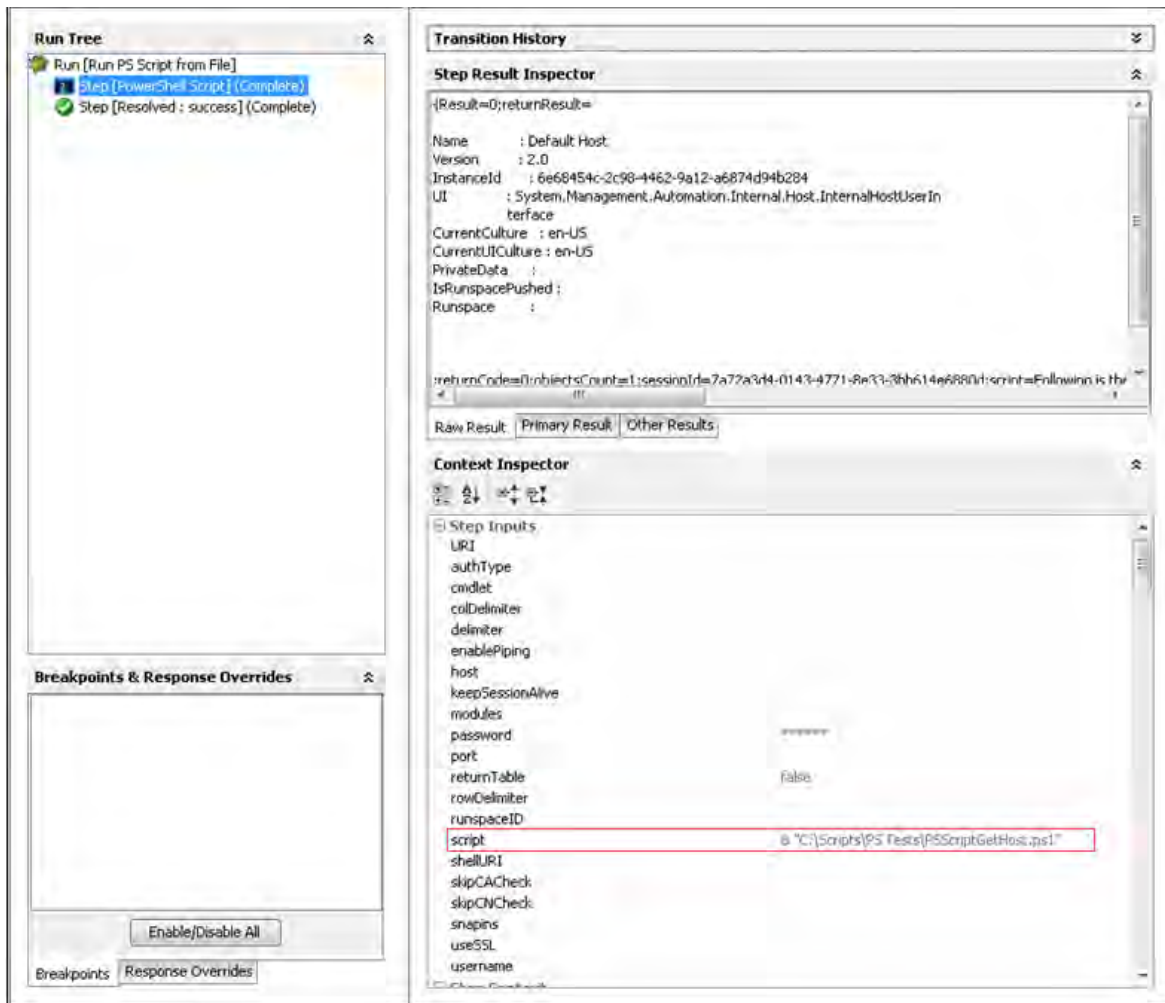
Figure 12: Running the PowerShell script operation from a file

To run scripts from a file, just provide the path to that file. This should work in most of the cases, when the path to the file contains white spaces, the operation fails. To fix this provide the path to that file like in the picture above.

Loading PowerShell Functions from Files
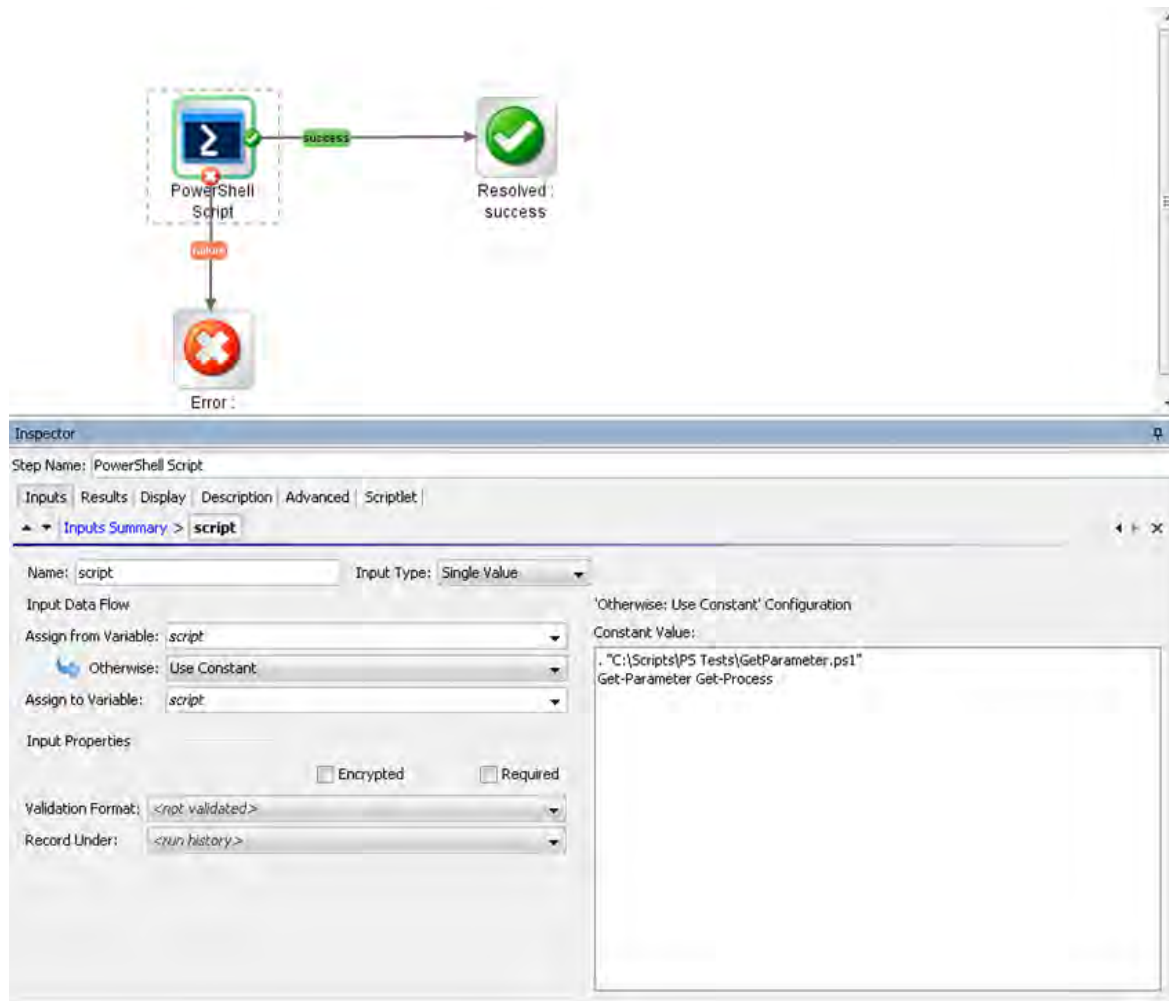
Figure 13: Using a script to enable all functions from a local file

In certain cases, a PowerShell script depends on functions from other file from the disk. The script from the picture above might help to load this file and enable all functions and cmdlets from it. Get-Parameter cmdlet is defined in the file named Get-Parameter.ps1.

Running a PowerShell Script on a Remote Host
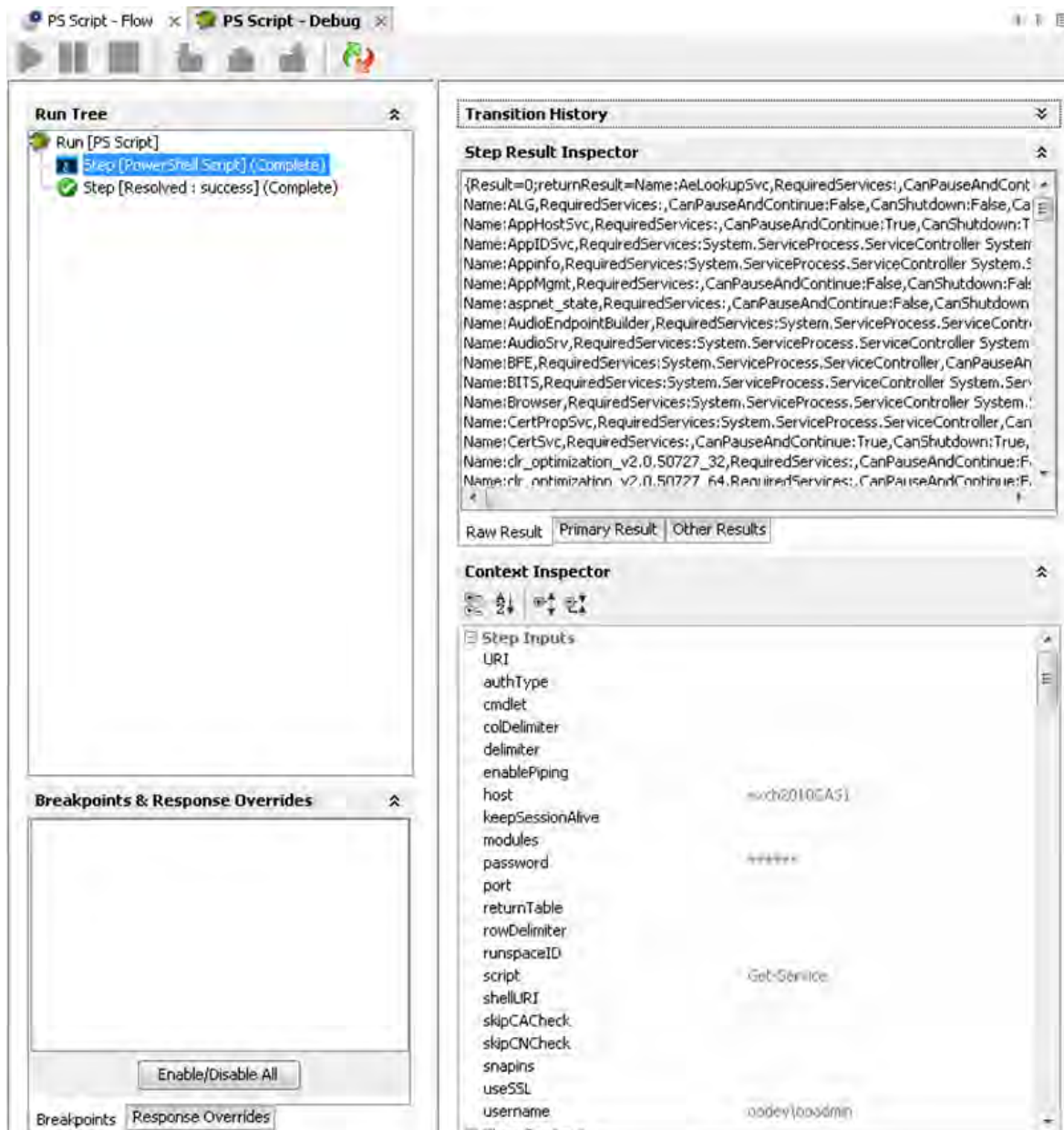
Figure 14: Running the PowerShell script operation on a remote host

First, execute the script providing only the required inputs: host, user name, password and the script.

Negotiate is the default authentication type if the host is provided as an IP address and Kerberos for host names. See PowerShell Remoting and Authentication Types if the connection fails.

Formatting the Result

The result can be formatted as a table or the same way it will be displayed in the PowerShell console. The format is decided by the **returnTable** input described above. If **returnTable** is set to false the operation will return the result as in the PowerShell console and like the old operation.

Use

```
Step Result Inspector

{Result=0;returnResult=
Status  Name            DisplayName
------  ----            -----------
Running AdobeARMservice   Adobe Acrobat Update Service
Stopped AeLookupSvc     Application Experience
Stopped ALG             Application Layer Gateway Service
Running AppHostSvc       Application Host Helper Service
Stopped AppIDSvc        Application Identity
Stopped Appinfo         Application Information
Stopped AppMgmt          Application Management
Running AudioEndpointBu... Windows Audio Endpoint Builder
Running AudioSrv         Windows Audio
Stopped AxInstSV         ActiveX Installer (AxInstSV)
Stopped BDESVC           BitLocker Drive Encryption Service
Running BFE             Base Filtering Engine
Running BITS            Background Intelligent Transfer Ser...
Running Browser          Computer Browser
Stopped bthserv          Bluetooth Support Service
Running CertPropSvc      Certificate Propagation
Stopped clr_optimizatio... Microsoft .NET Framework NGEN v2.0....
Stopped clr_optimizatio... Microsoft .NET Framework NGEN v2.0....
Stopped clr_optimizatio... Microsoft .NET Framework NGEN v4.0....
Stopped clr_optimizatio... Microsoft .NET Framework NGEN v4.0....
Stopped COMSysApp        COM+ System Application
Running CryptSvc         Cryptographic Services
Running CscService       Offline Files
Running DcomLaunch       DCOM Server Process Launcher
Stopped defragsvc        Disk Defragmenter
Running Dhcp            DHCP Client
Running Dnscache         DNS Client
Stopped dot3svc          Wired AutoConfig
Running DPS             Diagnostic Policy Service
Stopped EapHost          Extensible Authentication Protocol
Stopped EFS             Encrypting File System (EFS)
Stopped ehRecvr          Windows Media Center Receiver Service
Stopped ehSched          Windows Media Center Scheduler Service
Running eventlog         Windows Event Log
Running EventSystem      COM+ Event System
Stopped Fax             Fax
Stopped fdPHost          Function Discovery Provider Host

Raw Result | Primary Result | Other Results
```

Figure 15: The result if returnTable=false

The result is human readable, but the problem is that it is very difficult to parse, and does not contain
properties which could not be displayed on the screen.

The result can be displayed as a table.

PowerShell session considerations results are displayed as a table. Each PowerShell object (in this case each service) is displayed by default on a line. Each line contains different properties of the service (default delimiter is ",") and the key-value pairs are delimited by ":". All these delimiters can be changed, refer to the Inputs section for more information.

For example Get-Service returns the following result:

Figure 16: Raw result of the Get-Service flow

The result contains a full list of properties without any additional PowerShell script, for example, **Get-Service|fl, Get-Service|Select-Object Status)**. Therefore, the status of the services appears and can be parsed by writing two types filters on the result.
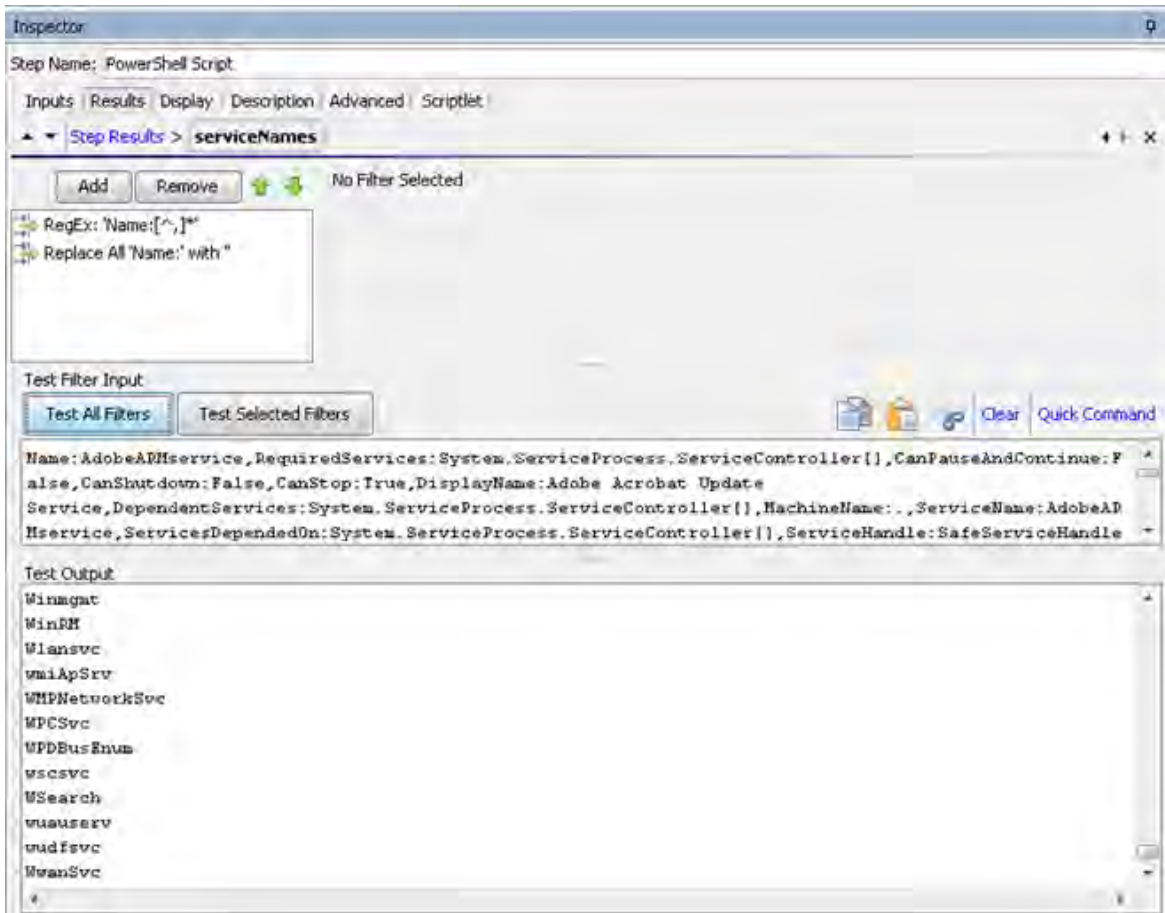
Figure 17: Filtering the results

Running Multiple PowerShell Cmdlets Scripts in the Same PowerShell Session

This section explains how to run multiple PowerShell Script steps in the same PowerShell session. When the PowerShell Script has to execute a single script on a remote server keeping the sessions alive is not necessary. The PowerShell Script connects to the remote host, creates a new PowerShell Runspace, for example, a new PowerShell session, runs the full script on the target and closes the runspace and the connection.

If you want to use the PowerShell Wizard and run the generated flows in a different sequence, you need to consider how much effort is required from the user and whether additional PowerShell Scripts need to be created in order to general flows. However, there are situations when you want to keep the PowerShell session alive.

For example, one uses the PowerShell Wizard, connects to a host which has PowerCLI installed on it and follows the wizard steps to generate OO flows for the PowerShell cmdlets to execute VMWare

tasks. Suppose that after the wizard finishes, the user wants to execute one simple cmdlet like Get VM.

To run cmdlets and keep the sessions alive:

1. Run the OOTB Get Datacenter flow generated with the PowerShell Wizard.

   Note: This cmdlet does not have any required inputs, however the flow fails to run as you need to run the Connect VIServer cmdlet. The PowerShell Wizard generates this flow to solve this problem.

2. Create a flow sequence as shown below.

   Do not modify the generated flow.

   The flow tries to execute Connect VIServer before Get VM.

   In the flow below, the parameters specific to the cmdlet were added as flow inputs. The names appear in capital letters.

Figure 18: Adding values to specific parameters of the cmdlet

The user runs the flow and provides inputs for the server, user name and password (Connect VIServer). No inputs are required for the Get Datacenter. The Connect VIServer succeeds, however, the Get Datacenter fails.

The Get Datacenter fails for the following reasons:

- Connect VIServer passes successfully and the connection to the VMWare server was established.

- Connect VIServer created a new PowerShell runspace, for example, PowerShell session and executed the cmdlet which established a valid connection to the server, however the runspace is closed after the flow runs and the connection is lost.

- Get Datacenter flow creates another PowerShell runspace which is different from the one created by Connect VIServer flow. Therefore, Get-Datacenter cmdlet fails.

3. The solution to the previous step is to keep the session alive during the execution of the two cmdlets.

   To do this:

   keepSessionAlive=true for the first flow which is Connect VIServer; the runspaceID must be added to the results of the Connect VIServer flow:

Figure 19: Adding a result to the Connect VIServer flow

4. The runspaceID of the Get Datacenter flow must get its value from the result of the Connect VIServer. This happens automatically because runspaceID input assigns its value from the flow variable.

At this point the flow completes successfully.
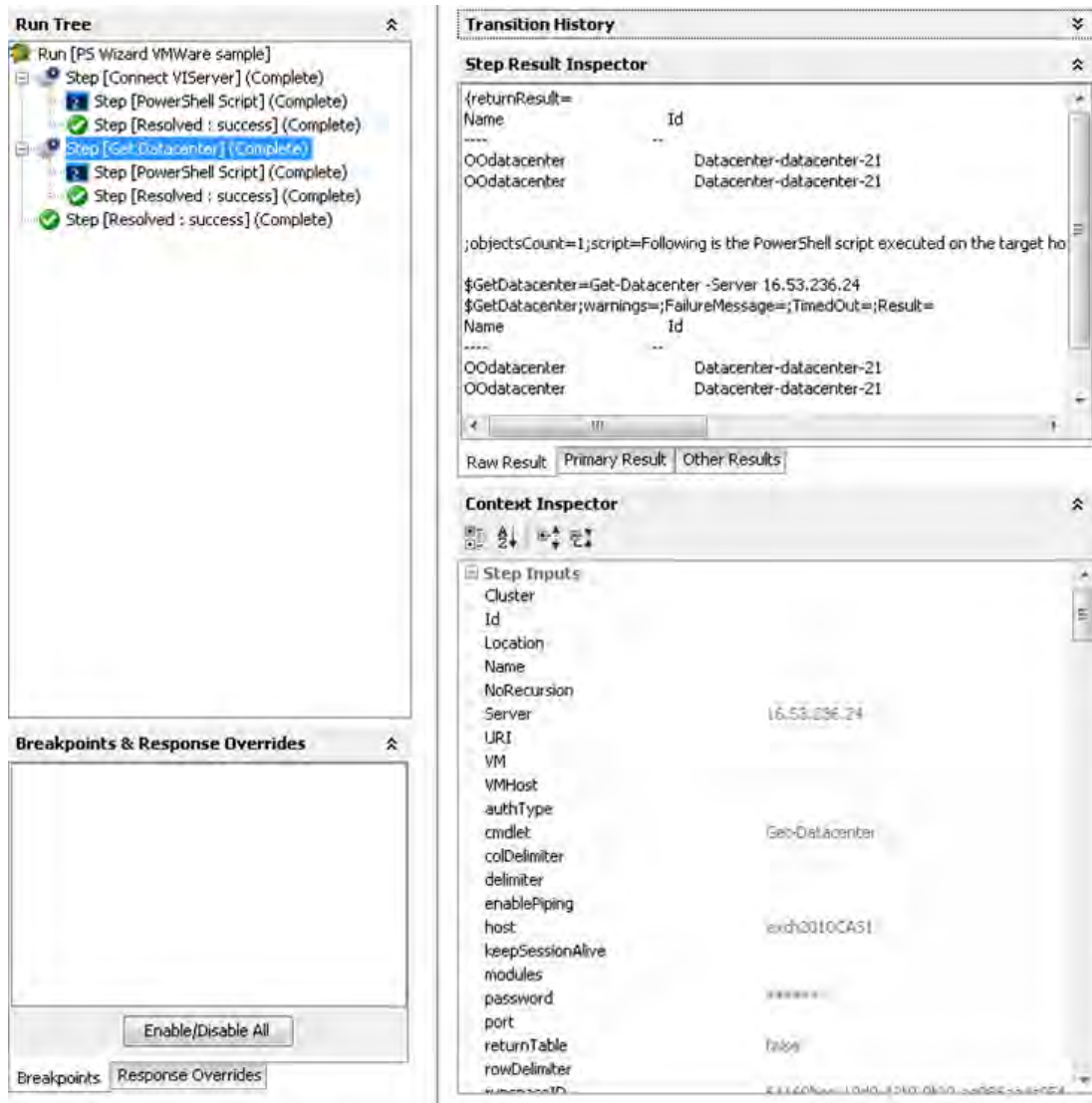
Figure 20: The execution of the PSWizard VMWare sample flow

Assigning the Result of One Cmdlet as a Parameter to Another Cmdlet

Most of the "get" flows generated through the PowerShell Wizard should work OOTB with minimum effort from the user. But there are some cmdlets, probably the "new" cmdlets, which require as parameters the result of another cmdlet. For example, one would like to create a new virtual machine using the generated flow New VM.

Even if the user follows all the steps described in the previous section, the flow can not be executed. The PowerShell Wizard generates the flow, but the user can run OOTB flows only if their parameters have a built-in type (for example, strings or integers). In case of cmdlets like Get-Help the parameters

can be passed as strings (for example, the name of the cmdlet to search for help information). The New VM flow parameters are below:
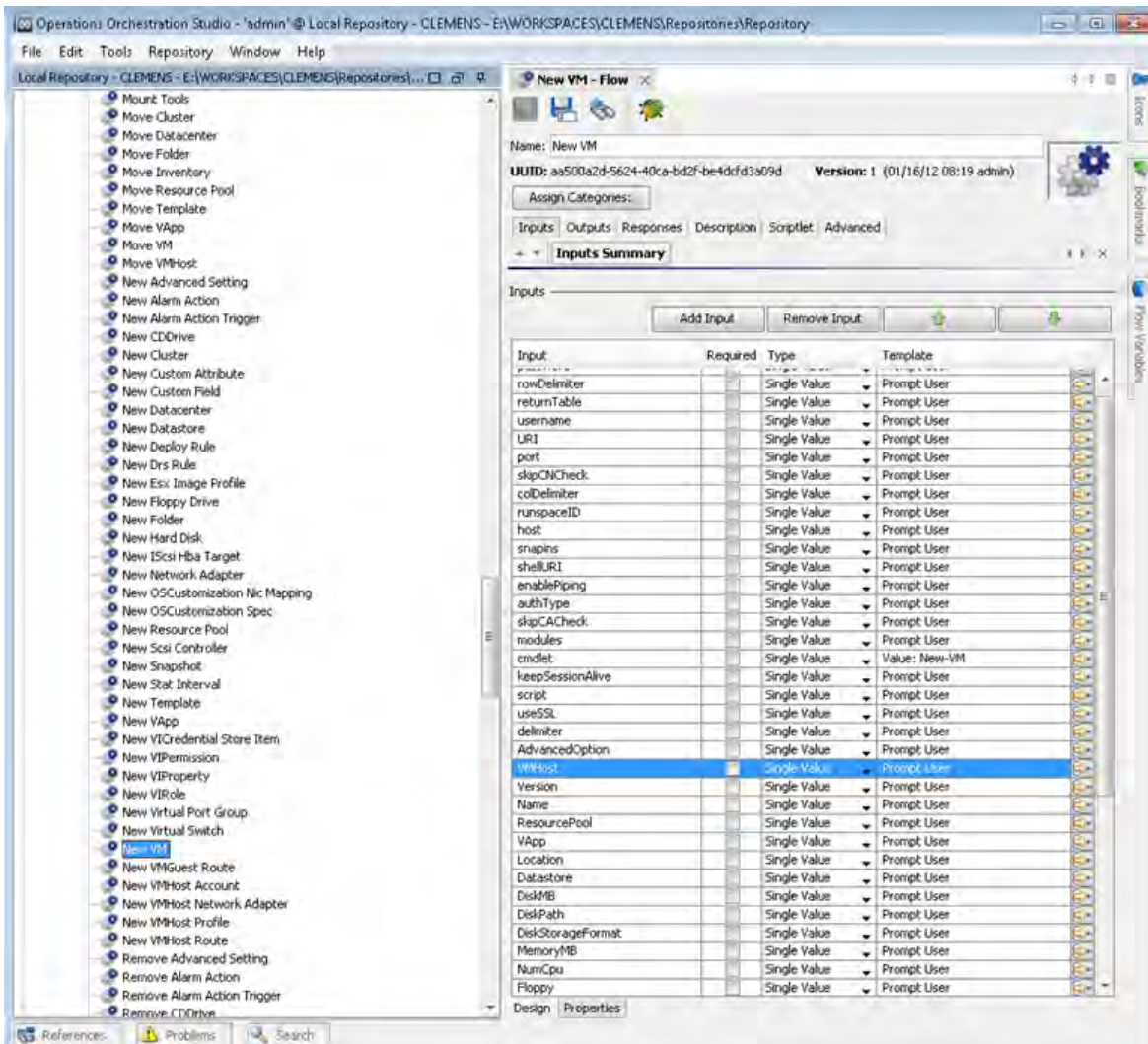
Figure 21: The inputs of the New VM flow

The New VM flow has an input named VMHost. What is the type of this parameter?

The description of the generated flows contains information about the PowerShell cmdlet, but from size reasons and other considerations we could not include the full description of the cmdlet as it is displayed when someone executes Get-Help New-VM –full. The description of the operation contains the original link where the user can find detailed information about the cmdlet.
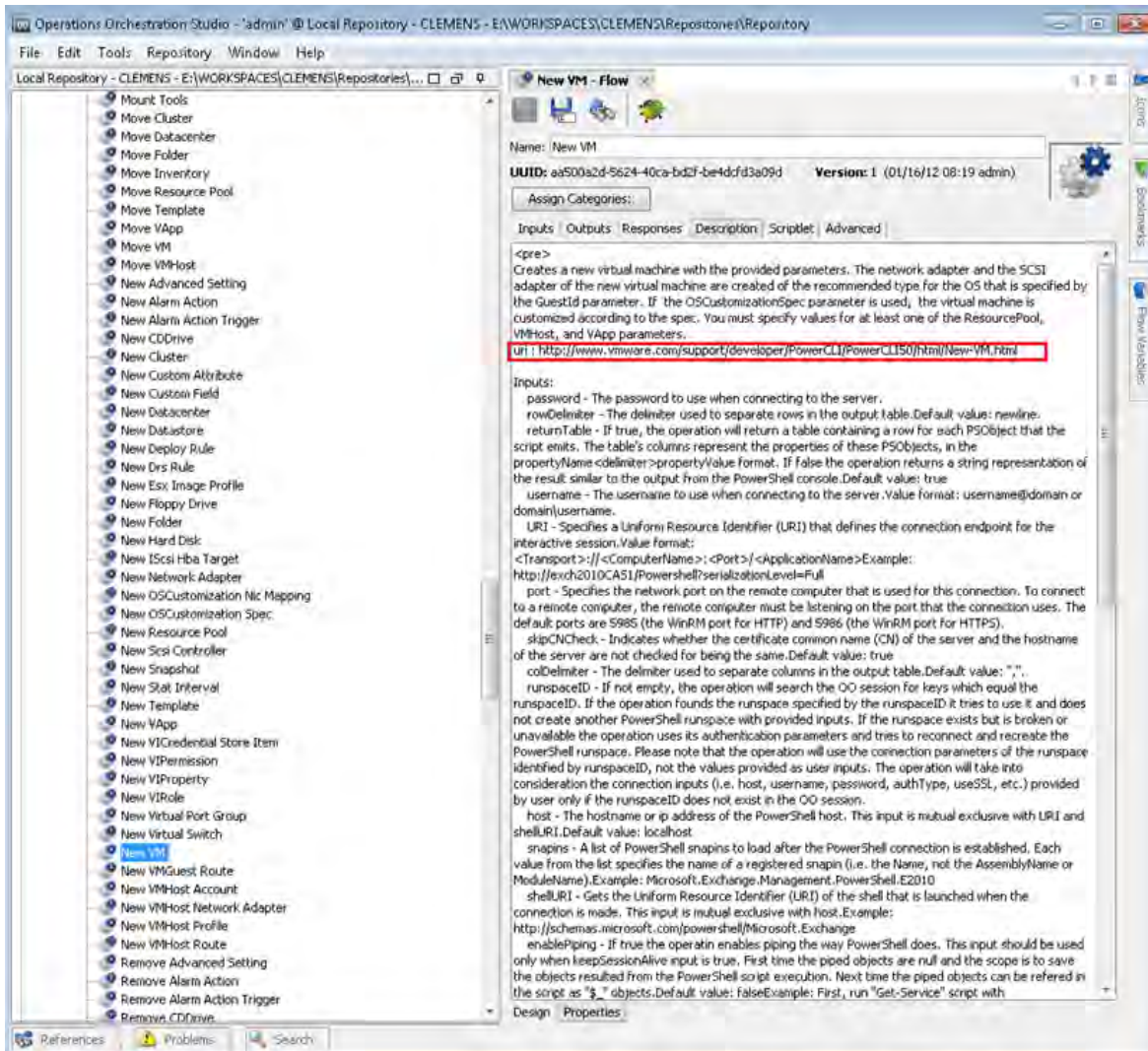
Figure 22: The description of the New VM flow

If the user opens the link in a browser he can observe the types of the cmdlet parameters. In our case, VMHost is of the same type as the name suggests.

Parameters

NAME	TYPE
VMHost	VMHost
Template	Template
AdvancedOption	AdvancedOption[]
AlternateGuestName	String
CD	SwitchParameter
Confirm	SwitchParameter
Datastore	StorageResource

Figure 23: The VMHost object: parameters

The problem is that VMHost is an object, not a built-in type which can be resolved as a string. Following the link provided for the VMHost type the user can find which cmdlets return VMHost objects as it can be observed below. In our case the type suggests that a cmdlet like Get-VMHost would return this kind of objects.

vSphere PowerCLI Reference

VMHost - Object

Property of
VMHostProfileInput, HostVMKernelVirtualNic, EsxCli, VirtualMachine, VMHostFirewallDefaultPolicy, Log, VMHostPatchResult, HostService, VMHostAu
HostVirtualNic, VMHostDiagnosticPartition, VirtualSwitch, VmHostModule, VMHostNetworkInfo, HostNic, IScsiHba, VMHostProfile, VMHostPatch, Hos
NicTeamingPolicy, VMHostProfileIncompliance, PciPassthroughDevice

Parameter to
Start-VMHost, Restart-VMHost, Get-VMHostHba, New-VMHostNetworkAdapter, Get-PassthroughDevice, Get-Datacenter, Get-Cluster, Remove-Dat
VMHostNtpServer, New-VMHostProfile, Get-VMHostService, New-VM, Import-VMHostProfile, Add-VmHostNtpServer, Suspend-VMHost, Stop-VMHos
VMHostFirewallDefaultPolicy, Get-VMHostStartPolicy, Get-VMHostProfile, Get-VMHostDiagnosticPartition, Get-VMHostNetworkAdapter, Set-VMHost
Datastore, Get-VMHostStorage, Get-VMHostNetwork, Get-VMHostFirmware, Get-VirtualSwitch, Test-VMHostProfileCompliance, Get-VirtualPortGrou
VMHostDisk, Get-EsxCli, Get-VMHostProfileRequiredInput, Get-VMHostAuthentication

Returned by
Start-VMHost, Add-VMHost, Set-VMHost, Suspend-VMHost, Stop-VMHost, Restart-VMHost, Get-VMHost, Move-VMHost, Get-HAPrimaryVMHost

Extends
VIContainer

Figure 24: The VMHost object : additional information

In the previous section, we explained how to execute in the same PowerShell session multiple OO flows generated with the PowerShell wizard. At this point we can imagine the following chain of cmdlets which need to be executed to create a new vm:

- **Connect-VIServer** – this must be executed before any VMWare cmdlet;
- **Get-VMHost** – we need the result of this cmdlet as parameter for the next cmdlet;
- **New-VM** – this cmdlet actually creates a new virtual machine.

New VM has other parameters beside VMHost which are not built-in, but we are going to explain how to solve the VMHost parameter, because the process is the same for the other parameters, too.

You need to execute three cmdlets in the same PowerShell session. You have generated flows for each of the cmdlets and executed them in the same session. The next step is to take the result of the **Get-VMHost cmdlet** and pass it to the **New-VM cmdlet?** Select from one of the following solutions:

Solution 1: Create a New PowerShell Script Step
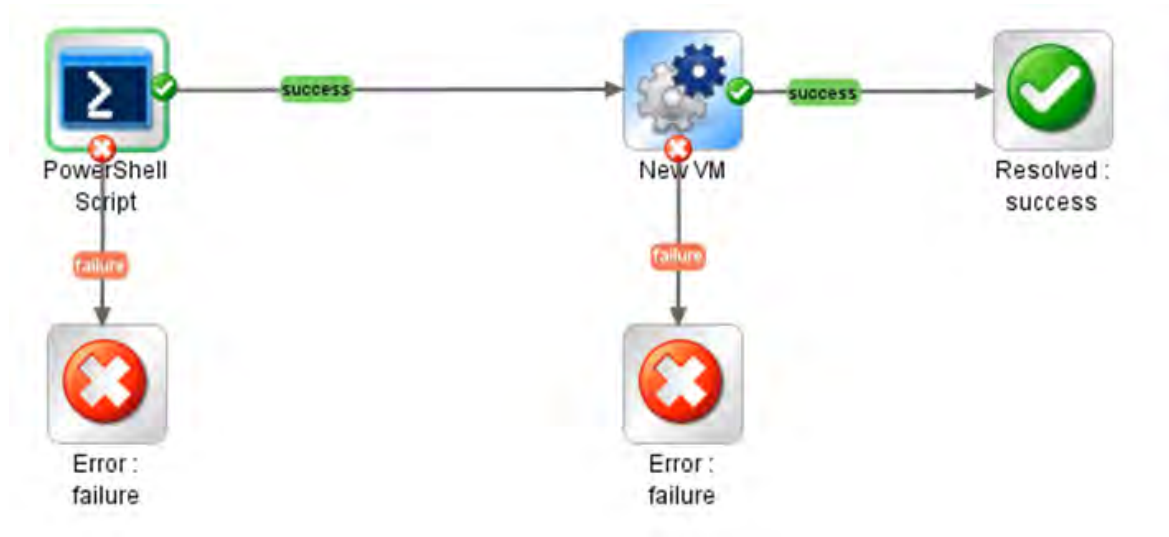
Figure 25: Add a PowerShell script step to the flow

Create a new PowerShell script in addition to the generated flow that you want to run. If you keep the session alive during the execution of the 2 PowerShell script steps, you can use the PowerShell script variables defined in the first step to pass them in the script of the second step or as parameters for the generated flow. In this case, you are not using the generated flows for **Connect-VIServer** and **Get-VMHost**; however, you need to write the script.

1. Execute the following script, then save the result of the **Get-VMHost** cmdlet in the PowerShell variable named **$vmHost**.

Use
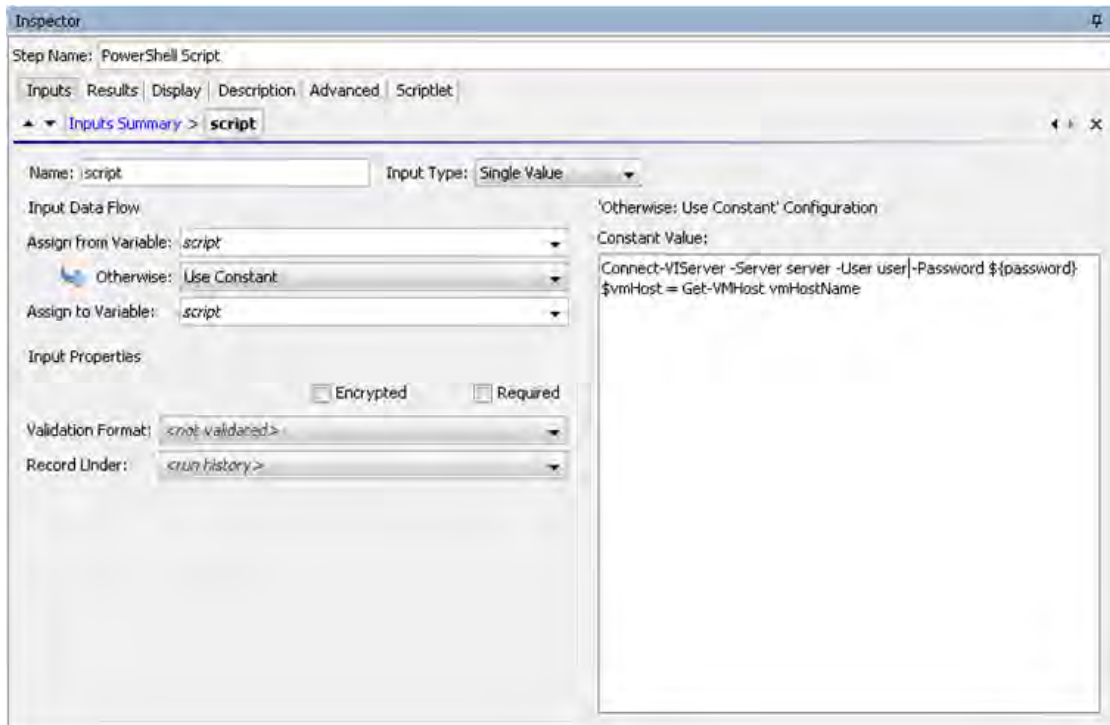
Figure 26: Use a variable to save the result of the step

2. The next step is to assign the value of the VMHost input from the **$vmHost** variable.

Use
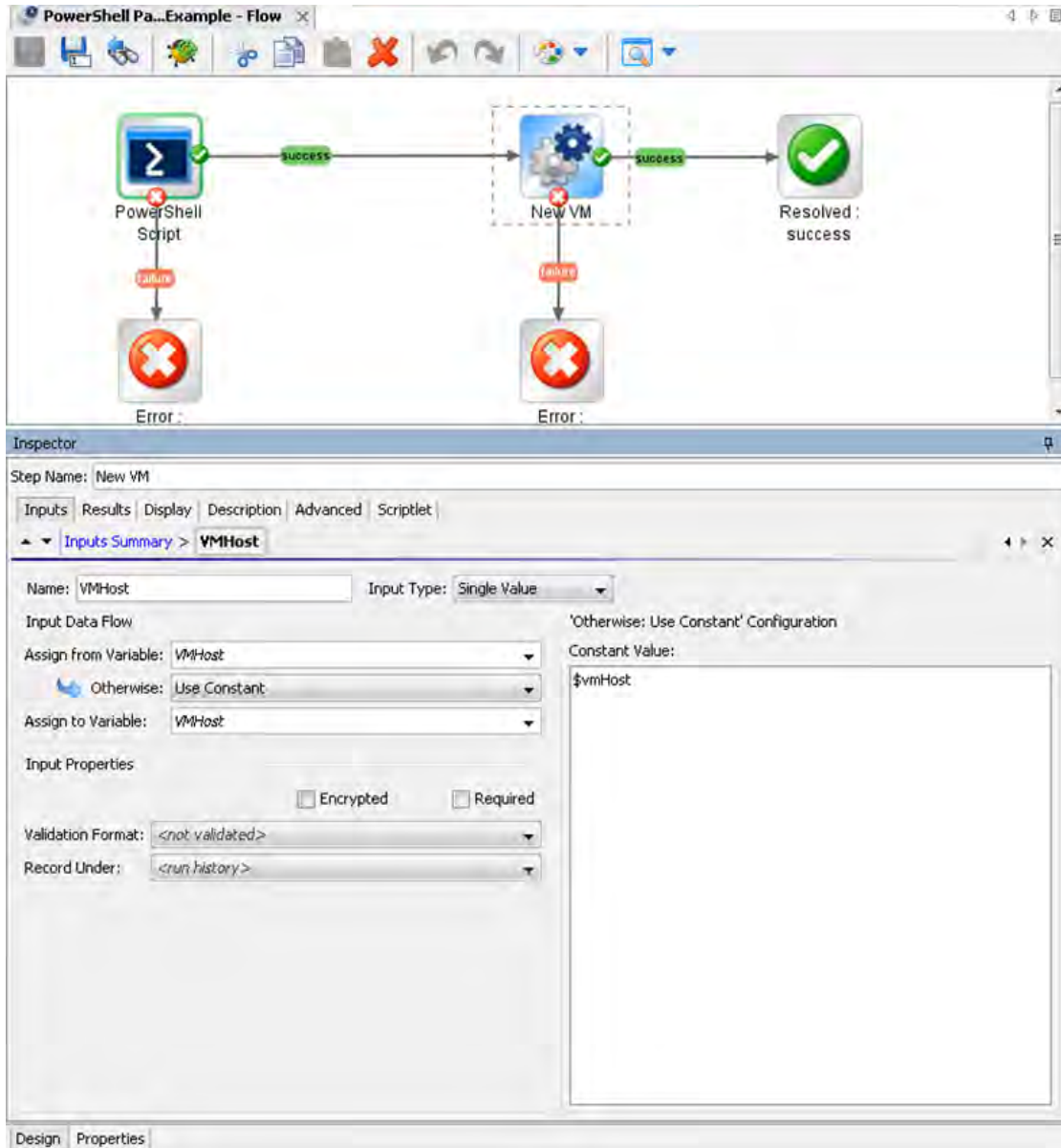
Figure 27: Assign the variable to the VMHost input of the flow

This way we managed to pass PowerShell cmdlets results between OO flows generated with the PowerShell wizard.

> **Note:** The **$var** refers to PowerShell variables and **${var}** refers to OO flow variables.

Solution 2: Run a PowerShell Script in the Generated Flow Context
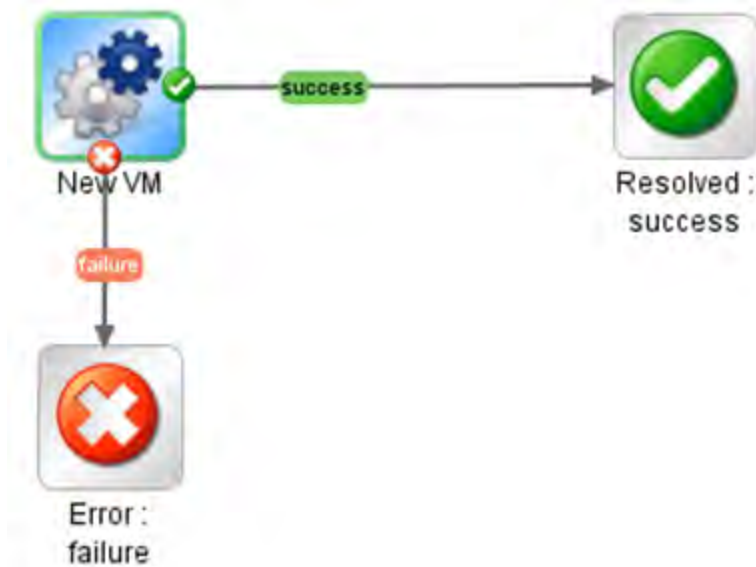
Figure 28: The NewVM Flow

The second solution is to execute the previous defined script in the context of the generated flow. The PowerShell script operation has two inputs which build the script that is going to be executed:

- **Script** – a PowerShell script to execute on target host;
- **Cmdlet** – the PowerShell cmdlet name. If the script input is not empty, than the PowerShell script defined by this input is going to be executed before the cmdlet. Although the script is executed in the same PowerShell runspace with no extra settings.

The solution is shown in the following flow:
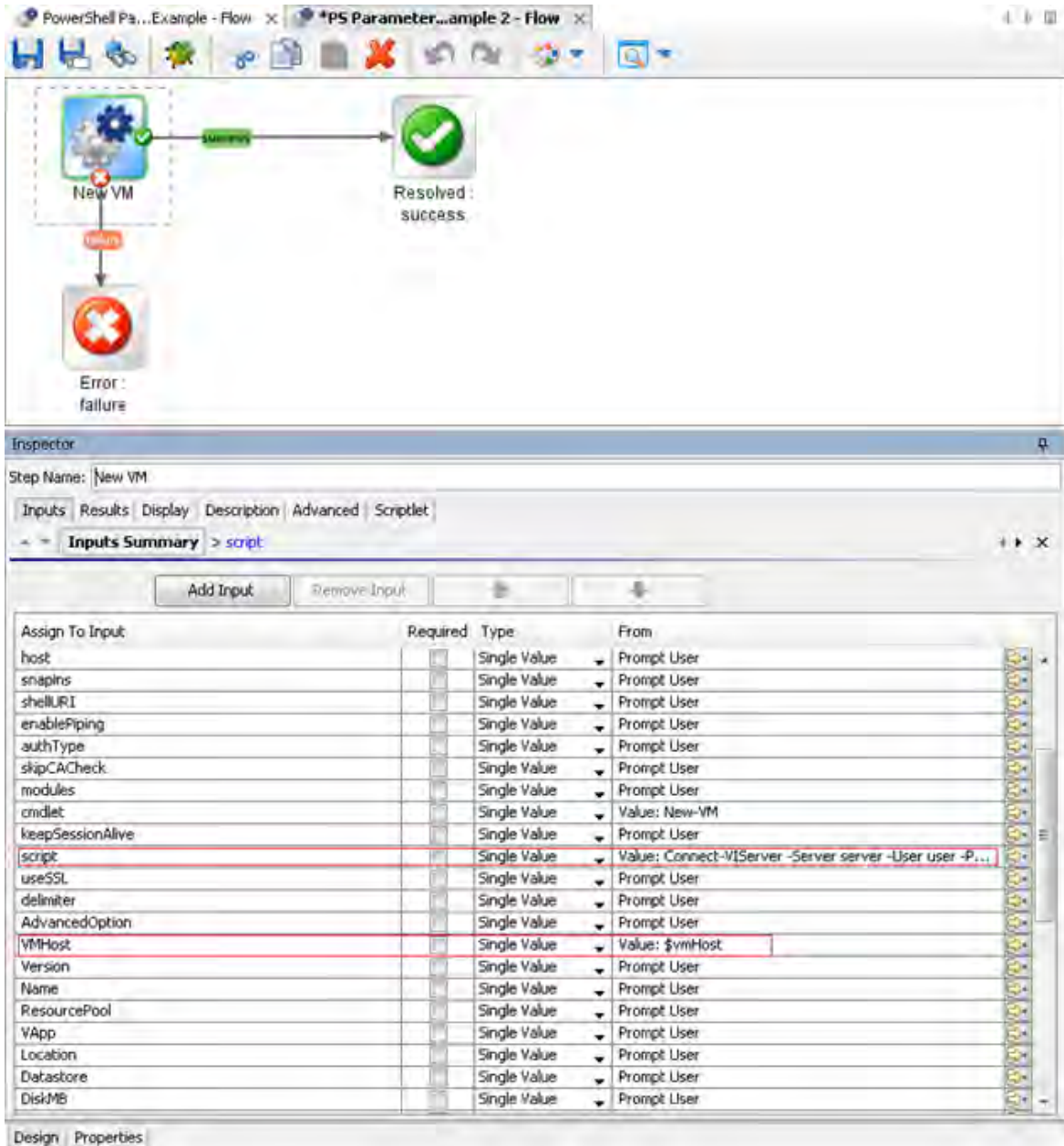
Figure 29: Add a script to the input of the flow

Solution 3: Use Generated Flows Only and Minimize the User Effort
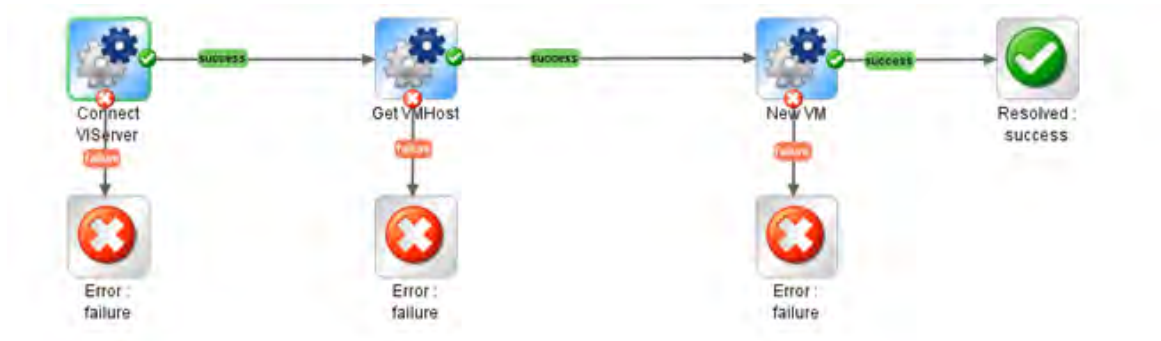
Figure 30: Using only generated flows

The result of each PowerShell cmdlet executed from a generated flow is saved in a PowerShell variable with the same name as the cmdlet, for example, for **Get-VMHost** the variable is $**GetVMHost**.

Without writing any PowerShell script, the user can execute **Get VMHost** generated flow and know that the result of this cmdlet is saved in the **$GetVMHost** variable. Pass the variable to the **VMHost** input of the **New VM** flow.

PowerShell Remoting

Overview

Enable PowerShell remoting, by running the following cmdlet: **Enable-PSRemoting**.

In workgroup environments, enable classic mode authentication for network logon:

1. Open **Local Security Policy** from the Control Panel and select **Administrative Tools**.

2. Navigate to **Local Policies \ Security Options**.

3. Double-click **Network Access: Sharing and Security Model for local accounts** and set it to **classic**.

Modify the WSMan trusted hosts setting, by adding the IP addresses of all remote clients to the list of trusted hosts. This can be done using one of the following commands:

- `Set-item wsman:localhost\client\trustedhosts -value *` (adds all computers as trusted hosts)

- `Set-item wsman:localhost\client\trustedhosts -value Computer` (only adds Computer to the trusted hosts)

- `Set-item wsman:localhost\client\trustedhosts -value *.domain.com` (adds all computers in the specified domain)

- `Set-item wsman:localhost\client\trustedhosts -value 10.10.10.1` (adds the remote computer with the IP address 10.10.10.1 to the trusted hosts list).

Enabling Remoting Using GPO (Group Policy Objects)

While remoting can be enabled manually using **Enable-PSRemoting**, it is recommended to use GPO management tools whenever it is possible. Use GPO to apply policies on a single host (for example, the target PowerShell host) or a group of servers.

Group Policy Configuration for a Single Host

To enable PowerShell remoting for a single host:

1. Open the Group Policy Management console. For example, **gpedit.msc**.

2. Go to **Local Computer Policy > Computer Configuration > Administrative Templates > Windows Components**.
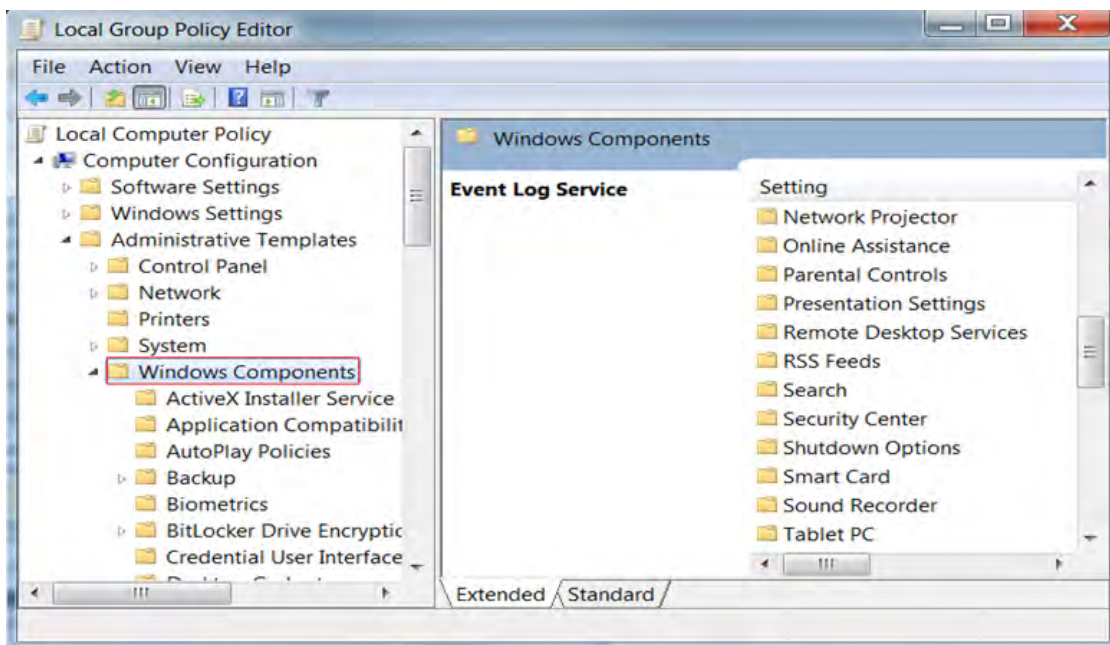
Figure 31: Group Policy Configuration : Windows components section

3. From the Windows Components listed in the right pane, we are interested in two of them. The first one is Windows Remote Management (WinRM) and the second one is Windows Remote Shell. The next step will refer to the first one, therefore browse the components and open Windows Remote Management (WinRM).
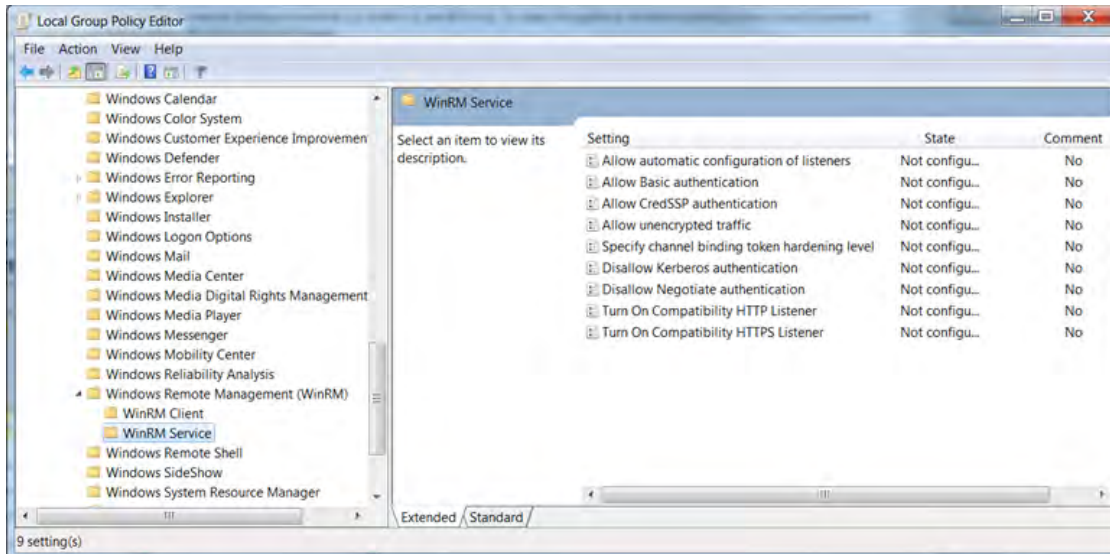
Figure 32: Group Policy Configuration: WinRM Service section

4. WinRM is the service that PowerShell uses for remote sessions. WinRM can be configured as client or service, depending on the role the host is going to have in a PowerShell connection (for example, request access to execute scripts on other hosts or allow other hosts to execute scripts on the current host). At this point, you can enable different authentication types, specify the trusted hosts, enable HTTP or HTTPS listeners, and so on.

5. There are some other default settings the user might want to change in a production environment. These settings can be found in Windows Remote Shell as shown below:
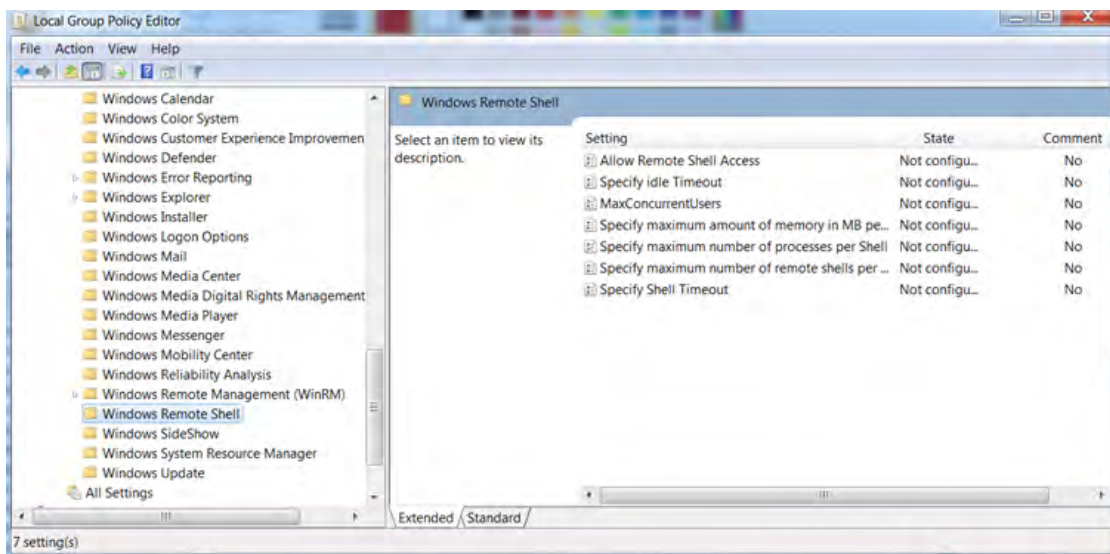
Figure 33: Group Policy Configuration : Windows Remote Shell section

6. You can specify the maximum number of remote shells per user (default is 2) or the maximum amount of memory in MB for shell (the default is 150).

7. After configuring the GPO, you might need to restart the computer in order to apply the policies or try to run the command **gpupdate**.

Group Policy Configuration for a Group of Servers

Sometimes the GPO policies must be applied on multiple server hosts and repeating the above steps on every server might not be the best solution. Therefore, you can create a new GPO policy, configure it and apply it on a list of servers.

1. Go to the domain controller or on a server where **gpmc.msc** is available and open it.

2. Right-click the **Group Policy Object** item as in the following image and choose **New**. Enter the name for the new GPO and select the policy to inherit from, and then go to the next step.
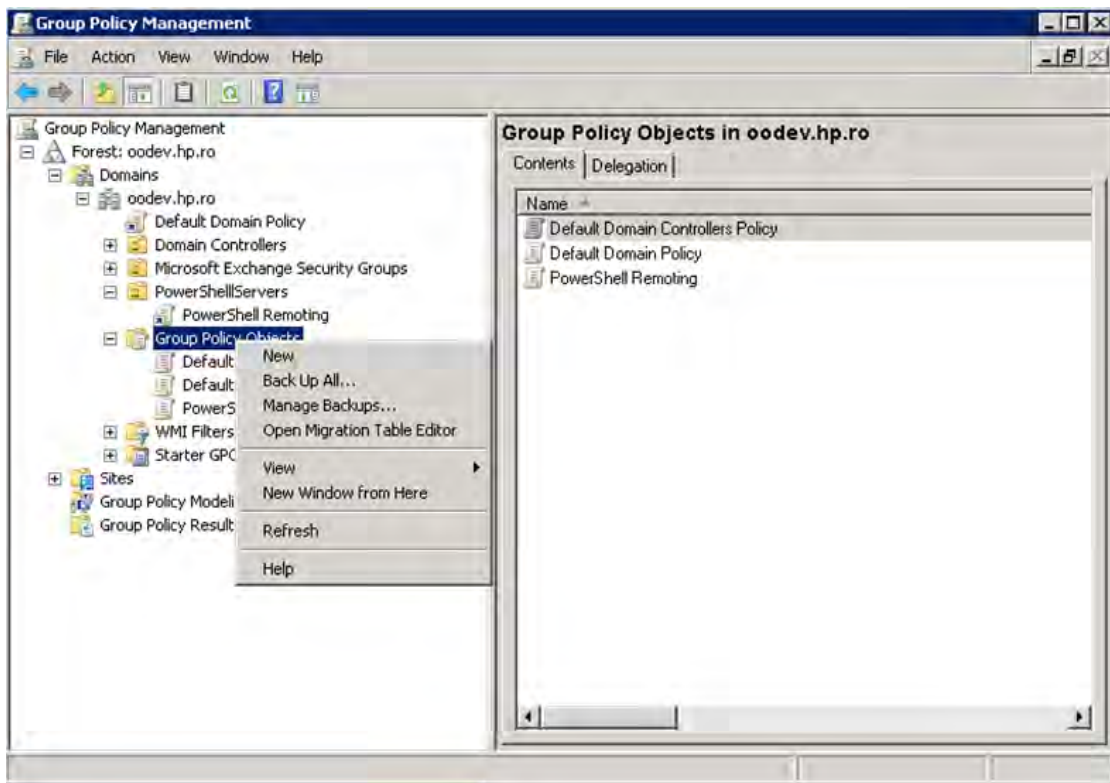
Figure 34: Using Group Policy Objects - step 1

3. Right-click the new GPO and select **Edit**.
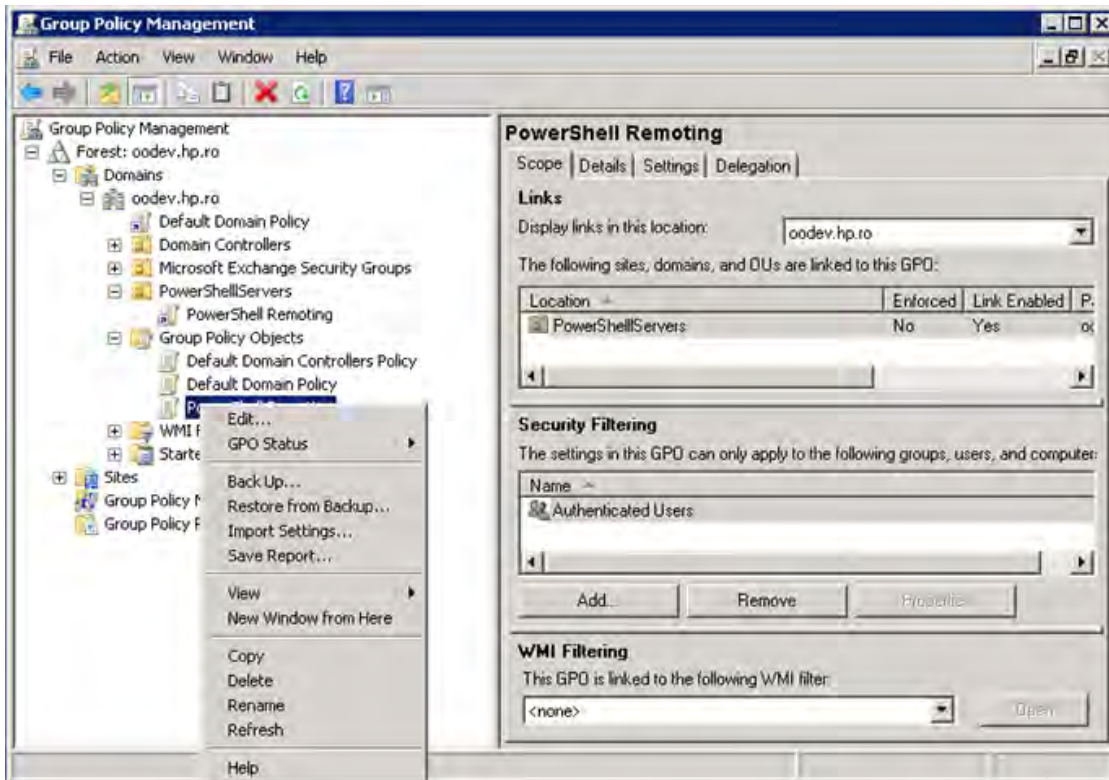
Figure 35: Using Group Policy Objects - step 2

4. Configure the GPO same way as for single hosts.

5. Now that there is a new GPO, you can apply it to a group of servers. The following example shows you how to link it on an existing OU from AD; however, this can be applied to other groups as well. The GPO interface displays the existing OUs from the domain controller AD.

   To link a GPO to an OU, go to that OU, right click it and select **Link an Existing GPO**.
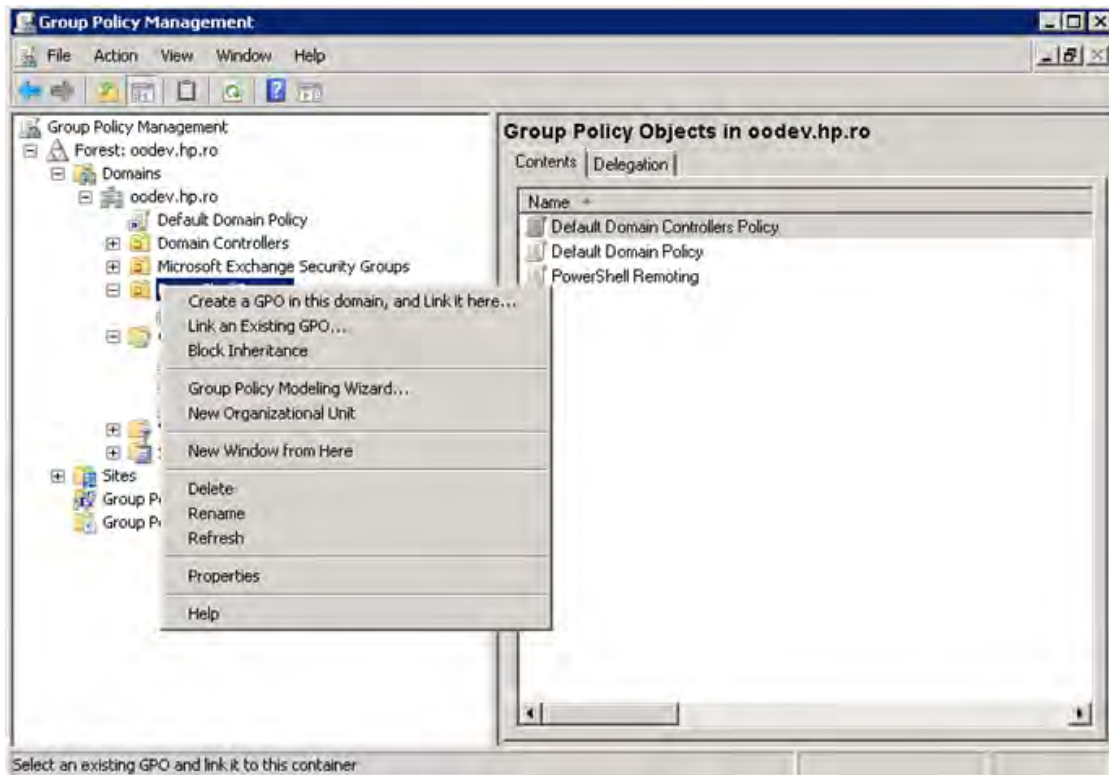
Figure 36: Using Group Policy Objects - step 3

The GPO settings will be applied to all servers contained in the selected OU. Although a GPO update might be required and also a reboot for the servers before the policies are actually applied.

> **Note:** Local policies overwrite domain policies.

Enabling Remoting for Non-Administrative Users

To establish a PSSession or run a command on a remote computer, you must have permission to use the session configurations on the remote computer.

By default, only members of the Administrators group on a computer have permission to use the default session configurations. Therefore, only members of the Administrators group can connect to the computer remotely.

To allow other users to connect to the local computer, give the user Execute permissions to the default session configurations on the local computer.

The following command opens a property sheet that lets you change the security descriptor of the default Microsoft.PowerShell session configuration on the local computer.

```
Set-PSSessionConfiguration Microsoft.Powershell –ShowSecurityDescriptorUI
```

Use
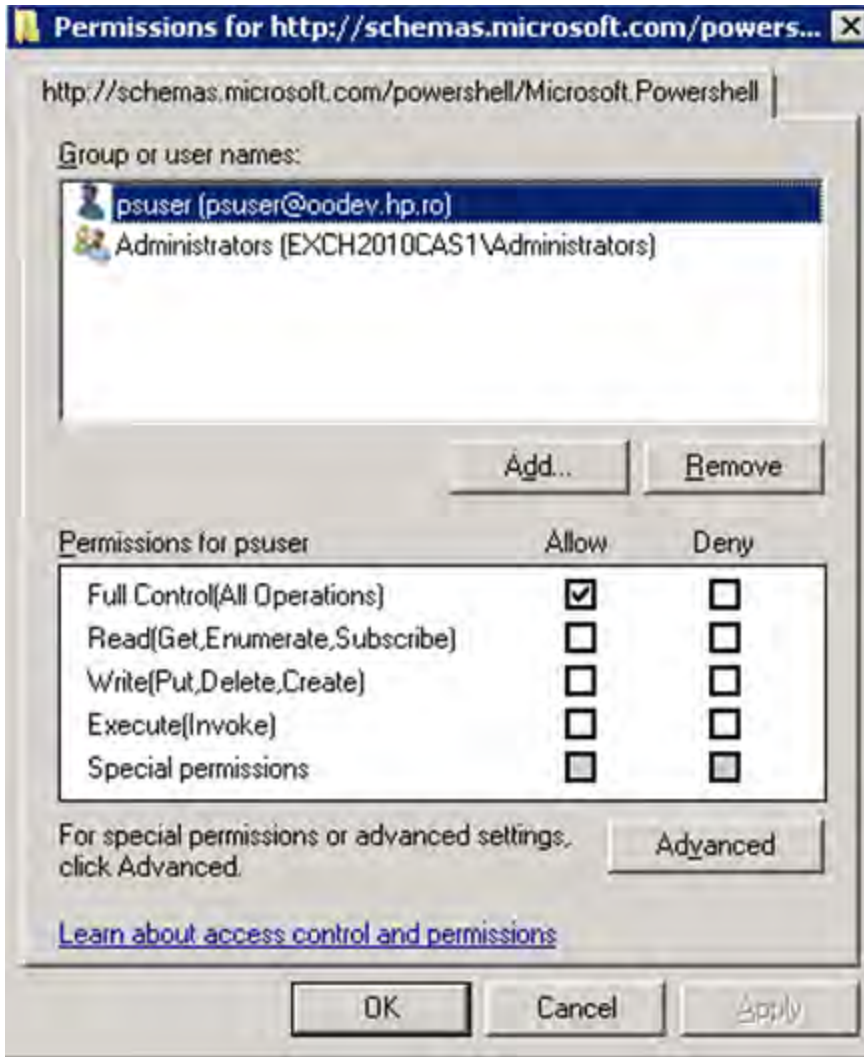
Figure 37: Adding a user to the PowerShell session configuration

Authentication Types

Basic

Client side steps:

- Allow unencrypted communication for the client, by running the following PowerShell command:

  ```
  set-item wsman:\localhost\client\AllowUnencrypted -value true
  ```

- Enable Basic authentication for the client, by running the following PowerShell command:

  ```
  set-item wsman:\localhost\client\auth\Basic -value true
  ```

Server side steps:

- Turn off encryption for the WinRM service, by running the following PowerShell command:

  ```
  set-item wsman:\localhost\service\AllowUnencrypted -value true
  ```

- Enable Basic authentication for the service, by running the following PowerShell command:

  ```
  set-item wsman:\localhost\service\auth\Basic -value true
  ```

  - The client and server can be in different domains.

  - When using Basic authentication, a local user account must be provided for authentication on the remote host.

  - Basic can be used when the destination is an IP address.

  - Basic can be used when the destination is one of the following: localhost, 127.0.0.1, [::1].

  - The cluster name, as well as the hostnames of the cluster nodes can be used for the destination host.

CredSSP

Client side steps:

- Enable CredSSP authentication for the client, by running the command:

  ```
  Enable-WSManCredSSP -Role Client -DelegateComputer WSMAN/*.
  ```

- Allow delegating fresh credentials by performing the following steps.

  a. Open gpedit.msc.

  b. Go to **Computer Configuration** > **Administrative Templates** > **System** > **Credentials Delegation**.

  c. Enable **Allow Delegating Fresh Credentials** and add the wsman hosts to the server list.

  d. Run **gpupdate /force** from command line to force policy update.

Server side steps:

- Enable CredSSP authentication, by running the following PowerShell command:

  ```
  Enable-WSManCredSSP -Role Server
  ```

- Create a new https listener by using the following commad:

  ```
  winrm create winrm/config/Listener?Address=*+Transport=HTTPS.
  ```

Domain Controller side steps:

Use

- If the NETWORK SERVICE doesn't have "Validated write to service principal name", do one of the following:

  - Try running the following command:

    ```
    dsacls "CN=AdminSDHolder,CN=System,DC=domain,DC=com" /G "Sn-1-5-20:WS;Validated write to service principal name"
    ```

  - Open ADUC, go to **Computers** > **DC object** > **Security**, select **Network Service** and give it **Validated write to SPN**.

Default

When **Default** authentication is used, the following situations can occur:

- Kerberos is the method of authentication used if the client is in the same domain as the destination host, and the value specified for that host is not one of the following: localhost, 127.0.0.1, [::1].

- Negotiate is the method of authentication used if the client is not in the same domain as the destination host, or the value specified for that host is one of the following: localhost, 127.0.0.1, [::1].

Digest

Digest authentication is not supported for remote connections. It cannot be configured for the WinRM server component.

Kerberos

Client side steps:

- Enable Kerberos authentication for the client, by running the following PowerShell command:

  ```
  set-item wsman:\localhost\client\auth\Kerberos -value true
  ```

Server side steps:

- Enable Kerberos authentication for the service, by running the following PowerShell command:

  ```
  set-item wsman:\localhost\service\auth\Kerberos -value true
  ```

- The client and server must be in the same domain.
- Either a local or a domain user account can be provided for authentication on the server host.
- Kerberos cannot be used when the destination is an IP address.
- Kerberos cannot be used when the destination is one of the following: localhost, 127.0.0.1, [::1].

- The cluster name cannot be used to specify the host. Only the hostnames of the cluster nodes can be used for the destination host.

Negotiate

Client side steps:

- Enable **Negotiate** authentication for the client, by running the following PowerShell command:

```
set-item wsman:\localhost\client\auth\Negotiate -value true
```

Server side steps:

- Enable **Negotiate** authentication for the service, by running the following PowerShell command:

```
set-item wsman:\localhost\service\auth\Negotiate -value true
```

- The client and server can be in different domains.
- Either a local or a domain user account can be provided for authentication on the server host. Local accounts can only be provided when connecting to the localhost.
- **Negotiate** can be used when the destination is an IP address.
- **Negotiate** can be used when the destination is one of the following: localhost, 127.0.0.1, [::1].
- The cluster name, as well as the host names of the cluster nodes can be used for the destination host.

NegotiateWithImplicitCredential

- When using **NegotiateWithImplicitCredentials**, no credentials should be provided. The current logged-on user account will be used for authentication. This can either be a local or a domain user account.
- **NegotiateWithImplicitCredential** can only be used when the destination is one of the following: localhost, 127.0.0.1, [::1].

Troubleshooting

This section provides troubleshooting procedures that you can use to solve problems you may encounter while using the wizard. It also includes an error message you may receive while using the integration and offers descriptions and possible fixes for the error.

Could not connect to the host

The possible reasons are:

- The user credentials are not correct.

- The user does not have permission to execute PowerShell scripts on the target host. Make sure the user has admin rights or see the section Enable Remoting for Non-Administrative Users.

- Authentication problems (most common). See "Run a PowerShell Script on a Remote Host" in "Using the PowerShell Wizard" on page 748.

- The WinRM service is stopped on the target host.

- WinRM default ports (5985 and 5986) were changed. You need to provide the correct port in the connection page of the wizard.

The wizard fails to load modules on a x64 localhost.

Some modules cannot be loaded using the wizard, but they are loaded from the PowerShell console. By default, the wizard runs in a x32 process (depends on the OO jre), which ends up calling x32 PowerShell. The x32 version of PowerShell cannot load some modules (for example, FailoverClusters); therefore, the wizard fails.

In order to fix this, do not leave the host input empty. Instead, you need to provide the "localhost". This way, the wizard will try to authenticate the localhost like any other remote host. Note that remoting rules should be satisfied for localhost in this case. If user is left empty, the wizard will connect using the **NegotiateWithImplicitCredential**. Otherwise, you need to provide user credentials and authentication type as for any other remote host.

The user has exceeded the maximum allowed number of remote shells

The user has exceeded the maximum allowed number of remote shells. This error would probably occur if the user stresses the wizard with too many "back and next" actions without running the wizard from start to end. See the "Enable Remoting Using GPO (Group Policy Objects)" section in "PowerShell Remoting" on page 778 in order to increase the allowed number of remote shells per user.

Use

Figure 38: Connection Error - number of remote shells has been exceeded.