HP Operations Orchestration Software

Software Version: 9.00.04

Microsoft Opalis Integration Guide

Document Release Date: January 2011

Software Release Date: January 2011

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

Trademark Notices

For information on open-source and third-party software acknowledgements, see in the documentation set for this release, Open-Source and Third-Party Software Acknowledgements (3rdPartyOpenNotices.pdf).

On the Web: Finding OO support and documentation

There are two Web sites where you can find support and documentation, including updates to OO Help systems, guides, and tutorials:

- The OO Support site
- HP Live Network

Support

Documentation enhancements are a continual project at Hewlett-Packard Software. You can obtain or update the HP OO documentation set and tutorials at any time from the HP Software Product Manuals Web site. You will need an HP Passport to log in to the Web site.

To obtain HP OO documentation and tutorials

1. Go to the HP Software Product Manuals Web site (*http://support.openview.hp.com/selfsolve/manuals*).
2. Log in with your HP Passport user name and password.

   OR

   If you do not have an HP Passport, click **New users – please register** to create an HP Passport, then return to this page and log in.

   If you need help getting an HP Passport, see your HP OO contact.
3. In the **Product** list box, scroll down to and select **Operations Orchestration**.
4. In the **Product Version** list, click the version of the manuals that you're interested in.
5. In the **Operating System** list, click the relevant operating system.
6. Click the **Search** button.
7. In the **Results** list, click the link for the file that you want.

HP Live Network

For support information, including patches, troubleshooting aids, support contract management, product manuals and more, visit the following site: *https://www.www2.hp.com/*.

This is the **HP Live Network** Web page. To sign in:

1. Click **Login**.
2. On the **HP Passport sign-in** page, enter your HP Passport user ID and password and then click **Sign-in**.
3. If you do not already have an HP Passport account, do the following:
   a. On the **HP Passport sign-in** page, click **New user registration**.
   b. On the **HP Passport new user registration** page, enter the required information and then click **Continue**.
   c. On the confirmation page that opens, check your information and then click **Register**.
   d. On the **Terms of Service** page, read the Terms of use and legal restrictions, select the **Agree** button, and then click **Submit**.
4. On the **HP Live Network** page, click **Operations Orchestration Community.**

The Operations Orchestration Community page contains links to announcements, discussions, downloads, documentation, help, and support.

Note: Contact your OO contact if you have any difficulties with this process.

In OO: How to find Help, PDFs, and tutorials

The HP Operations Orchestration software (HP OO) documentation set is made up of the following:

- Help for Central

  Central Help provides information to the following:

  - Finding and running flows
  - For HP OO administrators, configuring the functioning of HP OO
  - Generating and viewing the information available from the outcomes of flow runs

  The Central Help system is also available as a PDF document in the HP OO home directory, in the \Central\docs subdirectory.

- Help for Studio

  Studio Help instructs flow authors at varying levels of programming ability.

  The Studio Help system is also available as a PDF document in the HP OO home directory, in the \Studio\docs subdirectory.

- Animated tutorials for Central and Studio

  HP OO tutorials can each be completed in less than half an hour and provide basic instruction on the following:

  - In Central, finding, running, and viewing information from flows
  - In Studio, modifying flows

  The tutorials are available in the Central and Studio subdirectories of the HP OO home directory.

- Self-documentation for operations and flows in the Accelerator Packs and ITIL folders

  Self-documentation is available in the descriptions of the operations and steps that are included in the flows.

Table of Contents

Overview of Microsoft Opalis integration

With this integration, you can use and build HP Operations Orchestration (OO) flows that are integrated with Microsoft Opalis.

The Opalis integration integrates with OO to run and manage policies (workflows) defined on the Opalis management server. To use this integration successfully, you should have knowledge of the Opalis technology.

This document will explain how this integration has been implemented and how the operations included communicate between OO and Opalis.

Use cases and scenarios

The following are the major use cases for the Microsoft Opalis integration, and the operations that you can use to implement them.

1. Get information about Opalis policies:
   - Get All Policies
   - Get Policy Details
2. Get information about Opalis policy requests:
   - Get Policy Request Details
   - Get Policy Requests
3. Start Opalis policies:
   - Start Policy
4. Stop policy requests:
   - Stop All Policy Requests
   - Stop All Policy Requests For Policy
   - Stop Policy Request

Installation and configuration instructions

No special installation and configuration instructions are required for the Microsoft Opalis integration.

Versions

Operations Orchestration Version	Microsoft Opalis Version
9.00.04	6.2.2

Architecture

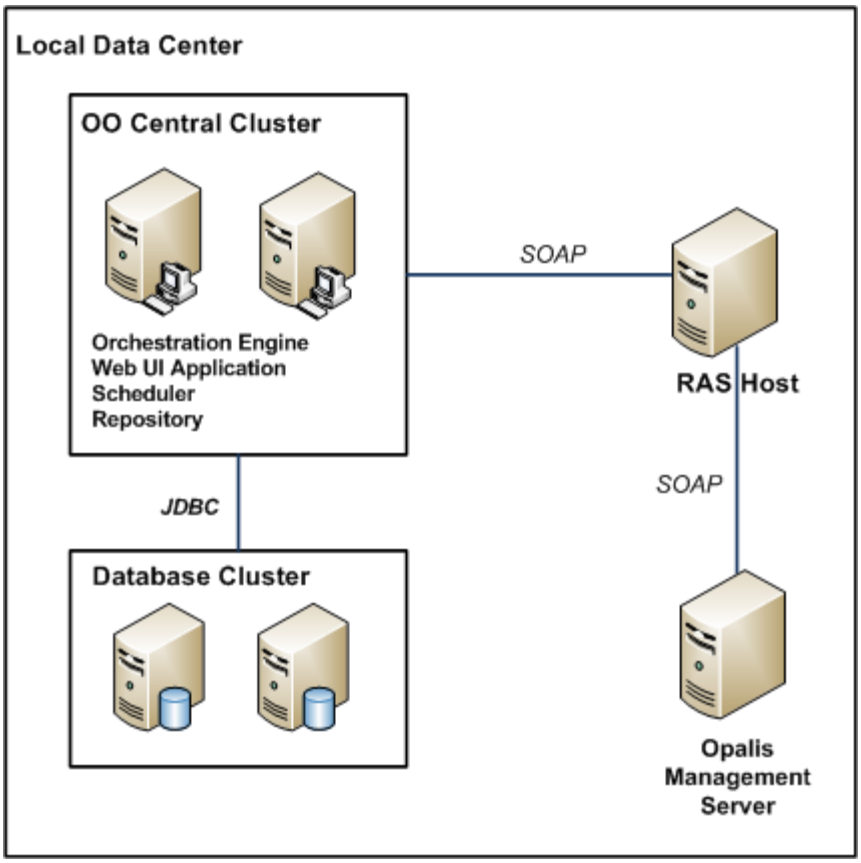

Figure 1 – Microsoft Opalis architecture

Opalis integration operation infrastructure

The EC2 integration includes the following operations in the OO Studio Library/Integrations/ Microsoft/Opalis/ folder.

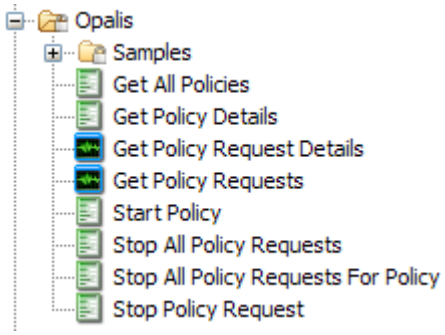

Figure 2 – Microsoft Opalis integration operation infrastructure

Common inputs in the integration

OO flows and operations use inputs to specify how they obtain the data that they need and when the data is obtained. The following inputs are used consistently throughout the Opalis integration's operations.

host

The host on which the Opalis management server runs. This can be a host name or an IP address. If you use a name, it must be able to be resolved to an IP address by a DNS server specified in your networking configuration. The **nslookup** command, available in Windows and most Unix operating systems, can be used to verify that the given host name can be resolved. This input is required.

port

The port on which the Opalis management server listens for Web service requests. This is normally the same port on which the Opalis operator console listens. When installing Opalis, the default port for HTTP is 5314 and the default port for HTTPS is 8443. If you do not specify a value for this input, it defaults to **5314**.

protocol

The protocol used to connect to the Opalis management service. The valid values are **http** and **https**. The default value is **http**.

username

The username used to connect to Opalis.

password

The password used to connect to Opalis.

Operation specifics

This section describes the Opalis integration's operations, including any operation- or flow-specific inputs. The sample flows in the OO Library/Integrations/Microsoft/Opalis/Samples/ folder use Opalis operations to perform some of the most common tasks that need to be automated when using Opalis, such as retrieving information about the policy requests for a given policy and then retrieving data on a specific policy. Each of these sample flows has a description that describes in detail what it does. You can use these flows as they are or as templates for new operations.

Get All Policies

The **Get All Policies** operation retrieves basic information about all policies defined in the Opalis management server. The basic information that is retrieved includes the policy IDs, descriptions, and names.

All of the operation's inputs are described in *Common inputs in the integration*.

The operation returns the following:

returnResult

When the operation is successful, this result contains a brief message stating that it was successful. In the event of a failure, this result contains details about the error.

jsPolicies

This is a JavaScript Object string that is an array of policy information objects. The folder Library/Utility Operations/Containers/JavaScript Objects/ contains operations to manipulate these objects, and the description of that folder has a more thorough discussion of the JavaScript Object format.

Each object in the **jsPolicies** array contains the following pieces of information about a policy:

- **policyId**

  The internal ID of the Opalis policy (its GUID). This ID is used as an input to other operations, such as **Get Policy Derails** and **Start Policy**. This is also the same ID that is used with Opalis' OIS5StartPolicy command line utility.

- **policyName**

  The name of the Opalis policy.

- **policyDescription**

  The description of the Opalis policy. This is null if no description has been entered.

For example, if there are two policies in Opalis, **MyPolicy** and **MyCustomPolicy**, then **jsPolicies** might contain:

```
[ { "policyName" : "My Custom Policy",
    "policyDescription" : "This is my custom policy",
    "policyId" : "{42FA4AB2-5B09-4A06-8416-B9C0D18619C7}" },
  { "policyName" : "My Policy",
    "policyDescription" : null,
    "policyId" : "{0D457CE5-389C-4EFA-A7DE-7F9419448C49}" }]
```

(Extra whitespace has been inserted for readability.)

See the **Get Policy Info** flow in the Library/Integrations/Microsoft/Opalis/Samples/ folder for an example of how to process this array of objects.

Note: The underlying Opalis Web services actually require a separate initial call to authenticate against Opalis, which returns an authentication token that is to be supplied in subsequent Web service calls. This authentication token is stored in the local context with a key whose name contains **OpalisAuthenticationToken** and the host and username, and this value is automatically supplied to any Opalis calls made within the same flow to avoid repeated authentication calls.

Get Policy Details

The **Get Policy Details** operation retrieves detailed information about a specific policy in the Opalis management server. This information includes the basic information about the policy (its ID, description, and name) as well as the details of any custom start parameters it may have.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

policyId

The internal policy ID (GUID). You can obtain this value with the **Get All Policies** operation. It is also the same ID that is used with Opalis' **OIS5StartPolicy** command line utility. This value is normally a long set of hexadecimal values surrounded by braces. For example, `{42FA4AB2-5B09-4A06-8416-B9C0D18619C7}`.

Note: If the operation fails, an error has occurred in retrieving the policy information. The most common errors are those due to an invalid combination of values in the inputs to this operation or

other errors communicating with the Opalis management server. This operation will also return a failure if the specified **policyId** value is not found in Opalis. The Opalis management server may also throw an internal error if the **policyId** value specified is not in the GUID format that it expects. In the event of a failure, the **returnResult** output normally has more details about the nature of the failure.

The operation returns the following:

returnResult

When the operation is successful, this output contains a brief message stating that it was successful. In the event of a failure, this output contains details about the error.

policyId

The internal policy ID (GUID). This value can be obtained using the **Get All Policies** operation. It is also the same ID that is used with Opalis' **OIS5StartPolicy** command line utility. This value is normally a long set of hexadecimal values surrounded by braces. For example, `{42FA4AB2-5B09-4A06-8416-B9C0D18619C7}`.

policyName

The name of the Opalis policy.

policyDescription

The description of the Opalis policy. This is null if there is no description.

jsParameters

This is a JavaScript Object string that is an array of parameter objects. The folder /Library/Utility Operations/Containers/JavaScript Objects/ contains operations to manipulate these objects. The description of this folder has a more thorough discussion of the JavaScript Object format.

Each object in the **jsParameters** array contains the following pieces of information about a parameter:

- **name**

  The name of the parameter.

- **description**

  The description of the parameter. This will be null if no description has been entered.

- **type**

  The type of the parameter. All parameters are passed to Opalis as strings (and all values in OO are stored as strings). Conversion of strings to this type will be performed within Opalis.

For example, a policy with a single parameter named **Message** may have the following value:

```
[ { "description" : null,
    "name" : "Message",
    "type" : "String" } ]
```

(Extra whitespace is inserted for readability.)

See the **Get Policy Info** flow in the Library/Integrations/Microsoft/Opalis/Samples/ folder for an example of how to process this array of objects.

Note: The underlying Opalis Web services require a separate initial call to authenticate against Opalis. This call returns an authentication token that is to be supplied in subsequent Web service calls. This authentication token is stored in the local context with a key whose name contains **OpalisAuthenticationToken** and the host and username, and this value is automatically supplied to any Opalis calls made within the same flow to avoid repeated authentication calls.

Get Policy Details

The **Get Policy Details** operation retrieves information about a specific policy request (a policy that has been requested to start). When you use the **Start Policy** operation to trigger a policy to start, a policy request ID is returned. You can then use this ID in the **policyRequestId** input of the **Get Policy Details** operation to inquire on the policy request's status, including the instances of the request.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

policyId

The ID of the policy request. This value is returned from the **Start Policy** operation. It is normally an integer value, for example 158.

Notes:

- If there are no policies defined in Opalis, this operation succeeds.
- If the operation fails, an error has occurred in retrieving the policy request. The most common errors are those due to an invalid combination of values in the inputs to this operation or other errors communicating with the Opalis management server. This operation will also return a failure if the specified **policyRequestId** value is not found in Opalis. The Opalis management server may also throw an internal error if the **policyRequestId** value is not a valid integer. In the event of a failure, the **returnResult** output normally has more details about the nature of the failure.

The operation returns the following:

returnResult

When the operation is successful, this output contains a brief message stating that it was successful. In the event of a failure, this output contains details about the error.

active

This is a boolean value, **true** or **false**, indicating whether the policy request is active.

jsInstances

This is a JavaScript Object string that is an array of instance objects. Each policy request may result in multiple instances, which is something you can see in the Opalis operator console. The folder /Library/Utility Operations/Containers/JavaScript Objects/ contains operations to manipulate these objects. The description of this folder has a more thorough discussion of the JavaScript Object format.

Opalis populates the **jsInstances** array if the policy request is active. Each object in the array contains the following pieces of information about an instance:

- **timeStarted**

  The time that the instance was started, in ISO 8601 format. For example, `2010-07-16T09:10:10.973-07:00`.

- **timeEnded**

  The time that the instance was ended, in ISO 8601 format. If **timeStarted** is populated but this value is null, it indicates that the policy is active.

For example, a policy with a single running instance may have the following value:

```
[ { "status" : null,
    "timeStarted" : 2010-07-16T09:10:10.973-07:00,
    "timeEnded" : null } ]
```

(Extra whitespace is inserted for readability.)

Note: The underlying Opalis Web services require a separate initial call to authenticate against Opalis. This call returns an authentication token that is to be supplied in subsequent Web service calls. This authentication token is stored in the local context with a key whose name contains **OpalisAuthenticationToken** and the host and username, and this value is automatically supplied to any Opalis calls made within the same flow to avoid repeated authentication calls.

Get Policy Requests

The **Get Policy Requests** operation retrieves information about all of the policy requests that are available for the given policy. A policy request is a record of the triggering of a policy to start, for instance using the **Start Policy** operation.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

policyId

The internal policy ID (GUID) of the policy whose requests are to be retrieved. This value can be obtained using the **Get All Policies** operation. It is also the same ID that is used with Opalis' **OIS5StartPolicy** command line utility. This value is normally a long set of hexadecimal values surrounded by braces. For example:

```
{42FA4AB2-5B09-4A06-8416-B9C0D18619C7}
```

Notes:

- If there are no policies defined in Opalis, this operation succeeds.
- If the operation fails, an error has occurred in retrieving the policy request. The most common errors are those due to an invalid combination of values in the inputs to this operation or other errors communicating with the Opalis management server. The Opalis management server may also throw an internal error if the **policyId** value specified is not in the GUID format that it expects. In the event of a failure, the **returnResult** output normally has more details about the nature of the failure.

The operation returns the following:

returnResult

When the operation is successful, this contains a brief message stating that it was successful. In the event of a failure, this output contains details about the error.

jsPolicyRequests

This is a JavaScript Object string that is an array of instance objects. The folder /Library/Utility Operations/Containers/JavaScript Objects/ contains operations to manipulate these objects, and the description of that folder has a more thorough discussion of the JavaScript Object format.

The **jsPolicyRequests** array contains an entry for each policy request that is present for the policy. Each object in the array contains the following pieces of information about a request:

- **active**

  This is a boolean value, **true** or **false**, indicating whether the policy request is active.

- **policyRequestId**

  The ID of the policy request. This value is returned from the **Start Policy** operation. This value is normally an integer value, such as 158.

For example, a policy with a two active requests may have the following:

```
value:
  [ { "active" : true,
      "policyRequestId" : 24 },
    { "active" : true,
      "policyRequestId" : 25 } ]
```

(Extra whitespace is inserted for readability.)

Note: The underlying Opalis Web services require a separate initial call to authenticate against Opalis, which returns an authentication token that is supplied in subsequent Web service calls. This authentication token is stored in the local context with a key whose name contains **OpalisAuthenticationToken** and the host and username. This value is automatically supplied to any Opalis calls made within the same flow to avoid repeated authentication calls.

Start Policy

The **Start Policy** operation triggers a policy to start on the Opalis management server. This operation can start both fixed policies (those without input parameters) and policies with custom starts (those having input parameters). When the policy has been triggered, this operation returns without waiting for the policy to complete. Note that Opalis policies that contain Monitor objects may continue to run indefinitely. When starting a custom policy, the operation will look for any extra inputs that have been added to the step and will forward them on as values of the inputs with the same names in Opalis.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

policyId

The internal policy ID (GUID) of the policy to start. This value can be obtained using the **Get All Policies** operation. It is also the same ID that is used with Opalis' **OIS5StartPolicy** command line utility. This value is normally a long set of hexadecimal values surrounded by braces. For example:

```
{42FA4AB2-5B09-4A06-8416-B9C0D18619C7}
```

Notes:

- When you add this operation to a step in a flow, you can add additional inputs to it to populate any custom parameters that the policy has. The step inputs should have the same names as the custom policy parameters. Alternatively, you can also prefix the step input with "Opalis_" to avoid overlap with other inputs. For example, if the policy accepts a password parameter and you don't want to give it the Opalis password, you can add an input with the name **Opalis_password**. All policy parameter values are sent as strings and are converted to the desired types by Opalis.
- If the operation fails, an error has occurred in retrieving the policy request. The most common errors are those due to an invalid combination of values in the inputs to this operation or other errors communicating with the Opalis management server. The Opalis management server may also throw an internal error if the **policyId** value specified is not in the GUID format that it expects. The Opalis management server may also throw an error if a policy parameter cannot be converted from the value sent; for example, if you supply a non-numeric value to a parameter that is a numeric type. In the event of a failure, the **returnResult** output normally has more details about the nature of the failure.

The operation returns the following:

returnResult

When the operation is successful, this contains a brief message stating that it was successful. In the event of a failure, this output contains details about the error.

policyRequestId

The ID of the policy request that was started. This value can be used to retrieve the information about the policy request using the **Get Policy Request** operation or to stop the policy using the **Stop Policy Request** operation. This value is normally an integer value, such as 158.

Note: The underlying Opalis Web services require a separate initial call to authenticate against Opalis, which returns an authentication token that is supplied in subsequent Web service calls. This authentication token is stored in the local context with a key whose name contains **OpalisAuthenticationToken** and the host and username. This value is automatically supplied to any Opalis calls made within the same flow to avoid repeated authentication calls.

Stop All Policy Requests

The **Stop All Policy Requests** operation triggers the Opalis management server to stop all active policy requests for all policies. This operation returns without waiting for the policy requests to be stopped.

All of the operation's inputs are described in *Common inputs in the integration*.

Notes:

- If there are no active policy requests, this operation succeeds.
- If the operation fails, an error has occurred in retrieving the policy request. The most common errors are those due to an invalid combination of values in the inputs to this operation or other errors communicating with the Opalis management server. In the event of a failure, the **returnResult** output normally has more details about the nature of the failure.

The operation returns the following:

returnResult

When the operation is successful, this contains a brief message stating that it was successful. In the event of a failure, this output contains details about the error.

Note: The underlying Opalis Web services require a separate initial call to authenticate against Opalis, which returns an authentication token that is supplied in subsequent Web service calls. This authentication token is stored in the local context with a key whose name contains **OpalisAuthenticationToken** and the host and username. This value is automatically supplied to any Opalis calls made within the same flow to avoid repeated authentication calls.

Stop All Policy Requests For Policy

The **Stop All Policy Requests For Policy** operation triggers all policy requests for a particular policy to stop on the Opalis management server. When the policy requests have been triggered to stop, this operation returns without waiting for the policy requests to be stopped.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

policyId

The internal policy ID (GUID) of the policy whose requests are to be stopped. This value can be obtained using the **Get All Policies** operation. It is also the same ID that is used with Opalis'

OIS5StartPolicy command line utility. This value is normally a long set of hexadecimal values surrounded by braces. For example:

```
{42FA4AB2-5B09-4A06-8416-B9C0D18619C7}
```

Note: If the operation fails, an error has occurred in stopping the policy requests. The most common errors are those due to an invalid combination of values in the inputs to this operation or other errors communicating with the Opalis management server. This operation will also return a failure if the specified **policyId** specified is not in the GUID format that it expects. In the event of a failure, the **returnResult** output normally has more details about the nature of the failure.

The operation returns the following:

returnResult

When the operation is successful, this contains a brief message stating that it was successful. In the event of a failure, this output contains details about the error.

Note: The underlying Opalis Web services require a separate initial call to authenticate against Opalis, which returns an authentication token that is supplied in subsequent Web service calls. This authentication token is stored in the local context with a key whose name contains **OpalisAuthenticationToken** and the host and username. This value is automatically supplied to any Opalis calls made within the same flow to avoid repeated authentication calls.

Stop Policy Request

The **Stop Policy Request** operation triggers a policy request to stop on the Opalis management server. When the policy request has been triggered to stop, this operation returns without waiting for the policy request to be stopped.

All of the operation's inputs except the following are described in *Common inputs in the integration*.

policyRequestId

The ID of the policy request. This value is returned from the **Start Policy** operation. This value is normally an integer value, such as 158.

Note: If the operation fails, an error has occurred in stopping the policy requests. The most common errors are those due to an invalid combination of values in the inputs to this operation or other errors communicating with the Opalis management server. This operation will also return a failure if the specified **policyId** is not found in Opalis. The Opalis management server may also throw an internal error if the **policyRequestId** value is not a valid integer. In the event of a failure, the **returnResult** output normally has more details about the nature of the failure.

The operation returns the following:

returnResult

When the operation is successful, this contains a brief message stating that it was successful. In the event of a failure, this output contains details about the error.

Note: The underlying Opalis Web services require a separate initial call to authenticate against Opalis, which returns an authentication token that is supplied in subsequent Web service calls. This authentication token is stored in the local context with a key whose name contains **OpalisAuthenticationToken** and the host and username. This value is automatically supplied to any Opalis calls made within the same flow to avoid repeated authentication calls.

Tools

Following are OO tools that you can use with the Microsoft Opalis integration:

- **RSFlowInvoke.exe** and **JRSFlowInvoke.jar**

  RSFlowInvoke (RSFlowInvoke.exe or the Java version, JRSFlowInvoke.jar) is a command-line utility that allows you to start a flow without using Central (although the Central service must be running). RSFlowInvoke is useful when you want to start a flow from an external system, such as a monitoring application that can use a command line to start a flow.

These tools are available in the Operations Orchestration home directory in the /Studio/tools/ folder.