HP Operations Orchestration Software

Software Version: 9.00.05

HP Operations Manager i Integration Guide

Document Release Date: April 2011

Software Release Date: April 2011

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 Operations Manager i integration

With this integration, you can build HP Operations Orchestration (OO) flows that are integrated into the HP Operations Manager i (OMi).

The OMi integration operations interact with OMi through its REST-based Event Web service, which is described more completely in the *BSM OMi Extensibility Guide*. You cannot use the Event Web Service to create or modify events, so there are no operations in this integration to perform these actions. The OMi Event Web service can be reached via the HP Business Service Management Gateway Server, which is the host that should be supplied in all of the operations. The supported versions are 9.0 and 9.01

Use cases and scenarios

The following are the major use cases for the HP OMi integration, and the operations and flows that you can use to implement them.

1. Samples
   - Custom Attributes Sample
2. Manage custom attributes:
   - Add Custom Attribute
   - Custom Attributes Sample
   - Delete Custom Attribute
   - Get Custom Attributes
   - Update Custom Attribute
3. Manage annotations:
   - Add Annotation
   - Delete Annotation
   - Get Annotations
   - Update Annotation
4. Manage symptoms:
   - Add Symptom
   - Delete Symptom
   - Get Symptoms
5. Manage actions:
   - Get Auto Action
   - Get User Action
   - Launch Auto Action
   - Launch User Action
   - Stop Auto Action
   - Stop User Action
6. Manage events:
   - Update Event
   - Get Event

Installation and configuration instructions

There are no special installation or configuration steps.

Versions

Operations Orchestration Version	OMi Version
9.00.05	9.0 and 9.01

Architecture

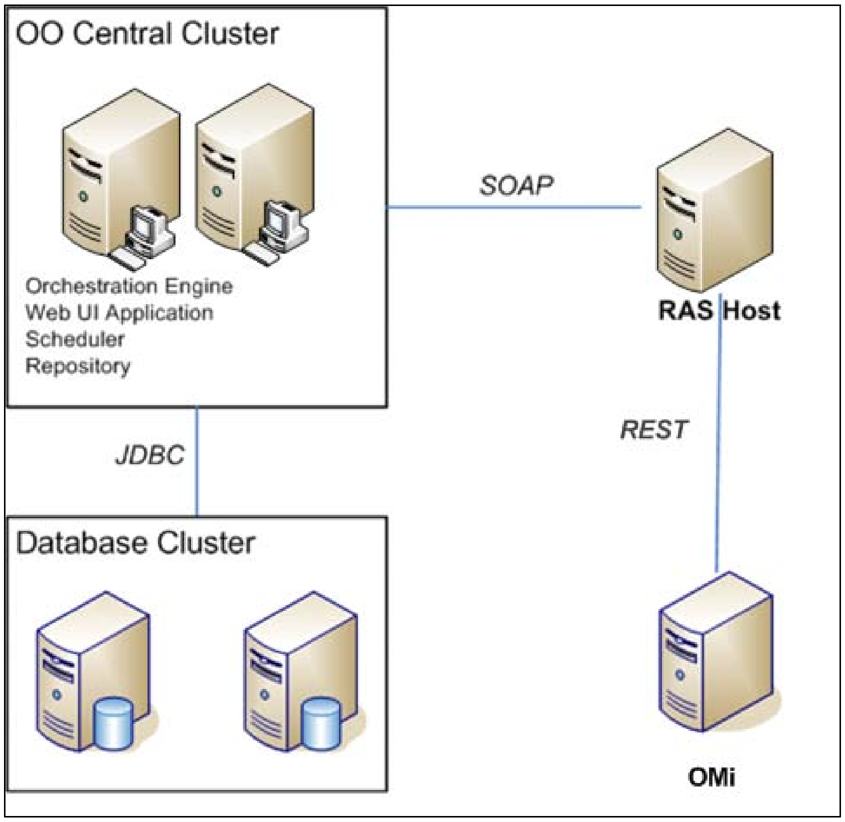

Figure 1 - OMi architecture

OMi integration operation and flow infrastructure

The OMi integration includes the following operations in the OO Studio Library/Integrations/Hewlett-Packard/Operations Manager i/ folder.

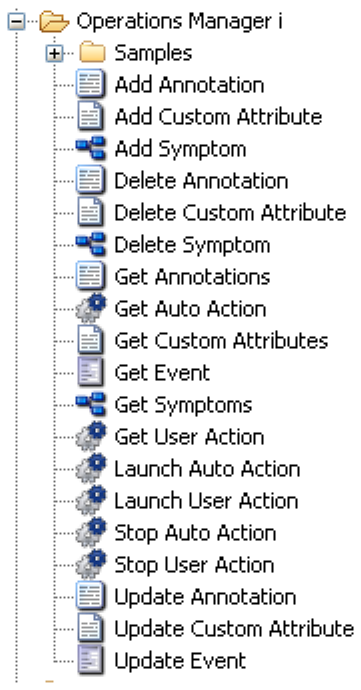

Figure 2 – OMi integration operation and flow infrastructure

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 OMi integration's operations and flows.

host

The host on which the Event Web service runs This should be the HP Business Service Management Gateway Server. The input value can be a host name or an IP address. If you specify a host name, it must be capable of being resolved to an IP address by a DNS server specified in your networking configuration. You can use the **nslookup** command, available in Windows and most UNIX systems, to verify that the given host name can be resolved.

port

The port on which the Event Web service listens for requests This is normally 80 for HTTP and 443 for HTTPS. If you do not specify a value for this input, it defaults to 80 or 443, depending on the value of the **protocol** input.

protocol

The protocol to use to connect to the Event Web service. The valid values are **HTTP** and **HTTPS**. The default is **HTTPS**.

username

In the OMi model, each user has a "user name" and a "login name" which may be different. The value for this input should be the OMi login name.

password

The password to use to connect to the Event Web service.

timeout

The length of time, in milliseconds, that can elapse before the connection times out. If the connection is not successfully made in this amount if time, the service gives up and the operation fails. If the connection is made within this time, this **timeout** value is the maximum amount of time that the operation waits to receive a reply. If you do not set a value for this input or set it to **0**, the default value of 60000 (one minute) is used.

id

The event identifier usually in UUID format —a string of hexadecimal digits and dashes (for example, **550e8400-e29b-41d4-a716-446655440000**).

Operation and flow specifics

This section describes the OMi integration’s operations and flows, including any operation- or flow-specific inputs.

The sample flow Custom Attributes Sample in the OO Library/Integrations/Hewlett-Packard/Operations Manager i/Samples/ folder perform some of the most common tasks that need to be automated when using EC2, such as starting instances, allocating elastic IPs, and working with images. 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.

Custom Attributes Sample

The **Custom Attributes Sample** flow (in the Library/Integrations/Hewlett-Packard/Operations Manager i/Samples/ folder) illustrates how to use all of the operations that manipulate custom attributes. This flow performs the following steps:

- Adds a new custom attribute to an event.
- Retrieves all custom attributes for the event.
- Builds a list of results, and displays it to the user.
- Updates the value of the newly created custom attribute to a constant value.
- Deletes the newly created custom attribute.

Add Annotation

The **Add Annotation** operation adds an annotation to an event.

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

annotationText

The annotation text to add to the event.

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Add Custom Attribute

The **Add Custom Attribute** operation adds a custom attribute to an event with the specified custom attribute name and value. If the event already has a custom attribute with the specified name, its value is updated with the new value.

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

caName

The custom Attribute name.

caValue

The custom attribute value.

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Add Symptom

The **Add Symptom** operation adds a symptom to the specified event. Symptoms are part of OMi's event correlation, where a symptom is the inverse of a cause. This operation triggers the following changes in OMi:

- The given event is marked as a "cause."
- The symptom event is marked as a "symptom."
- The correlation rule is marked as "Manually related."

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

symptomTargetId

The identifier of the event which is considered to be a symptom of the event in the **id** input. The **symptomTargetId** input value is usually in UUID format, a string of hexadecimal digits and dashes (for example, **550e8400-e29b-41d4-a716-446655440000**).

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Note: Here is an example:

Assume that OMi has received the following two events:

```
11111111-1111-1111-1111-111111111111 - Bad sector detected on disk 1
22222222-2222-2222-2222-222222222222 - Unable to save file 'myinfo.txt'
```

After determining that the event `11111111-1111-1111-1111-111111111111` is responsible for `00000000-0000-0000-0000-000000000000`, this operation could be called with the following inputs:

```
id: 11111111-1111-1111-1111-111111111111
symptomTargetId: 22222222-2222-2222-2222-222222222222
```

indicating that the second event is a symptom of the first. Event `11111111-1111-1111-1111-111111111111` is marked as a cause and event `22222222-2222-2222-2222-222222222222` is marked as a symptom.

Delete Annotation

The **Delete Annotation** operation deletes an annotation from an event. Annotation IDs can be obtained using the **Get Annotations** operation.

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

annotationId

The annotation ID to delete from the event. This ID is assigned by OMi when the annotation is created, and it may not be accessible via the OMi user interface. Annotation IDs are usually in UUID format.

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Delete Custom Attributes

The **Delete Custom Attributes** operation deletes a custom attribute from an event with the specified attribute name.

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

caName

The name of the custom attribute to delete.

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Delete Symptom

The **Delete Symptom** operation deletes a symptom from the specified event. Symptoms are part of OMi's event correlation, where a symptom is the inverse of a cause. This operation triggers changes in OMi, removing the "cause" from the given event and the "symptom" from the symptom event.

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

symptomTargetId

The identifier of the event that is a symptom of the event in the **id** input. The **symptomTargetId** is usually in UUID format, a string of hexadecimal digits and dashes (for example, **550e8400-e29b-41d4-a716-446655440000**).

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Get Annotations

The **Get Annotations** operation gets the annotations for an event.

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

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

jsAnnotations

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

Each object in the array contains the following information about an annotation:

id

The annotation ID. This ID is used as an input to other operations such as **Delete Annotation**.

author

The login name of the user that last modified this annotation.

text

The annotation text.

date

The date and time of the last modification of this annotation, in ISO 8601 format.

For example, if there are two annotations on the event, then **jsAnnotations** might contain:

```
[ { "id"    :"ad3d404e-0c23-463c-8910-ffd4466a785a",
      "author":"admin",
      "text"  :"This event was painful to track down",
      "date"  :"2010-12-07T21:37:14-0800" },
   { "id"    :"b345a3d7-302a-42e1-af47-292d8837dc2d",
      "author":"SYSTEM",
      "text"  :"automatic action\n started.\nOperator : SYSTEM",
      "date"  :"2010-12-02T06:35:24-0800" }]
```

(Extra whitespace inserted for readability.)

Get Auto Action

The **Get Auto Action** operation gets information about the auto action assigned to an event.

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

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

actionStatus

The state of the auto action. Values include **available** and **succeeded**.

actionCall

The command that is called when the action is run.

Get Custom Attributes

The **Get Custom Attributes** operation gets the custom attributes of an event.

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

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

jsCustomAttributes

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

Each object in the array contains the following pieces of information about a custom attribute:

caName

The custom attribute name

caValue

The custom attribute value

For example, if there are two custom attributes on the event, then **jsCustomAttributes** might contain:

```
[{ "caValue" :"Reviewed by",
   "caName"  :"operator2" },
 { "caValue" :"Region",
   "caName"  :"43"}]
```

(Extra whitespace inserted for readability.)

Get Event

The **Get Event** operation gets the details of an event.

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

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

application

The application that triggered the event.

assignedGroup

The assigned group of the event.

assignedUser

The login name of the user assigned to the event.

category

The category of the event.

description

The description of the event.

duplicates

The count of duplicates of the event.

eti

The Event Type Indicator.

id

The ID of the event.

node

The node on which the event occurred.

object

The object of the event.

priority

The priority of the event. Possible values are **none**, **low**, **medium**, **high**, and **highest**.

severity

The severity of the event. Possible values are **unknown**, **normal**, **warning**, **minor**, **major**, and **critical**.

sourceCI

The source CI.

state

The state of the event. Possible values are **open**, **in_progress**, **resolved**, and **closed**.

subcategory

The subcategory of the event.

timeCreated

The time the event was created, in ISO 8601 format.

timeReceived

The time the event was received by OMi, in ISO 8601 format.

title

The title of the event.

Get Symptoms

The **Get Symptoms** operation retrieves symptoms for an event. Symptoms are part of OMi's event correlation, where a symptom is the inverse of a cause.

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

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

jsSymptoms

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

For example, if the event has two symptoms, **jsSymptoms** might contain:

```
["cac7516a-b58b-48ff-b60e-0e23bc9886b5",
 "b5341300-fe28-71df-07f6-1039417e0000" ]
```

Get User Action

The **Get User Action** operation gets information about the user action assigned to an event.

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

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

actionStatus

The state of the user action. Values include **available** and **succeeded**.

actionCall

The command that is called when the action is run.

Launch Auto Action

The **Launch Auto Action** operation launches the auto action assigned to an event. Note that this operation does not wait for the action to complete.

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

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Launch User Action

The **Launch User Action** operation launches the user action assigned to an event. Note that this operation does not wait for the action to complete.

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

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Stop Auto Action

The **Stop Auto Action** operation stops the auto action assigned to an event.

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

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Stop User Action

The **Stop User Action** operation stops the user action assigned to an event.

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

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Update Annotation

The **Update Annotation** operation updates an annotation on an event. The specified annotation ID must already exist. Annotation IDs can be obtained using the **Get Annotations** operation.

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

annotationId

The annotation ID to update on the event. This ID is assigned by OMi when the annotation is created, and it may not be accessible via the OMi user interface. Annotation IDs are usually in UUID format.

annotationText

The annotation text to update on the event.

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Note: OMi 9.0 has a defect that causes this operation to always fail. This defect has been fixed in OMi 9.01.

Update Custom Attribute

The **Update Custom Attribute** operation updates the value of an existing custom attribute of an event.

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

caName

The custom attribute name to update.

caValue

The new custom attribute value.

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Update Event

The **Update Event** operation updates the details of an event. Note that only inputs that contain values are updated; others remain unchanged.

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

assignedGroup

The assigned group of the event.

assignedUser

The login name of the user to assign to the event. OMi expects this to be a login name (the one used to log on), which may be different from the display username. If the login name is unknown, OMi silently ignores this change.

description

The description of the event.

priority

The priority of the event. This value is case-insensitive. The valid values are **none**, **low**, **medium**, **high**, and **highest**.

severity

The severity of the event. This value is case-insensitive. The valid values are **unknown**, **normal**, **warning**, **minor**, **major**, and **critical**.

solution

The solution of the event.

state

The state of the event. This value is case-insensitive. The valid values are **open**, **in_progress**, **resolved**, and **closed**.

title

The title of the event.

The operation returns the following:

returnResult

If the operation is successful, this output contains a brief message stating that it was successful. If the operation fails, this output contains details about the error.

Troubleshooting

This section provides troubleshooting procedures and tools 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.

General troubleshooting procedures and tools

Section VI of the *BSM OMi Extensibility Guide,* "Automating Operator Functions and Event Change Detection," contains some useful tips for using a third-party REST client. This can be very helpful in troubleshooting connectivity or other problems.

Tools

Following are OO tools that you can use with the OMi 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 folder in /Studio/tools/.