



# Integration with Operations Orchestration

Software version: 4.80

Document release date: January 2017

Software release date: January 2017

# Contents

<b>Introduction .....</b>	<b>4</b>
Supported versions .....	4
Operations Orchestration requirements .....	4
Downloading OO releases and content .....	4
<b>Getting started .....</b>	<b>5</b>
Configuring system properties .....	5
Before you use the integration .....	5
CSA use cases .....	5
Integration architecture .....	6
CSA terminology .....	6
CSA interfaces .....	7
Location of CSA integration workflows and actions in OO Studio .....	7
<b>Integration operations and workflows .....</b>	<b>8</b>
General input parameters .....	8
Resource Provider Access Details workflow .....	10
Get Artifact Properties workflow .....	10
Update Service Component Property workflow .....	10
Add Child Service Component workflow .....	11
Retire Service Component workflow .....	12
Service Component Lifecycle State workflow .....	12
Add Resource Binding to Service Component workflow .....	12
Get User Identifier workflow .....	14
Update Process Instance State workflow .....	14
Get User Details workflow .....	15
Get Artifact Details workflow .....	15
Get Organization Details workflow .....	16
Get Service Subscription Details workflow .....	16
Add Public Actions workflow .....	16
Deploy Service Component Tree workflow .....	19
Execute Process Instance workflow .....	19
Finalize Service Component workflow .....	20
Invoke Clone Process Instance workflow .....	20
Poll Life Cycle Engine workflow .....	20
Poll Process Instance workflow .....	21
Get Candidate Providers workflow .....	21
Update Valid Providers workflow .....	21
Update Service Component Display Name workflow .....	22
Send Email Notification workflow .....	22
Update Approval Status workflow .....	23
Get Candidate Provider Pool .....	23
Get Resource Pool .....	24
Update Resource Pool Capacity .....	24
Update Resource Pool Utilization .....	25
Update Valid Provider Pool .....	25

<b>Troubleshooting</b> .....	<b>26</b>
General troubleshooting procedures and tools .....	26
Error messages .....	26
<b>Security</b> .....	<b>26</b>
<b>OO tools</b> .....	<b>27</b>
<b>Send documentation feedback</b> .....	<b>28</b>
<b>Legal notices</b> .....	<b>28</b>
Warranty .....	28
Restricted rights legend .....	28
Copyright notice .....	28
Trademark notices .....	28
Documentation updates .....	28
Support .....	28

# Introduction

With this integration, administrators can create HPE Operations Orchestration (OO) workflows that are integrated with HPE Cloud Service Automation (CSA).

To learn how to create OO workflows, see the Studio Guide to Authoring Operations Orchestration Workflows.

This integration uses the CSA web service and the Artifact API (primarily) to integrate with CSA.

This document explains how the integration has been implemented, and how the integration's operations and workflows communicate between OO and CSA.

The guide is intended for OO system administrators, CSA system administrators, or any user who intends to build and modify OO workflows that are integrated with CSA. This guide assumes that you have administrative access to both systems.

## Supported versions

CSA software and hardware requirements are documented in Cloud Service Automation Platform Support Matrix. OO software and hardware requirements are documented in Operations Orchestration System Requirements. You can find these documents at <http://support.openview.hp.com/selfsolve/manuals>.

## Operations Orchestration requirements

You must have the following OO patches and content packs installed:

- OO Central 9.00.0 (major version)
- OO Studio 9.00.00 (major version)
- OO 09.03.0001 or 09.05.0000 patch with OO hotfixes HF\_147063 (HTTPS Hot fix) and HF\_147063 (HTTPS Hot fix)
- OO Content Pack 9 (minor content pack, cumulative)
- OO – Server Automation 9.00.06 (minor OO-SA integration content pack)
- OO 9.00.006 JPN (minor content pack for Japanese versions, if needed)

Important: OO 09.03.0001 is a replacement for OO 9.03. If 9.03 is already installed, then uninstall 9.03 and install OO 09.03.0001.

## Downloading OO releases and content

To download OO patches, go to <http://support.openview.hp.com/selfsolve/patches> and navigate to Operations Orchestration > 9.03 > Operations Orches 09.03.0001.

To download CSA and SA content patches for OO:

1. Go to <https://hpln.hp.com/>.
2. Click the All Content tab.
3. In the Product field, select Operations Orchestration.
4. Select Operations Orchestration Content for Cloud Service Automation or Operations Orchestration Content for Server Automation.
5. Click Content.
6. The appropriate content packs are found in the 9.00 folder.

You may find the following documentation useful:

Online help for users and administrators in OO Studio and OO Central, which is found in the Help menu.

- Animated tutorials can be found in the \Central and \Studio subdirectories of the OO home directory.
- Documentation for CSA workflows is provided in the workflows.

# Getting started

## Configuring system properties

You must configure the following system properties.

Table 1 OO configuration

System property	Value
CSA_REST_URI	Set to the URI of the CSA instance. If OO and CSA are installed on same machine, then there is conflict with default ports.  If OO Studio uses 8081, you must change the default HTTP port from 8081 to another port on CSA:  https://<CSA Server IP or FQDN>:8444/csa/rest  or  http://<CSA Server IP or FQDN>:8081/csa/rest.
CSA_OO_USER	Set to the name of the OO service account, and the username for the basic authentication method. By default, this is oolboundUser. Consult your CSA administrator to obtain and set the password for this user.
CSA_REST_CREDENTIALS	Contains the credentials for the CSA_OO_USER configured above. The OO user and CSA user should be configured identically. You must set the password for this user before you use integrated workflows. The default values are:  User: oolboundUser  Password: blank

### Before you use the integration

The CSA integration content makes use of standard OO content and, in particular, the HTTP Client operation. Pay particular attention to the recommended versions of OO platform and content for use with CSA to ensure that the integration will work correctly.

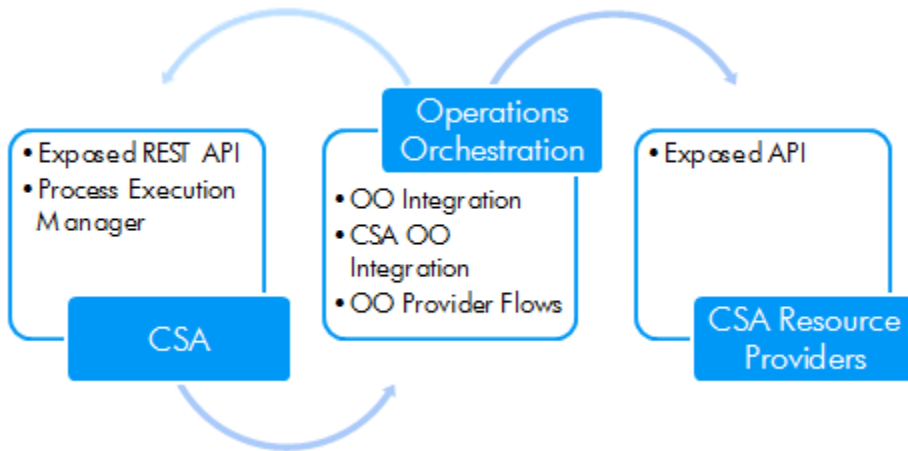
See the Rest Integration Developers Guide in the OO document set for more detailed information on the use of the HTTP client with application REST interfaces, such as CSA exposes.

The CSA platform requires that the correct URL and credentials are configured in OO to communicate with the API. In addition, you should consult the CSA Configuration Guide and the Operations Orchestration Software Development Kit Guide for information about configuring a secure communications channel between OO Central and CSA.

### CSA use cases

The primary use case for the CSA integration content is to provide a common, reusable set of operations for interacting with the CSA API from CSA resource provider workflows. CSA service subscriptions drive a sequence of automated actions from a service design which invoke OO workflows to interact with the CSA resource providers. The CSA process execution manager launches OO workflows, and tracks their completion and status. These workflows make extensive use of the OO integration content to communicate with resource providers to deploy elements of CSA services. These workflows, in turn, communicate with CSA to populate specific properties with resource information. The CSA integration content makes calls into the exposed CSA REST API to retrieve or update service instance information, create or delete service components, and drive lifecycle state changes.

Figure 1: Workflows

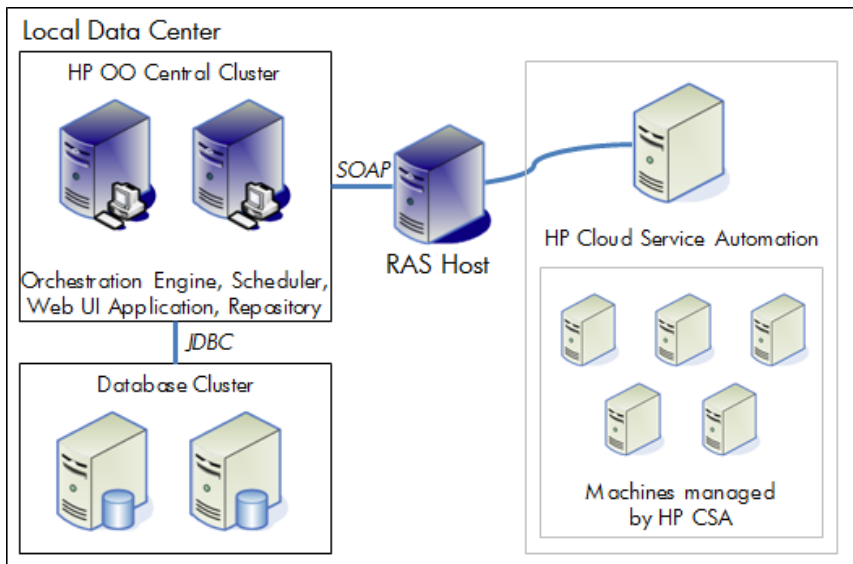


## Integration architecture

This integration allows administrators to build OO workflows that are integrated with Server Automation (SA).

The SA operations are found in the Central repository in the /Library/Integrations/Hewlett-Packard/Server Automation/ folder.

Figure 2: Architecture



## CSA terminology

The following terms are used in this guide. A complete glossary can be found in Cloud Service Automation Concepts Guide.

- **Lifecycle** - The stages of programmatically deploying a cloud service: initializing, reserving, and deploying. Conversely, the stages of removing a cloud service from deployment: un-deploying, un-reserving, and un-initializing. The service lifecycle also has a separate modification state.
- **Lifecycle action** - A function that is run automatically at a specified lifecycle state or sub-state. Lifecycle actions reference internal or external process definitions, which perform the specified action, such as initializing, reserving, or deploying a service subscription. Lifecycle actions can be applied to service components or resource offerings as part of the service lifecycle.
- **Lifecycle state** - A lifecycle state represents a step within the CSA service provisioning and de-provisioning lifecycles. States are either transition states or stable states.

- Process instance - A process instance is a specific invocation of a process definition, and is managed in CSA by the Process Execution Manager. Process definitions are created by synchronizing selected OO workflows into CSA to capture the calling signature – the workflow inputs, and the workflow path needed to programmatically launch the workflow from CSA.
- Resource binding - A link in a CSA service design between a resource offering and a service component. For example, a resource offering for a specific VMware vCenter VM template can be linked to a Server Group service component. The resource binding ensures that the resource offering is provisioned as part of the service component deployment.
- Resource provider - A management platform that provides either Infrastructure-as-a-Service (IaaS) or Software-as-a-Service (SaaS) to the cloud. For example, a provider such as CloudSystem Matrix deploys virtual machines, while a provider such as SiteScope monitors applications.
- Service component - An element of a CSA service design that has an associated component type that constrains its allowed children and its assignable resource categories.

## CSA interfaces

When using this integration, you may need to access the following CSA interfaces.

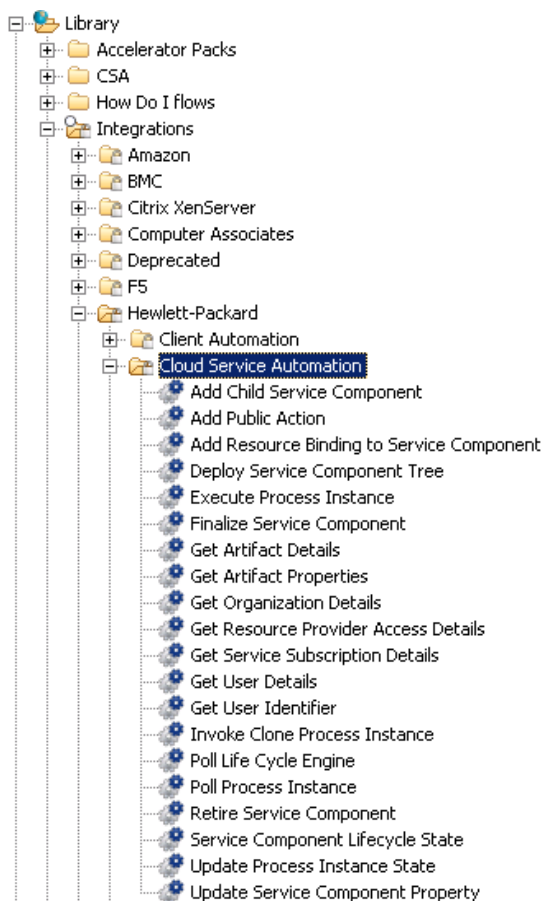
- CSA Management Console: This is an administrative user interface which provides a facility for creating resource offerings and service designs, and through which actions are assigned.

## Location of CSA integration workflows and actions in OO Studio

The integration includes both workflows and actions.

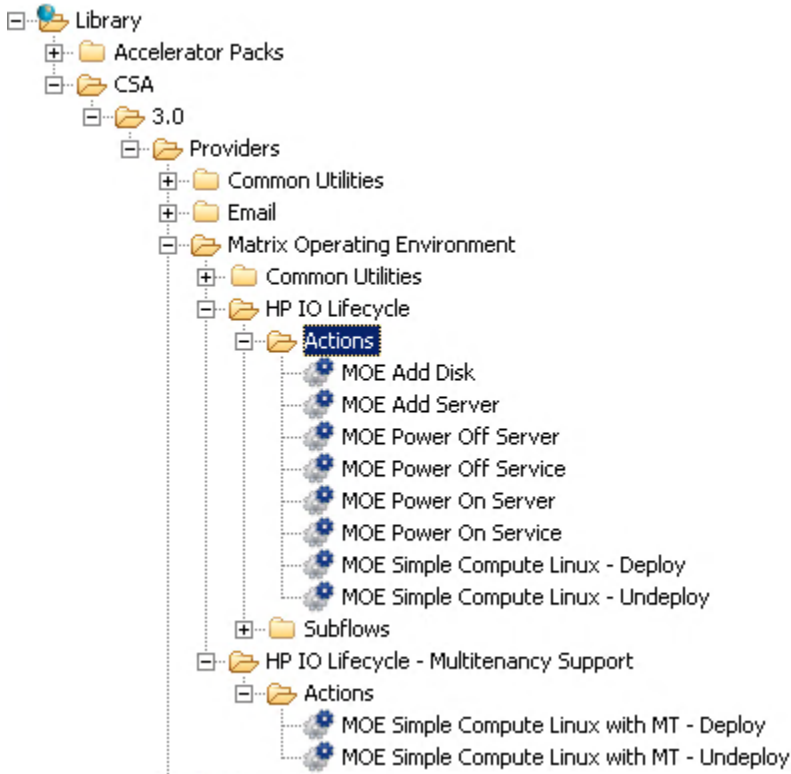
The workflows are located in the Library/Integrations/Hewlett-Packard/Cloud Service Automation folder.

Figure 3: Location of CSA workflows



The actions are located under the Library/CSA/3.0/Providers folder. The subfolders contain Actions folders, which is where the actions are located.

Figure 4: Location of an Actions folder



## Integration operations and workflows

### General input parameters

The following parameters are used in the workflows in this section. Each workflow may use some or all of these parameters.

Table 2 General input parameters

Parameter	Description
userIdentifier	The user identifier ID.
authType	The authentication type that this operation uses when trying to execute the request on the target server. Valid values are <code>basic</code> , <code>form</code> , <code>digest</code> , <code>ntlm</code> , <code>kerberos</code> , or <code>anonymous</code> (no authentication). Default is <code>anonymous</code> .
kerberosConfFile	The path to the Kerberos configuration file. If this input is empty the KDC (Key Distribution Center) and realm will get their values from the full domain name which is obtained from the URL of the web server.
timeout	The number of milliseconds to wait for a connection to be established. This input should be assigned a value greater than or equal to zero. A value of zero is treated as infinite. Default is 0.



socketTimeout	The number of milliseconds to wait for the data to be retrieved. This input should be assigned a value greater than or equal to zero. A value of zero is treated as infinite.  Default is 0.
useCookies	If <code>true</code> , enables cookie tracking. Cookies will be stored between consecutive calls. This is a Boolean input. If a non-Boolean value is specified, the default value is used.  Default is <code>true</code> .
followRedirects	If <code>true</code> , the GET command automatically follows redirects. This is a Boolean input. If a non-Boolean value is specified, the default value is used.  Default is <code>true</code> .
proxy	A proxy to use when accessing the web site.
proxyPort	The port to use when connecting to the proxy.
proxyUsername	The username to use when connecting to the proxy.
proxyPassword	The password to use when connecting to the proxy.
encodeURL	If <code>true</code> , encodes the URL. This is a Boolean input. If a non-Boolean value is specified, the default value is used.  Default is <code>false</code> .
userAgent	The value that should be used to override the HTTP header <code>USER_AGENT</code> parameter.
trustAllRoots	If <code>true</code> , enables weak security over SSL. A SSL certificate is trusted even it was not issued by a trusted certificate authority. This is a Boolean input. If a non-Boolean value is specified, the default value is used.  Default is <code>true</code> .
keystore	URL of the keystore file or local path to the file, such as <code>%JAVA_HOME%/jre/lib/security/cacerts</code> . It may be empty if HTTPS client authentication is not to be used.
keystorePassword	Password for the keystore.
trustKeystore	URL of the truststore file or local path to the file, such as <code>%JAVA_HOME%/jre/lib/security/cacerts</code> . It may be empty if HTTPS server authentication is not to be used.
trustPassword	Password for the truststore.

## Resource Provider Access Details workflow

This workflow returns the resource provider details.

Table 3 Input

Parameter	Description
providerId	The resource provider ID.

Table 4 Output

Parameter	Description
providerSAP	The provider access point.
providerUser	The provider user name.
providerPassword	The provider password.

## Get Artifact Properties workflow

This workflow returns the list of Artifact's properties and values.

Table 5 Input

Parameter	Description
providerId	The resource provider ID.

Table 6 Output

Parameter	Description
propertyNameValuesList	A list of property names and values. For example, <code>property 1:value1,value2 property 2:value3 property 3:value3,value4 ...</code>

## Update Service Component Property workflow

This workflow adds or updates the service component property using Artifact API.

Table 7 Input

Parameter	Description
componentId	The service component ID.
propertyName	The name of the property that you want to add or update.
onlyUpdate	Only update existing properties; do not create new properties. Valid values are either <code>yes</code> or <code>no</code> . If <code>yes</code> , then only existing properties will be updated. If <code>no</code> , then new properties will be added if a new property is sent.

valueType	The type of value for the property. This input is required only when onlyUpdate is set with a value of no. Valid values can be found in the <b>CSA – Property Value Type</b> selection list in OO.
separator	A delimiter that separates elements for the values parameter. Default is a comma.
values	One or more values delimited by a separator specified by separator parameter. If not specified, then the default is a comma.

Table 8 Output

Parameter	Description
document	The XML returned from CSA.

## Add Child Service Component workflow

This workflow adds or updates the child service component to parent service component using the Artifact API.

Table 9 Input

Parameter	Description
parentComponentId	Service component ID for the parent.
childComponentXML	Child component XML document.  Example: <pre>&lt;componentChild&gt;   &lt;isCriticalSystemObject&gt;false&lt;/isCriticalSystemObject&gt;   &lt;description&gt;Child 22&lt;/description&gt;   &lt;name&gt;Child22&lt;/name&gt;   &lt;state&gt;     &lt;name&gt;ACTIVE&lt;/name&gt;   &lt;/state&gt;   &lt;artifactType&gt;     &lt;name&gt;SERVICE_COMPONENT&lt;/name&gt;   &lt;/artifactType&gt;   &lt;disabled&gt;false&lt;/disabled&gt;   &lt;lifecycleProperties&gt;     &lt;lifecycleComponentOrder&gt;0&lt;/lifecycleComponentOrder&gt;     &lt;lifecycleState&gt;       &lt;name&gt;INITIALIZING&lt;/name&gt;     &lt;/lifecycleState&gt;     &lt;lifecycleSubstate&gt;       &lt;name&gt;PRE_TRANSITION&lt;/name&gt;     &lt;/lifecycleSubstate&gt;   &lt;/lifecycleProperties&gt;   &lt;componentType&gt;     &lt;name&gt;SERVER&lt;/name&gt;   &lt;/componentType&gt;   &lt;template&gt;false&lt;/template&gt;   &lt;createdFromTemplate&gt;false&lt;/createdFromTemplate&gt; &lt;/componentChild&gt;</pre>

Table 10 Output

Parameter	Description
componentId	The ID for the child service component that was added.
document	The XML returned from CSA.

## Retire Service Component workflow

This workflow deletes the service component from CSA.

Table 11 Input

Parameter	Description
componentId	Service component ID.

Table 12 Output

Parameter	Description
document	The XML returned from CSA.

## Service Component Lifecycle State workflow

This workflow updates the lifecycle state and substate of the service component.

Table 13 Input

Parameter	Description
componentId	The service component ID.
lifeCycleState	Gets the value from selection list CSA - Lifecycle State in OO.
lifeCycleSubState	Gets the value from selection list CSA - Lifecycle Substate in OO.

Table 14 Output

Parameter	Description
lifeCycleState	The updated lifecycle state of the service component.
lifeCycleSubState	The updated lifecycle sub-state of the service component.
document	The XML returned from CSA.

## Add Resource Binding to Service Component workflow

This workflow adds/updates the resource binding to the service component.

Table 15 Input

Parameter	Description
componentId	The service component ID.
resourceBindingXML	<p>Resource binding XML document</p> <p>Example:</p> <pre> &lt;resourceBinding&gt;   &lt;!--id is not required if creating a new resource   binding but is required for update --&gt;   &lt;isCriticalSystemObject&gt;&gt;false&lt;/isCriticalSystemObject&gt;   &lt;state&gt;     &lt;name&gt;ACTIVE&lt;/name&gt;   &lt;/state&gt;   &lt;artifactType&gt;     &lt;name&gt;RESOURCE_BINDING&lt;/name&gt;   &lt;/artifactType&gt;   &lt;disabled&gt;&gt;false&lt;/disabled&gt;   &lt;bindingState&gt;     &lt;name&gt;BOUND&lt;/name&gt;   &lt;/bindingState&gt;   &lt;bindingStatus&gt;bound&lt;/bindingStatus&gt;   &lt;resourceFilter&gt;This is the Resource Filter&lt;/resourceFilter&gt;   &lt;resourceOffering&gt;     &lt;!-- Provide either id or name, both are not     required --&gt;     &lt;name&gt;Test_SA_April 19, 2012 5:08:43 AM UTC&lt;/name&gt;     &lt;id&gt;8ab289a336cdcaf40136e35f1a2e21e8&lt;/id&gt;   &lt;/resourceOffering&gt;   &lt;lifeCycleProperties&gt;     &lt;lifecycleComponentOrder&gt;0&lt;/lifecycleComponentOrder&gt;     &lt;lifecycleState&gt;       &lt;name&gt;INITIALIZING&lt;/name&gt;     &lt;/lifecycleState&gt;     &lt;lifecycleSubstate&gt;       &lt;name&gt;PRE_TRANSITION&lt;/name&gt;     &lt;/lifecycleSubstate&gt;   &lt;/lifeCycleProperties&gt; &lt;/resourceBinding&gt; </pre>

Table 16 Output

Parameter	Description
resourceBindingId	The ID of the resource binding that was added or updated.
document	The XML returned from CSA.

## Get User Identifier workflow

This workflow returns the user identifier id for the given user.

Table 17 Input

Parameter	Description
csaUser	A CSA user name. Default is the <code>CSA_OO_USER</code> system property, which is set to <code>ooInboundUser</code> .

Table 18 Output

Parameter	Description
userIdentifier	The user identifier ID for the given user.

## Update Process Instance State workflow

This workflow updates the process instance state using the Process Instances API.

Table 19 Input

Parameter	Description
processInstanceId	The process instance ID.
processInstanceState	Gets the value from the selection list <code>CSA - Process Instance State</code> in OO.
processReturnCode	Optional. Gets the value from selection list <code>CSA - Process Instance Return Code</code> in OO.
processStatus	Optional. The text you want to use for the status update.

Table 20 Output

Parameter	Description
processInstanceState	The updated state of the process instance.
processReturnCode	The updated return code of the process instance.
processStatus	The updated status of the process instance.

document	The XML returned from CSA.
----------	----------------------------

## Get User Details workflow

This workflow returns the user details for given user.

Table 21 Input

Parameter	Description
userId	The CSA user ID for the user you want to get details.

Table 22 Output

Parameter	Description
userName	The user name for the provided user ID.
document	The XML returned from CSA.

## Get Artifact Details workflow

This workflow returns the details of an artifact.

Table 23 Input

Parameter	Description
artifactId	The artifact ID.

Table 24 Output

Parameter	Description
name	The name of the artifact.
displayName	The display name of the artifact.
artifactType	The artifact type.
state	The state of the artifact.
lifeCycleState	The lifecycle state of the artifact.
lifeCycleSubState	The lifecycle sub-state of the artifact.
document	The XML returned from CSA.

## Get Organization Details workflow

This workflow returns the details of an organization.

Table 25 Input

Parameter	Description
organizationId	The organization ID.

Table 26 Output

Parameter	Description
name	The name of the organization.
businessRole	The business role of the organization.
state	The state of the organization.
document	The XML returned from CSA.

## Get Service Subscription Details workflow

This workflow returns the details of a service subscription.

Table 27 Input

Parameter	Description
svcSubscriptionId	The service subscription ID.

Table 28 Output

Parameter	Description
subEndDate	The service subscription end date.
subStartDate	The service subscription start date.
document	The XML returned from CSA.

## Add Public Actions workflow

This workflow adds or updates the public action on service component using the Artifact API.

Table 29 Input

Parameter	Description
componentId	The service component ID.



actionXML

```
<action>
  <isCriticalSystemObject>>false</isCriticalSystemObject>
  <description>Power off the server</description>
  <name>Power Off Server</name>
  <errorOnTimeout>>false</errorOnTimeout>
  <failOnError>>false</failOnError>
  <processDefinition>
    <name>/Library/CSA/3.0/Providers/Matrix Operating Environment/IO
Lifecycle/Actions/MOE Power Off Server</name>
    </processDefinition>
    <stateConstraint>
      <lifecycleState>
        <name>DEPLOYED</name>
      </lifecycleState>
      <lifecycleSubstate>
        <name>TRANSITION</name>
      </lifecycleSubstate>
      <lifecycleExecOrder>0</lifecycleExecOrder>
    </stateConstraint>
    <synchronous>>false</synchronous>
    <timeout>0</timeout>
    <consumerVisible>>true</consumerVisible>
    <property>
      <name>RSC_PROVIDER_ID</name>
      <valueType>
        <name>STRING</name>
      </valueType>
      <values>
        <value>${RSC_PROVIDER_ID}</value>
      </values>
      <consumerVisible>>false</consumerVisible>
    </property>
    <property>
      <name>SVC_INSTANCE_ID</name>
      <valueType>
        <name>STRING</name>
      </valueType>
      <values>
        <value>[TOKEN:SVC_INSTANCE_ID]</value>
      </values>
    </property>
  </processDefinition>
</action>
```

```
</values>
<consumerVisible>>false</consumerVisible>
  </property>
</property>
<name>SVC_COMPONENT_ID</name>
<valueType>
  <name>STRING</name>
</valueType>
<values>
  <value>[TOKEN:SVC_COMPONENT_ID]</value>
</values>
  <consumerVisible>>false</consumerVisible>
  </property>
  <property>
<name>PRN_COMPONENT_ID</name>
<valueType>
  <name>STRING</name>
</valueType>
<values>
  <value>[TOKEN:PRN_COMPONENT_ID]</value>
</values>
<consumerVisible>>false</consumerVisible>
  </property>
  <property>
<name>REQ_USER_ID</name>
<valueType>
  <name>STRING</name>
</valueType>
<values>
  <value>[TOKEN:REQ_USER_ID]</value>
</values>
<consumerVisible>>false</consumerVisible>
  </property>
  <property>
<name>SVC_COMPONENT_TYPE</name>
<valueType>
  <name>STRING</name>
</valueType>
```

	<pre> &lt;values&gt;   &lt;value&gt;[TOKEN:SVC_COMPONENT_TYPE]&lt;/value&gt; &lt;/values&gt; &lt;consumerVisible&gt;&gt;false&lt;/consumerVisible&gt;   &lt;/property&gt; &lt;/action&gt; </pre>
--	--

Table 30 Output

Parameter	Description
document	The XML returned from CSA.

## Deploy Service Component Tree workflow

This workflow requests the lifecycle engine to execute and move the given service component and its children to the Deployed state.

Table 31 Input

Parameter	Description
componentId	The service component ID.
serviceInstanceId	The service instance ID.

Table 32 Output

Parameter	Description
document	The XML returned from CSA.

## Execute Process Instance workflow

This workflow starts a process instance execution.

Table 33 Input

Parameter	Description
processInstanceId	The process instance ID.

Table 34 Output

Parameter	Description
processInstanceId	If the execution started successfully, then the process instance ID is returned.
document	The XML returned from CSA.

## Finalize Service Component workflow

This workflow requests the lifecycle engine to execute and sets the service component to the Finalized state.

No unique input parameters are necessary. See the list of general input parameters at the beginning of this chapter.

Table 35 Output

Parameter	Description
document	The XML returned from CSA.

## Invoke Clone Process Instance workflow

This workflow requests the lifecycle engine to clone a process instance.

Table 36 Input

Parameter	Description
componentId	The component ID.
propertyName	The service component's property name which has the number of the clone server to be created.

Table 37 Output

Parameter	Description
processInstanceId	The ID of the process instance that was created.
document	The XML returned from CSA.

## Poll Life Cycle Engine workflow

This workflow returns the lifecycle engine status for a service instance and waits for the execution to be completed.

Table 38 Input

Parameter	Description
serviceInstanceId	The service instance ID.

Table 39 Output

Parameter	Description
status	The status of the lifecycle engine process.
document	The XML returned from CSA.

## Poll Process Instance workflow

This workflow returns a process instance status and waits for the execution to be completed.

Table 40 Input

Parameter	Description
processInstanceId	The process instance ID.
waitCounter	Maximum number of wait cycles. Default is 60.
waitTime	The wait time in seconds for each wait cycle. Default is 30.

Table 41 Output

Parameter	Description
status	The status of the lifecycle engine process.
document	The XML returned from CSA.

## Get Candidate Providers workflow

This workflow returns the candidate providers in the resource bindings.

Table 42 Input

Parameter	Description
resourceBindingId	The resource binding ID.

Table 43 Output

---

Parameter	Description
document	The XML returned from CSA.

---

## Update Valid Providers workflow

This workflow updates valid providers list to the resource binding and does not require input values.

Table 44 Output

Parameter	Description
resourceBindingId	The resource binding ID.
resourceBindingXML	Resource binding XML document

```

Example:
<ResourceBinding>
  <id>8ab289a336cdcaf40136e35f1a2e21e8</id>
  <ValidProvider>
    <id>8ab289a336cdcaf40136e35f1a2e21ef</id>
  </ValidProvider>
  <ValidProvider>
    <id>8ab289a336cdcaf40136e35f1a2e21ee</id>
  </ValidProvider>
  <ValidProvider>
    <id>8ab289a336cdcaf40136e35f1a2e21ed</id>
  </ValidProvider>
</ResourceBinding>

```

### Update Service Component Display Name workflow

This workflow updates service component display name.

Table 45 Input

Parameter	Description
componentId	The service component ID.
userIdentifier	The user identifier ID.
displayName	The display name of the artifact.

Table 46 Output

Parameter	Description
document	The XML returned from CSA.

### Send Email Notification workflow

This workflow sends an email notification to list of customers. Its output is an email rather than a parameter.

Table 47 Input

Parameter	Description
userIdentifier	The user identifier ID.
subject	The subject of the email.
body	The body of the email. For example : Welcome to the group {0} and your reporting manager will be {1}

customerUIDList	A comma delimited list of customer UUID tokens.
tokenList	List of token values separated by a delimiter that will be replaced in the email body text. For example: {0} and {1} will be replaced in the email body.
delimiter	Separator for token list of values. Default is a comma.

## Update Approval Status workflow

This workflow updates an external approval status (approve/deny) using Approval API.

Table 48 Input

Parameter	Description
componentId	The service component ID.
userIdentifier	The user identifier ID.
displayName	The display name of the artifact.
catalogId	The catalog ID
approvalId	The approval ID
approverName	The name of the approver
ApprovalResult	The approval status ( APPROVED or REJECTED)
organization	The organization ID

Table 49 Output

Parameter	Description
document	The XML returned from CSA.

## Get Candidate Provider Pool

Retrieve candidate pool xml document based on resource binding artifact.

Table 50 Input

Parameter	Description
resourceBindingId	Resource Binding Id

Table 51 Output

Parameter	Description
document	Resource Binding xml document from HPCSA

## Get Resource Pool

Get pool details from the resource pool artifact.

Table 52 Input

Parameter	Description
resourcePoolId	Resource pool ID.
resourceType	Resource type. For example: CPU, memory, storage, etc.

Table 53 Output

Parameter	Description
document	Resource Pool xml document from HPCSA
providerId	Resource Provider Id for the Pool
poolReference	Pool reference name / Known by Provider as
availabilityIndicator	Availability Indicator of the resource type
usedByCSA	Resource value consumed by CSA
Unit	Unit of the resource
availabletoCSA	Total resource available to CSA

## Update Resource Pool Capacity

This workflow updates available resource capacity value for a resource modeled in a Resource Pool.

Table 54 Input

Parameter	Description
resourcePoolId	Resource Pool Id
resourceType	Resource type. For example: CPU, Memory, Storage etc
availableValue	The capacity of the resource which is available to CSA.

Table 55 Output

Parameter	Description
document	Updated resource binding xml document.



## Update Resource Pool Utilization

This workflow updates consumed resource capacity value for a resource modeled in a Resource Pool.

Table 56 Input

Parameter	Description
resourcePoolId	Resource Pool Id
resourceType	Resource type. For example: CPU, Memory, Storage etc
usedValue	The capacity of the resource which is consumed in CSA.

Table 57 Output

Parameter	Description
document	Updated Resource Binding xml document from CSA

## Update Valid Provider Pool

Update valid provider pool ids for a resource binding.

Table 58 Input

Parameter	Description
resourceBindingId	Resource Binding Id
resourceBindingXml	Binding XML containing to update  <pre>&lt;ResourceBinding&gt; &lt;id&gt;\${RSC_BINDING_ID}&lt;/id&gt;   &lt;validProvider&gt;     &lt;resourceBinding&gt;       &lt;id&gt;         \${RSC_BINDING_ID}       &lt;/id&gt;     &lt;/resourceBinding&gt;     &lt;resourceProvider&gt;       &lt;id&gt;         \${resourceProviderId}       &lt;/id&gt;     &lt;/resourceProvider&gt;   &lt;validPool&gt;     &lt;id&gt;\${poolId}&lt;/id&gt;   &lt;/validPool&gt; &lt;/ResourceBinding&gt;</pre>

	<pre> &lt;/validPool&gt;  &lt;/validProvider&gt;  &lt;/ResourceBinding&gt; </pre>
--	---

Table 59 Output

Parameter	Description
document	Updated Resource Binding xml document from CSA

## Troubleshooting

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.

### General troubleshooting procedures and tools

This section describes the troubleshooting procedures and tools you can use to fix problems that you may experience while using this integration.

General troubleshooting for REST API calls into CSA should including reviewing the webapp logfiles, which are present by default in C:\Program Files\Hewlett-Packard\CSA\jboss-as-7.1.1.Final\standalone\log\

To enable additional logging for CSA REST calls, add this line to the log4j.properties file located here: C:\Program Files\Hewlett-Packard\CSA\jboss-as-7.1.1.Final\standalone\deployments\csa.war\WEB-INF\classes

```
log4j.logger.com.hp.csa.rest=INFO
```

Errors that occur with calls to the REST API are logged in the csa.log file in the logfile directory.

### Error messages

This section lists the error messages you may receive while using this integration.

As this integration content wraps calls into a REST API exposed by CSA, the error messages you'll find in the OO Central run logs from these calls will refer to standard REST API errors. Common errors for the CSA Artifact API include:

- 200 - OK (Returned an artifact. In this context, that's generally a service component)
- 401 - Unauthorized
- 404 - Object not found
- 500 - Server exception

## Security

This section describes how security is handled by this integration.

CSA REST API calls work through interfaces exposed by web applications running in a JBOSS application server. CSA accepts SSL connections secured with a basic authentication method. CSA integration content use credentials from the OO system accounts credential store to populate the user and password to authenticate calls.

The system account CSA\_REST\_CREDENTIALS contains a generic utility account for accessing the CSA REST API from OO integration workflows.

Other forms of authentication can be supported by both the OO HTTP client that the integration content uses, and by CSA. Configuration of those methods is not in scope for this document.

For authorization, calls to the CSA application require a "userIdentifier" parameter to establish a context for authorizing an operation. In order to obtain a userIdentifier, you must first use the CSA\_REST\_CREDENTIALS to make a call to the /login REST API.

In this integration content, the “Get User Identifier” subflow performs this function, and returns the userIdentifier required to make subsequent calls to the API.

## OO tools

You can use the following OO tools with this 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 workflow without using Central (although the Central service must be running). RSFlowInvoke is useful when you want to start a workflow from an external system, such as a monitoring application that can use a command line to start a workflow.

- Web Services Wizard (wswizard.exe)

When you run the Web Services Wizard, you provide it with the WSDL for a given Web service. The WSDL string you provide as a pointer can be a file’s location and name or a URL. The Web Services Wizard displays a list of the methods in the API of the Web service that you specify. When you run the wizard, pick the methods you want to use, and with one click for each method you have selected, the wizard creates an OO operation that can execute the method. This allows you to use the Web Services Wizard to create operations from your monitoring tool’s API.

These tools are available in the OO home folder in /Studio/tools/.

# Send documentation feedback

If you have comments about this document, you can send them to [clouddocs@hpe.com](mailto:clouddocs@hpe.com).

## 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

© Copyright 2017 Hewlett Packard Enterprise Development Company, L.P.

### Trademark notices

Adobe® is a trademark of Adobe Systems Incorporated.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

RED HAT READY™ Logo and RED HAT CERTIFIED PARTNER™ Logo are trademarks of Red Hat, Inc.

The OpenStack word mark and the Square O Design, together or apart, are trademarks or registered trademarks of OpenStack Foundation in the United States and other countries, and are used with the OpenStack Foundation's permission.

### Documentation updates

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

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

To check for recent updates or to verify that you are using the most recent edition of a document, go to the following URL and sign-in or register: <https://softwaresupport.hpe.com>.

Select Manuals from the Dashboard menu to view all available documentation. Use the search and filter functions to find documentation, whitepapers, and other information sources.

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

### Support

Visit the Hewlett Packard Enterprise Software Support Online web site at <https://softwaresupport.hpe.com>.