

HP Cloud Service Automation 4.00

Custom Resource Provider Selection



Table of contents

Introduction	2
Prerequisites	2
Concept of Provider Selection	2
Build Resource Provider List.....	2
Select Resource Provider	2
Provider Selection Example	2
Steps to configure Provider Selection	3
Custom Provider Selection	6
Custom Provider Selection Example.....	7
REST APIs for Custom Provider Selection.....	10
Get the list of candidate providers	10
Update the list of valid providers.....	10
Review: Custom Provider Selection.....	11
Appendix A - Custom Provider Selection HP OO Flow.....	12
Get Organization Details.....	15
Refine Valid Provider List based on Property	16
Select Provider based on Property	16
Update Valid Providers to Resource Binding	19
For more information	21

Introduction

This document describes how to implement custom resource provider selection. HP CSA comes with sample resource provider selection actions. These actions select a list of resource providers that are capable of providing a resource offering and then select one provider from this list. This selection functionality can be further refined by filtering the list of providers based on a desired condition, before the final resource provider is selected.

In this white paper, any reference to the term provider implies reference to a resource provider.

Prerequisites

This document is targeted at individuals who are familiar with the basic concepts of HP CSA. Experience with HP Operations Orchestration (HP OO) flow authoring is also required.

Note

The *HP CSA Concepts Guide* includes information on basic HP CSA concepts, including a master glossary of HP CSA-related terminology.

Concept of Provider Selection

Provider selection refers to the selection of a resource provider from the list of resource providers that are capable of providing a resource offering. This selection happens during service provisioning, when there is a subscription to a cloud service. HP CSA contains out-of-the-box internal actions to support provider selection. These actions are configured on a resource binding. A step by step example is presented in this document to illustrate the steps involved in configuring provider selection.

The following internal actions are used during provider selection:

- Build Resource Provider List
- Select Resource Provider

Build Resource Provider List

This action builds a candidate list of resource providers that meet the following requirements:

- They are capable of providing the resource offering referenced in a resource binding
- They belong to at least one of the resource environments that are associated with the corresponding service catalog. Corresponding service catalog refers to the catalog through which we publish the service offering related to the service design having this action.
- They have the availability option set to Enabled

This action also sets the valid list of resource providers to the list resulting from this action.

Select Resource Provider

This action selects a resource provider from the valid list of resource providers built by the previous action. The resource provider is selected randomly from the list of valid providers. The selected provider will be written to a property on the associated service component. Users specify the name of this property through a well-known property of this action.

Provider Selection Example

Let us assume the following scenario: we want to design a service to provision a simple compute server using HP Matrix Operating Environment (HP MOE). We have access to three resource providers for this purpose.

Using this scenario, we will walk through a step by step example. The purpose of this example is to help the reader understand how to configure provider selection actions and to see how a service subscription results in the selection of a provider. This example is not a best practice document of how to design a service, which is beyond the scope of this document.

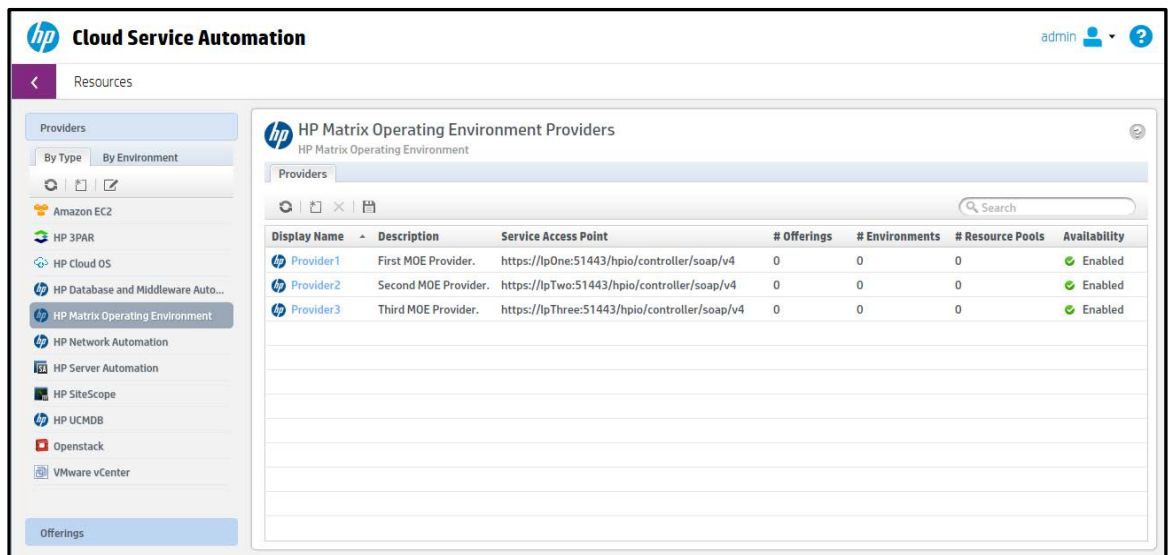
The steps will not illustrate basic HP CSA operations (for example adding a resource provider, publishing a service offering etc.) To perform these basic operations, refer to the *Cloud Service Management Console Help* document.

Since HP CSA contains out-of-the-box sample content for HP MOE, you will import it into the Designs area of the Cloud Service Management console and use it to illustrate the steps. Import the CSA_BP_MOE_COMPUTE_CUSTOM_PROVIDER_SELECTION content now. The content can be found at the location: %CSA_HOME%/CSAKit-<version>/Content Archives/matrix operating environment.

Steps to configure Provider Selection

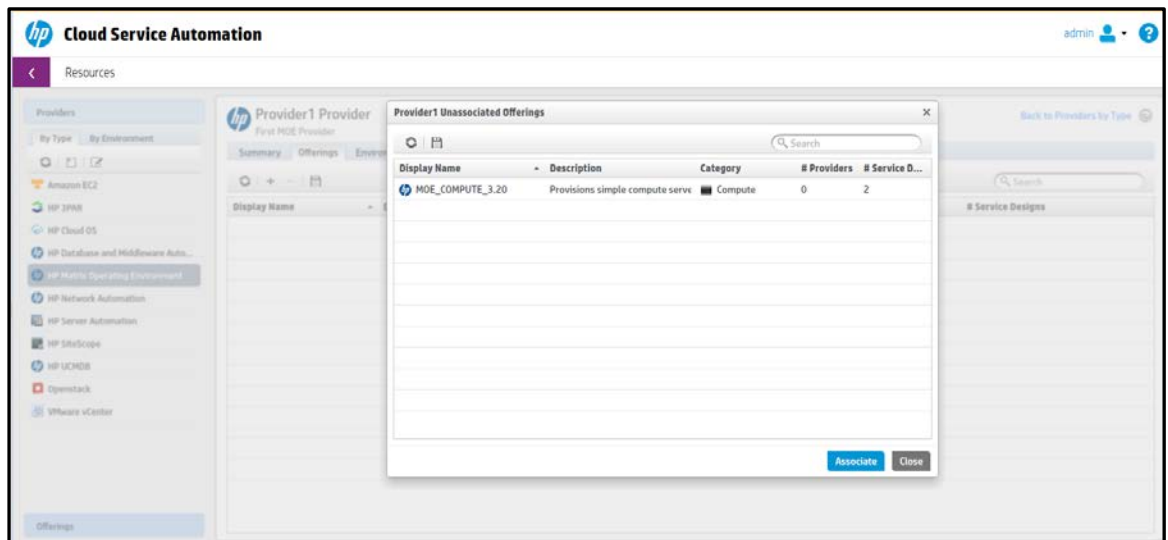
1. Add available resource providers.
Since we have three HP MOE providers, add them to the list of available providers, naming them Provider1, Provider2 and Provider3.

Figure 1. HP MOE Resource Providers



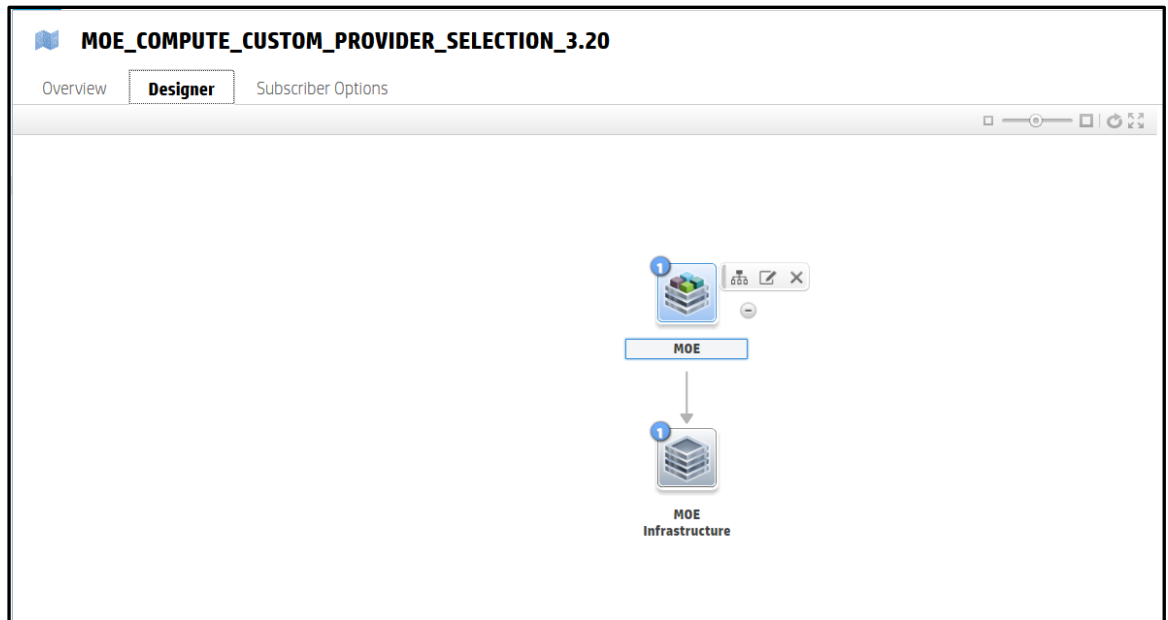
2. Associate resource offerings to resource providers capable of offering the resource.
Since we have three HP MOE providers capable of provisioning a simple compute server, let us associate these three providers with the MOE_COMPUTE resource offering, which is available as supplied content. This is done on a per provider basis. In Figure 2, the MOE_COMPUTE offering is being associated with Provider 1.

Figure 2. Provider 1 Association with the HP MOE Compute Resource Offering



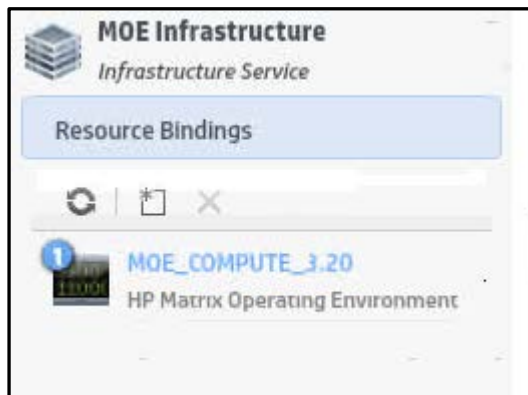
3. View the service to provision a simple compute server in the Designer.
A service that provisions a simple compute server using HP MOE is included as out-of-the-box content with HP CSA, and can be viewed in the Designer, as shown in Figure 3. Note that this design contains a service composite component (MOE) and an infrastructure service component (MOE Infrastructure) as its child.

Figure 3. Provided HP MOE Service Design



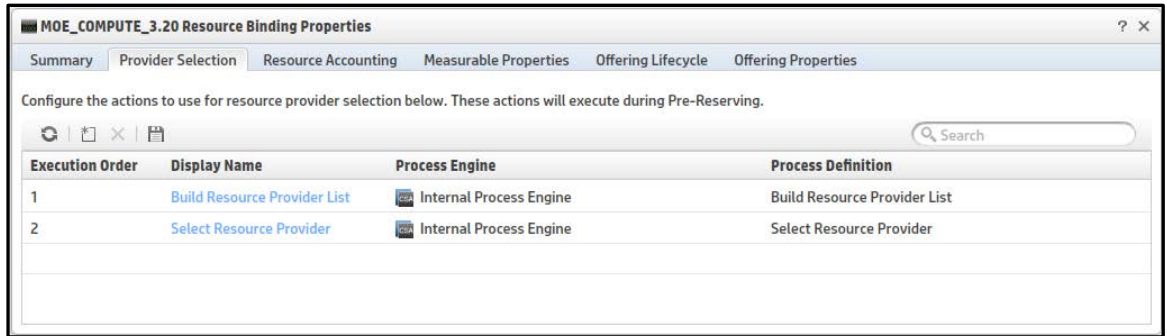
4. Associate the service component to the resource offering.
As stated in Step 2, the MOE_COMPUTE resource offering is available as supplied content with HP CSA. In this step, we associate this resource offering to the infrastructure service (MOE Infrastructure) component through a resource binding. A resource binding is a link between a resource offering and a service component. It ensures that the resource offering will be provisioned when the service component is deployed due to a user's subscription.

Figure 4. Resource Binding to the MOE_COMPUTE Resource Offering



5. Create provider selection actions for the resource bindings.
The next step is to create provider selection actions (discussed in the [Concept of Provider Selection](#) section) for the resource binding. When there is a subscription to the service, HP CSA creates a service instance which moves through the HP CSA lifecycle phases. During that time, provider selection actions will be executed during the Reserving – Pre-Transition lifecycle phase and will result in selection of a resource provider to provide the resource offering – the provisioning of the simple compute server.

Figure 5. Provider Selection Actions on the HP MOE Resource Binding

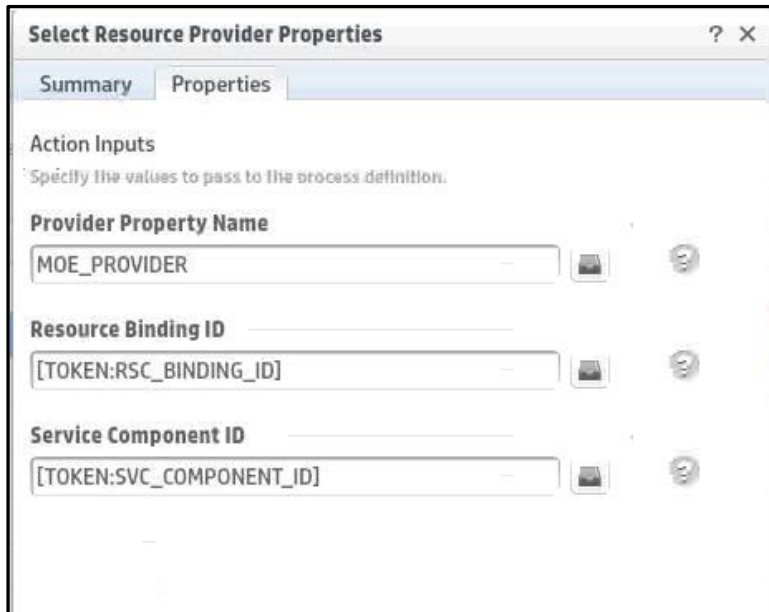


6. Specify the properties of the internal actions.

The Select Resource Provider action has a property named Provider Property Name. With this property, the user can specify the name of the property in which the selected resource provider ID will be set as a property of the infrastructure service component. For example, if the user specifies MOE_PROVIDER as the value for this property, then the selected resource provider will be set as a property on the service component as follows:

Provider=ID of Selected Provider.

Figure 6. Provider Property Name



After the above steps are performed, the service is ready to be associated with a service offering and published to a catalog. After a user subscribes to this service, HP CSA creates a service instance which moves through the HP CSA lifecycle phases. When the provider selection actions are executed during the Reserving – Pre-Transition lifecycle phase, then the following sequence of operations takes place:

- The Build Resource Provider List action runs, which does the following:
 - A. Gets the list of resource providers associated with the resource offering corresponding to the resource binding. This returns Provider1, Provider2, and Provider3.
 - B. Gets the list of resource environments associated with the Service catalog (which contains the service offering based on our HP MOE service design).
 - C. Filters the list returned above to include only those providers which are associated with the resource environments of the Service Catalog. For example, assume that the catalog is associated with a resource environment that contains two resource providers – Provider1 and Provider2. From the list returned above, Provider 3 will not be included. So the resulting list will include just Provider1 and Provider2.
 - D. Checks the list of providers returned above to see if each provider is enabled, and includes only the enabled providers. In our example, all providers are enabled, so both Provider1 and Provider2 will be returned.

- E. Finally sets the list of candidate providers to the list returned above. It also sets the list of valid providers to the same list. In our example, this will be set to Provider1 and Provider2.
- The Select Resource Provider action runs, which does the following:
 - A. Gets the list of valid providers built by the Build Resource Provider List action. This includes Provider1 and Provider2.
 - B. Selects a resource provider based on random selection from this list. Let us assume the selected provider is Provider1.
 - C. Sets the provider selected above as the selected resource provider of the resource binding. In our example, this is Provider1.
 - D. Sets the ID of the selected provider as a property of the service component. The name of this property is the user entered value for Provider Property Name when this action was added to the resource binding (See Figure 6). In our example, this is set as Provider= 90e72e4f3b622418013b686efbd200c7 (See Figure 7).

Figure 7. Provider Property of Service Component

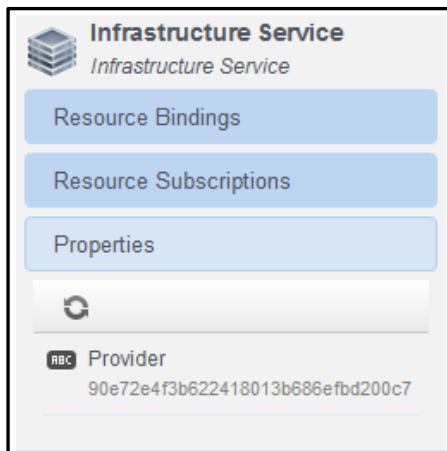
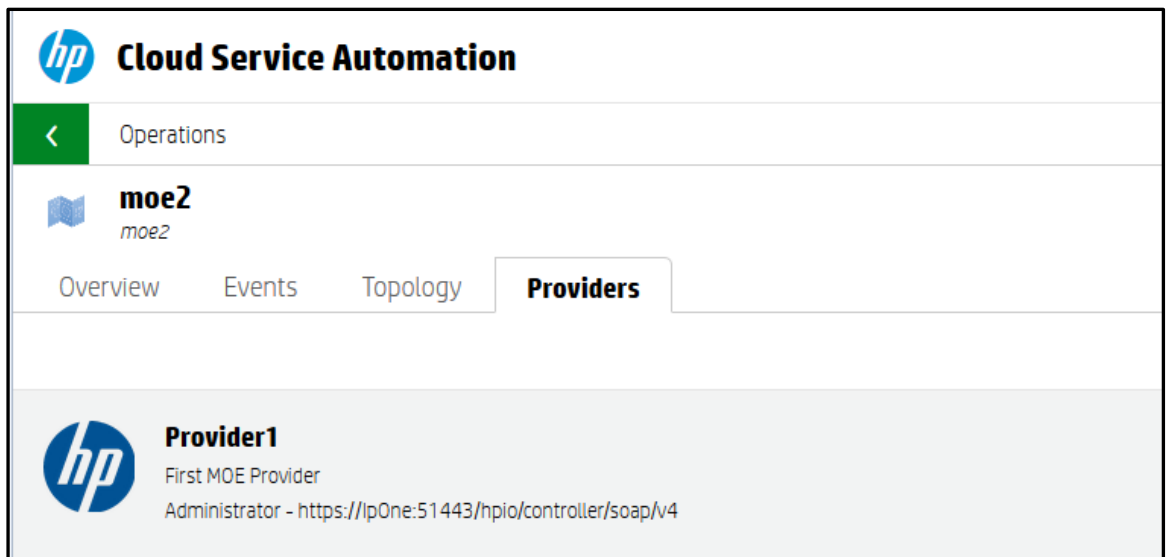


Figure 8. Selected Resource Provider for HP MOE subscription in Service Operations Console



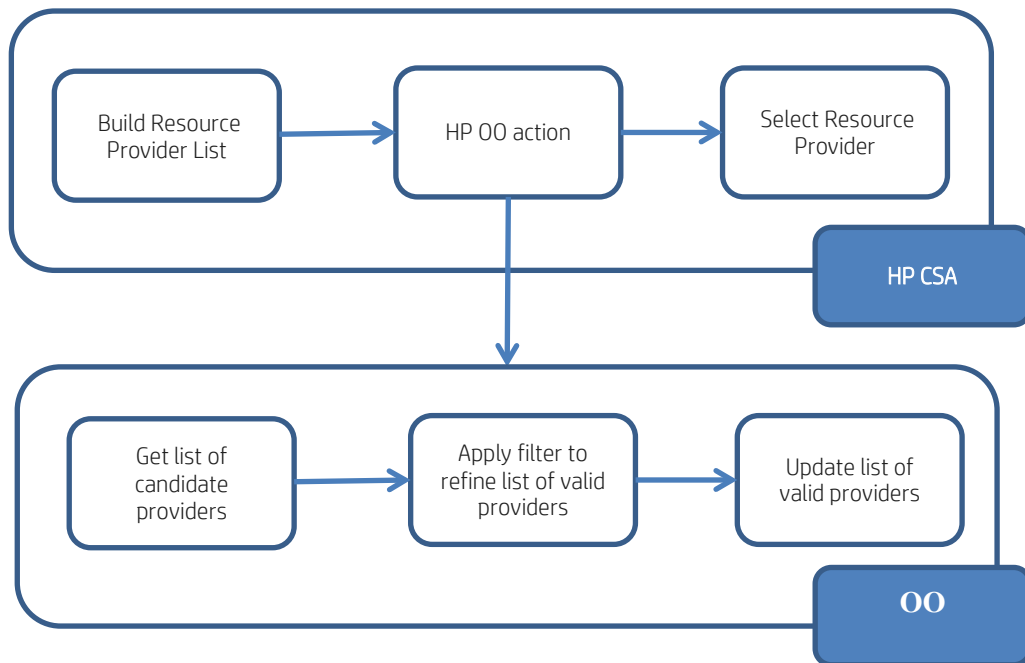
Custom Provider Selection

Using custom provider selection, the default provider selection functionality described above can be refined to further filter the list of resource providers based on some condition decided by the user.

This requires that the initial list of resource providers built by the Build Resource Provider List action be further refined before the Select Resource Provider action chooses the resource provider. This can be achieved using an HP OO flow which

performs the filtering functionality. After importing this flow into HP CSA, the corresponding HP CSA action can be inserted between the Build Provider List and Select Resource Provider actions. Figure 9 illustrates this concept.

Figure 9. Implementing Custom Provider Selection



As shown in Figure 9, the HP OO action performs three major functions:

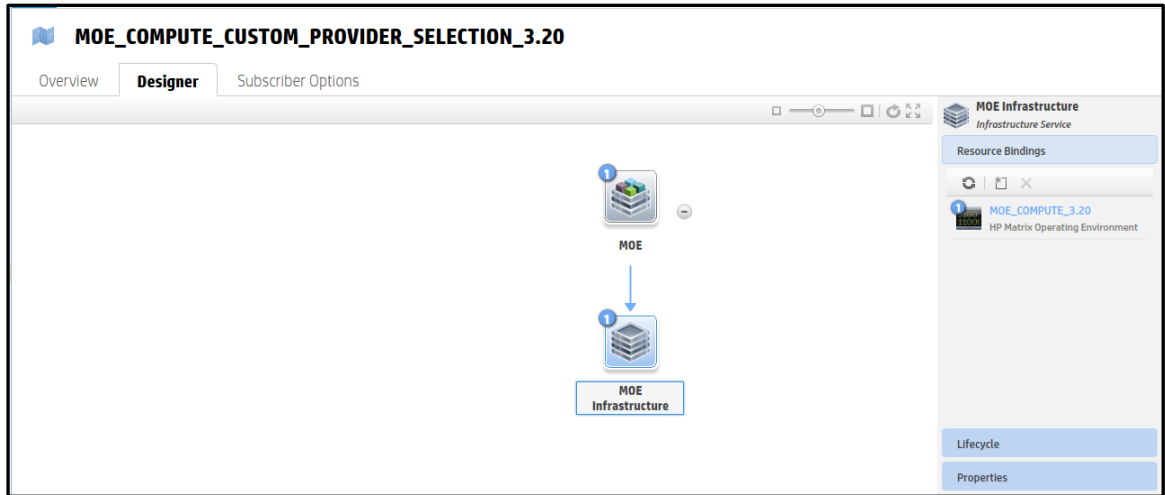
- Gets the list of candidate providers built by the Build Resource Provider List action.
- Refines the list with the filtering condition decided by the user.
- Updates the list of valid providers in HP CSA. This updated list is used as the basis for selection by the Select Resource Provider action.

To enable HP OO to interact with HP CSA, HP CSA exposes appropriate REST APIs. To get more information on REST APIs, refer to the *HP CSA API Reference*. The relevant REST APIs to implement custom provider selection are explained in the example in the next section.

Custom Provider Selection Example

For this example, we will use the MOE_COMPUTE_CUSTOM_PROVIDER_SELECTION service design, which is part of the sample content shipped with HP CSA.

Figure 10. Sample HP MOE Service Design for Custom Provider Selection



From Figure 10, we can see that the “MOE Infrastructure” component has a resource binding to the MOE_COMPUTE resource offering. Let us look at the provider selection actions configured on this resource binding.

Figure 11. Provider Selection Actions

Execution Order	Display Name	Process Engine	Process Definition
1	Build Resource Provider List	Internal Process Engine	Build Resource Provider List
2	User Organization based MOE Provider Selection	OO-MACHINE-NAME	User Organization based MOE Provider Selection
3	Select Resource Provider	Internal Process Engine	Select Resource Provider

Compared to our previous example, this resource binding has an additional action between the Build Resource Provider List and Select Resource Provider actions. This is an HP OO workflow which refines the list of providers further before the final resource provider is selected.

This action is available because the corresponding HP OO flow has been imported into HP CSA. To implement custom provider selection, we need to define our own HP OO flow and import it into HP CSA. For details on how to import HP OO flows into HP CSA refer to the *HP CSA Configuration Guide*.

At a broad level, the HP OO flow corresponding to this action performs three major functions:

1. Gets the list of candidate providers using the HP CSA REST API. Refer to the [REST APIs for Custom Provider Selection](#) section for more details.
2. Refines the list of providers based on some condition. This condition depends on the specific use case of the customer. For instance, taking our previous example, we may want to provide services to users from two organizations – Organization A, and B. We want Provider1 and Provider2 to provide resources for users from Organization A and Provider3 to provide resources to users from Organization B. To implement this condition, we can define a property on all resource providers to indicate the acceptable organization (Organization=A for Provider1 and Provider2, Organization=B for Provider3). In the HP OO flow, the list returned in Step1 should be filtered to include only those providers whose Organization property value includes the requesting user’s organization. If the requesting user belongs to Organization A, then only Provider1 and Provider2 will be set as the valid providers. At the end of the refinement process, a subset of the providers will be valid choices.
3. Updates the list of valid providers with the list returned in Step 2 using the HP CSA REST API. Refer to the [REST APIs for Custom Provider Selection](#) section for more details.

Details of how to create an HP OO flow are beyond the scope of this document. To learn how to create HP OO workflows, see the *HP Operations Orchestration Studio Authoring Guide*. A basic overview of the HP OO flow (User Organization based MOE Provider Selection) is presented in Appendix A. This flow is shipped as sample content with HP CSA.

REST APIs for Custom Provider Selection

Get the list of candidate providers

To get the list of candidate providers, use the GET artifact API.

URI:

```
https://localhost:8444/csa/rest/artifact/<Resource_Binding_ID>?userIdentifier=<User_ID>&scope=view&view=candidateproviders
```

Method: GET

Response:

```
<ResourceBinding>
  <id>90e72e4f3b622418013b6d1cde3e0162</id>
  <objectId>90e72e4f3b622418013b6d1cde3e0162</objectId>
  <isCriticalSystemObject>>false</isCriticalSystemObject>
  <description>binding_December 5, 2012 7:22:43 PM UTC</description>
  <name>binding_December 5, 2012 7:22:43 PM UTC</name>
  <displayName>binding_December 5, 2012 7:22:43 PM UTC</displayName>
  .....
  <candidateProvider>
    <id>90e72e4f3b622418013b686efbd200c7</id>
    <objectId>90e72e4f3b622418013b686efbd200c7</objectId>
    <isCriticalSystemObject>>false</isCriticalSystemObject>
    <name>Provider1_December 5, 2012 12:20:01 AM UTC</name>
    <displayName>Provider1</displayName>
    <property>
      <id>90e72e4f3b622418013b6df6fc440210</id>
      <objectId>90e72e4f3b622418013b6df6fc440210</objectId>
      <isCriticalSystemObject>>false</isCriticalSystemObject>
      <name>ORGANIZATIONS</name>
      <displayName>Organizations</displayName>
      <values>
        <value>A</value>
      </values>
      <maxOccurs>1</maxOccurs>
      <minOccurs>1</minOccurs>
      <orderIndex>0</orderIndex>
      <confidential>>false</confidential>
      <encrypted>>false</encrypted>
      <consumerReadOnly>>true</consumerReadOnly>
      <consumerVisible>>false</consumerVisible>
    </property>
    <disabled>>false</disabled>
    <numberOfResourceOffering>0</numberOfResourceOffering>
    <numberOfEnvironment>0</numberOfEnvironment>
    <numberOfPools>0</numberOfPools>
  </candidateProvider>
  <candidateProvider> ... </candidateProvider>
  <candidateProvider> ... </candidateProvider>
</ResourceBinding>
```

Update the list of valid providers

To update the list of valid providers, use the PUT artifact API.

URI:

```
https://localhost:8444/csa/rest/artifact/<RESOURCE_BINDING_ID>?userIdentifier=<User_ID>&scope=view&view=validproviders
```

METHOD: PUT

Body: Include the refined list of valid providers. In this example, we have refined the list to remove Provider3.

```
<ResourceBinding>
  <id>90e72e243992d89e013992fc10370056</id>
  <validProvider>
    <id>90e72e4f3b622418013b686efbd200c7</id>
  </validProvider>
  <validProvider>
    <id>90e72e4f3b622418013b686f5b5e00ca</id>
  </validProvider>
</ResourceBinding>
```

Response:

```
<ResourceBinding>
  <id>90e72e4f3b622418013b6d1cde3e0162</id>
  <objectId>90e72e4f3b622418013b6d1cde3e0162</objectId>
  <isCriticalSystemObject>false</isCriticalSystemObject>
  <name>binding_December 5, 2012 7:22:43 PM UTC</name>
  <displayName>binding_December 5, 2012 7:22:43 PM UTC</displayName>
  .....
  <validProvider>
    <id>90e72e4f3b622418013b686efbd200c7</id>
    <objectId>90e72e4f3b622418013b686efbd200c7</objectId>
    <isCriticalSystemObject>false</isCriticalSystemObject>
    <name>Provider1_December 5, 2012 12:20:01 AM UTC</name>
    <displayName>Provider1</displayName>
    <disabled>false</disabled>
    <numberOfResourceOffering>0</numberOfResourceOffering>
    <numberOfEnvironment>0</numberOfEnvironment>
    <numberOfPools>0</numberOfPools>
  </validProvider>
  <validProvider>
    <id>90e72e4f3b622418013b686f5b5e00ca</id>
    <objectId>90e72e4f3b622418013b686f5b5e00ca</objectId>
    <isCriticalSystemObject>false</isCriticalSystemObject>
    <name>Provider2_December 5, 2012 12:20:26 AM UTC</name>
    <displayName>Provider2</displayName>
    <disabled>false</disabled>
    <numberOfResourceOffering>0</numberOfResourceOffering>
    <numberOfEnvironment>0</numberOfEnvironment>
    <numberOfPools>0</numberOfPools>
  </validProvider>
</ResourceBinding>
```

Review: Custom Provider Selection

Following are the major steps that occur to set up HP CSA for custom provider selection:

- User decides the condition on which the resource providers should be filtered.
- User creates the corresponding HP OO flow to implement refinement of the list of valid providers, based on the above condition. The HP OO flow does the following:
 - Gets the list of candidate providers through an HP CSA REST API. The list is built by the Build Resource Providers List action.
 - Filters providers based on the condition.
 - Updates the list of valid providers through an HP CSA REST API. This list will be used by the Select Resource Provider action to select the final resource provider.
- User imports the HP OO flow into HP CSA. This makes the HP OO flow available as an HP CSA action.
- The user inserts the action in the Provider Selection actions of the resource binding. The action is inserted between the Build Resource Provider List action and the Select Resource Provider action.

Appendix A - Custom Provider Selection HP OO Flow

This section provides a brief overview of how custom provider selection is implemented in the User Organization based MOE Provider Selection HP OO flow. This flow is provided as sample content with HP CSA. Refer to the *HP CSA Configuration Guide* for information on how to configure HP OO to obtain the flows.

Figure 12. Custom Provider Selection HP OO flow for HP MOE

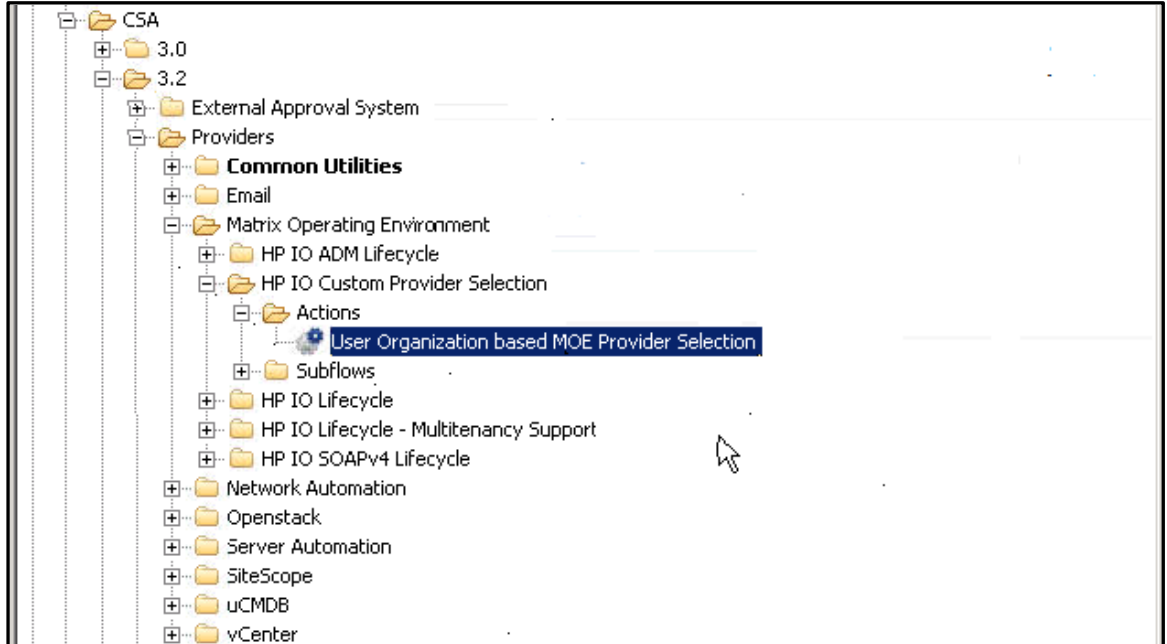


Figure 13 shows the HP CSA action used in the HP MOE design for performing custom provider selection. The Process Definition field shows the details of the corresponding HP OO flow.

Figure 13. HP CSA Action for Custom Provider Selection

User Organization based MOE Provider Selection Properties ? X

Summary Properties

Action Properties
Configure the properties for this action below

Process Engine
OO-MACHINE-NAME

Process Definition
User Organization based MOE Provider Selection
/Library/CSA/3.2/Providers/Matrix Operating Environment/HP IO Custom...

Display Name *
User Organization based MOE Provider Selection

Description
[Empty text box]

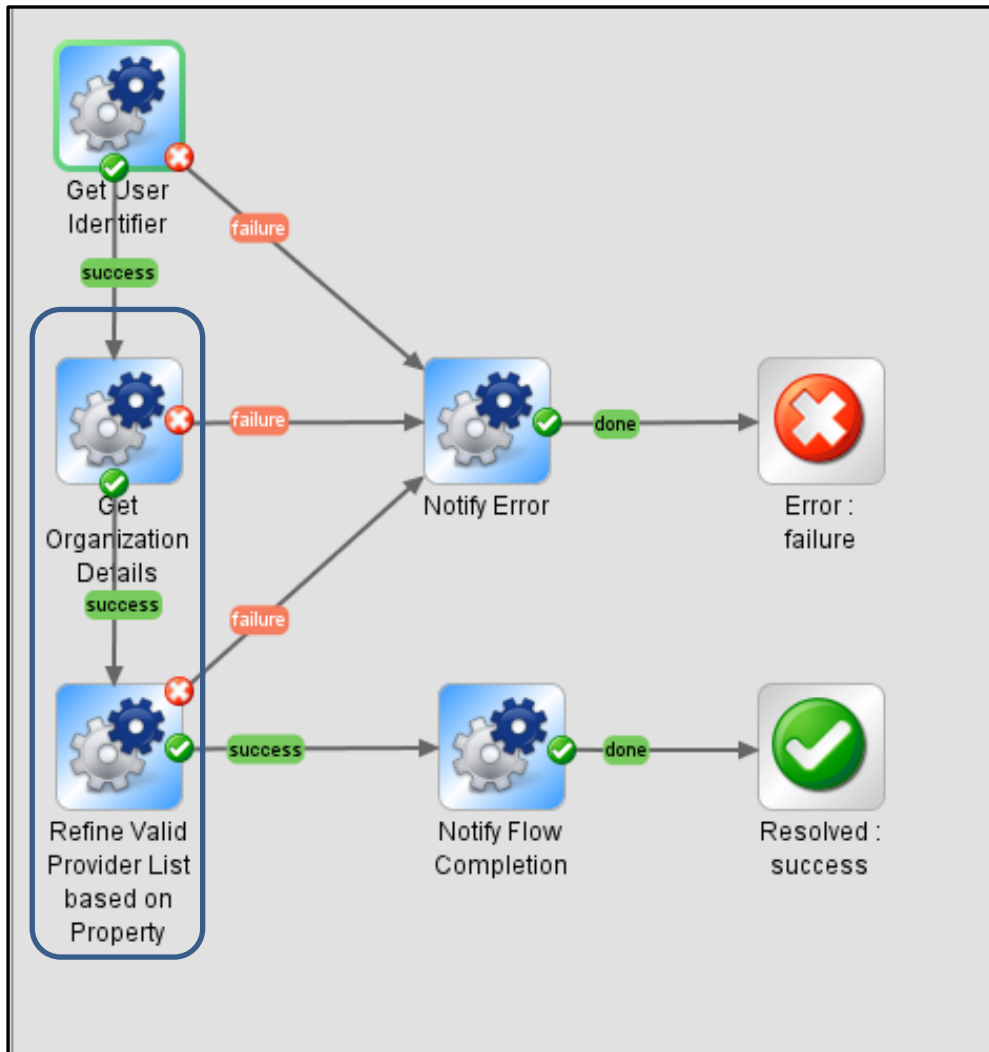
Execution Order
2

Execution Properties
 Fail on Error Error on Timeout

Save Reset Close

Figure 14 shows the HP OO flow invoked from HP CSA corresponding to the action shown in Figure 13. The steps which implement the refinement logic are shown in the blue box in Figure 14. For this flow, the refinement condition is based on the requesting user's organization. Each resource provider has an HP CSA property called ORGANIZATIONS which specifies a list of organizations. A provider will be included in the list of valid providers only if its ORGANIZATIONS property includes the requesting user's organization.

Figure 14. User Organization-Based MOE Provider Selection Flow



Get Organization Details

This step gets the requesting user's organization (organization name). This is implemented by using the Organization REST API from HP CSA. Figure 15 shows the URI used by the sub flow corresponding to this step.

Figure 15. Get Organization Details Sub flow

The screenshot displays a workflow editor interface. At the top, a sub-flow diagram shows an 'Http Client' step (represented by a globe icon) with a 'success' label. This step branches into two paths: 'Error: failure' (represented by a red 'X' icon) and 'Resolved: success' (represented by a green checkmark icon). Below the workflow is an 'Inspector' panel for the 'Http Client Get' step. The 'Inspector' panel shows the following configuration:

- Step Name: Http Client Get
- Inputs Summary: url
- Name: url
- Input Type: Single Value
- Input Data Flow: Assign from Variable: <not assigned>, Otherwise: Use Constant, Assign to Variable: <not assigned>
- Input Properties: Encrypted (unchecked), Required (checked)
- Validation Format: <not validated>
- Record Under: <run history>

The 'Otherwise: Use Constant' configuration is expanded, showing the following constant value:

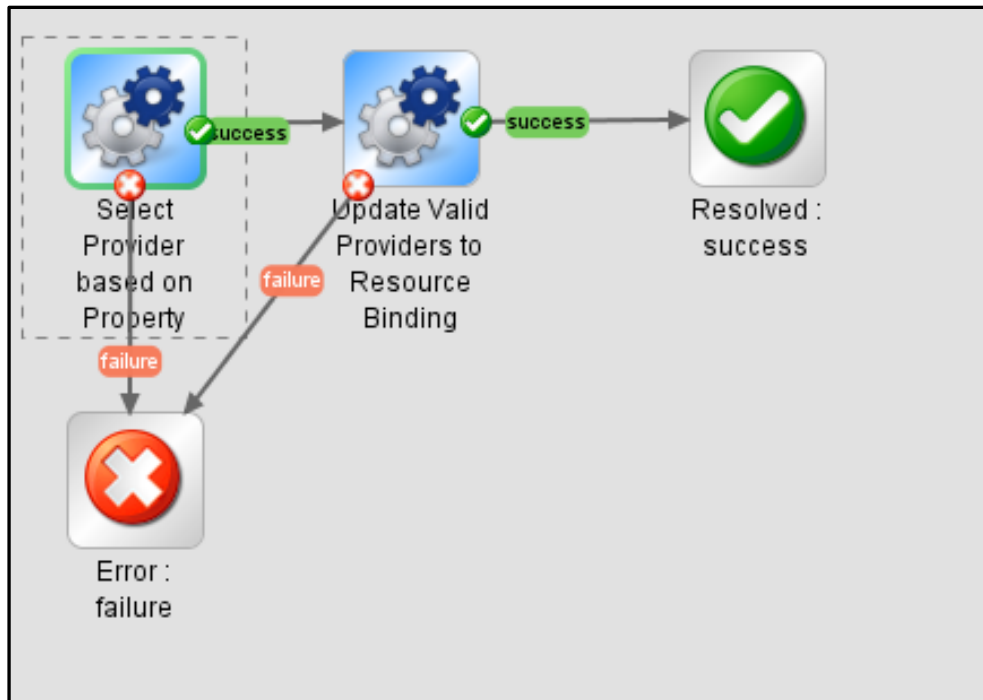
```
$(CSA_REST_URI)/organization/${organizationId}?userIdentifier=${userIdentifier}&scope=base&detail=basic
```

Refine Valid Provider List based on Property

The requesting user's organization value is used by this step to filter only those providers that can support the user's organization. Figure 16 shows the sub flow corresponding to this step:

- *Select Provider based on property:* This sub flow step gets the list of candidate providers and filters the providers based on the value of their ORGANIZATIONS property.
- *Update Valid providers to Resource binding:* This sub flow step updates the list of valid providers in HP CSA with the list built from the previous step.

Figure 16. Refine Valid Provider List based on Property Sub flow

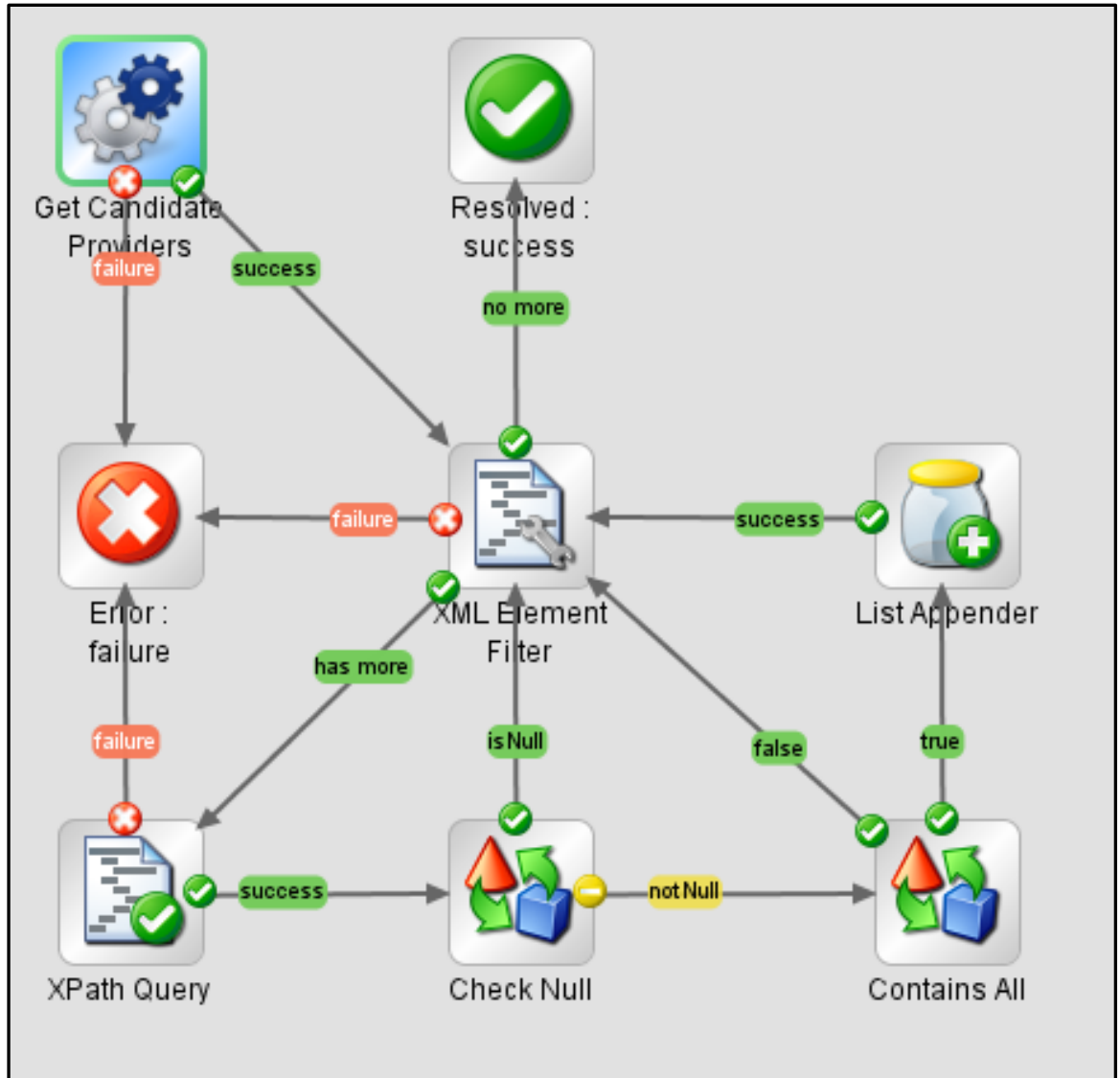


Select Provider based on Property

The sub flow corresponding to this step is shown in Figure 17 and has the following major steps:

- *Get Candidate Providers:* This step uses the HP CSA Artifact REST API to get the list of candidate providers for the resource binding.
- *XML Element Filter:* This step looks for the ResourceBinding/candidateProvider element of the result from the previous step.
- *XPath Query:* For each candidate provider, this step queries the ORGANIZATIONS property value.
- *Contains All:* This step checks if the requesting user's organization value is contained in the ORGANIZATIONS property of the resource provider.
- *List Appender:* This step appends the resource provider to the list of valid resource providers if the previous step returns true (when the resource provider's ORGANIZATIONS property has the requesting user's organization name).

Figure 17. Select Provider Based on Property Sub flow



The Get Candidate Providers sub flow uses the HTTP Client GET operation. Figure 18 shows the HP CSA REST URI used with the candidateproviders view.

Figure 18. Get Candidate Providers Sub flow

The image shows a workflow diagram at the top and its configuration inspector below. The workflow consists of three steps: 'Http Client Get' (a globe icon with a green checkmark), 'Resolved: success' (a green checkmark icon), and 'Error: failure' (a red 'X' icon). A green arrow labeled 'success' points from 'Http Client Get' to 'Resolved: success', and a red arrow labeled 'failure' points from 'Http Client Get' to 'Error: failure'.

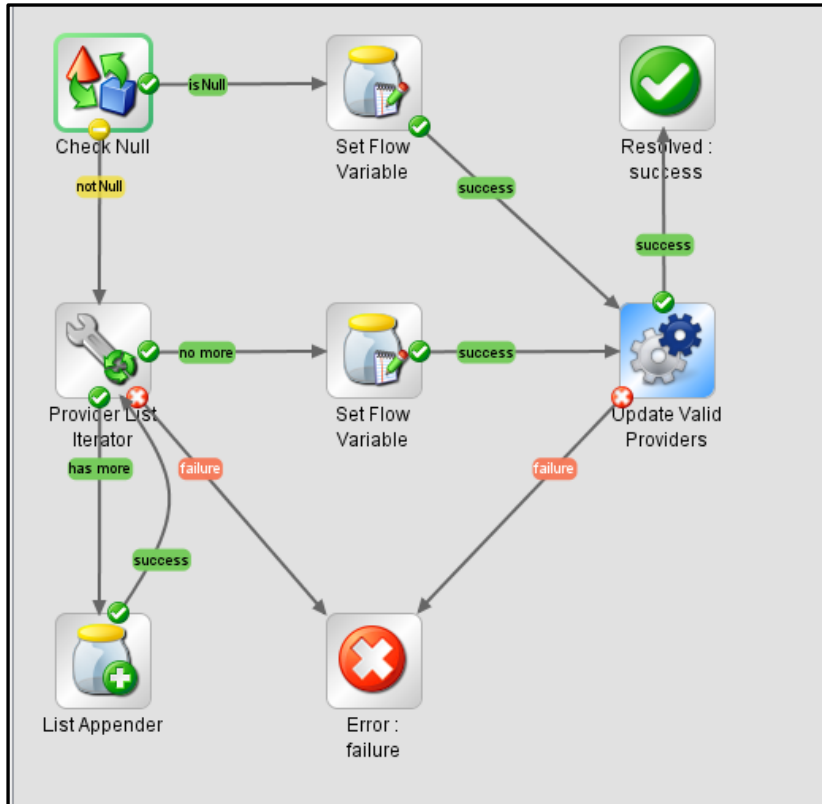
The 'Inspector' window below shows the configuration for the 'Http Client Get' step. The 'Step Name' is 'Http Client Get'. The 'Inputs' tab is active, showing the 'url' input. The 'Name' is 'url' and the 'Input Type' is 'Single Value'. Under 'Input Data Flow', 'Assign from Variable' is '<not assigned>', 'Assign to Variable' is '<not assigned>', and 'Otherwise: Use Constant' is selected. The 'Constant Value' is `$(CSA_REST_URI)/artifact/${resourcebindingId}?userIdentity=${userIdentity}&scope=view&view=candidateproviders`. Under 'Input Properties', 'Encrypted' is unchecked and 'Required' is checked. 'Validation Format' is '<not validated>' and 'Record Under' is '<run history>'.

Update Valid Providers to Resource Binding

The sub flow corresponding to this step is shown in Figure 19 and has the following major steps:

- *Provider List Iterator* and *List Appender*: Together these two steps iterate over each of the valid providers filtered by the Select Providers based on Property sub flow and create the body of the PUT request to update the resource binding.
- *Update Valid Providers*: This step uses the HP CSA Artifact REST API to update the list of valid providers for the resource binding.

Figure 19. Update Provider List to Resource Binding Sub flow



The Update Valid Providers sub flow uses the HTTP Client PUT operation. Figure 20 shows the HP CSA REST API with the validproviders view, used by this operation to update the list of valid providers.

Figure 20. Update Valid Providers Sub flow

The image shows a workflow diagram at the top and its configuration inspector below. The workflow diagram consists of three nodes: an 'Http Client' node (represented by a globe icon), a 'Resolved: success' node (represented by a green checkmark icon), and an 'Error: failure' node (represented by a red 'X' icon). A green arrow labeled 'success' points from the 'Http Client' node to the 'Resolved: success' node. A red arrow labeled 'failure' points from the 'Http Client' node to the 'Error: failure' node.

The configuration inspector below shows the details for the 'Http Client Put' step. The 'Name' is 'url' and the 'Input Type' is 'Single Value'. Under 'Input Data Flow', the 'Assign from Variable' is '<not assigned>', the 'Assign to Variable' is '<not assigned>', and the 'Otherwise: Use Constant' is selected. The 'Constant value' is highlighted with a blue box and contains the following REST API URL: `${CSA_REST_URI}/artifact/${resourceBindingId}?userIdentity=${userIdentity}&scope=view&view=validproviders`. Under 'Input Properties', the 'Required' checkbox is checked. The 'Validation Format' is '<not validated>' and the 'Record Under' is '<run history>'.

For more information

HP software product manuals and documentation for HP CSA can be found at <http://h20230.www2.hp.com/selfsolve/manuals>. You will need an HP Passport to sign in and gain access.

Note

General-access documentation requires that you register for an HP Passport and sign in. In some cases, access to the documentation is restricted and requires that you have an active HP support agreement ID (SAID) and an HP Passport sign-in.

Table 1. Document Revision History

Date or Version	Changes
June 2013	Initial release of document.
February 2014	Updated for HP CSA 4.00.

To help us improve our documents, please send feedback to CSAdocs@hp.com.

Learn more at
hp.com/go/csa

Sign up for updates
hp.com/go/getupdated

© Copyright 2013–2014 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. 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.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. RED HAT READY™ Logo and RED HAT CERTIFIED PARTNER™ Logo are trademarks of Red Hat, Inc.

February 2014

