Server and Storage Workflows for HP Insight Dynamics

Technical white paper

Table of Contents

Table of Figures

Introduction

This document is intended to provide information regarding usage and customization of the Insight Dynamics server and storage workflows. The *HP Insight Orchestration User Guide* contains basic information on using the server and storage HP Operations Orchestration (OO) workflows delivered with Insight Dynamics. This document provides information needed to utilize the reference OO workflows included with Insight Dynamics and/or to customize the supplied reference OO workflows for particular customer environments. Using the bundled OO workflows is a matter of suitable configuration and editing various files. Extending the workflows to other environments requires additional work, as described in this paper. Those sections of this paper have been written for users with considerable experience designing, developing and testing HP Operations Orchestration workflows.

The standard Insight Dynamics solution provides out-of-the-box support for HP BladeSystem c-Class servers using Virtual Connect, as well as various virtual machine hosts. The standard Insight Dynamics solution provides out-of-the-box support for the HP EVA storage solution (which can be bundled as part of HP BladeSystem Matrix). OO workflows are the means to extend that support, and Insight Dynamics includes reference implementations of OO workflows for both servers and storage.

- The server workflows included with Insight Dynamics include support for ProLiant iLO2 or later servers not using Virtual Connect. The Insight Dynamics 6.2 release adds support for Integrity iLO2 or later servers not using Virtual Connect. For HP servers not using Virtual Connect, file customizations are necessary to ensure server information can be gathered and power operations performed. Additional customizations can enable integration with a CMDB or more sophisticated handling of iLO passwords. The workflows may be extended to support other HP server types.
- The storage workflows included with Insight Dynamics include support for XP arrays. The workflows demonstrate use of the XP CLI and can be customized for other storage types (e.g., EMC) or other means of interacting with storage (e.g., SMI-S versus CLI).

The remainder of this document reviews the supplied server and storage workflows, their configuration, and provides information on how they can be customized to meet the needs of a specific customer environment.

Overview

The standard Insight Dynamics solution provides out-of-the-box support for HP BladeSystem c-Class servers using Virtual Connect and the HP EVA storage solution (which can be bundled as part of HP BladeSystem Matrix). Insight Dynamics is able to gather information regarding servers and perform power operations through Virtual Connect. Insight Dynamics is able to gather information regarding storage and, when used in conjunction with the HP Storage Provisioning Manager, perform operations such as changing hostmode and LUN masking via SMI-S.

In addition to this out-of-the-box support, Insight Dynamics supported servers and storage can be extended via HP Operations Orchestration (OO) workflows. The Insight Dynamics solution for infrastructure orchestration includes reference implementations of OO workflows for both servers and storage.

- The server workflows included with Insight Dynamics include support for ProLiant and Integrity iLO2 or later servers not using Virtual Connect, and can be customized for other HP server types as appropriate. The support for Integrity iLO2 servers not using Virtual Connect was included with the Insight Dynamics 6.2 release.

- The storage workflows included with Insight Dynamics include support for XP arrays. The workflows demonstrate use of the XP CLI and can be customized for other storage types (e.g., EMC) or other means of interacting with storage (e.g., SMI-S versus CLI).

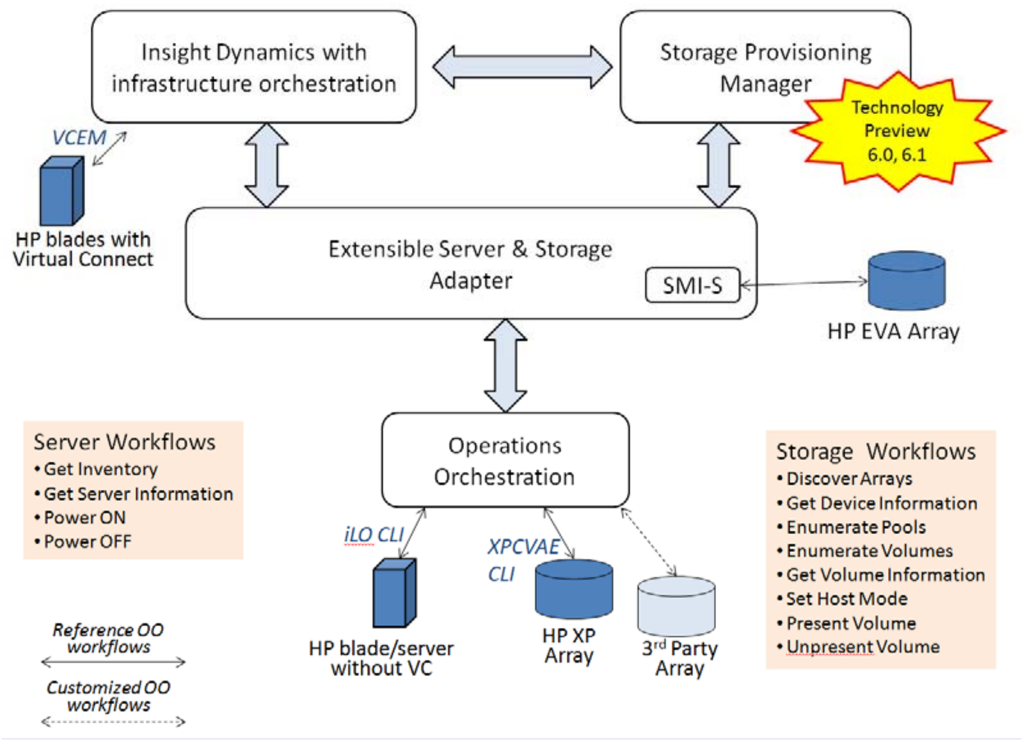
The following sections describe the Extensible Server and Storage Adapter (ESA) component of the Insight Dynamics solution, configuration steps, and how the server and storage workflows can be customized.

Extensible Server and Storage Adapter

The Insight Dynamics solution includes logical server management and infrastructure automation capabilities and an Extensible Server and Storage Adapter layer which enables support for a wide range of servers and storage. Figure 1 represents the high-level interaction between various Insight Dynamics components.

As noted, the HP Storage Provisioning Manager feature was available as a technology preview with the Insight Dynamics 6.0 and 6.1 releases, and is fully supported in the 6.2 release.

Figure 1: Insight Dynamics component interactions



The Extensible Storage and Server Adapter uses HP Operations Orchestration (OO) workflows to perform tasks (such as the server and storage workflows shown in the diagram above and discussed below). Insight Dynamics includes reference implementations which can be customized for specific environments.

The ESA server OO workflows support:

1. Discovery of servers not using Virtual Connect
2. Retrieving details of servers not using Virtual Connect
3. Powering ON the HP servers not using VC (via iLO)
4. Powering OFF the HP servers not using VC (via iLO)

The ESA storage OO workflows support:

1. Discover arrays
2. Get device information
3. Enumerate pools on device
4. Enumerate volumes on device
5. Get volume information
6. Present LUN
7. Unpresent LUN
8. Set hostmode
9. ValidateManagement Path

ESA will call OO in an asynchronous fashion (and OO workflows could return quickly or run for longer periods of time). ESA will invoke a workflow, passing the appropriate input data and an ESA_ID (for correlation). Once the specific OO workflow is completed, it does a callback to ESA to send back the response (including the ESA_ID).

Figures 2 and 3 below show the OO workflow and the interactions with ESA to receive the input and invoke the callback mechanism.

Figure 2: Reference OO workflow asynchronous implementation

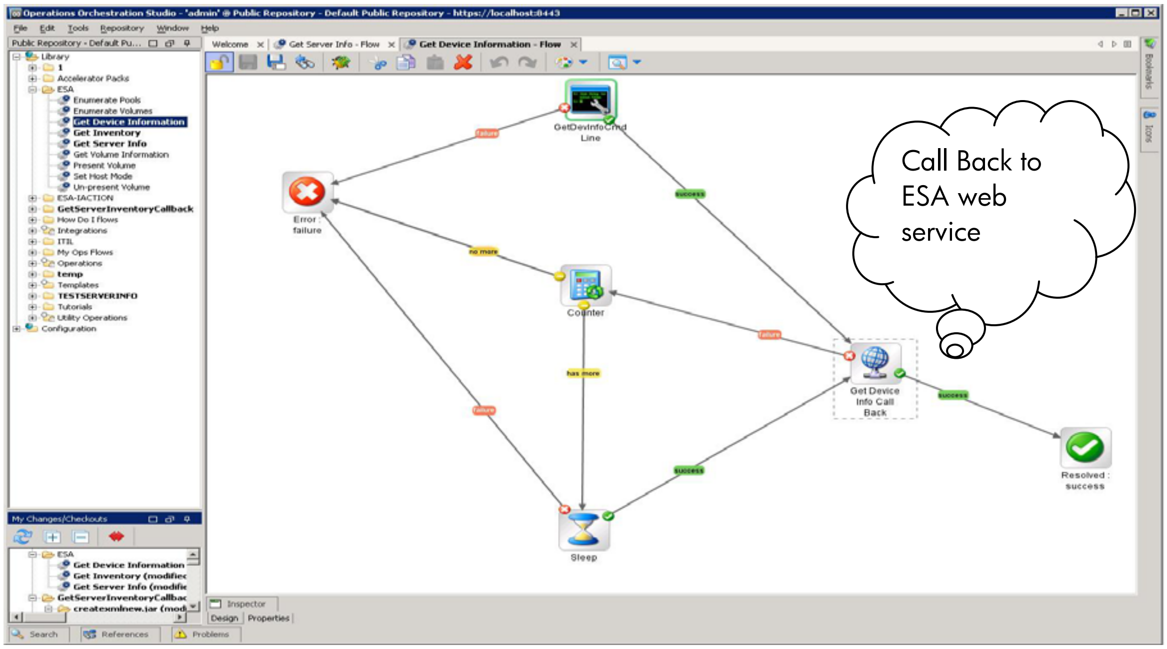
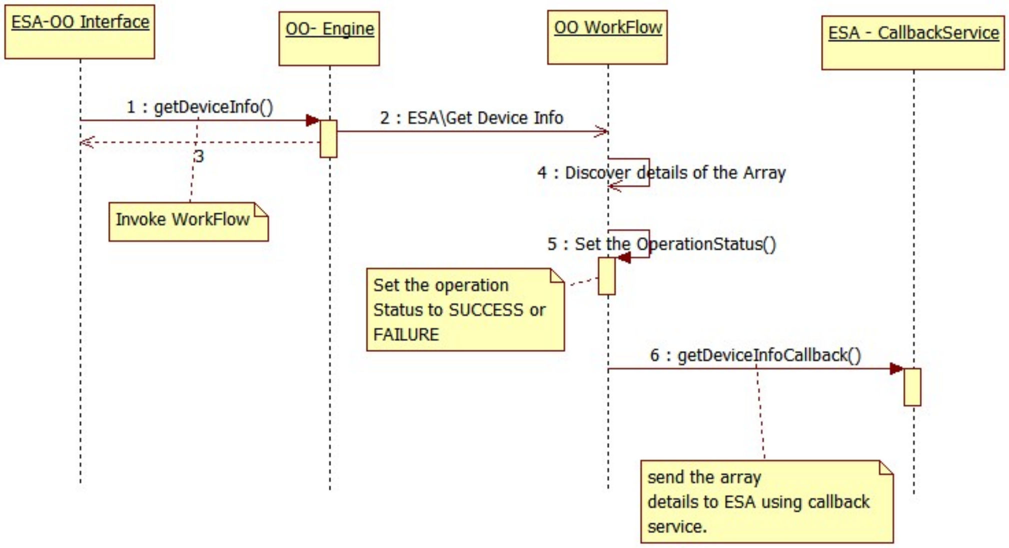


Figure 3: Interaction between ESA and OO


ESA-OO Interface
OO- Engine
OO WorkFlow
ESA - CallbackService
1 : getDeviceInfo()
2 : ESA\Get Device Info
3
Invoke WorkFlow
4 : Discover details of the Array
5 : Set the OperationStatus()
Set the operation Status to SUCCESS or FAILURE
6 : getDeviceInfoCallback()
send the array details to ESA using callback service.

The following sections provide more details on the server and storage OO workflows, configuration and use of the reference implementations included with Insight Dynamics, and information on how to customize the workflows for specific environments.

ESA and OO configuration

For ESA to invoke OO workflows (for either server or storage workflows), it is necessary to configure the OO credentials.

The assumption is that OO and ESA are installed on same CMS (Central Management Server). The user needs to update the OO password and restart ESA service from the Service console.

Open the *esa.properties* file which is located under C:\Program Files\HP\Virtual Server Environment\conf. Change the following and restart the ESA service from Windows service console.

```
#
#OO - Configuration
#
esa.oo.admin.password=<oo password>
```

Customers using the 6.0 update 1 release of HP Insight Dynamics with infrastructure orchestration have additional configuration steps outlined in the *Configuring ESA for 6.0 Updater 1* appendix.

Subsequent sections provide specific configuration information for the workflow (e.g., configuring iLO credentials for servers not using Virtual Connect).

Locations of XSD and WSDL Files

Once ESA is installed as part of Insight Dynamics, the WSDL is available at http://localhost:52000/esa/services/serverservice?wsdl.

The XSD file (*esa-server-model-1.0.xsd)* can be found at location C:\Program Files\HP\Virtual Server Environment\esa\etc\xsd\ esa-server-model-1.0.xsd.

Supporting servers not using Virtual Connect

The Insight Dynamics solution provides support for a range of servers, including blades using Virtual Connect, blades not using Virtual Connect, and rack-mount HP servers. The support for Integrity servers not using Virtual Connect is included in the Insight Dynamics 6.2 release. Information regarding Virtual Connect blades can be gathered automatically. Control operations can be performed through Virtual Connect, or (for customers using the infrastructure orchestration aspects of Insight Dynamics) via OO workflows (using the HP iLO2 interface for HP servers not using Virtual Connect). OO refers to HP Operations Orchestration, a workflow engine included with the Insight Dynamics solution (and a broader product offering is available from HP with significant additional capabilities).

The following sections describe how to take advantage of the range of server support in Insight Dynamics, and customize it as appropriate for your environment. The Insight Dynamics solution includes integrated support for

- HP blades Virtual Connect
- HP blades not using Virtual Connect using iLO2 or later
- HP Rack-mount servers using iLO2 or later

The XML files used to supply information about non-Virtual Connect servers can be customized for a specific customer environment (e.g., describing the HP blades and HP rack-mount servers not using Virtual Connect).

The ESA server OO workflows support:

1. Discovery of servers not using Virtual Connect
2. Retrieving details of servers not using Virtual Connect
3. Powering ON the HP servers not using Virtual Connect (via iLO)
4. Powering OFF the HP servers not using Virtual Connect (via iLO)

The interaction happens as follows

1. The Insight Dynamics logical server management layer requests an update of its catalog through ESA, which invokes the OO *GetInventory* workflow. This returns the basic server data which is loaded into the inventory of servers.
2. The logical server management layer requests more information on each of the servers, using ESA to invoke the *GetServerInfo* workflow. This returns the more detailed server information (including LAN and SAN connectivity).
3. When Insight Dynamics needs to power on a server, it performs the operation thru Virtual Connect, or uses ESA to call the OO Server *Power ON* workflow which in turn uses the server's remote management interface (iLO2) to power on the specific server.
4. When Insight Dynamics needs to power off a server, it performs the operation thru Virtual Connect, or uses ESA to call the OO Server *Power OFF* workflow which in turn uses the server's remote management interface (iLO2) to power off the specific server.

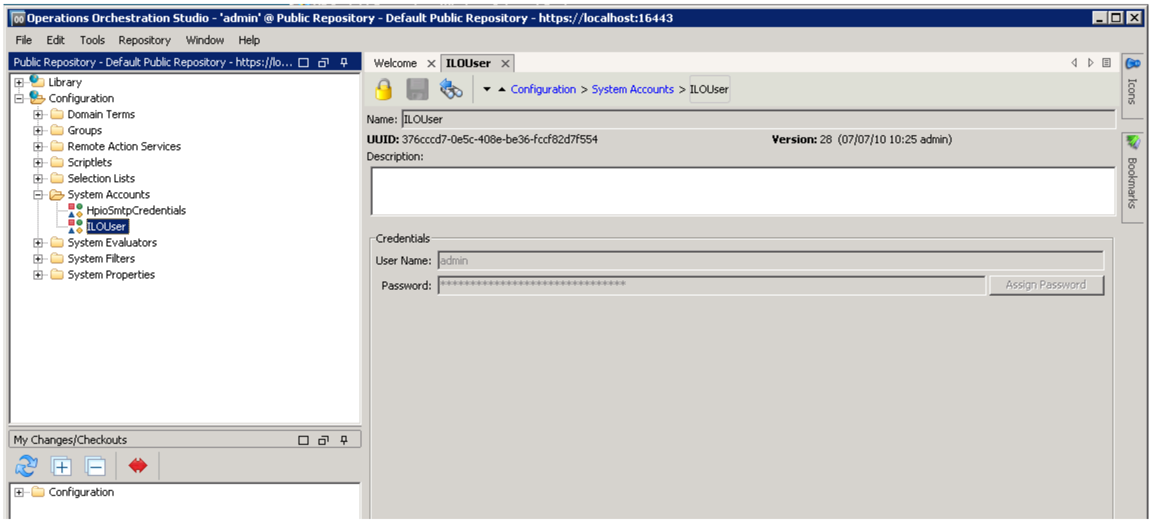
Deployment and configuration of server OO workflows

The four server OO workflows (i.e. *GetInventory*, *GetServerInfo*, *PowerON*, and *PowerOFF*) are deployed as part of the Insight Dynamics installation.   Three XML files (described in subsequent sections) need to be modified to supply appropriate information about the iLO2 servers not using Virtual Connect. If Virtual Connect is being used, the servers will be automatically discovered and information gathered, so no XML file adjustments are required. The following operations configure the OO workflows.

In OO Studio, configure the following System Account.

- ILOUser - This is the management console user credentials for HP servers not using Virtual Connect (servers using Virtual Connect are managed by Insight Dynamics using Virtual Connect)

Figure 4: Configuring the system account



Name: ILOUser

UUID: 376cccd7-0e5c-408e-be36-fccf82d7f554 **Version:** 28 (07/07/10 10:25 admin)

Description:

Credentials

User Name: admin

Password: ******************************

Figure 4 shows the use of OO Studio to create the ILOUser. The credentials are stored in an encrypted form within Operations Orchestration.

Note:

1. Username and password are the server's remote management interface user credentials.
2. The provided reference implementation mandates that this username and password be the same for all the remote management stations of the same server type. If they need to be different (e.g., separate users per model type, or even per individual server), the OO workflow logic will need to be modified accordingly to use the appropriate credentials.
3. To add a workflow for an additional HP server type (e.g., for separate workflows for model type), create an account as shown in Figure 4. Additional steps to add support for a new server type are covered in the appropriate sections below.

Reference implementation and extensions

As mentioned earlier, there are four workflows which have been implemented for reference (and can be extended as appropriate to meet the needs of a specific customer environment). This section provides details on each of the workflows implemented in the reference implementation, and notes on how they can be used, and extended to support additional environments.

Get Inventory

Description: The *GetInventory* workflow (shown in Figure 5) reads the server inventory information from a pre-designated file and returns the list to ESA. The logical server management aspect of Insight Dynamics calls this workflow on periodic basis to maintain an inventory of server information.

Implementation:
The server inventory file *inventoryList.xml* is located under the C:\Program Files\HP\Insight Orchestration\esa_extensions\server directory.
The file has the basic server information such as UUID and handle.
The workflow reads this file and sends the file contents back to ESA.

A sample inventoryList.xml file is listed below.

```
 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ROOT xmlns="http://model.server.com">
      <NodeList>
             <nodeAttributes>
                    <handle>37393434-3836-5532-5838-343130384152</handle>
                    <uuid>37393434-3836-5532-5838-343130384152</uuid>
             </nodeAttributes>
             <nodeAttributes>
                    <handle>38313933-3533-5355-4538-32304E334151</handle>
                    <uuid>38313933-3533-5355-4538-32304E334151</uuid>
             </nodeAttributes>
             <nodeAttributes>
                    <handle>37393434-3836-5532-5838-343130384150</handle>
                    <uuid>37393434-3836-5532-5838-343130384150</uuid>
             </nodeAttributes>
      </NodeList>
</ROOT>
```

Note: The output of this workflow must adhere to *esa-server-model-1.0.xsd* available under C:\Program Files\HP\Virtual Server Environment\esa\etc\xsd

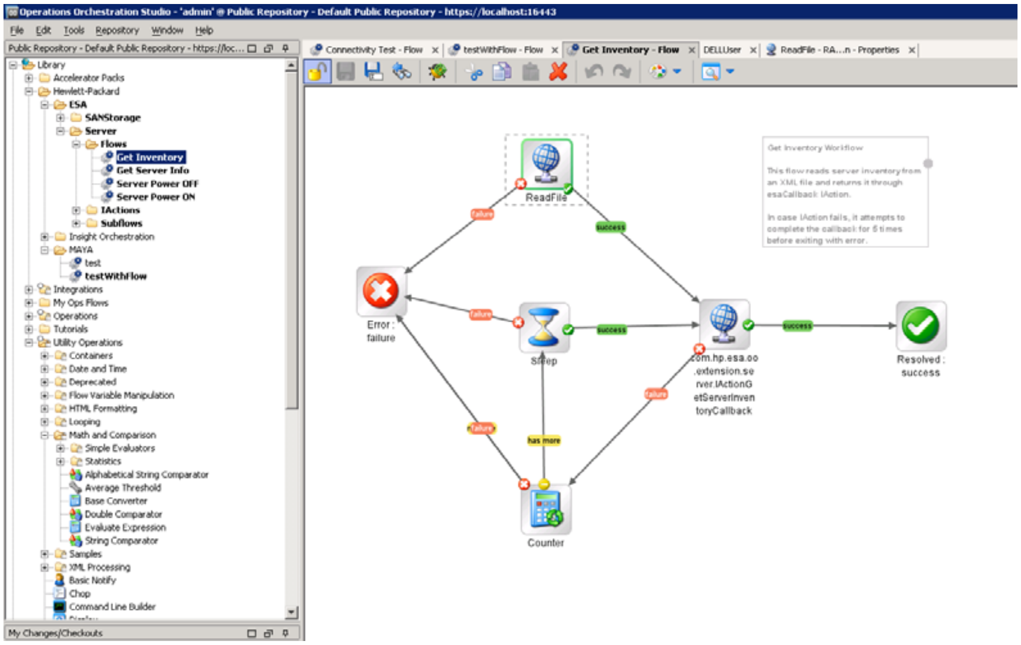
In the example above, a UUID is used for both the <handle/> and the <uuid/> elements. This is the recommended form. Typically, the UUID is directly extracted from the managed server or from a centralized configuration management database (CMDB) within the data center.

Note: It is ***important*** that <uuid/> be expressed in the canonical form shown above. The canonical form is 32 hexadecimal digits represented as 5 hyphen-separated groups in the form of:

8-4-4-4-12 or xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

comprising a total of 36 characters (32 hexadecimal digits and 4 hyphens). If this standard is not followed for all UUIDs within both the inventoryList.xml file and the serverInfo.xml file, Insight Dynamics will not be able to successfully manage the servers contained by the server inventory.

Figure 5: Get Inventory workflow



Get Inventory Workflow

This flow reads server inventory from an XML file and returns it through esaCallback IAction.

In case IAction fails, it attempts to complete the callback for 5 times before exiting with error.

Extensibility:

1. To add more servers not managed through Virtual Connect, edit the *inventoryList.xml* adding new *<nodeAttributes>* as appropriate.
2. The *Adding a new server type* section summarizes the changes for a new server type across the workflows.

Further Improvements:
The reference implementation uses a file-based approach for inventory purposes. If there is need for a large pool of servers, the information could be held in a database (perhaps a formal CMDB, or information extracted from a CMDB). The workflows would need to be modified to handle database operations appropriately (i.e., retrieving UUID information and using it to index into other server information such as model information, LAN/SAN connectivity, remote management IP and credentials, etc.).

Get Server Info

Description: The *GetServerInfo* workflow (shown in Figure 6) looks up the server detailed information based on the supplied UUID (which was obtained by Insight Dynamics via the *GetInventory* workflow). This *GetServerInfo* workflow is invoked on a periodic basis to maintain a view of server information.

Implementation:

- The server inventory file, *serverinfo.xml*, located under C:\Program Files\HP\Insight Orchestration\esa_extensions\server has the detailed information for the servers.
- The details of the server information are based on esa-server-model-1.0.xsd.

- Before passing back the server information for a given UUID, the workflow retrieves the power status of the server and updates the *serverinfo.xml* file. This information enables Insight Dynamics to know the current power status of a server (useful when performing power operations).
- In case the power status command fails to retrieve power status information, the old power status in the serverinfo.xml file is retained and will be returned.
- The expected values for powerOn status in the serverInfo.xml file are:
    - true: If the server power is ON
    - false: If the server power is OFF

A sample *serverinfo.xml* is shown below (containing one server). The format of serverInfo.xml should adhere to the schema defined in *esa-server-model-1.0.xsd* (located at C:\Program Files\HP\Virtual Server Environment\esa\etc\xsd). It provides details about permissible values and required formatting. Failing to adhere to the XML schema will result in failure of the Get Server Information operation.

The following example follows the recommended convention of setting the handle values to be the same as the UUID values. The server has a direct attach boot disk (indicated by the <localStorageList> element); for maximum flexibility, boot-from-SAN is recommended, as the servers will be able to access their boot volumes when moved to alternate physical servers.

```
<?xml version="1.0" encoding="UTF-8"?>
<ROOT xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="http://model.server.com ./esa-server-model-1.0.xsd">
 <ServerList>
  <ooServers>
       <handle>37393434-3836-5532-5838-343130384152</handle>
       <uuid>37393434-3836-5532-5838-343130384152</uuid>
       <serialNumber>F6BWDH1</serialNumber>
       <serverName>F6BWDH1</serverName>
       <hostname/>
       <localStorageList>
              <localStorageVolumes>
                     <id>1</id>
                     <sizeGB>67</sizeGB>
                     <volumeIdentifier>1</volumeIdentifier>
              </localStorageVolumes>
       </localStorageList>
       <networkInterfaceList>
              <networkInterfaces>
              <nicID>nic0</nicID>
              <macAddress>00:1E:4F:FC:61:21</macAddress>
              <ipAddress>
                     <address>127.0.0.1</address>
                     <dnsName>myName</dnsName>
                     <allocationType>STATIC</allocationType>
              </ipAddress>
              <networkInterfaceName>vlan51-dhcp-pxe-sa</networkInterfaceName>
              <pxeEnabled>true</pxeEnabled>
       </networkInterfaces>
       </networkInterfaceList>
       <storageInterfaceList>
              <storageInterfaces>
                     <hbaID>QLE2460-01</hbaID>
                     <portType>FC</portType>
                     <nodeWWN>21:00:00:1B:32:0F:76:8F</nodeWWN>
                     <portWWN>21:00:00:1B:32:0F:76:8F</portWWN>
                     <sanFabric>san3</sanFabric>
              </storageInterfaces>
       </storageInterfaceList>
       <serverModel>ProLiant DL385 G5</serverModel>
       <osType>Windows</osType>
```

```
      <memoryMB>16000</memoryMB>
      <coreCount>4</coreCount>
      <cpuFrequencyMHz>2100</cpuFrequencyMHz>
      <cpuCount>2</cpuCount>
      <cpuFamily>HPProLiant</cpuFamily>
      <powerOn>true</powerOn>
      <rackName>FtC-Bldg-R3</rackName>
      <rackPosition>RackPos-5</rackPosition>
      <extendedAttributes>
            <keyValuePairs>
                  <keyName>Enclosure</keyName>
                  <keyValue>FtC-Bldg3-R3</keyValue>
            </keyValuePairs>
            <keyValuePairs>
                  <keyName>EnclosurePosition</keyName>
                  <keyValue>RackPos-5</keyValue>
            </keyValuePairs>
      </extendedAttributes>
 </ooServers>
</ServerList>
```

Note: The <rackName> and <rackPosition> elements were added in the 6.0 update 1 release and appear in subsequent releases (but are not present in release 6.0). The attributes are not used by the server OO WF, but do need to be present.

Note: The Extended Attributes provide key-value pairs such as Enclosure and EnclosurePosition. If the attributes are set, infrastructure orchestration will read the information and report location information of the servers.

Insight Dynamics will determine the LAN and SAN connectivity for the server based on the networkInterfaceName and sanFabric values from this file. Thus, it is key these values match what will be used when specifying resource requests which may be fulfilled by these servers (e.g., the named networks in the Insight Orchestration Designer "Networks" tab). The servers will be visible in the Insight Orchestration Designer "Servers" interface (with suitable details).

Each UUID specified in *inventoryList.xml* must have corresponding server details in the *serverinfo.xml* file (else the workflow will fail to return details). Any UUID in the *serverinfo.xml* file but not in the *inventoryList.xml* file will not be seen (as the Get Server Info workflow would not be invoked).

Note: The <serverModel> and <cpuFamily> information must be specified correctly to ensure the appropriate Insight Dynamics license is made available for the server being described. The following sub-sections show the licenses and <serverModel> and <cpuFamily> values for the blade and rack-mount ProLiant and Integrity servers.

Once the servers are appropriately described in the file, the following steps must be taken for the servers to be successfully and used by HP Insight Dynamics infrastructure orchestration:

1. Edit C:\Program Files\HP\Insight Orchestration\conf\blade_models
2. Add the appropriate X86_64_MODELS or IA_64_MODELs entry depending on the server type that is being added.

   For example:

   X86_64_MODELS=ProLiant DL 580 G6
   IA_64_MODELS=HPIntegrity BL 860c

3. Repeat the entries above using the SUPPORTED_MODELS property. For example:

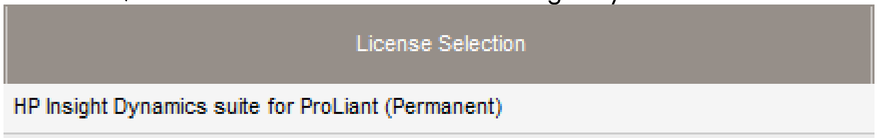
```
SUPPORTED_MODELS=ProLiant DL 580 G6
SUPPORTED_MODELS=HPIntegrity BL 860c
```

It is necessary to restart the “HP Insight Orchestration” Windows service after making changes to this file.

HP ProLiant c-Class blades not using Virtual Connect and rack-mount servers

As noted earlier, Insight Dynamics is able to automatically gather information regarding (and perform power operations on) blades using Virtual Connect. OO workflows are needed for ProLiant c-Class blades not using Virtual Connect, or ProLiant rack-mount servers.

The blades/servers should use the standard Insight Dynamics suite for ProLiant license:



License Selection

HP Insight Dynamics suite for ProLiant (Permanent)

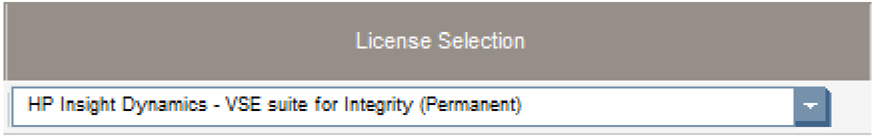
- ProLiant c-Class blades not using Virtual Connect
    - <serverModel>ProLiant BLXXXc GX</serverModel>
    - <cpuFamily>HPProLiant c-Class</cpuFamily>

- ProLiant rack-mount servers
    - <serverModel>ProLiant DL/MLXXX GX</serverModel>
    - <cpuFamily>HPProLiant</cpuFamily>

Integrity c-Class blades not using Virtual Connect and rack-mount servers

As noted earlier, Insight Dynamics is able to automatically gather information regarding (and perform power operations on) blades using Virtual Connect. OO workflows are needed for Integrity c-Class blades not using Virtual Connect, or Integrity rack-mount servers.

The blades/servers should use the standard HP Insight Dynamics – VSE suite for Integrity license:



License Selection

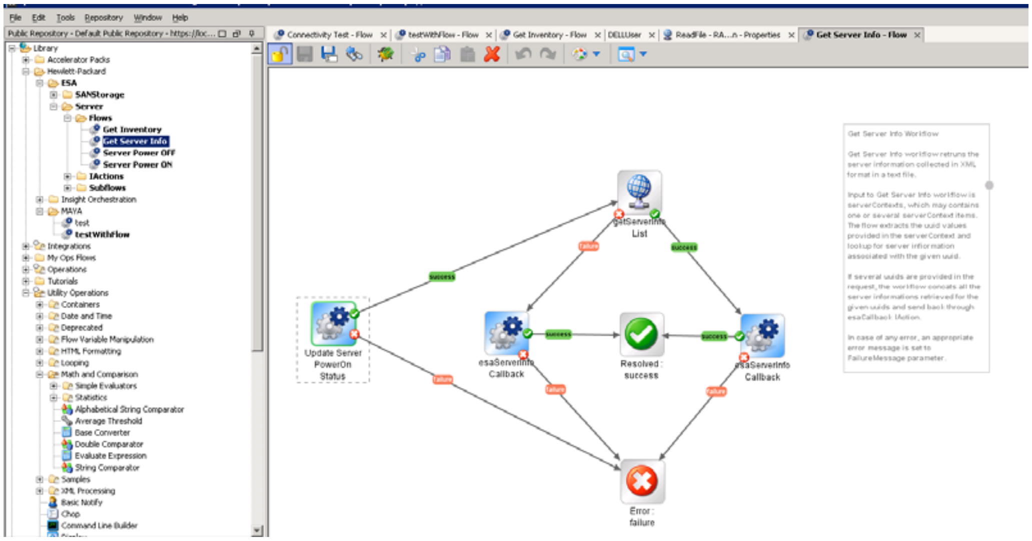
HP Insight Dynamics - VSE suite for Integrity (Permanent)

- Integrity c-Class Blades not using Virtual Connect
    - <serverModel>Integrity BLXXXc GX<serverModel>
    - <cpuFamily>HPIntegrity</cpuFamily>

- Integrity rackmount
    - <serverModel>Integrity rxXXX iX<serverModel>
    - <cpuFamily>HPIntegrity</cpuFamily>

The mechanism of XML file annotation enables the specification of additional servers which can be managed with the existing workflows. Managing additional types of servers (e.g., HP servers with power commands different than the iLO2 commands in the existing workflow) will involve adding a branch in the existing flow to handle that specific set of servers. Figure 6 shows the main *Get Server Info* workflow, and Figure 7 shows the subflow to gather the power status of a server.

Figure 6: Get Server Info main workflow



Get Server Info Workflow

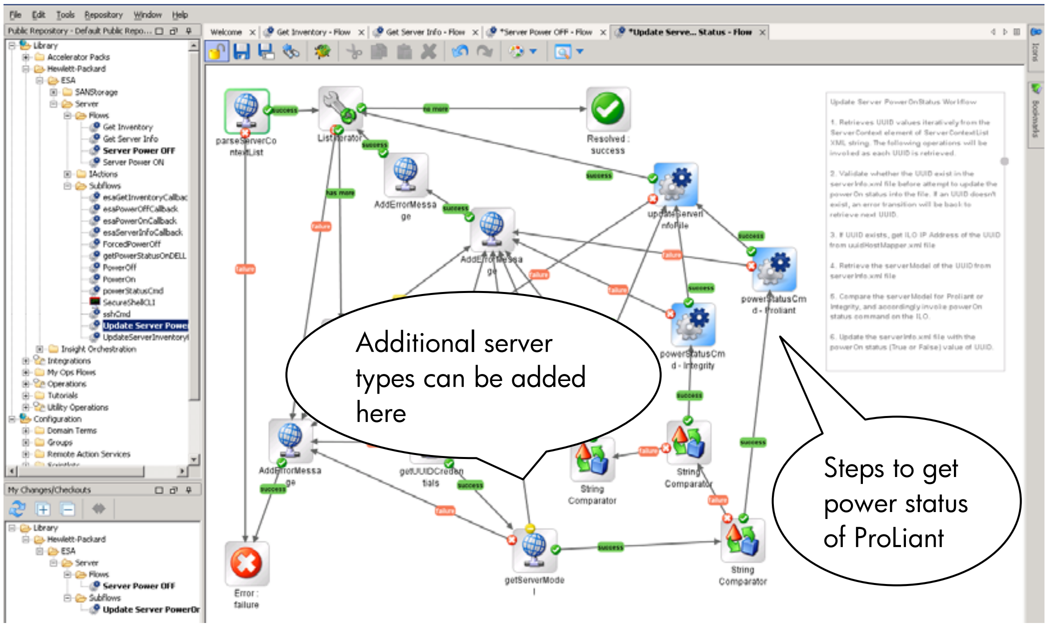
Get Server Info workflow retruns the server information collected in XML format in a text file.

Input to Get Server Info workflow is serverContexts, which may contains one or several serverContext items. The flow extracts the uuid values provided in the serverContext and lookup for server information associated with the given uuid.

If several uuids are provided in the request, the workflow concats all the server informations retrieved for the given uuids and send back through esaCallback IAction.

In case of any error, an appropriate error message is set to FailureMessage parameter.

Figure 7: Subflow to get the power status of the server



Update Server PowerOnStatus Workflow

1. Retrieves UUID values iteratively from the ServerContext element of ServerContextList XML string. The following operations will be invoked as each UUID is retrieved.

2. Validate whether the UUID exist in the serverInfo.xml file before attempt to update the powerOn status into the file. If an UUID doesn't exist, an error transition will be back to retrieve next UUID.

3. If UUID exists, get ILO IP Address of the UUID from uuidHostMapper.xml file

4. Retrieve the serverModel of the UUID from serverInfo.xml file

5. Compare the serverModel for Proliant or Integrity, and accordingly invoke powerOn status command on the ILO.

6. Update the serverInfo.xml file with the powerOn status (True or False) value of UUID.

Extensibility:

1. To add new servers not managed by Virtual Connect (Insight Dynamics is able to gather information directly for servers using Vitual Connect), add its UUID in the *inventoryList.xml*, IP address of its remote management interface in the *uuidHostMapper.xml* and the server specific details in the *serverinfo.xml* file.
2. Determine the appropriate server operation(s) to retrieve power status information and implement a subflow to get the power status (as shown in Figure 7). The *Adding a new server type* section summarizes the changes for a new server type across the workflows.
3. Finally, you must declare any new server models to be managed by taking the <ServerModel/> string supplied in the "serverInfo.xml" file above and adding it to C:\Program Files\HP\Insight Orchestration\conf\blade_models.properties. After making the changes, the Insight Orchestration Windows service needs to be restarted for the changes to take effect.

Note: Insight Dynamics performs power operations via iLO2 on the HP blades/servers not using Virtual Connect. Thus, the HP iLO IP address and credentials are needed. The credentials were supplied in the user configuration step described earlier. The remote management interface IP addresss (e.g. the iLO IP address for ProLiant servers) must be entered into the *uuidHostmapper.xml* file located under C:\Program Files\HP\Insight Orchestration\esa_extensions\server.

A sample file is shown below (giving IP information for the remote management interfaces of four servers).

```
<CredMap>
        <map>
                <uuid>37393434-3836-5532-5838-343130384152</uuid>
                <host>15.154.108.198</host>
        </map>
        <map>
```

```
              <uuid>37393434-3836-5532-5838-343130384150</uuid>
              <host>15.154.109.214</host>
       </map>
       <map>
              <uuid>37393434-3836-5532-5838-343130384153</uuid>
              <host>15.154.109.214</host>
       </map>
       <map>
              <uuid>37393434-3836-5532-5838-343130384154</uuid>
              <host>15.154.108.195</host>
       </map>
</CredMap>
```

The UUID value in this file must match the UUID value specified in the inventory file; it is the means of uniquely identifying a server and correlating the management information regarding that server.

Further Improvements:
The reference implementation uses a file-based approach for inventory purposes. If there is need for a large pool of servers, the information could be held in a database (perhaps a formal CMDB, or information extracted from a CMDB). The workflows would need to be modified to handle database operations appropriately (i.e., retrieving UUID information and using it to index into other server information such as model information, LAN/SAN connectivity, remote management IP and credentials, etc.). As noted earlier, the workflows could also be modified to handle different iLO passwords (versus the current assumption that the same password is used for a given server type).

Server power ON

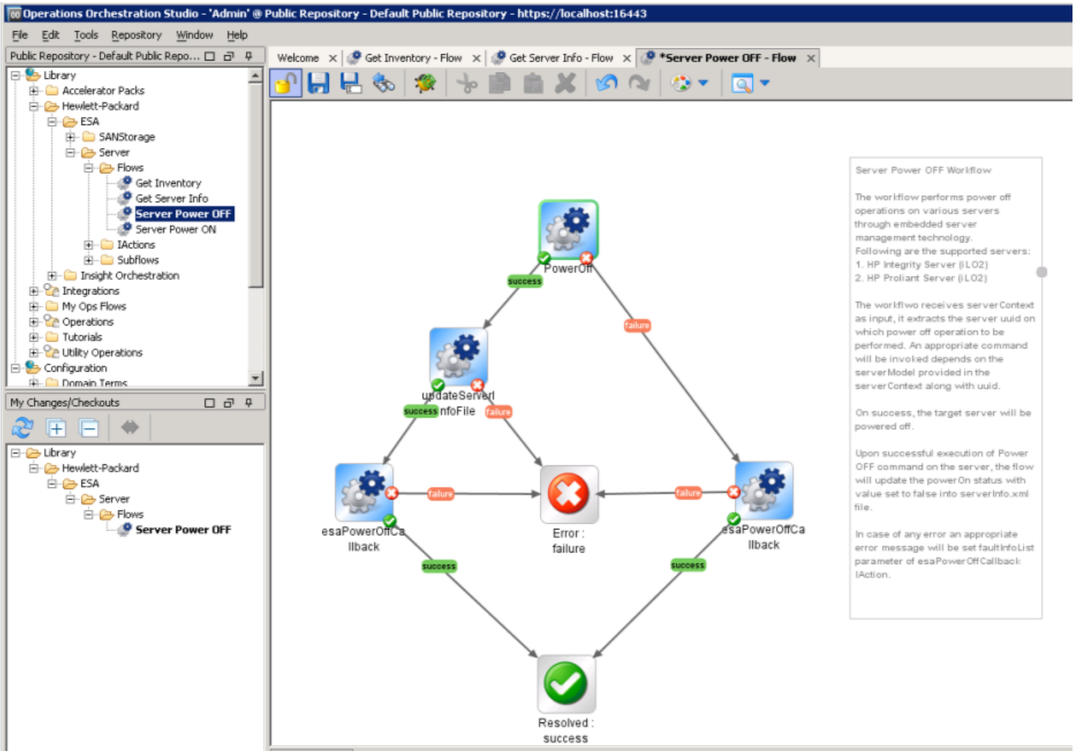
Description: The Server *PowerON* workflow (shown in Figure 8) receives the UUID and model of the server (through ESA) and invokes a Server Power ON operation through the server's remote management interface (iLO2).

Implementation:

1. The workflow gets the server's remote management interface IP address from the *uuidHostMapper.xml* file.
2. The workflow obtains the server's remote management interface user credentials (configured in the flow).
3. Using the IP address and credentials, the workflow connects to the management station using SSH and executes the appropriate command. For example, to power on a ProLiant server, the "power on" command is used; to power on an Integrity server, the "PC –on" command is used.
4. Based on the response, update the power status in the *serverinfo.xml* file (powerOn status set to true).
5. Return back success or failure as appropriate.

Figure 8 shows the main Server Power ON workflow, while Figure 9 shows the subflow to execute the power ON operation, and Figure 10 shows the command to be invoked by SSH ("power on").

Figure 8: Server power ON workflow



Server Power OFF Workflow

The workflow performs power off operations on various servers through embedded server management technology.
Following are the supported servers:
1. HP Integrity Server (iLO2)
2. HP Proliant Server (iLO2)

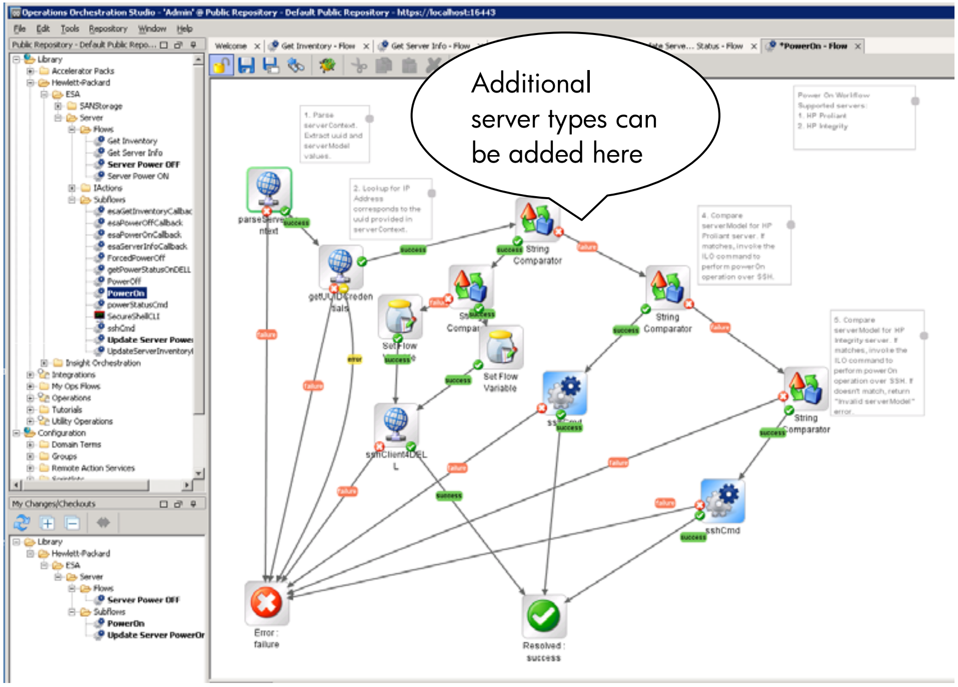
The workflwo receives serverContext as input, it extracts the server uuid on which power off operation to be performed. An appropriate command will be invoked depends on the serverModel provided in the serverContext along with uuid.

On success, the target server will be powered off.

Upon successful execution of Power OFF command on the server, the flow will update the powerOn status with value set to false into serverInfo.xml file.

In case of any error an appropriate error message will be set faultInfoList parameter of esaPowerOffCallback IAction.

Figure 9: Server power ON workflow: subflow


Operations Orchestration Studio - 'Admin' @ Public Repository - Default Public Repository - https://localhost:16443

Additional server types can be added here

1. Parse serverContext. Extract uuid and serverModel values.

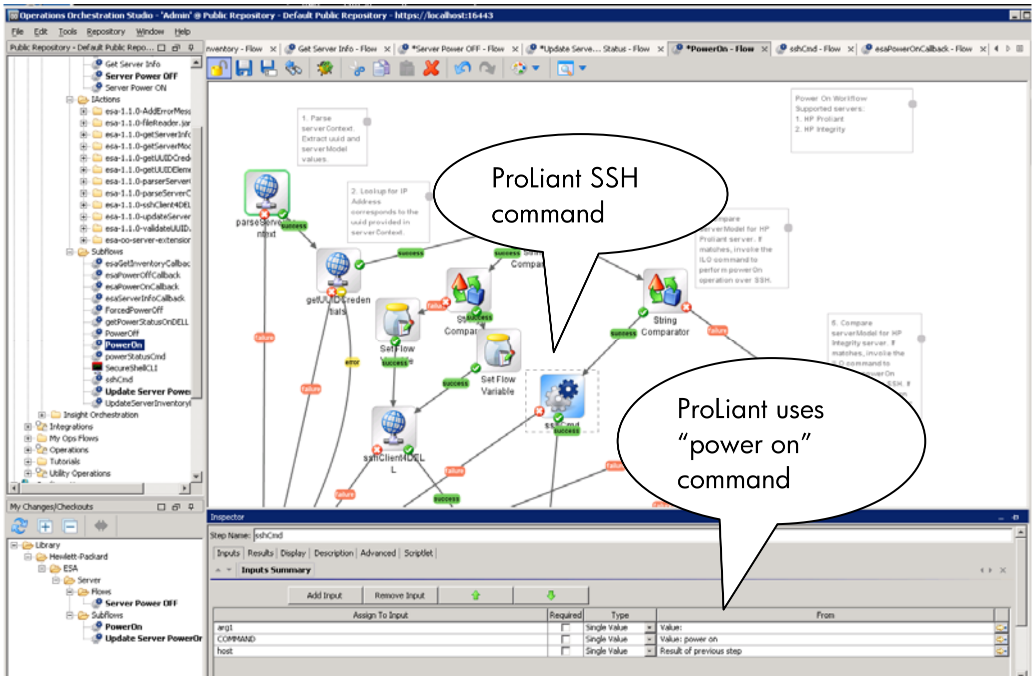
2. Lookup for IP Address corresponds to the uuid provided in serverContext.

4. Compare serverModel for HP Proliant server. If matches, invoke the ILO command to perform powerOn operation over SSH.

5. Compare serverModel for HP Integrity server. If matches, invoke the ILO command to perform power On operation over SSH. If doesn't match, return "invalid serverModel" error.

Power On Workflow
Supported servers:
1. HP Proliant
2. HP Integrity

Figure 10: Server power ON workflow: subflow SSH command ("power on" for ProLiant)


Operations Orchestration Studio - 'Admin' @ Public Repository - Default Public Repository - https://localhost:16443

ProLiant SSH command

ProLiant uses "power on" command

Inspector

Step Name: sshCmd

Inputs | Results | Display | Description | Advanced | Scriptlet

Inputs Summary

Add Input | Remove Input

Assign To Input	Required	Type	From
arg1	□	Single Value	Value:
COMMAND	□	Single Value	Value: power on
host	□	Single Value	Result of previous step

The OO workflow subflow for PowerOn invokes the appropriate iLO command via SSH.

Figure 10 shows "power on" being used for ProLiant iLO2 servers; Integrity servers use the "PC –on" command.

Extensibility:

1. Understand the Power ON OO workflow
2. Add a new check in the subflow for the corresponding server type; the *Adding a new server type* section summarizes the changes for a new server type across the workflows.
3. The reference implementation uses OO's internal SSH to invoke the command.
4. Alternatively, one can create a new IAction class (the OO documentation provides details on developing IAction extensions)

Server power OFF

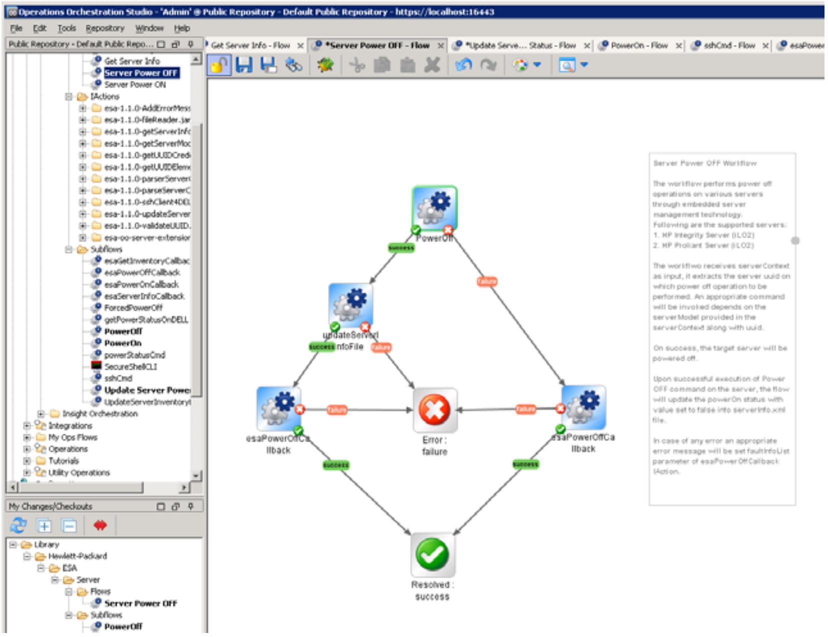
Description: The Server *PowerOFF* workflow (shown in Figure 11) receives the UUID and model of the server (through ESA) and invokes a Power OFF operation through the server's remote management interface (iLO2).

Implementation:

1. The workflow gets the server's remote management interface IP address from the *uuidhostMapper.xml* file.
2. Based on the server model, the workflow obtains the remote management user credentials.
3. Using the IP address and credentials, the workflow connects to the remote management interface using SSH and executes the appropriate command. For example, to power off a ProLiant server, the "power off"command is used, and to power off an Integrity server, the "PC –graceful" command is used.
4. Based on the command response, the workflow updates the power status in the *serverinfo.xml* file (PowerOn status set to false).
5. Return back success or failure as appropriate.

Figure 11 shows the main Server Power OFF workflow, while Figure 12 shows the subflow to execute the power OFF operation, and Figure 13 shows the SSH command invoked ("power off" for ProLiant servers).

Figure 11: Server power OFF workflow



Server Power OFF Workflow

The workflow performs power off operations on various servers through embedded server management technology.
Following are the supported servers:
1. HP Integrity Server (iLO2)
2. HP Proliant Server (iLO2)

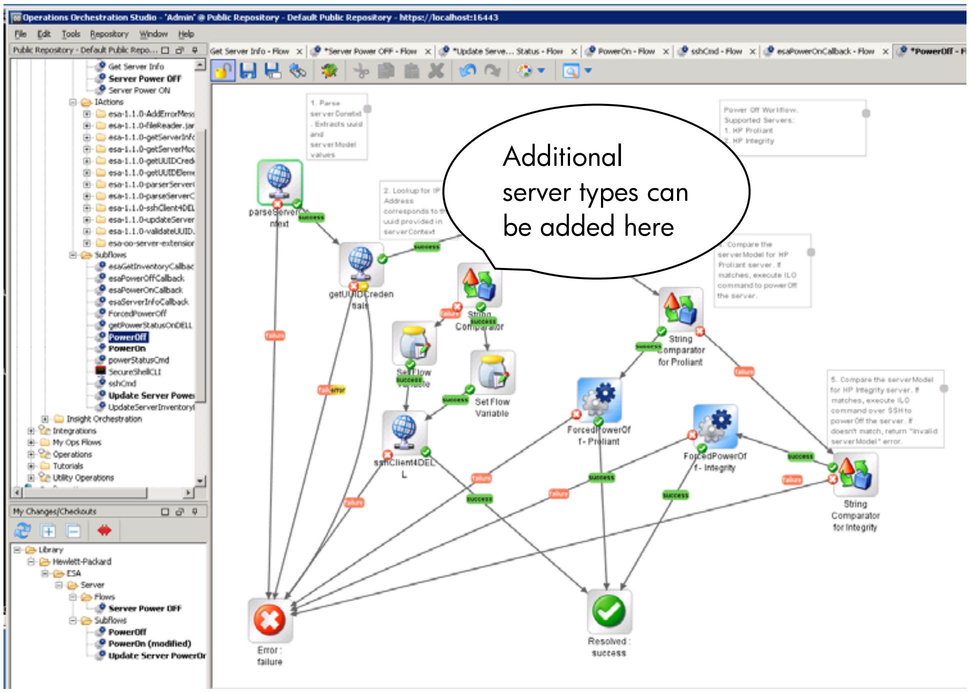
The workflwo receives serverContext as input, it extracts the server uuid on which power off operation to be performed. An appropriate command will be invoked depends on the serverModel provided in the serverContext along with uuid.

On success, the target server will be powered off.

Upon successful execution of Power OFF command on the server, the flow will update the powerOn status with value set to false into serverInfo.xml file.

In case of any error an appropriate error message will be set faultInfoList parameter of esaPowerOffCallback IAction.

Figure 12: Server power OFF workflow: subflow



Operations Orchestration Studio - 'Admin' @ Public Repository - Default Public Repository - https://localhost:16443
File Edit Tools Repository Window Help
Get Server Info - Flow
*Server Power OFF - Flow
*Update Serve... Status - Flow
PowerOn - Flow
sshCmd - Flow
esaPowerOnCallback - Flow
*PowerOff - F
PowerOff
PowerOn (modified)
Update Server PowerOn
parseServerContext
getUUIDCredentials
String Comparator
Set Flow Variable
Set Flow Variable
sshClient4DELL
String Comparator for Proliant
ForcedPowerOff - Proliant
ForcedPowerOff - Integrity
String Comparator for Integrity
success
failure
error
Error : failure
Resolved : success

1. Parse serverConetxt . Extracts uuid and serverModel values

2. Lookup for IP Address corresponds to th uuid provided in serverContext

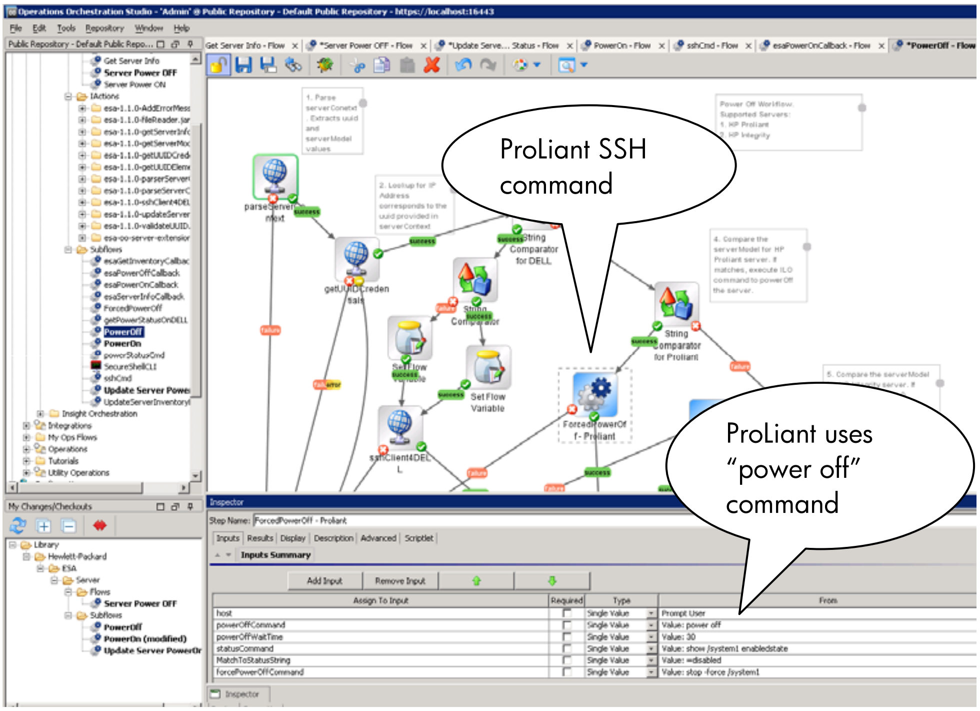
Power Off Workflow.
Supported Servers:
1. HP Proliant
2. HP Integrity

. Compare the serverModel for HP Proliant server. If matches, execute ILO command to power Off the server.

5. Compare the serverModel for HP Integrity server. If matches, execute ILO command over SSH to powerOff the server. If doesn't match, return "Invalid serverModel" error.

Additional server types can be added here

Figure 13: Server power OFF workflow: subflow SSH commands


ProLiant SSH command
ProLiant uses "power off" command
parseServerContext
getUUIDCredential
String Comparator for DELL
String Comparator
Set Flow Variable
sshClient4DELL
ForcedPowerOff - Proliant
String Comparator for Proliant
success
failure
Inputs | Results | Display | Description | Advanced | Scriptlet
Inputs Summary
Add Input
Remove Input

Step Name: ForcedPowerOff - Proliant

Assign To Input	Required	Type	From
host	□	Single Value	Prompt User
powerOffCommand	□	Single Value	Value: power off
powerOffWaitTime	□	Single Value	Value: 30
statusCommand	□	Single Value	Value: show /system1 enabledstate
MatchToStatusString	□	Single Value	Value: =disabled
forcePowerOffCommand	□	Single Value	Value: stop -force /system1

The OO workflow subflow for PowerOff invokes the appropriate iLO command via SSH. Figure 13 shows HP ProLiant servers using the iLO2 "power off" command for a standard power off, and "stop –force /system1" for a forced power off. Integrity servers use the "pc –graceful" and "pc –off" commands.

Extensibility:

1. Understand the Power OFF OO workflow
2. Add a new check in the subflow shown above for the corresponding server type (typically included in the server model string); the *Adding a new server type* section summarizes the changes for a new server type across the workflows.
3. The reference implementation uses SSH to invoke the command.
4. Alternatively, one can create a new IAction class (the OO documentation provides details on developing IAction extensions).

As noted earlier, the power on and off operations for HP servers are done through iLO (or Virtual Connect, if present). The following power off sequence is used for those servers:

- Invoke a graceful power off command
- Wait for 30 seconds
- Check the powerOff status
- If the server power is still on, invoke a forced power off command

These steps are performed on HP servers using iLO2 or later.

IActions

To facilitate development of OO workflows, the Insight Dynamics implementation includes IAction classes for server operations. The dependent jars for the server workflows are esa-server-model-x.jar, esa-oo-server-rservice-x.jar.

The details of the server-related IAction classes are listed below.

Iaction Class	Inputs	Description
IActionGetServerInventoryCallback	esaMessageId	esaMessageID (reused from Workflow input)
	inventory	List of servers in the inventory as XML. (see server_schema for details)
	opStatus	opStatus as XML.
	faultInfoList	Fault Info List as XML
	replytoURI	replyToURI.(reused from Workflow input)
IActionGetServerInfoCallback	esaMessageId	esaMessageID (reused from Workflow input)
	serverInfo	List of servers in the inventory as XML. (see server_schema for details)
	opStatus	opStatus as XML.
	faultInfoList	Fault Info List as XML
	replytoURI	replyToURI.(reused from Workflow input)
IActionPowerOnCallback	esaMessageId	esaMessageID (reused from Workflow input)
	opStatus	opStatus: SUCCESS or FAILED
	faultInfoList	Fault Info List: Error string or “No fault” string.
	replytoURI	replyToURI.(reused from Workflow input)
IActionPowerOffCallback	esaMessageId	esaMessageID (reused from Workflow input)
	opStatus	opStatus: SUCCESS or FAILED
	faultInfoList	Fault Info List: Error string or “No fault” string
	replytoURI	replyToURI.(reused from Workflow input)

Ading a new server type

A new server type can be supported via the reference implementation by adding and/or modifying the flows into the current implementation. This section details how to add a new subflow and modify the existing flows to support new server types.

In this section, the term “new server” is used to indicate that a server of a new type (not the existing ProLiant iLO2 or later servers). **NOTE: HP only supports the Insight Dynamics product server OO workflows with HP servers.**

It is essential to identify and collect relevant data before adding support for a new server into the reference implementation.

The following information is required:

1. Server Data, in the format as defined in the *serverInfo.xml* file. The UUID and server model are essential to execute the flows successfully, and information such as network and SAN

connectivity are essential to successfully utilize the servers from Insight Dynamics with infrastructure orchestration.

2. Remote management interface to perform power operations on the new server
3. Credentials required to login into remote management interface
    a. IP Address
    b. User Login ID
    c. Password
4. Commands to execute following power operations:
    a. Power ON
    b. Power OFF
    c. Power Status (optional)

Once the information is collected, the following operations can be supported on the new server:

- Get Inventory
- Get Server Info
- Server Power ON
- Server Power OFF

The following sub sections explain implementation details of adding these operations for the new server. It is a summarization of information noted earlier in the Extensibility section of the discussion of each workflow.

Get Inventory operation

In order to support Get Inventory operation, UUID of the new server must be added into *inventoryList.xml* file. Once the *inventoryList.xml* file is updated with the new UUID, the subsequent calls to Get Inventory operation will retrieve the new server's UUID.

No additional changes are required in the current implementation to support Get Inventory operation for the new server.

Get Server Info operation

The new server data must be added to the *serverInfo.xml* file.
If the new server remote management interface supports a command to get the power status then a new logic needs to be implemented in a subflow. In order to invoke the power status command on the remote management interface, the IP address of the remoter management interface should be added into *uuidHostMapper.xml* file. However, if power status command is not supported on the remote management interface, no additional changes are required to the current Get Server Info implementation.

The following steps explain how to add a new subflow and modify the existing implementation:

- Implement the logic to retrieve the power status of the server
    - Make sure that the new subflow will receive all necessary input parameters:
        - IP Address of the remote management interface
        - If necessary, add login (username/password) credentials into OO System Accounts
        - Command to execute
        - Any other parameters to fulfill the flow logic
    - Ensure the implemented logic will give the result (true or false) in the form of string into the result field parameter.
        - true indicates that server power is ON

        - false indicates that server power is OFF
- Once the new subflow is implemented and tested, it can be called by modfying the subflow "Update Server PowerOn Status".
    - Add a string comparator to compare the new server model. If matches, call the new subflow logic to get the power status from the new server.
    - Add the newly implemented subflow as a step operation upon matching the server model.
        - Provide all necessary inputs to the new subflow operation added.
        - Define the result field parameter of the subflow operation by name CmdOutput. This will hold the result of the logic implemented to get the power status of the new server.
    - The successful transition should invoke "UpdateServerInventoryFile" operation.
    - The failure transition should return back to the ListIterator.

After performing the above steps, the Get Server Info operation can be invoked from ESA by providing the new server's UUID in the serverContextList parameter. The returned server details for the new UUID will contain the new server data along with the last known (if not most recent) power status data.

Server Power ON operation

To perform the Power ON operation on the new server, the following changes should be done to the current implementation:
A new subflow should be created with the logic to execute the Power ON operation on the remote management interface.
Modify "PowerOn" subflow to add the newly created subflow.
The IP address of the remote management interface should be added into uuidHostMapper.xml file.

The following steps explain how to support the Power ON operation on a new server:

- Implement the logic to execute Power ON operation on the remote management interface into a new subflow
    - Make sure the new subflow will receive all the necessary input parameters:
        - IP Address of the remote management interface
        - If necessary, add login (username/password) credentials into OO System Accounts
        - Power ON command to execute on remote management interface
        - Any other parameters to fulfill the flow logic
    - Capture the results
        - Command output in case of success
        - Error message in case of failure
- Once the newly created subflow has been successfully tested, modify the PowerOn subflow to call it upon matching the server model string.
    - Add a String comparator to compare server model of the new server
    - Add the newly created subflow as a step operation. This subflow will be called upon successfully matching the server model string from the new string comparator.
        - Provide all necessary inputs to the newly added subflow operation
        - Define the following result field parameters to hold the result of the subflow
            - CmdOutput: To capture the command output if the subflow executes successfully
            - ErrMsg: To capture failure message if the subflow execution failed
    - The successful transition of the new subflow should return success from PowerOn subflow.
    - The failure transition of the new subflow should return failure from PowerOn subflow.

Note that the main flow (Server Power ON) will also update the *serverInfo.xml* file with powerOn status set to false upon successful execution. Thus, even if the server does not support an explicit power status command, the power status is set based on any power operations performed through the OO workflows.

After performing the steps outlined above, the Server Power ON operation can be invoked from ESA by providing the new server's UUID and server model in the serverContext parameter. A successful execution will turn on the server if it is turned off.

Server Power OFF operation

To perform Power OFF operation on the new server, following changes should be made to the current implementation:

- Create a new subflow with the logic to execute the Power OFF operation using the server's remote management interface
- Modify the "PowerOff" subflow to add the newly created subflow

Make sure that the IP address of the server's remote management interface is added into uuidHostMapper.xml file.

The following steps explain how to support the Power OFF operation on a new server:
Implement the logic to execute a Power OFF operation on the server's remote management interface into a new subflow

- Make sure the new subflow will receive all the necessary input parameters:
    - IP Address of the remote management interface
    - If necessary, add login (username/password) credentials into OO System Accounts
    - Power OFF command to execute on remote management interface
    - Any other parameters to fulfill the flow logic
- Make sure that the logic/command will turn off the system after successful execution.
- Capture the results
    - Command output in case of success
    - Error message in case of failure
- Once the newly created subflow has been tested successfully, modify the PowerOff subflow to call it upon matching the server model string.
    - Add a String comparator to compare server model of the new server
    - Add the newly created subflow as a step operation. This subflow will be called upon successfully matching the server model string from the new string comparator.
        - Provide all necessary inputs to the newly added subflow operation
        - Define the following result field parameters to hold the result of the subflow
            - CmdOutput: To capture the command output if the subflow executes successfully
            - ErrMsg: To capture failure message if the subflow execution failed
    - The successful transition of the new subflow should return success from PowerOff subflow.
    - The failure transition of the new subflow should return failure from PowerOff subflow.

Note that, the main flow (Server Power OFF) will also update the *serverInfo.xml* file with powerOn status set to false upon successful execution. Thus, even if the server does not support an explicit power status command, the power status is set based on any power operations performed through the OO workflows.

After performing the above steps, the Server Power OFF operation can be invoked from ESA by providing the new server's UUID and server model in the serverContext parameter. A successful execution will turn on the server if it is turned off.

Supporting additional storage types

The standard Insight Dynamics solution provides out-of-the-box support for the HP EVA storage solution (which can be bundled as part of HP BladeSystem Matrix).   The 6.0 and 6.1 Insight Dynamics releases provide a technology preview of using OO workflows and the new Storage Provisioning Manager, with full support in the Insight Dynamics 6.2 release. OO workflows are the means to extend storage support, and Insight Dynamics includes an OO workflow reference implementation adding support for XP arrays. The workflows demonstrate use of the XP CLI and can be customized for other storage types (e.g., EMC) or other means of interacting with storage (e.g., SMI-S versus CLI).

The Insight Dynamics solution requires pre-provisioned storage volumes (the storage administrator has created the storage and performed appropriate SAN zoning). These workflows provide the ability to gather storage property information, change hostmode, and adjust SAN zoning for storage other than HP EVA storage solutions. It is not necessary to use these workflows to simply use existing storage by manually creating Insight Dynamics storage pool entries; they are used to enable Insight Dynamics to use the storage catalog provided by the HP Storage Provisioning Manager (which can automatically collect information and perform certain storage operations such as changing hostmode and LUN masking).

Overview

The Extensible Server and Storage Adapter layer of the Insight Dynamics solution supports storage through two mechanisms:

1. SMI-S: used for out-of-the-box storage discovery and configuration
2. OO workflows: used for providing an extensible means of gathering storage information and performing configuration of additional storage types

That storage support is made visible to the administrator through use of the HP Storage Provisioning Manager (SPM) to maintain a catalog of storage which can be used when defining storage pool entries for use by Insight Dynamics. The 6.0 and 6.1 releases of Insight Dynamics include a technology preview of SPM, providing support for the EVA array solution via SMI-S and use an OO workflow for XP array solutions.   The 6.2 release provides full support for the HP Storage Provisioning Manager solution. The storage operations supported include gathering storage details and (if authorized by the storage administrator using the SPM), changing hostmodes and doing LUN masking. Insight Dynamics currently require pre-provisioned Fibre Channel storage (so the volume creation, presentation, and zoning operations are currently done independently from the Insight Dynamics solution, perhaps via Command View or Storage Essentials or other array and fabric management interfaces).

ESA is used to gather information about that storage, through SMI-S or OO workflows. For SMI-S management, the array credentials are configured in the Storage Provisioning Manager and array and volume information can be automatically discovered and imported into the storage catalog. For OO workflows, it is assumed that Insight Dynamics will not have any knowledge on the access credential required by the workflow to access the storage capabilities and services. It is expected that the workflow writers will manage the access rights and credential for managing the storage services external to the workflows. OO allows the workflow writers to store the credentials in a secure fashion (as discussed in the upcoming *Security credential configuration* section.

As noted in the *ESA and OO configuration* section, ESA does require the credentials of OO engine to invoke the required workflows.

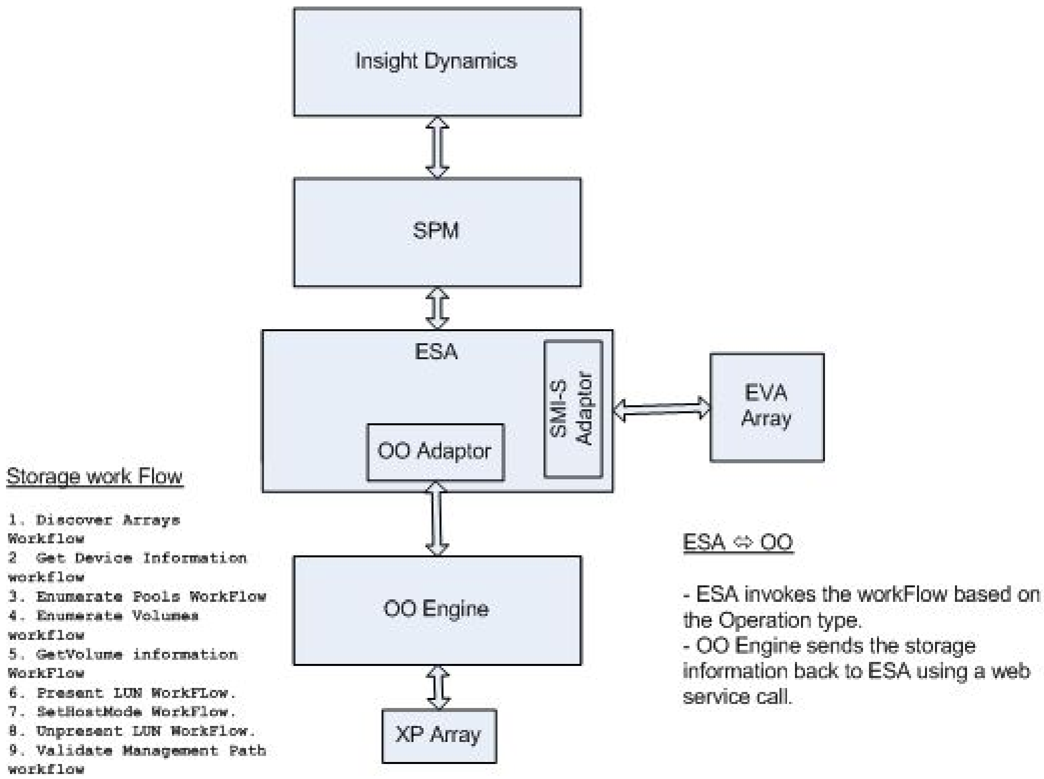
The ESA storage OO workflows support:

1. Discover Arrays
2. Get device information
3. Enumerate pools on device
4. Enumerate volumes on device
5. Get volume information
6. Present LUN
7. Unpresent LUN
8. Set hostmode
9. Validate Management Path

The storage administrator can use the Storage Provisioning Manager interface to specify authorizations for arrays, groups of volumes, or individual volumes. Some volumes may be view only (from a management perspective), while others may allow the Insight Dynamics software to adjust hostmode or perform LUN masking. The ability to change hostmode enables more flexible use of storage resources, as an available volume can be customized for the hostmode needed by a given logical server, versus having to maintain available volumes with a variety of hostmodes. The ability to perform LUN masking enables all the storage to be created and presented initially, and yet ensure OS installation is directed to the correct volume by unmasking data volumes (making them invisible) prior to install and masking them after the install (making them visible again).

Figure 14 demonstrates the flow of operations between Insight Dynamics with infrastructure orchestration, the HP Storage Provisioning Manager (providing a programmatic and graphical storage catalog interface), the Extensible Server and Storage Adapter layer, and the OO workflow or SMI-S mechanisms out to the storage arrays.

Figure 14: Flow of Operations



Storage work Flow

1. Discover Arrays Workflow
2 Get Device Information workflow
3. Enumerate Pools WorkFlow
4. Enumerate Volumes workflow
5. GetVolume information WorkFlow
6. Present LUN WorkFLow.
7. SetHostMode WorkFlow.
8. Unpresent LUN WorkFlow.
9. Validate Management Path workflow

ESA ⇔ OO

- ESA invokes the workFlow based on the Operation type.
- OO Engine sends the storage information back to ESA using a web service call.

ESA asynchronously invokes the appropriate workflow based on the operation type, providing an ESA_ID (for correlation) and appropriate input data. The OO workflows complete their tasks and make a call back to ESA's response web service. The OO callback to ESA includes the ESA_ID which was passed during the invocation of the workflow along with the appropriate data and status.

The following sections describe how to use the included XP reference implementation, and customize the workflows to support additional array types. The *HP Storage Provisioning Manager User Guide* (available at http://www.hp.com/go/insightdynamics/docs) explains use of the storage catalog GUI interface. Insight Dynamics will use the programmatic interface (but it is necessary to use the SPM GUI to configure the array credentials for SMI-S management; as noted earlier credentials management for OO workflows is assumed to be independent of the Insight Dynamics solution).

Deployment and configuration of storage OO workflows

The storage workflows are packaged and installed as part of the Insight Dynamics solution. The reference implementation uses the CLI (Command Line Interface) provided by the HP StorageWorks Command View XP Advanced Edition software to implement the various operations for XP arrays (XP 12000 and 24000). The XP Command View AE CLI provides XML formatted output of the commands, which can be parsed by the OO workflows.

The storage workflows for XP arrays can be configured for usage by performing the steps listed below. It is assumed the customer environment has the XP Command View Advanced Edition software available for management.

- XPCLI Installation
  - Login to XP Command View Device Manager Web Client.
  - Choose Go -> Download in Global Task Bar Area of Device Manager Web Client.
  - Choose the Download link for the CLI of Windows(R) in the Device Manager Software Deployment dialog.
  - Change directory into the location of where the CLI is downloaded.
  - Extract the DVMCLI*.zip to C:\DVMCli_win

- XPCLI Configuration
  - Configure the hdvmcli.properties file to execute commands on the XP Command View AE server.

```
# set the server url
hdvmcli.serverurl=http://<XP AE Server host>:2001/service
#####  OPTIONS ####
user=system
# password can be provided directly, or from a password file
password=xxxxxxx
#password=@D:\\DeviceManager\\.passwd
# provide copy of raw xml request & response in MessageTrace.log file
messagetrace=true
```

- Configuration of ESA use of XP OO Workflow

The XP-CLI must be installed and configured on the OO server to execute ESA-OO Storage workflows.

The following configuration steps need to be done

1. Configure the XP CLI configuration file to point to the right CVXPAE Server.
2. Configure the ESA Properties file with the credentials of the OO server and its credentials.

Reference implementation and extensions

The storage OO workflows included with Insight Dynamics are a reference implementation for the HP StorageWorks XP12000 and HP StorageWorks XP24000. These reference storage OO workflows can be extended or modified to meet the needs of a specific customer environment.

The reference implementation uses the CLI (Command Line Interface) provided by XP Command View Advanced Edition to implement the various operations. This CLI also applies to various other models of the XP array. When invoking the appropriate OO workflow, ESA passes the ArrayID and ArrayType. This information must have been configured in the Storage Provisioning Manager by the storage administrator. Based on this information, the workflow can obtain the array management station’s credentials (through whatever means chosen by the workflow developer) and proceed to execute management operations. Management operations are invoked via the CLI and the workflows convert the resultant XML data to the required data model (using XSL) and send the data back to ESA. Inclusion of ArrayType enables the OO workflow to handle any differences in CLI syntax or semantics between various array types.

This section provides some best practices for writing OO workflows, and discusses each of the OO storage workflows in detail, noting how they can be extended to support additional environments.

Discover Arrays

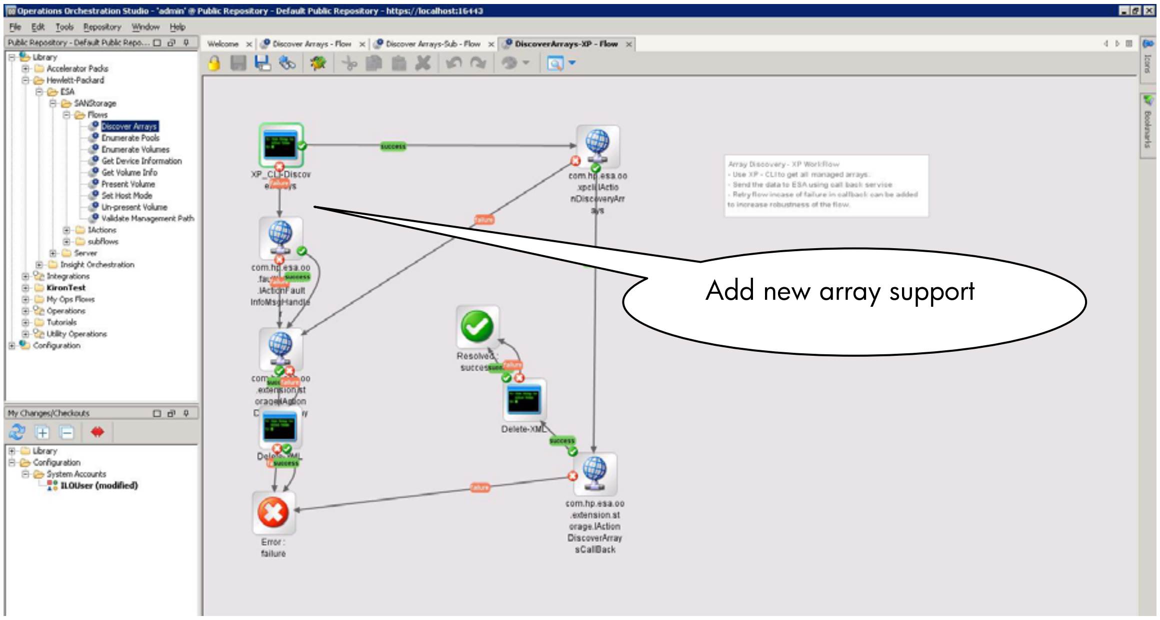
Description: The *DiscoverArrays* workflow (shown in Figure 15) is used to discover the various arrays. It is invoked by the HP Storage Provisioning Manager interface when the storage administrator specifies the array is to be managed via OO workflows. This inventory data is used to populate the storage catalog (with array discovery preceding volume discovery and optional import). The table below describes the inputs and outputs.

Discover Arrays

Details	This workflow is used to discover the various arrays which have to be managed by SPM. It is invoked by the Storage Provisioning Manager interface (when the storage administrator specifies the array is to be managed via OO workflows) This inventory data is used to populate the storage catalog (with array discovery preceding volume discovery and optional import).	
Call Type	SOAP Web Service	
Called By	ESA	
Input Parameters	esaMessageId	EsaMessage ID
	arrayContext	See ArrayContext
	replytoURI	URI for sending the information back to ESA.
Output	Call ESA SOAP WebService :-GetDeviceInfoCallback	
Output Parameters	esaMessageId	Message ID as sent during invocation
	sanArrayList	See storageSchema.Do Not populate controllerList .
	operationStatus	Send OperationStatus.SUCCESS if succeeded ELSE send OperationStatus.FAILED.See OperationStatus
	faultInfoList	Should be populated if OperationStatus.FAILED.See FaultInfoList

Extensibility:
To add support for new arrays, new IAction blocks or CLI blocks (based on the customer scenario) can be inserted at the location shown in the workflow in Figure 15. That subflow would return the list of arrays intended to be managed by Insight Dynamics via OO workflows through SPM/ESA. For the XP arrays, the information is gathered via the `getstoragearray` command. An EMC environment might use the `symcfg list` command (or SMI-S instrumentation) to get the arrays and their SymmID values (to be used as Array UUIDs).

Figure 15: Discover Arrays workflow



Array Discovery - XP WorkFlow
- Use XP - CLI to get all managed arrays.
- Send the data to ESA using call back service
- Retry flow incase of failure in callback can be added to increase robustness of the flow.

Add new array support

Get Device Information

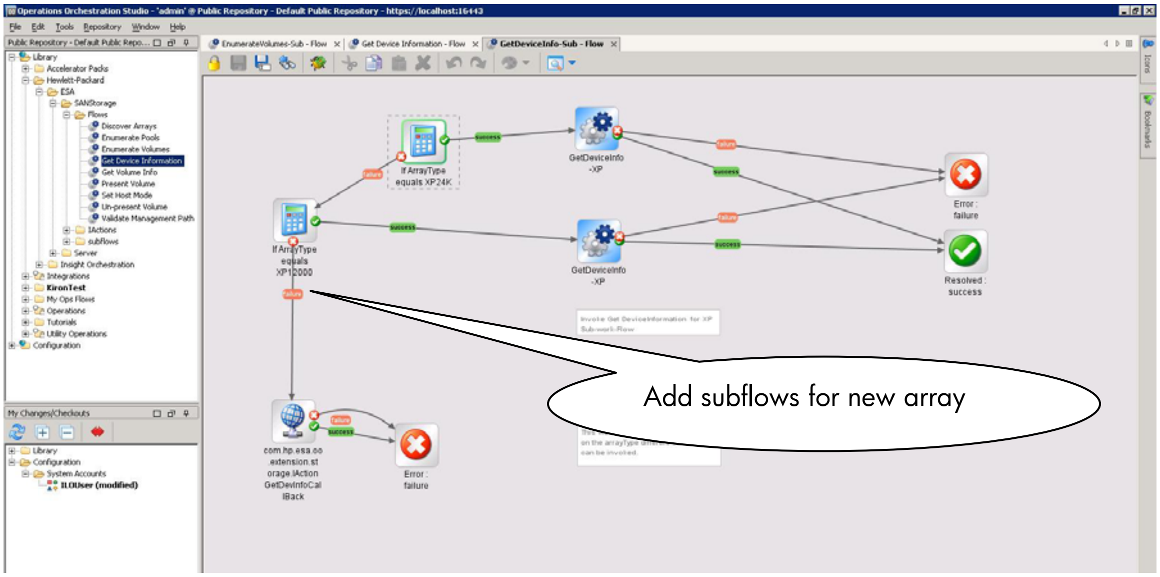
Description: The *Get Device Information* workflow (shown in Figure 16) is used to obtain detailed information about a particular array (given its array ID). The table below describes the inputs and outputs.

Get Device Info

Details	This workflow is used to get information about the storage array. Only array device information must be populated. Pools and Volume information must not be passed back.	
Call Type	SOAP Web Service	
Called By	ESA	
Input Parameters	esaMessageId	EsaMessage ID
	arrayContext	See ArrayContext
	replytoURI	URI for sending the information back to ESA.
Output	Call ESA SOAP WebService :-GetDeviceInfoCallback	
Output Parameters	esaMessageId	Message ID as sent during invocation
	sanArray	See storageSchema.Do Not populate controllerList .
	operationStatus	Send OperationStatus.SUCCESS if succeeded ELSE send OperationStatus.FAILED.See OperationStatus
	faultInfoList	Should be populated if OperationStatus.FAILED.See FaultInfoList

Extensibility:
To add support a new array such as EMC, the workflow writer can insert subflows at the location shown in Figure 16. The subflow would gather the appropriate information from the array, using a CLI or SMI-S or another mechanism of gathering data, which would be transformed to conform to the ESA XML schema. For XP, the `getstoragearray` command is used. For an EMC environment, the `symcfg` command might be used (or SMI-S information collection).

Figure 16: Get Device Information workflow



Add subflows for new array

Enumerate Pools on device

Description: The *Enumerate Pools* workflow (shown in Figure 17) is used to list out the various pools inside a given array. The pool-ID is later used by the *Enumerate Volumes* workflow to list out the volumes for that pool. In the HP StorageWorks XP24000 array reference implementation each pool corresponds to a disk-group type. In case of storage arrays such as EMC Clarion which do not have pools, a dummy pool can be created which can encompass all the volumes. The table below describes the inputs and outputs.

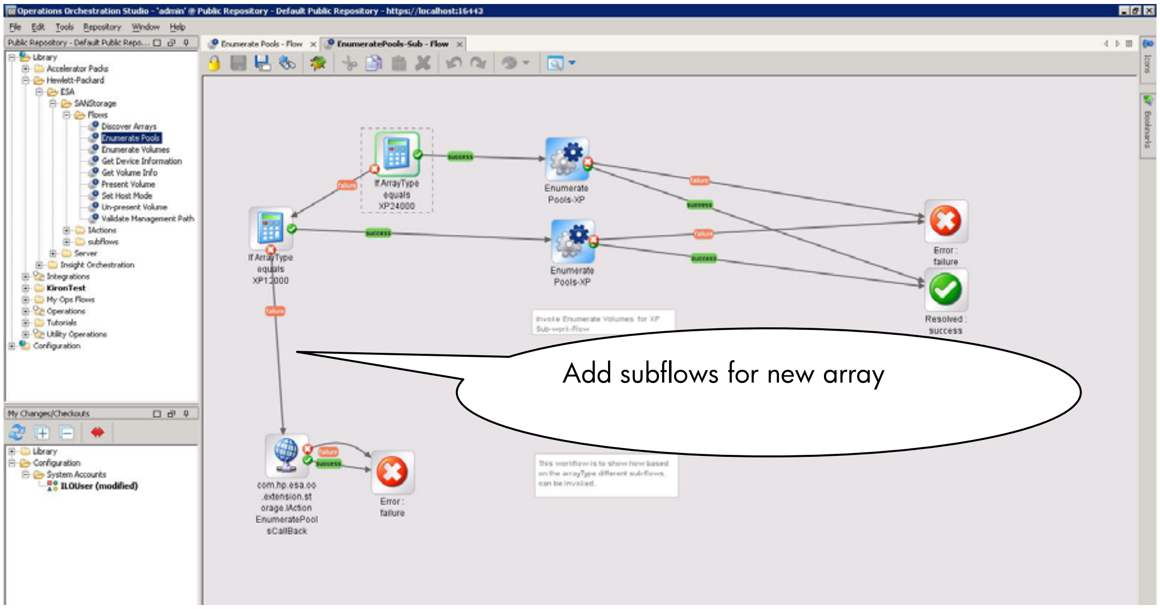
Enumerate Pools on Device

Details	This workflow is used to get information about disk pools inside a storage array. Only pool related information is populated. Volume information must not be passed back.	
Call Type	SOAP Web Service	
Called By	ESA	
Input Parameters	esaMessageId	EsaMessage ID
	arrayContext	See ArrayContext
	replytoURI	URI for sending the information back to ESA.
Output	Call ESA SOAP WebService :- enumeratePoolsCallback	
Output	esaMessageId	Message ID as sent during invocation

Parameters	arrayPoolList	See the storageSchema for details. The volumeList is not populated.
	operationStatus	Send OperationStatus.SUCCESS if succeeded ELSE send OperationStatus.FAILED. See OperationStatus
	faultInfoList	Should be populated if OperationStatus.FAILED. See FaultInfoList

Extensibility:
To add support a new array such as EMC, the workflow writer can insert subflows at the location shown in Figure 17. As noted earlier, if the array does not have the concept of pools to contain volumes, a dummy pool can be created which contains all volumes on the array. For XP, the `getstoragearray` command is used. For an EMC environment, the `symdg –sid <ArrayUID> list –v` command might be used (or SMI-S information collection).

Figure 17: Enumerate Pools workflow



Invoke Enumerate Volumes for XP Sub-work-flow

This workflow is to show how based on the arrayType different sub-flows can be invoked.

Add subflows for new array

Enumerate Volumes on device

Description: The *Enumerate Volumes* workflow (shown in Figure 18) is used to list out the various volumes for a given arrayID and PoolID. The table below describes the inputs and outputs.

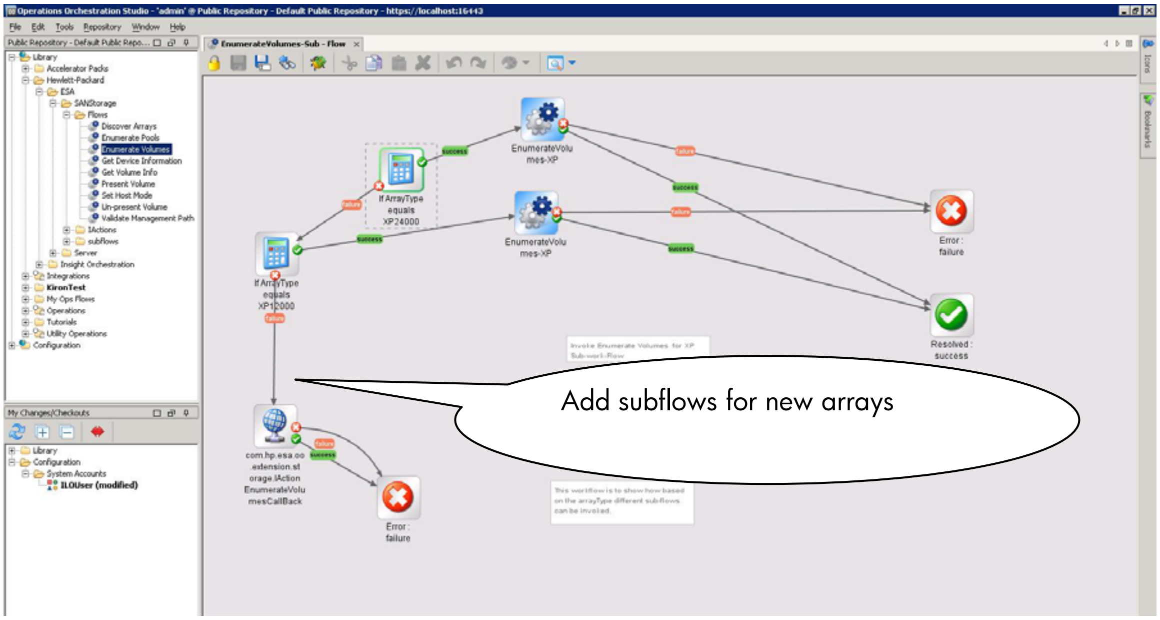
Enumerate Volumes

Details	This workflow is used to get information about disk volumes for a given pool inside a storage array. The detailed information of the volumes is not populated. Only the volumes deviceId is populated.
Call Type	SOAP Web Service

Called By	ESA	
Input Parameters	esaMessageId	EsaMessage ID
	arrayContext	See ArrayContext
	replytoURI	URI for sending the information back to ESA.
	poolUID	Pools Unique ID
Output	Call ESA SOAP WebService :- enumerateVolumesCallback	
Output Parameters	esaMessageId	Message ID as sent during invocation
	arrayVolumeList	See the storageSchema for details.
	operationStatus	Send OperationStatus.SUCCESS if succeeded ELSE send OperationStatus.FAILED. See OperationStatus
	faultInfoList	Should be populated if OperationStatus.FAILED. See FaultInfoList

Extensibility:
To add support a new array such as EMC, the workflow writer can insert subflows at the location shown in Figure 18. For XP, the `getstoragearray` command is used. For an EMC environment, the `symdev -sid <ArrayUID> list -v` command might be used (or SMI-S information co

Figure 18: Enumerate Volumes workflow



Invoke Enumerate Volumes for XP Sub-work-flow

Add subflows for new arrays

This workflow is to show how based on the arrayType different sub-flows can be invoked.

Get volume information

Description: The *Get Volume* workflow (shown in Figure 19) is used to gather details about a given volume inside a given array. The details include host information such as the initiator WWN of the host HBA (Host Bus Adapter), its host mode, and the controller port through which it is visible to the host. The ESA WSDL provides details on the data model.

Handling of hostmode will probably be unique to each array. The requests from Insight Dynamics (thru the Storage Provisioning Manager) will note the hostmode required by the appropriate logical server. Some arrays may have explicit hostmodes for each OS, others may have a "standard" mode which works for several OS flavors (such as Windows and Linux), and specialized OS modes for HP-UX or other variants.

In the HP StorageWorks XP12000 and HP StorageWorks XP24000 reference implementation, a host mode "Standard" was mapped to the Linux hostmode (but SPM is aware of this fact and can match appropriately). This mapping logic was implemented using XSL logic.

Standard hostmodes on other arrays can be handled in the manner the workflow writer deems appropriate (e.g., maybe it makes sense for all the Standards to map to Windows, or 50% to Windows and the rest to Linux, or based on other environmental knowledge).
Hostmode support on EMC arrays may vary based on microcode version, with earlier arrays requiring hostmodes suitable for HP-UX (with VSA enabled) to be at a per-port versus a per volume basis). Later versions enable specification per volume (with the –set heterogeneous option). Good knowledge of the array being supported will enable the workflow writer to make the appropriate choice.

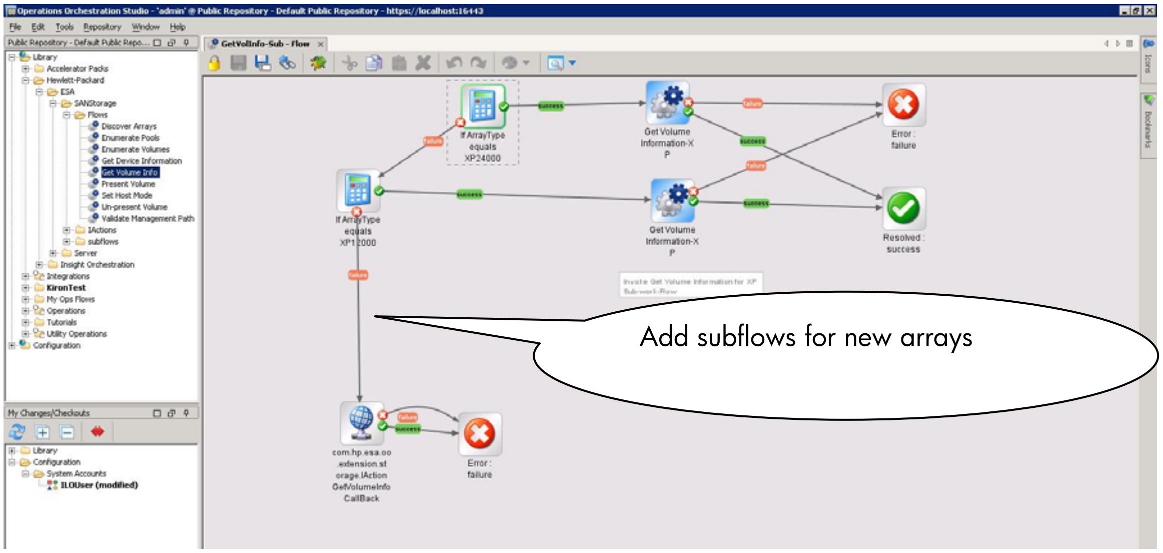
The table below describes the inputs and outputs.

Get Volume Information

Details	This workflow is used to get detailed information about a volume for given a storage array.	
Call Type	SOAP Web Service	
Called By	ESA	
Input Parameters	esaMessageId	EsaMessage ID
	arrayContext	See ArrayContext
	replytoURI	URI for sending the information back to ESA.
	lunUID	LUN's Unique ID
Output	Call ESA SOAP WebService :- getVolumeInfoCallback	
Output Parameters	esaMessageId	Message ID as sent during invocation
	sanArrayVolume	See the storageSchema for details.
	operationStatus	Send OperationStatus.SUCCESS if succeeded ELSE send OperationStatus.FAILED. See OperationStatus
	faultInfoList	Should be populated if OperationStatus.FAILED. See FaultInfoList

Extensibility:
To add support a new array such as EMC, the workflow writer can insert subflows at the location shown in Figure 19. For XP, the `getstoragearray` command is used. For an EMC environment, the `symdev` and `symmaskdb` commands might be used (or SMI-S information collection).

Figure 19: Get Volume Information workflow



Add subflows for new arrays

Present LUN

Description: The *Present LUN* workflow (shown in Figure 20) is used to present a volume to host(s) given the initiator WWN(s) and the volume ID This is an active call which modifies the configuration of the array. This operation enables the LUN to be visible to the host. Recall that the storage administrator can use SPM to control which volumes offer this capability (and the workflow writer could decide to always return failure if they did not want to support this operation). Without support for this operation, the Insight Dynamics solution cannot programmatically unmask the data LUNs to ensure they are invisible during OS install and later mask them back to visibility.

The table below describes the inputs and outputs.

Present LUN

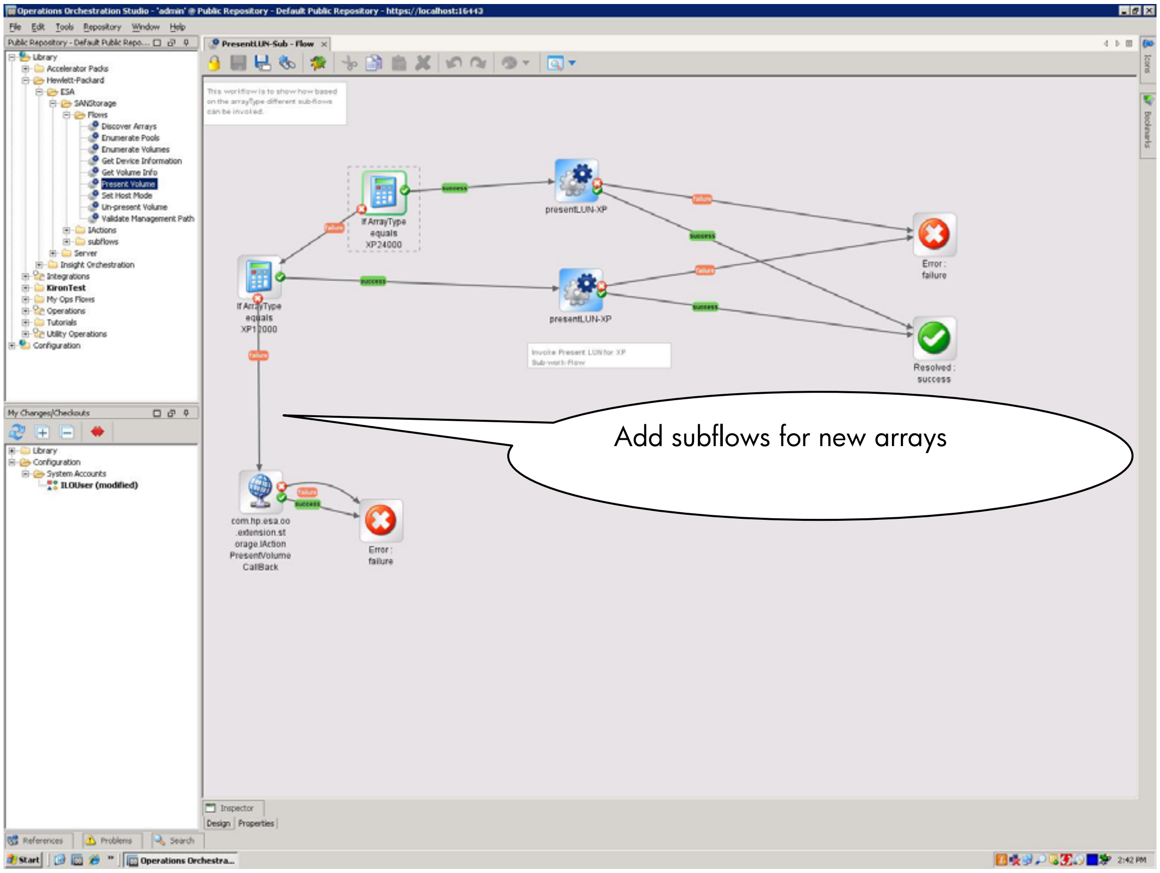
Details	This workflow is used to present a LUN/volume to a list of hosts.	
Call Type	SOAP Web Service	
Called By	ESA	
Input Parameters	esaMessageId	EsaMessage ID
	arrayContext	See ArrayContext
	replytoURI	URI for sending the information back to ESA.
	lunUID	LUN's Unique ID
	hostMappers	XML String of Host Mapper information. See the storageSchema for details.
Output	Call ESA SOAP WebService :- presentVolumeCallback	
Output Parameters	esaMessageId	Message ID as sent during invocation
	operationStatus	Send OperationStatus.SUCCESS if succeeded Send OperationStatus.FAILED if the operation fails for all the hosts. Send OperationStatus. PARTIAL_SUCCESS if the operation succeeds only for a few hosts.

			See OperationStatus	
		faultInfoList	Should be populated if OperationStatus.FAILED or OperationStatus.PARTIAL_SUCCESS. See FaultInfoList	

For the HP StorageWorks XP 12000 and HP StorageWorks XP24000 reference implementation in Insight Dynamics 6.0, the *Present LUN* operation will enable the LUN to be visible to the host through the pre-configured array controller port(s). The array controller port configuration is pre-designed by the storage administrator and made visible to SPM though Get Device Information. With the 6.1 and later releases, the only requirement is to use Storage Provisioning Manager to identify the suitable fabric for each disk array controller port which is to be used; ports not being used should have no fabric assignment.

Extensibility:
To add support a new array such as EMC, the workflow writer can insert subflows at the location shown in Figure 20. For XP, the `addLUN` command is used. For an EMC environment, the `symmask` command might be used (or SMI-S information collection).

Figure 20: Present LUN workflow



This workflow is to show how based on the arrayType different sub-flows can be invoked.

Invoke Present LUN for XP Sub-work-flow

Unpresent LUN

Description: The *Unpresent LUN* workflow (show in Figure 21) is used to unpresent the volume visible to a host through a given array controller port. This is an active call which modifies the configuration of the array.

In the XP24000 reference implementation, the unpresent LUN operation only removes the connection between the Controller port of the array and the Host HBA. The internal configuration of the LUN being connected to the array controller port is kept intact. The implementation will also validate if any other LUNs are using the host definition which may have been created when presenting the LUN to the specified initiator WWN. If there are no other users of the host definition created by ESA, the host definition is removed (since arrays often limit the number of host definitions they support).

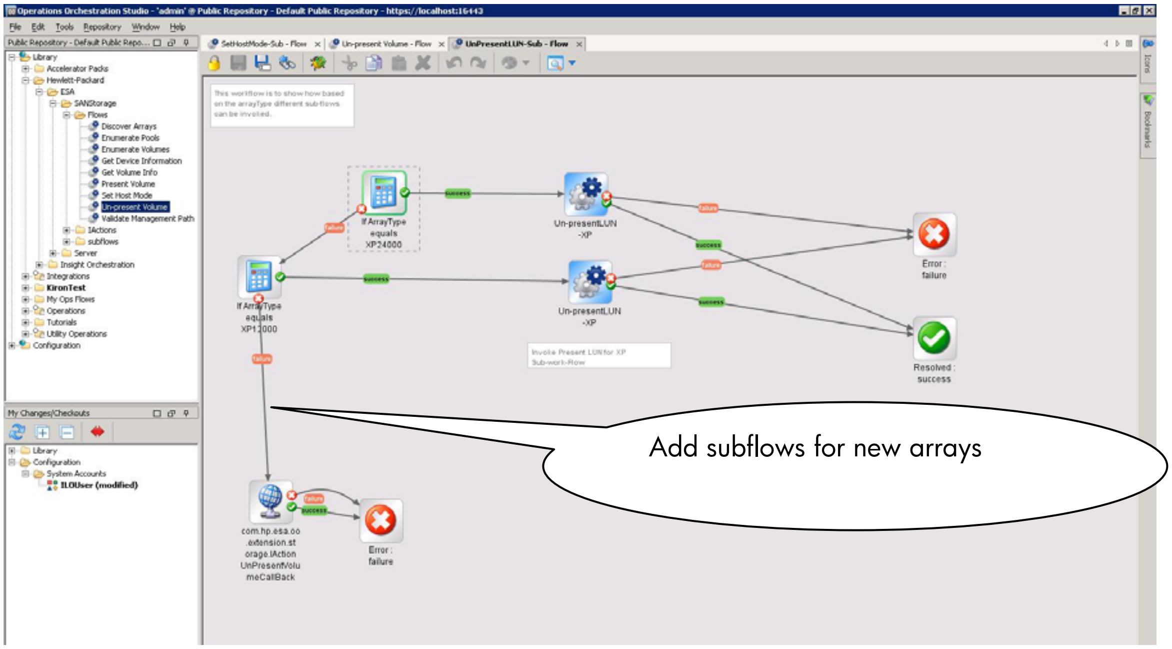
The table below describes the inputs and outputs.

Unpresent LUN

Details	This workflow is used to un-present a LUN/volume to a list of hosts.	
Call Type	SOAP Web Service	
Called By	ESA	
Input Parameters	esaMessageId	EsaMessage ID
	arrayContext	See ArrayContext
	replytoURI	URI for sending the information back to ESA.
	lunUID	LUN's Unique ID
	hostMappers	XML String of Host Mapper information. See the storageSchema for details.
Output	Call ESA SOAP WebService :- unpresentVolumeCallback	
Output Parameters	esaMessageId	Message ID as sent during invocation
	operationStatus	Send OperationStatus.SUCCESS if succeeded Send OperationStatus.FAILED if the operation fails for all the hosts. Send OperationStatus. PARTIAL_SUCCESS if the operation succeeds only for a few hosts. See OperationStatus
	faultInfoList	Should be populated if OperationStatus.FAILED or OperationStatus. PARTIAL_SUCCESS. See FaultInfoList

Extensibility:
To add support a new array such as EMC, the workflow writer can insert subflows at the location shown in Figure 21. For XP, the `deletedLUN` command is used. For an EMC environment, the `symmask` command might be used (or SMI-S information collection).

Figure 21: Unpresent LUN workflow



This workflow is to show how based on the arrayType different sub-flows can be invoked.

Invoke Present LUN for XP Sub-work-flow

Add subflows for new arrays

Set hostmode

Description: The *Set Hostmode* workflow (shown in Figure 22) is used to change the hostmode of a given host presentation of a volume. This is an active call which modifies the configuration of the array. Recall that the storage administrator can use SPM to control which volumes offer this capability (and the workflow writer could decide to always return failure if they did not want to support this operation). Without support for this operation, the Insight Dynamics solution cannot programmatically change the hostmode of a volume, and thus the storage catalog needs to have volumes of all possible hostmodes available (versus having a set of volumes flexibly changes to the hostmode requried by the logical server).

In the HP StorageWorks XP12000 and HP StorageWorks XP24000 reference implementation, the *Set Hostmode* operation with a hostmode of LINUX is mapped to 'Standard' hostmode using the XP-CLI. (Note: this mapping between Standard to Linux host mode mapping also done in the Get Volume Information workflow).

The table below shows the inputs and outputs.

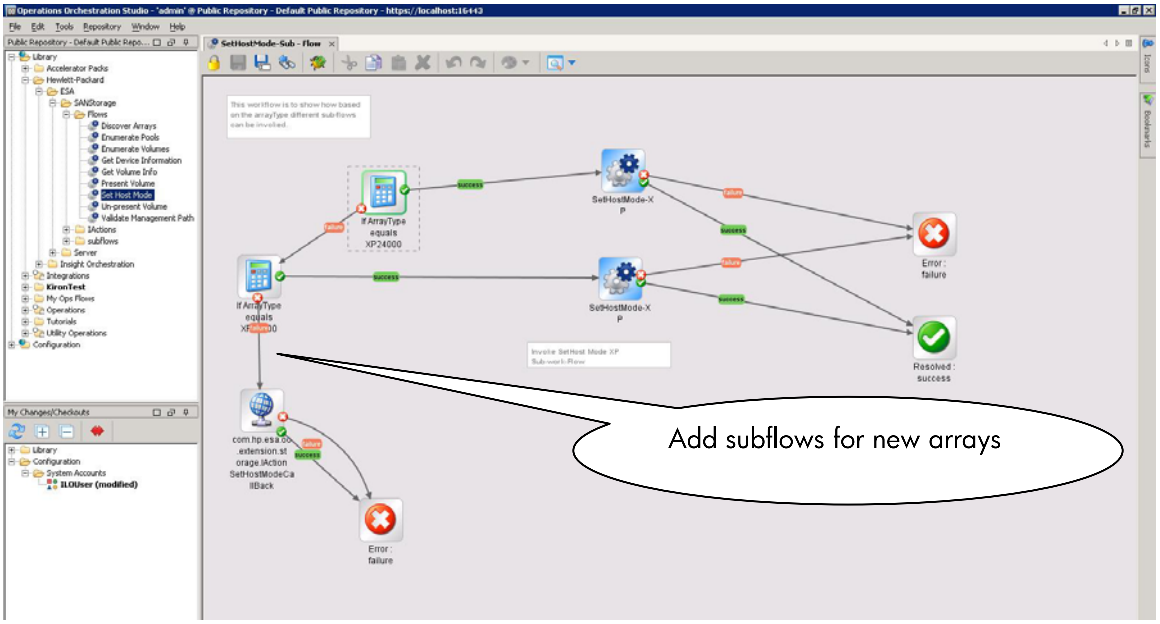
Set hostmode

Details	This workflow is used to set host mode of a LUN/volume to a list of hosts.	
Call Type	SOAP Web Service	
Called By	ESA	
Input Parameters	esaMessageId	EsaMessage ID
	arrayContext	See ArrayContext
	replytoURI	URI for sending the information back to ESA.
	lunUID	LUN's Unique ID
	hostMappers	XML String of Host Mapper information.

		See the storageSchema for details.
Output	Call ESA SOAP WebService :- setHostModeCallback	
Output Parameters	esaMessageId	Message ID as sent during invocation
	operationStatus	Send OperationStatus.SUCCESS if succeeded Send OperationStatus.FAILED if the operation fails for all the hosts. Send OperationStatus. PARTIAL_SUCCESS if the operation succeeds only for a few hosts. See OperationStatus
	faultInfoList	Should be populated if OperationStatus.FAILED or OperationStatus.PARTIAL_SUCCESS. See FaultInfoList

Extensibility:
To add support a new array such as EMC, the workflow writer can insert subflows at the location shown in Figure 22. For XP, the `addhost,addhoststoragedomain, addWWNforhoststoragedomain,`and `modifyport` commands are used (adding the host and domain if necessary before the port modify). For an EMC environment, the `symcfg -sid <ArrayUID> list -v` command might be used (or SMI-S information collection).

Figure 22: Set Hostmode workflow


This workflow is to show how based on the arrayType different sub flows can be invoked
If ArrayType equals XP24000
SetHostMode-X P
SetHostMode-X P
Error : failure
Resolved : success
Invoke SetHost Mode XP Sub-work-flow
com.hp.esa.oo.extension.storage.IAction SetHostModeCallBack
Error : failure
Add subflows for new arrays

Validate Management Path

Description: The *Validate Management Path* workflow (shown in Figure 23) is used to validate the management path of the array (validating that the necessary storage operations can be performed on the array). The command used here is same as that used for the getDeviceInfo call.

In the HP StorageWorks XP12000 and HP StorageWorks XP24000 reference implementation, the *Validate Management Path* workflow is used to verify if the XP-CLI is able to access storage using the XP CommandView AE CLI management path.

Note: The main reason for this functionality is to enable the Storage Provisioning Manager to maintain similar user experiences across arrays managed by OO workflows and SMI-S. SPM will check communication with the SMI-S client, while this workflow is used for communication through whatever means is used by the OO workflow.

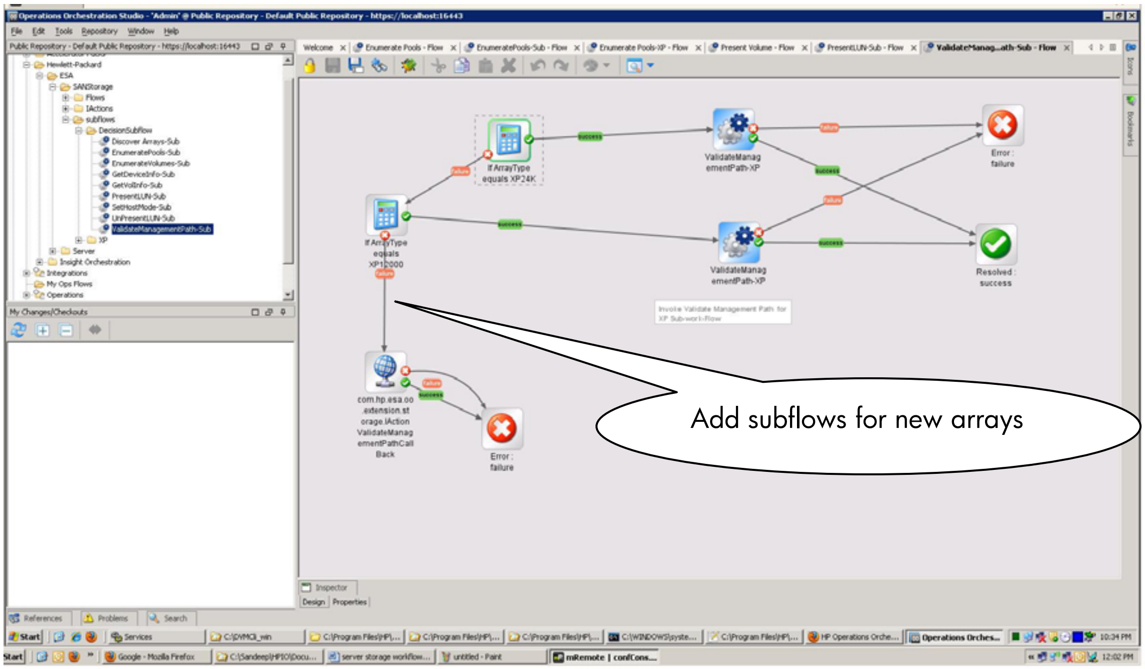
The table below shows the inputs and outputs.

Validate Management Path

Details	This workflow is used to validate the management path.	
Call Type	SOAP Web Service	
Called By	ESA	
Input Parameters	esaMessageId	EsaMessage ID
	arrayContext	See ArrayContext
	replytoURI	URI for sending the information back to ESA.
Output	Call ESA SOAP WebService :- validateManagementPathCallback	
Output Parameters	esaMessageId	Message ID as sent during invocation
	operationStatus	Send OperationStatus.SUCCESS if succeeded Send OperationStatus.FAILED if the operation fails for all the hosts. See OperationStatus
	faultInfoList	Should be populated if OperationStatus.FAILED or OperationStatus.PARTIAL_SUCCESS. See FaultInfoList

Extensibility:
To add support a new array such as EMC, the workflow writer can insert subflows at the location shown in Figure 23. For XP, the `getstoragearray` command (similar to getDeviceInfo) is used. For an EMC environment, the mechanism being used for array operations would be used (e.g., perhaps the `symcfg` command, or SMI-S information collection, or Storage Essentials operations).

Figure 23: Validate Management Path workflow



Add subflows for new arrays

Notes regarding the XP reference implementation

There are a few limitations or specific features of the XP12000 and XP24000 reference implementation:

- After Unpresent LUN, the host entry information and Host Storage domain related entries are deleted as a part of the 'CleanUp Operation'. This clean up operation will only happen if the Host/Host StorageDomain has no path related information.
- If the user credentials used by the CLI are being used by any other UI (User Interface)/CLI then none of the active calls (Present, Unpresent and SetHost Mode) operations will go through. A suitable error message is sent to ESA.
- The XP-AE is assumed to be synchronized with actual XP array hardware, otherwise a XP-synchronization error is sent back to ESA.
- During Enumerate Volumes in a pool, the following type of volumes are listed: "continuousAccessVolumeType"=Simplex, "businessCopyVolumeType"=Simplex, "snapshotVolumeType"=Simplex, "journalVolumeType"=Simplex.,"path = true.
- If an already presented LUN is presented to the same WWN again, then the workflows return an error. This is true with Un-present operation also.
- For setHost mode operations with LINUX as an input, the host mode is set as Standard in the OO-Workflow Implementation. During getVolume operation a Standard host mode is mapped to LINUX host mode.
- The HP StorageWorks XP12000 and HP StorageWorks XP24000 reference implementation has been implemented to handle only one XP-AE station. The XP-AE CLI used can handle multiple XP-AE stations. This can be achieved by passing the URL of the XP-AE station to the

CLI using –URL option. (Please refer the XP –AE CLI documentation for more details.) This implementation can be extended to handle multiple XP-AE stations by having a lookup implemented inside OO handling the various XP-AE URLs versus the array Unique IDs.

Adding support for additional storage types

The storage OO workflows are designed to extend to additional storage types (still required to use pre-provisioned Fibre Channel storage). For example, it may be appropriate to change the reference implementation to work with EMC storage rather than XP storage (or perform validation/modification for various models of XP). The following sections outline the steps necessary to add that storage support.

Prerequisites

It is key to first ensure a sufficient understanding array to be supported, its management interfaces, the ESA data model and reference implementation, and OO workflows. Customizing the XP reference implementation to support additional arrays requires more than a simple command substitution in the OO workflows, as described below.

1. Deep understanding of array concepts. Each array often has unique ways to represent the common concepts of pooling storage, allocating volumes, and presenting LUNs (including hostmode considerations). Once these concepts are known for the given array, they can be mapped to the ESA data model (and the XP reference implementation provides a useful reference)

2. Understanding of the ESA's Web Service data models. The ESA data model is based on SMI-S and is documented in the WSDL (see the earlier *Locations of XSD ad WSDL files* section for location information). Note the inclusion of Extended Attributes which can be used to extend the data model with information that remains opaque to ESA and Insight Dynamics (but may be useful in supporting particular arrays). For example, in the XP reference workflow, the “Domain ID of the LUN” is being passed using extended attributes.

3. XSL knowledge to transform the data. Often the array information may be returned to the OO workflow in XML format, and use of XSL is helpful to transform the data. The XP reference implementation uses XML and XSL; the EMC symcli interface provides an option to get the results in XML format.

4. Good understanding of OO workflows (e.g., use of IAction classes, deployment, and testing). This document provides some high-level information and details are available in the OO documentation.

5. Understanding of the XP reference implementation (as explained in this document and the supplied WSDL information).

Steps to write a storage OO workflow

This section outlines the various steps, and subsequent sections provide more details.

1. Understand the inputs and outputs of the workflow to be written (e.g., *GetDeviceInformation* for an EMC array rather than an XP array). For a complete implementation, it is necessary to write all of the storage workflows (although the workflow writer may choose to have some simply return a failure status if that operation is not to be supported, e.g., if not allowing hostmode changes or LUN presentation). Privileges within the HP Storage Provisoining Manager can be used to ensure certain workflows are not invoked (e.g., setting a catalog entry to view only means SPM will not invoke the workflow to change hostmode or perform LUN maksing); **but** be aware that the catalog owner can perform all operations, so it may still be possible the workflow would be invoked and should return an appropriate response (perhaps failure if you don't support that operation).

2. Branch off another subflow in the reference implementation based on the ArrayType parameter. The HP StorageWorks XP12000 and HP StorageWorks XP24000 reference implementation uses branching because of similar CLI and commands. It is recommended to create a separate subflow if the command lines are different. (The OO documentation provides details on creating subflows)

3. Based on the array knowledge, determine the appropriate commands to be called for the given Array (Figure 24 shows an example of mapping XP commands to EMC symcli equivalents). It is not necessary to use commands to manage the new array; the workflow writer could choose to use SMI-S or a scripting language or some other appropriate means of managing the array.

4. Execute the commands, programs, or scripts and gather outputs.

5. Identify the data which you need to send back to ESA (based on the required inputs and outputs). Note that there is also support for extended attributes, should the workflow writer have the need to pass data not explicitly included in the ESA data model.

6. Use XSL or other mechanisms to convert the data to the format which Output APIs accepts. The *Get Device Information* section discusses the Output parameters for the *GetDeviceInformation* workflow.

7. Test the data manipulation independently, then incorporate the command/script invocation into the workflow and test full solution (e.g., does the Storage Provisioning Manager have the ability to discover the array and see the volumes, change hostmodes, do LUN presentation). See the *Testing and troubleshooting* section for details.

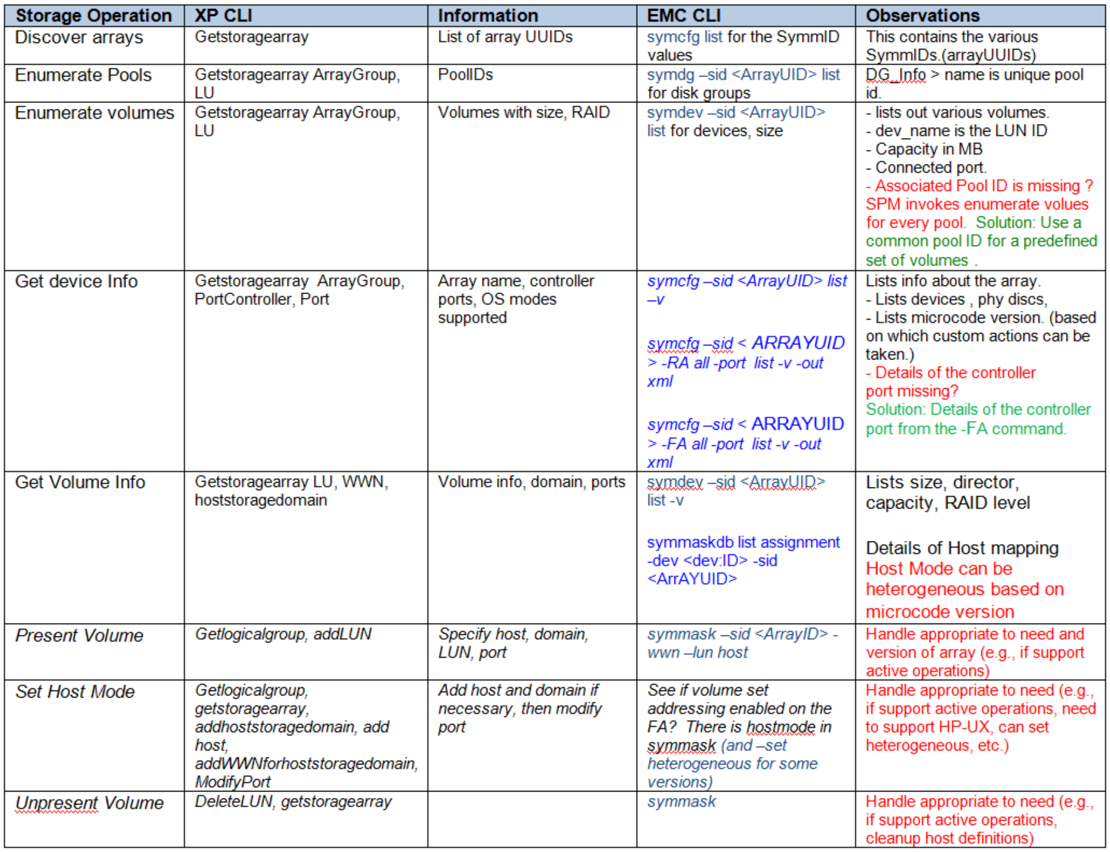
Determine appropriate interfaces

As noted, one of the key prerequisites of implementing a set of OO workflows for Insight Dynamics to successfully manage a new array is a strong understanding of the array, its management interfaces, and how those concepts map to the ESA data model.

The following table shows an example of the various storage operations, how they are implemented in the XP reference implementation via the CLI, the information required, and a sample mapping to a new array type (in this case an EMC array using the symcli environment).

Note: The included reference implementation **DOES NOT** support EMC. The following information is supplied to demonstrate the process that would be used to consider another array type.

Figure 24: Sample mapping of XP commands to EMC array environment



Storage Operation	XP CLI	Information	EMC CLI	Observations
Discover arrays	Getstoragearray	List of array UUIDs	symcfg list for the SymmID values	This contains the various SymmIDs.(arrayUUIDs)
Enumerate Pools	Getstoragearray ArrayGroup, LU	PoolIDs	symdg –sid <ArrayUID> list for disk groups	DG_Info > name is unique pool id.
Enumerate volumes	Getstoragearray ArrayGroup, LU	Volumes with size, RAID	symdev –sid <ArrayUID> list for devices, size	- lists out various volumes. - dev_name is the LUN ID - Capacity in MB - Connected port. - Associated Pool ID is missing ? SPM invokes enumerate volues for every pool. Solution: Use a common pool ID for a predefined set of volumes .
Get device Info	Getstoragearray ArrayGroup, PortController, Port	Array name, controller ports, OS modes supported	*symcfg –sid <ArrayUID> list –v* *symcfg –sid < ARRAYUID > -RA all -port list -v -out xml* *symcfg –sid < ARRAYUID > -FA all -port list -v -out xml*	Lists info about the array. - Lists devices , phy discs, - Lists microcode version. (based on which custom actions can be taken.) - Details of the controller port missing? Solution: Details of the controller port from the -FA command.
Get Volume Info	Getstoragearray LU, WWN, hoststoragedomain	Volume info, domain, ports	symdev –sid <ArrayUID> list -v symmaskdb list assignment -dev <dev:ID> -sid <ArrAYUID>	Lists size, director, capacity, RAID level Details of Host mapping Host Mode can be heterogeneous based on microcode version
Present Volume	*Getlogicalgroup, addLUN*	*Specify host, domain, LUN, port*	*symmask –sid <ArrayID> -wwn –lun host*	Handle appropriate to need and version of array (e.g., if support active operations)
Set Host Mode	*Getlogicalgroup, getstoragearray, addhoststoragedomain, add host, addWWNforhoststoragedomain, ModifyPort*	*Add host and domain if necessary, then modify port*	*See if volume set addressing enabled on the FA? There is hostmode in symmask (and –set heterogeneous for some versions)*	Handle appropriate to need (e.g., if support active operations, need to support HP-UX, can set heterogeneous, etc.)
Unpresent Volume	*DeleteLUN, getstoragearray*		*symmask*	Handle appropriate to need (e.g., if support active operations, cleanup host definitions)

The workflow writer would use their array knowledge to determine the appropriate mechanisms to perform the necessary storage operations.   Depending on the customer environment, active operations such as presenting LUNs and changing hostmode may or may not be supported. Differences in various arrays would need to be handled (e.g,. supported hostmodes, which may vary by array or even microcode versions).   There are often subtleties in handling OS modes for HP-UX and Windows 2008.

The table above shows some considerations in red text, along with green text outlining possible solutions.   As demonstrated, in some situations it may be necessary to combine the results of several commands in order to gather all the required information. As noted earlier, a workflow writer can decide not to support certain operations (but needs to realize that may preclude use of certain Insight Dynamics features – such as using LUN masking to hide data volumes prior to OS installation).

The workflow writer would invoke the appropriate commands (or call an API or the SMI-S interfaces) and examine the results, determining how that information can be transformed to conform to the ESA data model.   If the responses are in XML format, XSL transforms can be used to manipulate the information (and tested independently of the OO workflow interactions).

Modify storage OO workflows

Given an understanding of how to perform the necessary storage operations and gather and transform the appropriate information, the workflow writer will proceed to modify the storage workflows. To minimize the amount of changes, it is suggested new arrays be supported with additional subflows, as noted in the sections on each storage OO workflow. The XP implementation can be left unaltered, and it would not be invoked unless the customer environment contains XP arrays with the model information that triggers those workflows.

When there is great commonality in CLI/commands/interfaces for a variety of array models, it may be appropriate to combine those into a single subflow, using branching to handle small differences when appropriate. Typically, supporting a new array will require a significantly new set of interfaces, and thus creation of a separate subflow is recommended.

Each workflow would be modified to branch to the new subflow based on the ArrayType parameter. Those subflows would invoke the appropriate interfaces to perform storage operations on the arrays, conforming to the ESA data model (described in the WSDL).

Security credential configuration

OO allows workflow developers to store the credentials in a secure fashion. This feature can be used by the workflow developer to store the credentials of the storage array or storage management station's credentials.

To store the credentials the developer has to create a new entry inside the System Accounts folder (as was shown earlier for server OO workflows in Figure 4).

For the HP StorageWorks XP12000 and HP StorageWorks XP24000 reference implementation, the 'Systems Account' feature of OO was not used as the credentials of the XP-CVAE (XP Command View Advanced Edition) management station was stored in the configuration files of the XP-CLI. (Please refer to the XP-CVAE documentation for more details.)

IAction classes

For ease of the workflow writers, the XP reference implementation bundled with Insight Dynamics includes IAction classes to help the workflows callback to the ESA SOAP web service.

The details of the Storage related IAction classes are listed below.

Iaction Class	Inputs	Description
IActionGetDevInfoCallBack	esaMessageId	esaMessageID (reused from Workflow inp
	arrayXml	Array Details as XML. (see Storage Schema details).Do not populate controllerList
	opStatus	opStatus as XML.
	faultInfoList	Fault Info List as XML
	replytoURI	replyToURI.(reused from Workflow input)
IActionEnumeratePoolsCallBack	esaMessageId	esaMessageID (reused from Workflow inp
	arrayPoolListXml	Array Pool List as XML
	opStatus	opStatus as XML.
	faultInfoList	Fault Info List as XML
	replytoURI	replyToURI.(reused from Workflow input)

IActionEnumerateVolumesCallBack	esaMessageId	esaMessageID (reused from WorkFlow inp
	arrayVolumeListXml	Array Volume List as XML
	opStatus	opStatus as XML.
	faultInfoList	Fault Info List as XML
	replytoURI	replyToURI.(reused from Workflow input)
IActionGetVolumeInfoCallBack	esaMessageId	esaMessageID (reused from Workflow inp
	arrayVolumeXml	Array Volume details as XML.
	opStatus	opStatus as XML.
	faultInfoList	Fault Info List as XML
	replytoURI	replyToURI.(reused from Workflow input)
IActionPresentVolumeCallBack	esaMessageId	esaMessageID (reused from Workflow inp
	opStatus	opStatus as XML.
	faultInfoList	Fault Info List as XML
	replytoURI	replyToURI.(reused from Workflow input)
IActionUnPresentVolumeCallBack	esaMessageId	esaMessageID (reused from Workflow inp
	opStatus	opStatus as XML.
	faultInfoList	Fault Info List as XML
	replytoURI	replyToURI.(reused from Workflow input)
IActionSetHostModeCallBack	esaMessageId	esaMessageID (reused from Workflow inp
	opStatus	opStatus as XML.
	faultInfoList	Fault Info List as XML
	replytoURI	replyToURI.(reused from Workflow input)
IActionValidateManagementPathCallBack	esaMessageId	esaMessageID (reused from Workflow inp
	arrayXml	Array Details as XML. (see Storage Schema details).Do not populate controllerList
	opStatus	opStatus as XML.
	faultInfoList	Fault Info List as XML
	replytoURI	replyToURI.(reused from Workflow input)

Notes:
i. All the xml Strings follow the ESA schema (found in the WSDL). Esa-san-model-x.jar can be used to creat xml.
ii. ESA throws a MessageException in case of invalid data/improper data.
iii. Retry Mechanism to handle ESA being down implemented in the workflow using the default operation of

The extensions from ESA are available in the CMS C:\Program Files\HP\Virtual Server Environment\esa\OO Extensions

1. esa-oo-storage-extensions-x.jar (x indicates the version of the jar)
   It contains all the callback operations for ESA.
   This can be imported into the OO for usage inside the workflow.
2. esa-san-model-x.jar, esa-oo-storage-rservice-x.jar
   These are the dependant jars. These jars can also be used by the OO integrators to write their own custom IAction classes. For example, it can be used to create XML strings according to ESA's XML schema.

Testing and troubleshooting

This section provides information on testing any changes made to the server and/or storage OO workflows.

Testing can be done at various levels. The workflow writer may test the data collection and transformation aspects independently from Insight Dynamics and OO (to validate the array operations and data transformations work as expected). The workflow writer may then proceed to edit the workflows, adding their specific subflows and contents. From that point, testing may be done within OO Studio. Finally, once testing within OO is successful, the workflow writer will perform end-to-end testing, using the Storage Provisioning Manager to validate the arrays and volumes can be discovered and imported, and utilizing that storage as the foundation for logical servers (which may take advantage of LUN masking and hostmode changes). For example, the volume may have a hostmode of Windows, but when selected for an HP-UX logical server, the appropriate workflow would be invoked to adjust the hostmode (assuming the catalog entry within SPM is not view only).

Testing a workflow or subflow

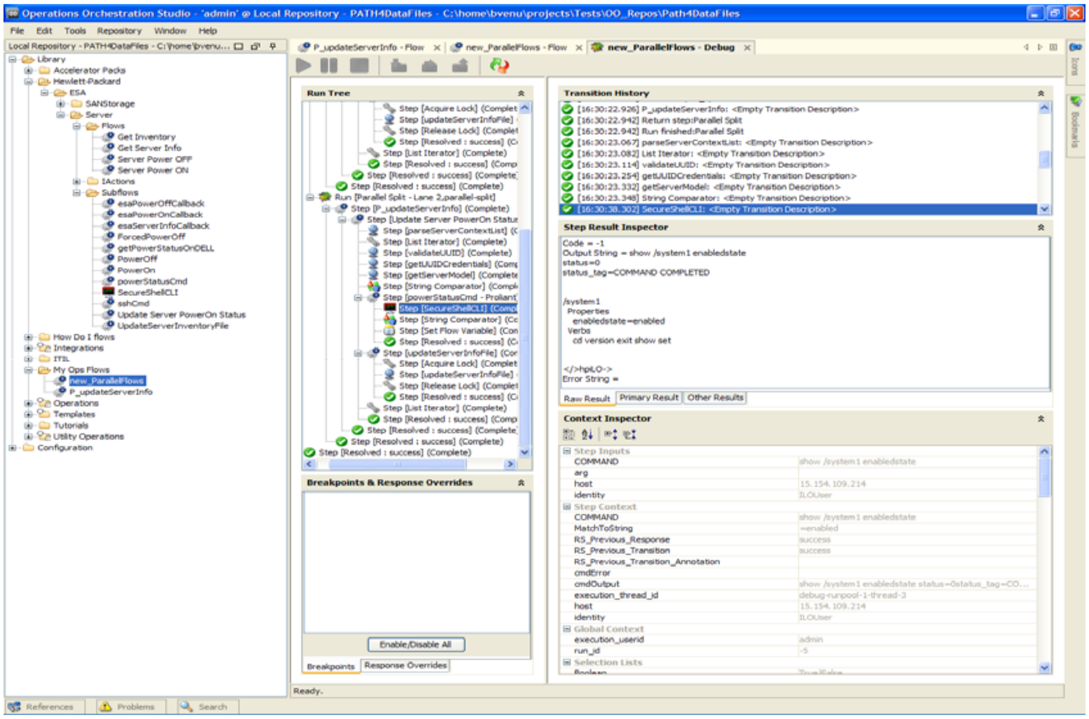
During the implementation of a flow (main or subflow) on OO Studio, OO Studio keeps checking for the errors. Once the developer successfully implements the execution logic into the flow, the behavior of the flow can be tested on the Studio using "Debug flow" tool, by providing appropriate parameters during execution.

The steps below explain how to debug a flow:

- Double click on the flow name in the explorer window to open the flow
- Click on the "Debug flow" icon on the tool bar.
- On the <flow name> - Debug window, select the "Play" button to execute the flow.
    - During execution, provide appropriate values for the flow parameters.
    - The flow execution can be paused and/or stopped any time for debugging the intermediate steps.
    - The execution results can be monitored on the "Transition History" pane and the result can be viewed for each transition step "Step Result Inspector" pane.
    - The flow can be re-executed by clicking the "reset" and "play" buttons

For more information on the flow execution in OO Studio, refer to "Authors Guide" document that comes along with OO installation (located at C:\Program Files\HP\Operations Orchestration\Studio\docs).

Figure 25: Running a workflow in a debug window


Operations Orchestration Studio - 'admin' @ Local Repository - PATH4DataFiles - C:\home\bvenu\projects\Tests\OO_Repos\Path4DataFiles
File Edit Tools Repository Window Help
P_updateServerInfo - Flow
new_ParallelFlows - Flow
new_ParallelFlows - Debug
Run Tree
Breakpoints & Response Overrides
Enable/Disable All
Breakpoints
Response Overrides
References
Problems
Search
Ready.

Transition History

```
[16:30:22.926] P_updateServerInfo: <Empty Transition Description>
[16:30:22.942] Return step:Parallel Split
[16:30:22.942] Run finished:Parallel Split
[16:30:23.067] parseServerContextList: <Empty Transition Description>
[16:30:23.082] List Iterator: <Empty Transition Description>
[16:30:23.114] validateUUID: <Empty Transition Description>
[16:30:23.254] getUUIDCredentials: <Empty Transition Description>
[16:30:23.332] getServerModel: <Empty Transition Description>
[16:30:23.348] String Comparator: <Empty Transition Description>
[16:30:38.302] SecureShellCLI: <Empty Transition Description>
```

Step Result Inspector

```
Code = -1
Output String = show /system1 enabledstate
status=0
status_tag=COMMAND COMPLETED


/system1
 Properties
  enabledstate=enabled
 Verbs
  cd version exit show set


</>hpiLO->
Error String =
```

Raw Result | Primary Result | Other Results

Context Inspector

Step Inputs	
COMMAND	show /system1 enabledstate
arg	
host	15.154.109.214
identity	ILOUser
Step Context	
COMMAND	show /system1 enabledstate
MatchToString	=enabled
RS_Previous_Response	success
RS_Previous_Transition	success
RS_Previous_Transition_Annotation	
cmdError	
cmdOutput	show /system1 enabledstate status=0status_tag=CO...
execution_thread_id	debug-runpool-1-thread-3
host	15.154.109.214
identity	ILOUser
Global Context	
execution_userid	admin
run_id	-5
Selection Lists	
Boolean	True|False

Troubleshooting

This section outlines troubleshooting steps for some of the possible issues that may arise when implementing extensions to the server and storage OO workflows.

Issues covered include

1. ESA not invoking the OO workflow
2. OO callback to ESA failing
3. OO not able to connect to a server's remote management interface
4. Failure due to specific workflow issues (covered for each server and storage workflow)

Note: ESA logs are enabled by default, no specific settings are required. The log file is located at: C:\Program Files\HP\Virtual Server Environment\logs\esa.log.

1. ESA not calling OO

   A Call from ESA to OO workflow can fail for following reasons:

   - HP SIM service is not running or not responding
   - OO RSCentral service is not running or not responding
   - OO RSJRAS service is not running or not responding
   - ESA (esa.properties) is not configured properly

   To rectify the problem, follow the steps below:

   - Start/Restart the following services:
     - HP SIM
     - RSCentral
     - RSJRAS

- Check and ensure the correct values for ESA configuration in esa.properties file on following parameters:
    - esa.oo.admin.username
    - esa.oo.admin.password
    - esa.oo.host
    - esa.oo.port
    - esa.oo.service.ws.path
- Check and ensure that the correct path is configured in esa.properties file for the workflow being invoked,

2. OO Callback to ESA is failing

   OO Call back to ESA can fail for following reasons:

   - Wrong configuration for esa.oo.server.replytoURI property in esa.properties file
   - Invalid XML response from workflow execution
   - LSA service is not responding or not running
   - ESA service is not responding or not running

   To rectify the problem, follow the steps below:

   - Check and ensure that esa.oo.server.replytoURI property has correct value to locate ESA service
   - Check the OO Central report for any erroneous step performed by the workflow during execution
       - Ensure the Workflow response include valid XML string and is compatible to the XSD defined for server services.
   - Start/Restart LSA and/or ESA service

3. OO not able to connect to server's remote management interface

   During workflow execution, some of the operations try to connect to the remote server management interfaces (iLO2). These operations can fail for following reasons:

   - The remote server management interface is not accessible due to network issues
   - Connection time out due to slow network speed
   - Wrong IP address configuration in the *uuidHostMapper.xml* file
   - Authentication failures due to invalid username/password.
   - The service (ssh) being invoked is disabled or running with non-default port on the remote server management interface

   To rectify the problem, follow the steps below:

   - Ensure the network connectivity and accessibility of the remote server management interface
   - Check and ensure the user credentials configured in OO System Account is valid for the remote server management interface (and appropriate use of the server UUID)
   - Check and ensure that the service being invoked through workflow is enabled and running on the default port.

4. GetInventory is failing

   The Get Inventory workflow can fail for following reasons:

   - The *inventoryList.xml* file is missing or not readable from the configured location: C:\Program Files\HP\Insight Orchestration\esa_extensions\server

   To rectify the problem, follow the steps below:

   - Ensure that the *inventoryList.xml* file is available in the configured location: C:\Program Files\HP\Insight Orchestration\esa_extensions\server
   - Check the OO Central reports or central_wrapper.log file for any fatal errors during execution of the flow. In this case, ESA would have sent a "Timeout" response back to LSA.

5. GetServerinfo is failing

   The Get Server Info flow can fail for following reasons:

   - Missing *serverInfo.xml* file from the configured location:

C:\Program Files\HP\Insight Orchestration\esa_extensions\server

- Unsupported server model
- Invalid UUID
- Network issues while connecting to remote server management interface
- Wrong IP address configuration in the *uuidHostMapper.xml* file
- Invalid user credentials while connecting to the remote server management interface
- Fatal errors from OO Central and/or OO RSJRAS services

To rectify the problem follow the steps below:

- Ensure *serverInfo.xml* file is available and readable from the configured location: C:\Program Files\HP\Insight Orchestration\esa_extensions\server
- Ensure a supported Server Model configured in the *serverInfo.xml* file and validate the server model string contains the anticipated server type or name expected in the workflow
- Ensure a valid UUID is received as input parameter to the flow
- Ensure the UUID is configured in the *uuidHostMapper.xml* file.
- Ensure the IP address is configured for the remote server management interface is valid for the given UUID
- Ensure network connectivity and accessibility to the remote server, management interface
- Ensure valid user credentials are configured for the remote server management interface in the OO System Accounts
- Check the OO Central reports or central_wrapper.log file for any fatal errors during execution of the flow. In case of Fatal errors on OO logs and/or Central reports, ESA would have sent "Timeout" response when invoked by Insight Dynamics.

6. Power ON is failing

The Server Power ON workflow can fail for following reasons:

- Missing server data files – *inventoryList.xml*, *serverInfo.xml* and *uuidHostMapper.xml* – from the configured location C:\Program Files\HP\Insight Orchestration\esa_extensions\server
- Invalid or missing data from the server data files
- Unsupprted Server model
- Invalid UUID
- Network issues while connecting to the remote server management interface
- Invalid user credentials for the remote server management interface configured in the OO System Account
- Unsupported/invalid command execution on the remote server, management interface
- Fatal errors from OO Central and/or OO RSJRAS services

To rectify the problem, follow the steps below:

- Ensure that the server data files are available and readable/writeable from the configured location C:\Program Files\HP\Insight Orchestration\esa_extensions\server
- Ensure that the server data files contain all valid data
    - Ensure a valid Server Model is configured in *serverInfo.xml* file
    - Ensure a valid IP address is configured for Server Management Station for the given UUID
- Ensure that a valid UUID is provided as input to the workflow
- Ensure that network connectivity and accessibility to connect to the remote server management interface
- Ensure that a valid user credentials are configured in the OO System Account
- Ensure that the command being invoked on the remote server management interface is valid

- Check the OO Central reports or central_wrapper.log file for any fatal errors during execution of the flow. In case of Fatal errors on OO logs and/or Central reports, ESA would have sent "Timeout" response when invoked by Insight Dynamics.

7. Power OFF is failing

   The Server Power OFF workflow can fail for following reasons:

   - Missing server data files – *inventoryList.xml*, *serverInfo.xml* and *uuidHostMapper.xml* – from the configured location C:\Program Files\HP\Insight Orchestration\esa_extensions\server
   - Invalid or missing data from the server data files
   - Unsupprted Server model
   - Invalid UUID
   - Network issues while connecting to remote server management interface
   - Invalid user credentials configured in the OO System Account
   - Unsupported/invalid command execution on the remote server management interface
   - Fatal errors from OO Central and/or OO RSJRAS services

   To rectify the problem, follow the steps below:

   - Ensure that the server data files are available and readable/writeable from the configured location C:\Program Files\HP\Insight Orchestration\esa_extensions\server
   - Ensure that the server data files contain all valid data
     - Ensure a valid Server Model is configured in *serverInfo.xml file*
     - Ensure a valid IP address is configured in the *uuidHostMapper.xml* file for the remote server management interface for the given UUID
   - Ensure that a valid UUID is provided as input to the workflow
   - Ensure that network connectivity and accessibility to connect to the remote server management interface
   - Ensure that a valid user credentials are configured in the OO System Account
   - Ensure that the command being invoked on the remote server management interface is valid
   - Check the OO Central reports or central_wrapper.log file for any fatal errors during execution of the flow. In case of Fatal errors on OO logs and/or Central reports, ESA would have sent "Timeout" response to LSA.

8. XP CLI calls to populate the storage catalog are failing

   Possible reasons for this failure include:

   - Invalid login credentials
   - Invalid inputs to the CLI

   To rectify the problem, follow the steps below

   - Make sure that the login credentials are configured correctly for the CLI.
   - The inputs to the CLI depend on the commands used. Ensure that the input to the CLI is as mentioned in the CLI documentation.

9. XP CLI calls fail for active storage operations.

   Possible reasons for this failure include:

   - Insufficient permissions
   - The reference implementation does not handle the wiring of the LDEV to the controller port. If this mapping is not configured by the storage administrators the active calls might fail.

   To rectify the problem, follow the steps below

   - The login credentials configured for the CLI are insufficient. Use the XP-AE to configure the right permissions.
   - Ensure the storage administrators have configured the mapping of LDEV to the controller.

Glossary

Central management server (CMS)	A system in the management domain that executes the HP Systems Insight Manager software. All central operations within HP Systems Insight Manager are initiated from this system.
ESA	Extensible Server and Storage Adapter (a software layer which enables Insight Dynamics support to expand beyond blades using Virtual Connect and SPM support of EVA storage solutions).
IAction	OO's internal way of creating customized java classes
iLO	Integrated Lights Out management (a means of remote server management)
OO	HP Operations Orchestration
SPM	HP's Storage Provisioning Manager (a software solution enabling the storage administrator to create a catalog of storage used by the server administrator, with the controllable ability to change hostmode and adjust LUN masking).
VC	Virtual Connect

For more information

The following links provide additional information HP Operations Orchestration:

- http://www.hp.com/go/oo
- http://support.openview.hp.com/selfsolve/manuals (select Operations Orchestration)
- Operations Orchestration Software Development Kit (SDK) available online at http://support.openview.hp.com/selfsolve/document/KM759197/binary/OO7.51_SDKGuide.pdf?searchIdentifier=1970798b:12697172e06:-5235&resultType=document

For more information on the flow execution in OO Studio, refer to "Authors Guide" document that comes along with OO installation (located at C:\Program Files\HP\Operations Orchestration\Studio\docs).

Insight Dynamics with infrastructure orchestration

- http://www.hp.com/go/insightdynamics
- http://www.hp.com/go/insightdynamics/docs
- Insight Orchestration User Guide http://docs.hp.com/en/490653-001/490653-001.pdf

Information regarding HP XP Command View Advanced Edition is available at http://h20000.www2.hp.com/bizsupport/TechSupport/DocumentIndex.jsp?contentType=SupportManual&lang=en&cc=us&docIndexId=179111&taskId=101&prodTypeId=12169&prodSeriesId=1099451

Appendix: Configuring ESA for 6.0 Update 1

Overview

The 6.0 Update 1 release of Insight Dynamics requires additional configuration steps to successfully use the ESA integration of servers and the HP Storage Provisioning Manager with infrastructure orchestration. **These steps are NOT required for the 6.0, 6.1, or 6.2 releases.**

The ESA core jars delivered in 6.0 update 1 will not work with the ESA-OO content provided for integration of ESA with HP SW Operation Orchestration (referred to as OO) delivered in 6.0.0; therefore, the customer will need to upgrade content of ESA-OO with the 6.0 update 1 deliverables.

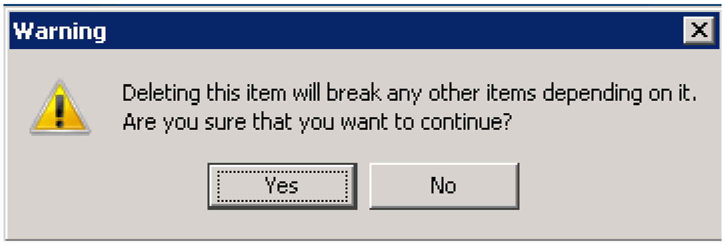
Generally, the customer will have to:

1. Remove old ESA-OO repository content
2. Extract the new 6.0 update 1 version of the ESA-OO repository content
3. Import the new content into OO

Remove 6.0.0. ESA-OO repository content

The first step in upgrading the ESA-OO content to 6.0 update 1 is to remove the ESA-OO content from the 6.0.0 version of the OO repository. Use the following procedure to remove the 6.0.0 ESA-OO repository content.

1. Start up OO Studio Designer
2. Log-in as Admin
3. In the library, expand the "Hewlett-Packard" Tree
4. Select the ESA folder
5. Right click on the ESA folder and select "Delete"
6. A warning dialog appears:



Deleting this item will break any other items depending on it.
Are you sure that you want to continue?

7. Select Yes
8. After deleting ESA from the "Hewlett-Packard" Tree, perform a Check In Tree by right clicking on the Library->Hewlett-Packard tree and selecting Repository->Check In Tree
    a. This finalizes the delete of the ESA folder.

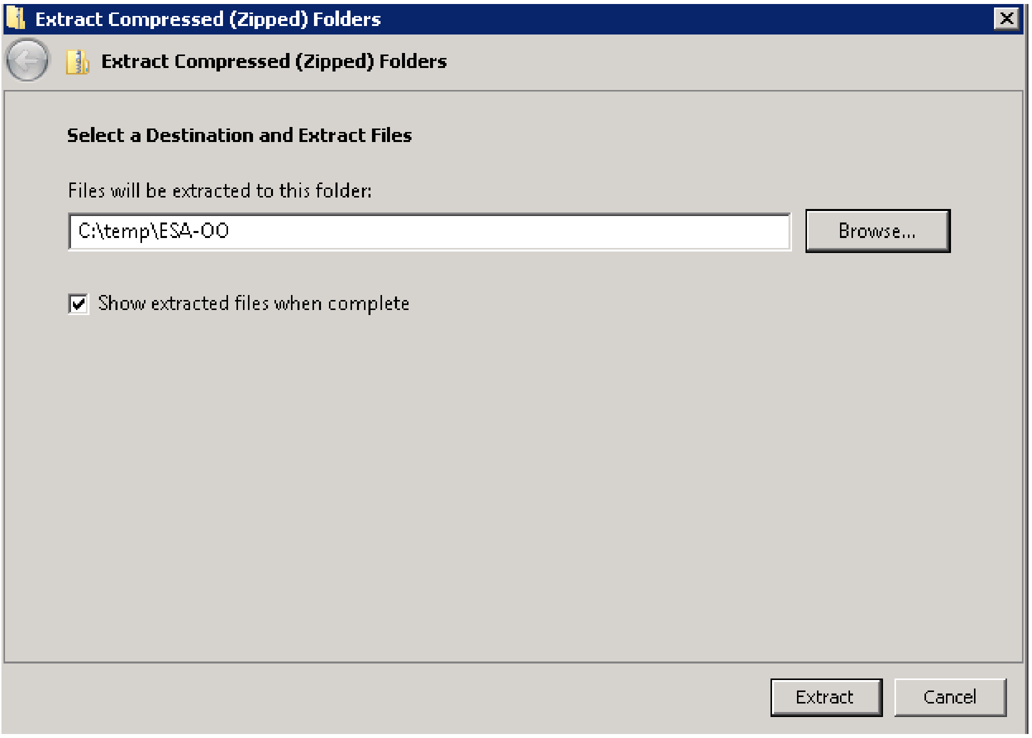
The ESA-OO content is now removed from the OO repository.

Extract the 6.0 update 1 ESA-OO repository content

As part of installing the HP Insight Dynamics 6.0 Update 1 patch, the new ESA-OO reference implementation is delivered into the "..\ESA\REF IMPL" folder of "Virtual Server Environment". To extract the contents of the reference implementation, do the following:

1. Locate the file ESA-OO-RefImpl.zip in the "Virtual Server Environment" directory:
    a. <Program Files>\HP\Virtual ServerEnvironment\esa\ref impl
2. Right click on the .zip file and select "Extract All"

3. Select a Destination folder for the extracted files (C:\temp\ESA-OO)



Extract Compressed (Zipped) Folders
Extract Compressed (Zipped) Folders
Browse...
Extract
Cancel

Select a Destination and Extract Files

Files will be extracted to this folder:

C:\temp\ESA-OO

☑ Show extracted files when complete

4. Extract the files by selecting Extract

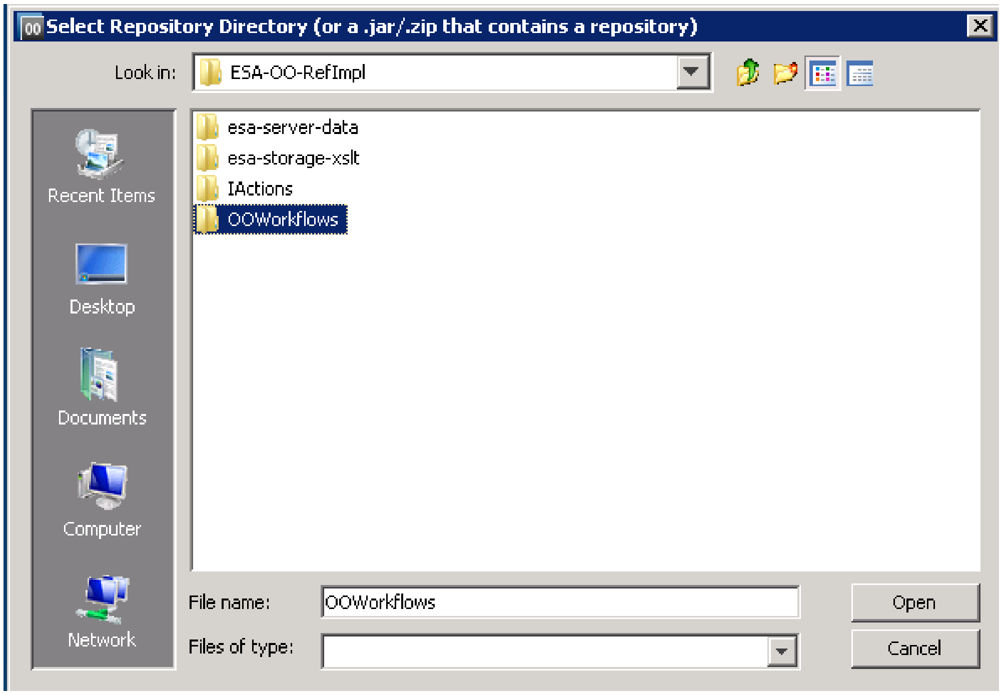
The required files for the upgrade are now extracted and ready for the remaining configuration tasks.

Import the new content into OO

Once the extraction is complete, the new ESA-OO content needs to be imported into the OO repository. This is done as follows:

1. Stop RSJRAS service
2. Delete all the esa 1.0 jars from RAS\java\default\repository path and RAS\java\default\repository\lib path
3. Go to the directory where ESA-OO-RefImpl was extracted (C:\temp\ESA-OO\ESA-OO-RefImpl) and run the batch script “setupESA_IActionJars.bat” to copy IAction jars and ESA libs into RAS path
4. Copy the .xslt files in the esa-storage-xslt directory into the <Program Files>HP\Insight Orchestration\esa_extensions\storage directory
5. Restart RSJRAS service, if it failed to do so through the batch script.
6. Import the workflows into OO Studio from the ESA-OO-RefImpl kit.
    a. Start OO Studio designer
    b. Login as Admin
    c. Select Repository->Import Repository

d. Import the OOWorkflows directory (C:\temp\ESA-OO\ESA-OO-RefImpl\OOWorkflows)



Select Repository Directory (or a .jar/.zip that contains a repository)

Look in: ESA-OO-RefImpl

- esa-server-data
- esa-storage-xslt
- IActions
- OOWorkflows

File name: OOWorkflows

Files of type:

e. Once imported, check in the “Hewlett-Packard” tree by right clicking on Library->Hewlett-Packard and selecting Repository->Check In Tree

   i. This finalizes the import into the repository

Share with colleagues

Created September 2010