

Accelerator - BMC Atrium Orchestrator Adapter

User Guide

| Version | Description of change | Updated by | Date |
|-----------|-------------------------|------------|-----------------|
| Version 1 | 1 st Release | T Scott | 17 October 2011 |
| | | | |
| | | | |
| | | | |

Contents

| | | |
|------------|---|-----------|
| 1.0 | Introduction | 3 |
| 2.0 | Installation | 3 |
| 3.0 | Application Configuration | 4 |
| 4.0 | Front Office Configuration | 5 |
| 4.1 | Configure External List Boxes | 5 |
| 4.2 | Configure Request Update Hooks | 5 |
| 4.3 | Configure Requests | 5 |
| 5.0 | Mapping Configuration | 6 |
| 5.1 | Requests | 6 |
| 5.2 | External List Boxes | 8 |
| 5.3 | Request Update Hooks..... | 11 |
| 6.0 | Further Information | 13 |

1.0 Introduction

The BMC Atrium Orchestrator Adapter is a suite of Biomni Front Office (BFO) integration tools that allows Front Office to execute synchronous workflow processes within BMC Atrium Orchestrator (AO).

It includes:

- Request fulfilment hook - to enable request information to be passed to an AO workflow.
- External list box adapter - retrieves information from AO (via an AO workflow) then populates a BFO external list box.
- Request Update hook to pull information from AO (via an AO workflow) into a BFO request form.

It supports AO platform versions 7.5.x / 7.6.x

2.0 Installation

The application is a Microsoft .NET 4 Web application, requiring IIS 5.1+ and .Net Framework 4 to be installed on the target server.

Note: If installing on IIS7 ensure that Role Services for 'IIS 6 Management Compatibility' is checked in Server Manager.

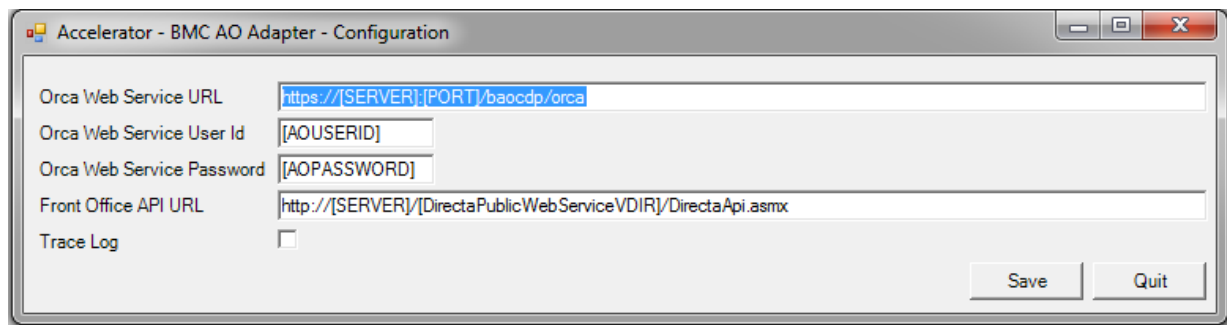
To install:

- log in on the target installation server with administrator privileges
- Extract the files from the zip file into a temporary folder
- Using Windows explorer open the temporary folder, then run setup.exe (on Windows Server 2008 right click then "Run as Administrator")
- Click the next button
- Set the target web site
- Set the virtual directory (default AcceleratorBMCAOAdapter)
- Set the target application pool
- Click the next button, then again to start the installation
- Grant the NETWORK SERVICE account (or the account which the selected application pool is running under) "Full Control" rights to
 - \inetpub\wwwroot\<selected virtual directory>\logs

The server that the application is installed on requires visibility of the Front Office and Atrium Orchestrator servers.

3.0 Application Configuration

During installation a shortcut is created on the desktop “Accelerator - BMC AO Adapter - Configuration”, click on the shortcut and a window like this should appear:



For “Orca Web Service URL”, replace [SERVER] with the server name/ip of the AO server. Replace [PORT] with the port number configured within AO. **Note the AO web service must be configured to use HTTPS as per page 12 of the AO Web Services User Guide.**

For “Orca Web Service User Id”, [AUSERID] with the user Id required to authenticate with AO.

For “Orca Web Service Password”, [AOPASSWORD] with the password required to authenticate with AO.

For “Front Office API URL”, replace [SERVER] with the ip/server name of the Front Office server, [DirectaPublicWebServiceVDIR] with the Front Office public web server virtual folder.

If “Trace Log” is checked the application will keep a trace of all input / output (good for support issues). **This should not be enabled for live operation.**

Once all values are set click on the “Save” button.

4.0 Front Office Configuration

4.1 Configure External List Boxes

External list boxes can be configured to pull data from an AO workflow into a selectable drop down list. To do this, for each external list box required create an external system definition of type External List Box (Version 6), use the following URL:

[http://<Server>/<SelectedVirtualDirectory>/AcceleratorBMCAOAdapterELB.asmx?ID=\[CODE\]](http://<Server>/<SelectedVirtualDirectory>/AcceleratorBMCAOAdapterELB.asmx?ID=[CODE])

Replace [CODE] with a unique code for each external list box being configured e.g. GetUsers, GetServers.

Then for each external system definition configured create a request field (type External List Box) for use in the desired request types.

4.2 Configure Request Update Hooks

Request update hooks can be configured to pull data from an AO workflow into request fields. To do this, for each external list box required create an external system definition of type Request Update Hook (Version 6), use the following URL:

[http://<Server>/<SelectedVirtualDirectory>/AcceleratorBMCAOAdapterRUH.asmx?ID=\[CODE\]](http://<Server>/<SelectedVirtualDirectory>/AcceleratorBMCAOAdapterRUH.asmx?ID=[CODE])

Replace [CODE] with a unique code for each request update hook being configured e.g. GetUserInfo, GetServerInfo.

Then for each external system definition configured, link to the desired request types update button.

4.3 Configure Requests

Configure Front Office with one request type per target AO workflow. Each request type must capture the necessary AO workflow input parameters.

Create an external system definition (of type Fulfilment Hook), set URL to:

<http://<Server>/<SelectedVirtualDirectory>/AcceleratorBMCAOAdapter.asmx>

Then for each request type configured set fulfilment to the external system configured above.

5.0 Mapping Configuration

5.1 Requests

There is a mapping XML file that is used by the adapter to map Front Office request fields into AO Workflows.

Edit the Config.xml file (using an xml editor), found here:

inetpub\<SelectedWebFolder>\<SelectedVirtualDirectory>\xml\Config.xml

For each request type configured repeat the `<OrcaConfig>` xml block, and configure the necessary settings, e.g.:

```
<?xml version="1.0" encoding="UTF-8"?>
<bXML xmlns="http://www.biomni.com/Schemas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <OrcaConfig>
    <RequestType>REBOOTVM</RequestType>
    <GridName>GRID1</GridName>
    <ProcessName>>:ServerProcesses:Reboot_Server</ProcessName>
    <InputParameters>
      <ParameterCount>3</ParameterCount>
      <Parameter>
        <Name>ServerIP</Name>
        <Type>xs:string</Type>
        <Value>[CVM001]</Value>
      </Parameter>
      <Parameter>
        <Name>HardReboot</Name>
        <Type>xs:string</Type>
        <Value>[CVM002]</Value>
      </Parameter>
      <Parameter>
        <Name>requestuserid</Name>
```

```

        <Type>xs:string</Type>
        <Value>[n:bXML/n:RequestTask/n:Request/n:RequestHeader/n:RequestUsers/n:RequestedBy/n:UserID]</Value>
    </Parameter>
</InputParameters>
<OutputParameters>
    <Parameter>
        <RequestFieldCode>CVM3</RequestFieldCode>
        <Name>Confirmation</Name>
    </Parameter>
</OutputParameters>
</OrcaConfig>
<OrcaConfig>
    <RequestType>SHUTDVM</RequestType>
    <GridName>GRID1</GridName>
    <ProcessName>ShutDown_Server</ProcessName>
    <InputParameters>
        <ParameterCount>3</ParameterCount>
        <Parameter>
            <Name>ServerIP</Name>
            <Type>xs:string</Type>
            <Value>[CVM001]</Value>
        </Parameter>
    </InputParameters>
    <OutputParameters>
        <Parameter>
            <RequestFieldCode>CVM3</RequestFieldCode>
            <Name>Confirmation</Name>
        </Parameter>
    </OutputParameters>
</OrcaConfig>
</bXML>

```

In the example above there are two Front Office request types configured. “REBOOTVM”, consists of 3 fields, CVM001 is used for the server IP address, CVM002 is a “Yes/No” drop down, the third parameter uses an XML Xpath statement to extract the Requested By UserID from the request XML. Request type “SHUTDVM” just has a server IP address (CVM001).

Both include an output parameter, request field CVM3 will be updated with the AO output parameter called Confirmation.

XML elements that need to be configured for each `<OrcaConfig>` element are:

| Element | Description |
|---------------------------------------|--|
| <code><RequestType></code> | Front Office Request Type Code |
| <code><GridName></code> | Name of the target AO Grid |
| <code><ProcessName></code> | Name of the target AO workflow |
| <code><InputParameters></code> | Collection of input parameters |
| <code><ParameterCount></code> | Number of parameters passed to AO |
| <code><Parameter></code> | Repeat for each parameter required in the target AO workflow |
| <code><Name></code> | Name of the AO parameter |
| <code><Type></code> | Xml Type of the AO parameter |
| <code><Value></code> | Value of the AO parameter (1) |
| <code><OutputParameters></code> | Collection of output parameters |
| <code><Parameter></code> | Repeat for each parameter returned by the AO workflow |
| <code><RequestFieldCode></code> | Front Office Request Field Code |
| <code><Name></code> | Name of AO output parameter |

(1) To pass the value of a Front Office request field to AO, put the name of the field in square brackets. If you want to pass a constant value omit the brackets. You can also use an XML xpath statement to extract data from anywhere in the request xml (see the xsd schema OMC_RequestTaskOUT_3_0.xsd included in the installed schemas folder)

5.2 External List Boxes

Edit the ELBConfig.xml file (using an xml editor), found here:

inetpub\<SelectedWebFolder>\<SelectedVirtualDirectory>\xml\ELBConfig.xml

For each external list box configured repeat the `<ExternalListBoxConfig>` xml block, and configure the necessary settings, e.g.:

```
<?xml version="1.0" encoding="UTF-8"?>
<bXML xmlns="http://www.biomni.com/Schemas" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ExternalListBoxConfig>
    <Code>GetSnapshots</Code>
    <GridName>GRID1</GridName>
```



```

<ProcessName>:NBS_vCloud:vCloudAPI:Wrapper:vCloud_API_Wrapper</ProcessName>
<ResultsNodeName>snapshotrecord</ResultsNodeName>
<KeyColumnName>Name</KeyColumnName>
<Columns>
  <Column>
    <Name>Name</Name>
    <Heading>Snapshot Name</Heading>
    <Width>300</Width>
    <Type>String</Type>
    <ColumnDataSource>[Snapshot_Name] (ID:[Snapshot_ID])</ColumnDataSource>
  </Column>
  <Column>
    <Name>Description</Name>
    <Heading>Snapshot Description</Heading>
    <Width>300</Width>
    <Type>String</Type>
    <ColumnDataSource>[Snapshot_Desc]</ColumnDataSource>
  </Column>
</Columns>
<InputParameters>
  <ParameterCount>2</ParameterCount>
  <Parameter>
    <Name>action</Name>
    <Type>xs:string</Type>
    <Value>listsnapshots</Value>
  </Parameter>
  <Parameter>
    <Name>vmurl</Name>
    <Type>xs:string</Type>
    <Value>[SELSVRS]</Value>
  </Parameter>
</InputParameters>
</ExternalListBoxConfig>
</bXML>

```

| Element | Description |
|--------------------|---|
| <Code> | A unique code for the external list box (see 4.1) |
| <GridName> | Name of the target AO Grid |
| <ProcessName> | Name of the target AO workflow |
| <ResultsNodeName> | Name of the target XML record node returned from AO (1) |
| <KeyColumnName> | The name of the column that will be stored on the BFO request |
| <Columns> | Container for multiple column elements |
| <Column> | Container for column elements |
| <Name> | The name of the column |
| <Heading> | The heading of the column (displayed in BFO) |
| <Width> | The width of the column (in pixels) |
| <Type> | The type of the column (String or Image) |
| <ColumnDataSource> | Source of the data used for the column (2) |
| <InputParameters> | Collection of input parameters |
| <ParameterCount> | Number of parameters passed to AO |
| <Parameter> | Repeat for each parameter required in the target AO workflow |
| <Name> | Name of the AO parameter |
| <Type> | Xml Type of the AO parameter |
| <Value> | Value of the AO parameter (3) |

(1) It is important that the AO workflow is configured to return list box results in an XML format. The <ResultsNodeName> should be set to the name of the XML element that contains each result record, each record should contain a number of XML attributes that contain the relevant information e.g.

```
<snapshotrecords>
  <snapshotrecord Snapshot_Desc="Test Snapshot for W2K3TEST1 - 01" Snapshot_Name="W2K3TEST1-01/>
  <snapshotrecord Snapshot_Desc="Test Snapshot for W2K3TEST1 - 02" " Snapshot_Name="W2K3TEST1-02" />
</snapshotrecords>
```

In the example above the <ResultsNodeName> would be set to **snapshotrecord**.

(2) Specify the name of an XML attribute (enclosed in square brackets) returned in the AO XML results. More than one attribute can be configured per column data source if required, e.g:

```
<ColumnDataSource>[Snapshot_Name] (ID:[Snapshot_ID])</ColumnDataSource>
```

(3) To pass the value of a Front Office request field to AO, put the name of the field in square brackets. If you want to pass a constant value omit the brackets. You can also use an XML xpath statement to extract data from anywhere in the request xml (see the xsd schema ExternalListboxOutbound_600.xsd included in the installed schemas folder)

5.3 Request Update Hooks

Edit the RUHConfig.xml file (using an xml editor), found here:

inetpub\<SelectedWebFolder>\<SelectedVirtualDirectory>\xml\RUHConfig.xml

For each request update hook required repeat the `<RequestUpdateHookConfig>` xml block, and configure the necessary settings, e.g.:

```
<RequestUpdateHookConfig>
  <Code>GetSnapshotServerInfo</Code>
  <GridName>GRID1</GridName>
  <ProcessName>NBS_vCloud:vCloudAPI:Wrapper:vCloud_API_Wrapper</ProcessName>
  <ResultsNodeName>vminfo</ResultsNodeName>
  <InputParameters>
    <ParameterCount>7</ParameterCount>
    <Parameter>
      <Name>apiuser</Name>
      <Type>xs:string</Type>
      <Value>nationwide</Value>
    </Parameter>
    <Parameter>
      <Name>apiorg</Name>
      <Type>xs:string</Type>
      <Value>nationwide</Value>
    </Parameter>
    <Parameter>
      <Name>apipassword</Name>
      <Type>xs:string</Type>
      <Value>nationwide</Value>
    </Parameter>
    <Parameter>
      <Name>action</Name>
      <Type>xs:string</Type>
      <Value>getvminfo</Value>
    </Parameter>
    <Parameter>
      <Name>vmurl</Name>
      <Type>xs:string</Type>
    </Parameter>
  </InputParameters>
</RequestUpdateHookConfig>
```

```

        <Value>[SELSVRS]</Value>
    </Parameter>
</InputParameters>
<OutputParameters>
    <Parameter>
        <RequestFieldCode>HOSTNAME</RequestFieldCode>
        <Name>@vmname</Name>
    </Parameter>
    <Parameter>
        <RequestFieldCode>VIRTID</RequestFieldCode>
        <Name>@vmid</Name>
    </Parameter>
    <Parameter>
        <RequestFieldCode>IPADD</RequestFieldCode>
        <Name>@ipaddress</Name>
    </Parameter>
    <Parameter>
        <RequestFieldCode>OPSYS</RequestFieldCode>
        <Name>@osfamily</Name>
    </Parameter>
</OutputParameters>
</RequestUpdateHookConfig>

```

| Element | Description |
|--------------------|--|
| <Code> | A unique code for the external list box (see 4.2) |
| <GridName> | Name of the target AO Grid |
| <ProcessName> | Name of the target AO workflow |
| <ResultsNodeName> | Name of the target XML record node returned from AO (1) |
| <InputParameters> | Collection of input parameters |
| <ParameterCount> | Number of parameters passed to AO |
| <Parameter> | Repeat for each parameter required in the target AO workflow |
| <Name> | Name of the AO parameter |
| <Type> | Xml Type of the AO parameter |
| <Value> | Value of the AO parameter (2) |
| <OutputParameters> | Collection of output parameters |
| <Parameter> | Repeat for each parameter returned by the AO workflow |
| <RequestFieldCode> | Front Office Request Field Code |
| <Name> | Name of AO output parameter (3) |

(1) It is important that the AO workflow is configured to request update results in an XML format. The **<ResultsNodeName>** should be set to the name of the XML element that contains each result record, each record should contain a number of XML attributes / elements that contain the relevant information e.g.

```
<vmrecords>
  <vmrecord vmname="Windows 2008 R2 Server" vmid="W2K8TEST1-01"/>
  <vmrecord vmname="Windows 2003 Server" vmid="W2K3TEST1-01"/>
</vmrecords>
```

In the example above the **<ResultsNodeName>** would be set to **vmrecord**.

(2) To pass the value of a Front Office request field to AO, put the name of the field in square brackets. If you want to pass a constant value omit the brackets.

(3) Specify the name of an XML attribute (preceded by the @ character) or XML Element name returned in the AO XML results.

6.0 Further Information

This adapter utilises the `executeProcess` method of the AO Web Service. For further information see the BMC Atrium Orchestrator Web Services User Guide.