

An Oracle White Paper
February 2017

Combining Siebel IP 2016 and native OPA 12.x Interviews

Purpose

This whitepaper is a guide for Siebel customers that wish to take advantage of OPA 12.x functionality that is not supported by the Siebel Open UI OPA Interview Applet provided with Siebel IP2016. It provides detailed instructions for how to extend the Siebel IP2016 OPA integration to support native OPA interviews, with support for all the latest control styles, dynamic screen rendering, document attachments and checkpoints. Included as an appendix is an example of how to integrate the OPA Answer Service.

Executive Overview

This whitepaper describes a low risk approach for Oracle customers to adopt the latest versions of Siebel and Oracle Policy Automation, while also taking advantage of all the latest OPA interview features.

To use Oracle Policy Automation versions up to 10.4.x with Siebel CRM, Oracle provides a product called Oracle Policy Automation Connector for Siebel. This connector supports the native OPA advice experiences called *interviews* that load and save data from Siebel, and making decisions in Siebel workflows with calls to OPA Determinations API Assess web service.

For Oracle Policy Automation 12 and later, the integration between Siebel and OPA moved into the core Siebel product, and no separate OPA connector is needed. Siebel IP2014 released the first integration to OPA version 12, and with the release of Siebel IP2016 both interviews and web service integration with OPA v12 are supported. An OpenUI OPA Interview Applet is provided, which renders OPA interviews as Siebel OpenUI controls. Custom configuration of an OPA interview is required before it can be used with this applet, and only some OPA interview features are supported.

This whitepaper explains in detail how to extend the functionality of the Siebel IP2016 OPA integration to support *native* OPA interviews in a similar fashion to OPA Connector for Siebel 10.4, but with full compatibility with OPA 12.x. In particular, it explains how to add support to Siebel for the standard OPA 12 web service connector operations Load, Submit, GetCheckpoint and SetCheckpoint.

The referenced .zip file includes workflows, integration objects and other Siebel repository objects that extend the functionality of the Siebel IP2016 OPA integration. These changes can be maintained during subsequent Siebel upgrades.

[Contents](#)

Purpose..... 1

Executive Overview 1

Introduction 2

How to add an OPA v12.2.x Interview into Siebel IP 2016..... 4

Installation of required Assets 4

Adding Siebel Inbound Web Service 4

OPA Hub Connection for Siebel Inbound Web Service 7

Testing the Siebel Connection in OPM 8

Adding the Load Operation 9

Simulate the workflow from Siebel Tools 9

Deploying the Load Method..... 10

Deploying the Save Method..... 11

Launching the Interview 11

Checkpoints..... 18

Appendix – How to use the OPA Answer Service..... 22

 Deploy the OPA Policy Model..... 22

 Deploy the Siebel Workflow..... 24

Conclusion 26

Introduction

When is the Siebel Open UI OPA Interview Applet not enough?

The Open UI OPA Applet currently supports a subset of OPA interview features. Unlike native OPA interviews, it cannot use OPA rules to dynamically change the appearance of a screen as data is entered on that screen, nor support control types such as sliders, image buttons and signatures. It also does not currently support checkpoints that allow saving and resuming OPA interviews, nor file attachments for uploaded documents or generated forms. In general, the Open UI OPA Applet only adds support for certain missing OPA interview features in each Siebel innovation pack release.

By contrast, implementing native OPA interviews allows for immediate and complete adoption of all new OPA front-end functionality as it is released, without any custom screen configuration to support the Siebel Open UI OPA Applet.

The tables below compare native OPA interviews with the Siebel IP2016 Open UI OPA applet.

	Siebel Open UI OPA Applet	Native OPA Interviews
100% Open UI Compatibility	Yes	No
Siebel industry examples	Yes	No
No Siebel customization	Yes	No
Fully supported by Oracle Support	Yes	No*
All OPA controls supported	No	Yes
Dynamic screen behavior	No	Yes
Attachments, signatures and custom controls	No	Yes
Checkpoints	No	Yes
Standard OPA screen authoring	No	Yes
Simple upgrade from OPA 10.4	No	Yes

*All standard Siebel and OPA functionality is fully supported. The recommended customizations outlined in this whitepaper must be supported by in-house Siebel developers, Oracle Consulting Services or Siebel-certified partners.

The purpose of this whitepaper is to eliminate most of the risk of adopting native OPA interviews with Siebel IP2016 and later, by providing the community with clear steps that can be followed by any qualified Siebel personnel. The recommended path outlined is supported with current and later versions of Siebel and Oracle Policy Automation.

How to add an OPA v12.2.x Interview into Siebel IP 2016

This whitepaper will set out the steps required to add, test and verify native OPA interviews in a Siebel IP 2016 environment. It will show how re-use of much of the existing specific Siebel OPA integration framework can be utilized.

You will need a Siebel IP 2016 environment (client and server components) with the EAI Object Manager running, Siebel Tools and an OPA Hub; for deploy of your policy models. Other useful tools to assist in testing your deployment would be SoapUI, TCPMon and an Xml Editor.

Installation of required Assets

1. Unzip the attached OPA Siebel Generic WS Connector.zip
2. Add Archive OPAGenericWSConnector.sif in Siebel Tools
3. Full Compile and release SRF to Server and Client
4. Publish and Activate the OPA Get Checkpoint, OPA Load For PUB Sample Intake Contact, OPA Process Submit Data, OPA Set Checkpoint and OPA Submit For PUB Sample Intake Contact workflows
5. Deploy all the xsl files in the subfolder XSLT to the XSLT folders for Server (and client for debug) also to the C:\\temp\\ folder.
6. Import the OPA Generic Connection.XML Inbound Web Service definition and clear cache.
7. Deploy any required policy models.
8. Import the OPA Generic Connection.XML Inbound Web Service definition and Clear Cache + Generate WSDL to check valid

Note: Some file logs are output to C:\\temp\\. This may cause issue when deployed to UNIX servers.

Note: Metadata IO was changed to add the supports-checkpoints attribute and the XSL sets the value to True

Note: The Web Service is not yet generic as no way to pass in the IO Name to the OPA Hub's Connection so you will need to adjust the workflow processes if using another IO definition.

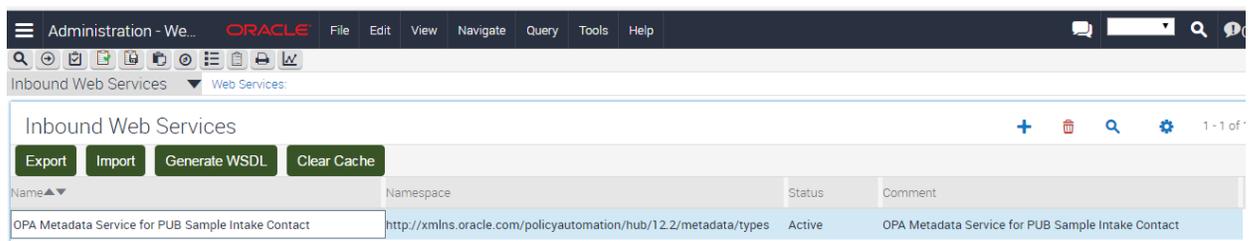
Adding Siebel Inbound Web Service

To get started we need to set up a Generic Web Service Connection between Siebel and OPA so we can get the metadata we need to map the entities and attributes for our rules, based on

the Integration Object in Siebel we plan to use. For this exercise we will use an existing Integration Object that is defined OOTB 'PUB Sample Intake Contact'.

In Siebel we need to look at the inbound web services and we will need 2 methods initially to allow us to set up the connection and get at the Siebel metadata. These are 'CheckAlive' and 'GetMetadata'.

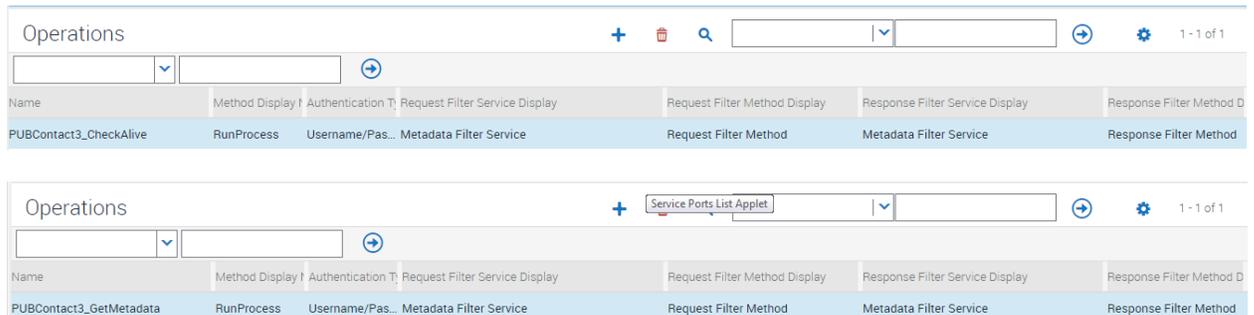
Navigate to Administration - Web Services – Inbound Web and search for '*OPA*' and locate the Web Service named OPA Generic Connection. The complete Web Service definition is available in the package. Note, you will need to rename the original web service if you import from the package rather than use the existing.



Notice it has two service ports defined 'CheckAlivePort' and 'GetMetaData' both of which invoke a Workflow Process



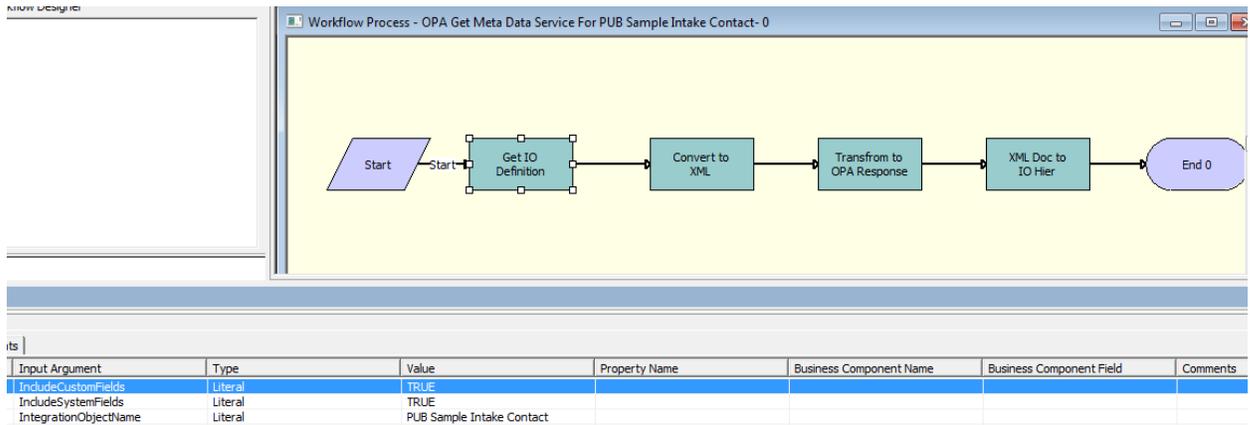
Each port also has a method defined.



The Request Filter Method and Response Filter Method are Siebel Business Service calls that are used to clean up the Request and Response XML.

The CheckAlive Operation and workflow is generic and should not need to be amended for re-use with other Integration Objects.

The GetMetaData Operation workflow 'OPA Get Meta Data Service For PUB Sample Intake Contact' is specific for this Integration Object (IO) so if you are using a different IO you will need to make a copy of this workflow and amend the step 'GetIODefinition' to reference the correct IO.



If using a different IO you would also need to copy the Inbound Web Service and reference your new workflow processes in the service ports and operations.

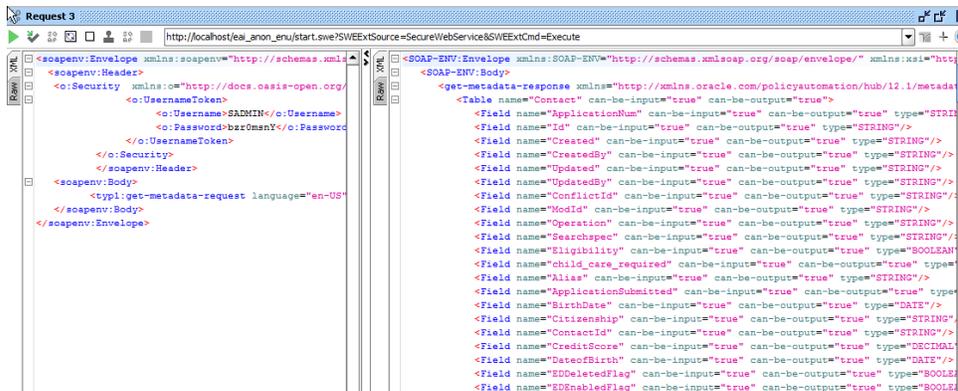
Once completed, generate the WSDL from the Inbound Web Service definition in Siebel after clearing the cache and test the responses using SoapUI.

CheckAlive

```

Request 1
http://localhost/ea1_anon_enu/start.swe?SWExtSource=SecureWebService&SWExtCmd=Execute
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <o:Security xmlns:o="http://docs.oasis-open.org/ws-sx/ws-security/2004-08" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
      <o:UsernameToken>
        <o:Username>ADMIN</o:Username>
        <o:Password>b3r0m3nY</o:Password>
      </o:UsernameToken>
    </o:Security>
  </soapenv:Header>
  <soapenv:Body>
    <type:check-alive-request/>
  </soapenv:Body>
</soapenv:Envelope>
  </SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <check-alive-response xmlns="http://xmlns.oracle.com/policyautomation/hub/12.1/metadata">
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
  
```

GetMetaData – showing the metadata items from your Integration Object



A SoapUI project is included in the package.

OPA Hub Connection for Siebel Inbound Web Service

Log onto the OPA Hub and go to 'Connections'. Add a new connection similar to below

Connection : Siebel Actions Save and Close Cancel

Name:

Type:

Collection Access:

Status:

Web Service Details

URL:

Version:

SOAP Action Pattern: (optional)

OAUTH for Data Operations

Provide OAUTH bearer token in HTTP header on Load and Save actions

URL Parameter:

e.g. http://localhost:7003/november/web-determinations/startsession/<deployment>?<URL Parameter>=<OAUTH token>

WS-Security

Provide WS-Security Username token in SOAP actions

Applies to:

Username:

New Password:

Include timestamp with a 5 minute expiration

Give the connection a name, type of 'Web Service'. Choose a collection, if required.

Input the url connection for the specific EAI Object Manager and server instance.

Version: 12.2.5

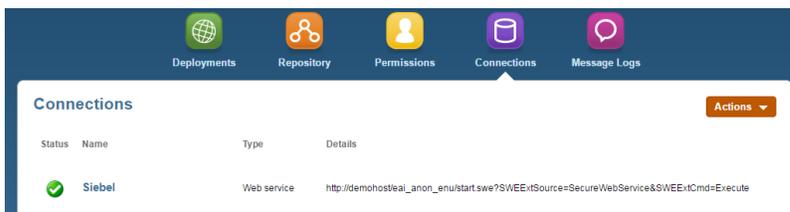
SOAP Action Pattern:

'document/http://xmlns.oracle.com/policyautomation/hub/12.2.5/metadata/types:Conn_{0}'

Note: The highlighted text should match the prefix in the Operation Name of your Siebel Inbound Web Service port. This tells Policy Automation Hub to look for the prefix Conn_ in this case, and {0} represents the actual, expected Operation Name. Thus Conn_GetMetadata etc.

Ensure the OPA Hub has access to the Siebel environment and there are no firewalls in between.

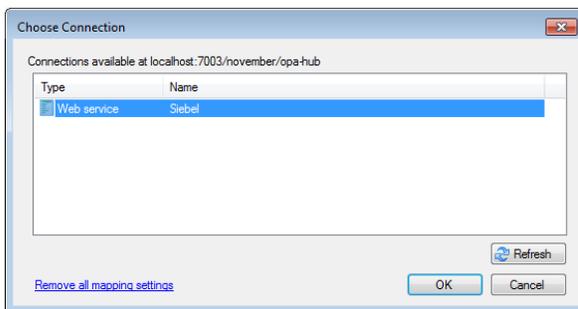
Tick 'Provide WS-Security username token in SOAP Actions' and input the username and password for the Siebel user that will run the web service e.g. SADMIN/password. Tick 'Include timestamp' and save the connection. You should see a green tick appear against the connection if all is correct.



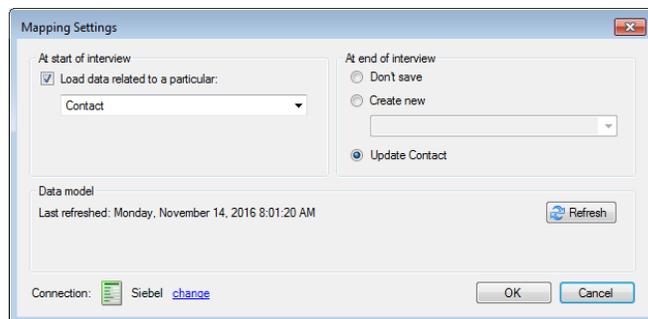
Testing the Siebel Connection in OPM

Now that the connection is active you can open OPM and get the metadata to start creating the data model and rules.

Create a new project, connect to the Hub and open the Data Tab and from Mapping Settings pick the connection you just set up



Then set up the initial mappings of the integration object



Now you can map your Entities to Integration Components and Attributes to Fields using your mapped IO data definitions.

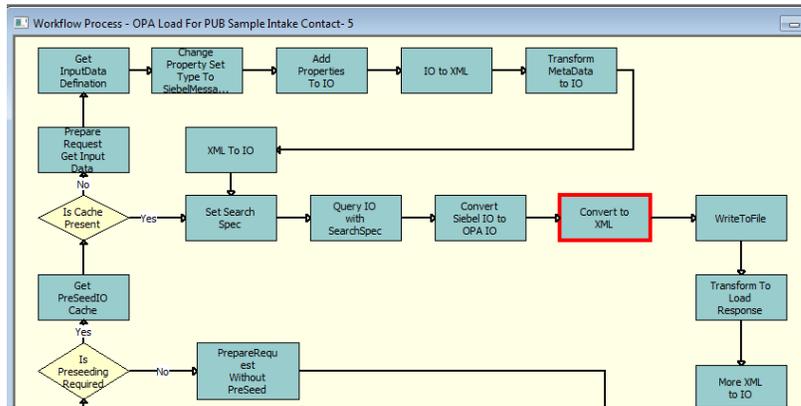
For this example, we are using the 'ApplyForBenefits' Policy Model (included in the package) that is defined against the PUB Sample Intake Contact IO. So far we have added the connection and tested that we can get the metadata for the IO. To use native OPA interviews we will also need to define both a load and submit operation and add a way to start the interview session from Siebel.

Adding the Load Operation

To Load data at the start of the interview we will need to create a workflow that will be used in the Inbound Web Service Load operation. You should have added and released all required workflows in the installation section. The provided workflow is based on the 'OPA Get PreSeed Data Workflow'

Simulate the workflow from Siebel Tools

Now we need to test the workflow. In order to do this, we will need to add in a default value into the ContactId property. Find a contact row id and add in. Ensure the xsl files (GetMetaDataResp.xsl and StartInterviewToLoadResponse.xsl, OPAMsgToSiebelMsgForSubmit.xsl and OPAMsgToSiebelMsgForPreseed.xsl) are in the correct locations locally. These are required to transform from Siebel to OPA format. Also place a copy of the files in the server directory in the siebsrvr directory under the XSLT folder.



Note: If you get an error message about missing DLL's when simulating copy the the 3 DLL's from the Siebsrvr\BIN directory into the client BIN and tools BIN directories.

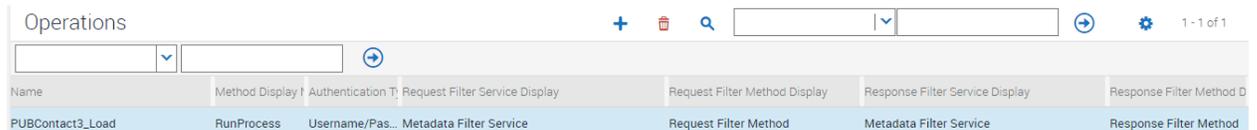
Check the file created in the WriteToFile step to ensure expected data is present. Once tested and verified Publish and Activate the workflow.

Deploying the Load Method

Now we need to add a new service port to the Inbound Web Service to initiate the load (unless you already have this imported).

IntakeConCheckAlivePort	Workflow Process	OPA Metadata Service - CheckAlive	HTTP	http://<webserver>/eai_anon_<lang>/start.swe?SWEExtSource=SecureWe
IntakeConGetMetadata	Workflow Process	OPA Get Meta Data Service For PUB Sample Intake Contact	HTTP	http://<webserver>/eai_anon_<lang>/start.swe?SWEExtSource=SecureWe
IntakeLoad	Workflow Process	OPA Load For PUB Sample Intake Contact	HTTP	http://<webserver>/eai_anon_<lang>/start.swe?SWEExtSource=SecureWe

Add the workflow name we just simulated. Type IntakeLoad into Name field and in the shuttle applet click the plus symbol to add a new inbound web service port type. Give it a Name and Type of workflow process and pick in the workflow process name just simulated. Binding needs to be SOAP_DOC_LITERAL . Then add the operation, as below and clear the cache:



Check that IO's, Workflows and srf have been deployed to the server and that the server has been restarted.

After deploying the rulebase to the hub we can now test the load in SoapUI Load method and in an OPA interview.

Open a browser and type the url for the deployed rulebase, adding a Siebel contact row-id on the end of the URL e.g.

http://localhost:7003/november/web-determinations/startsession/ApplyForBenefits?id=0-1

You should see the first screen load and any pre-seeded data loaded

Personal Infor... Household me... Primary Contac... Household Me...

Personal Information

First Name

Middle Name

Last Name

Date of birth (MM/DD/YYYY)

SSN

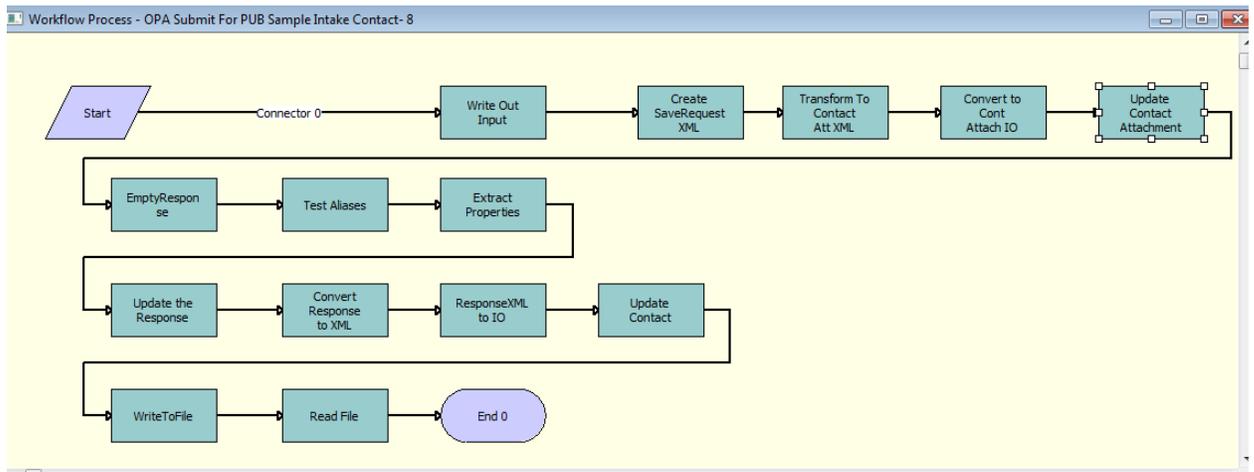
Gender
 Male
 Female

Deploying the Save Method

We need to add the ability to submit and save any data back to Siebel.

For this we need another workflow and another Inbound Web Service port and operation.

Create a new workflow process (or import from the package). Our example is named 'OPA Submit for PUB Sample Intake Contact', which also calls a subprocess OPA Process Submit Data.



It looks quite complex but is mainly transforming, updating and writing out the responses.

Launching the Interview

The next task is to add a way of initiating the OPA interview from Siebel. To do this we will add a 'Launch' button and add a Symbolic URL definition.

Add the Symbolic URL definition first – Navigate to Administration – Integration/Symbolic URL Administration and add a new record – ours is named Web Determinations Popup. Add the url to the policy model and append the query string with id=[id]. Add values as shown for Fixup Name and SSO Disposition

WI Symbolic URL List Symbolic URL Administration:

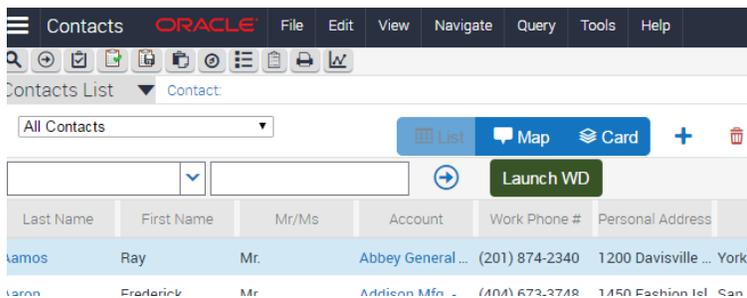
Name	URL	Host Name	Fixup Name	Multivalue Treat	SSO Disposition	Web App
Web Determinations	http://localhost/siebel-wd-embedded/startsession/[rulebase]/[locale]?user=[UserID]&caseID=[SessionID]	demohost			IFrame	
Web Determinations Context	http://demohost/siebel-wd-embedded/startsession/[rulebase]/[locale]?user=[UserID]&caseID=[SessionID]	demohost			IFrame	
Web Determinations Intake	http://localhost:7001/siebel-wd-embedded/startsession/[rulebase]/[locale]?user=username&caseID=C...		Default		IFrame	
Web Determinations Popup	http://localhost:7003/november/web-determinations/startsession/ApplyForBenefits?id=[id]		OutsideApplica...		Form Redirect	

Then add the following arguments, which will make the OPA interview open as a popup and add the current contact id to the query string.

Symbolic URL Arguments

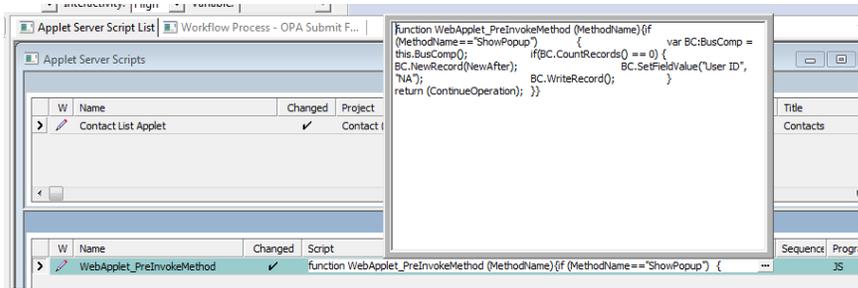
Name	Required Argument	Argument Type	Argument Value	Append as Argur	Substitute in Tex	Sequence #▲▼
FreePopup	<input checked="" type="checkbox"/>	Command	TRUE	<input checked="" type="checkbox"/>		1
[id]	<input checked="" type="checkbox"/>	Field	Id		<input checked="" type="checkbox"/>	1
FullWindow	<input checked="" type="checkbox"/>	Command	TRUE	<input checked="" type="checkbox"/>		2

Now we need to add a button to Launch the URL. Ours is located on the Contact List Applet. You can import this rather than follow the instruction. Remember to compile a srf and add to the server.

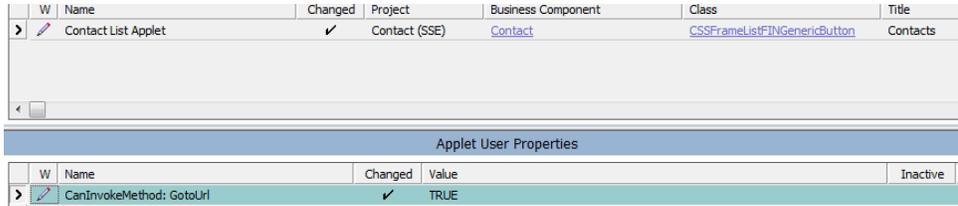


In Siebel Tools Navigate to the 'Contact List Applet' and lock. Add the following to the Applet Server Script

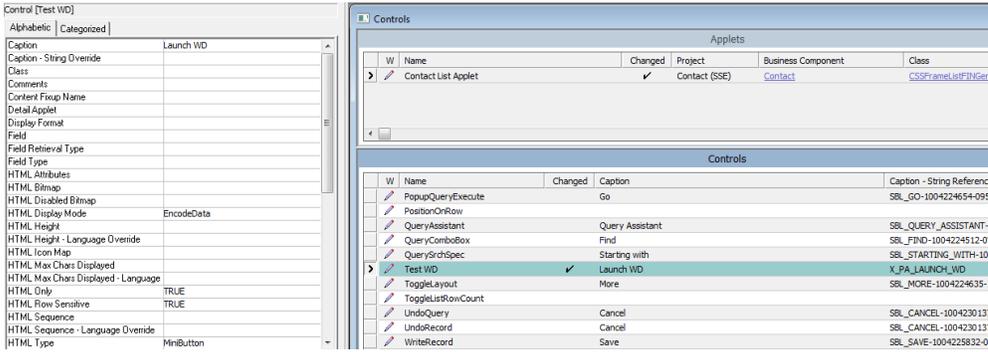
Combining Siebel IP 2016 and OPA v12 Interviews



Under Applet User Properties add the following:



Under Applet Control add a button control as follows:



Additional properties definitions for the control

Properties

Control [Test WD]

Alphabetic | Categorized

HTML Width	
HTML Width - Language Override	
Inactive	FALSE
MVG Applet	
Method Invoked	ShowPopup
Module	
Name	Test WD
Parent Name	Contact List Applet
Pick Applet	
Prompt	FALSE
Prompt Text	
Prompt Text - String Override	
Read Only	FALSE
Runtime	FALSE
Show Popup	FALSE
Sort	TRUE
Target View Frame	
Text Alignment	Left
Text Alignment - Language Override	
Text Alignment-Label	Right
Text Alignment-Label - Language Overrid	
Tooltip Text - String Override	
Visible	TRUE
Visible - Language Override	

Under Control User Properties add the following for Control Test WD

The screenshot shows the 'Control User Props' configuration table. The 'Test WD' property is highlighted, showing it is set to 'Launch WD' with a value of 'X_PA_LAUNCH_WD'. Below this, a table lists the 'Control User Props' for the 'PopUp' control.

W	Name	Changed	Value	Inactive	Comments
<input checked="" type="checkbox"/>	PopUp	<input checked="" type="checkbox"/>	Contact Web Determinations Test Popup		

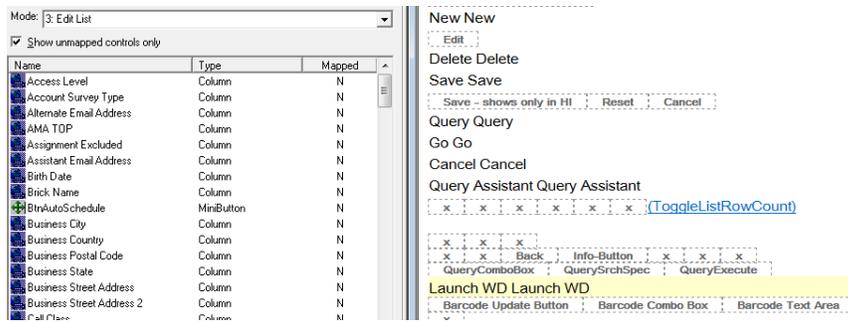
Under Applet Web Template Item for the Edit List template, add the following:

The screenshot shows the 'Applet Web Template Items' configuration table. The 'Test WD' item is highlighted, showing it is set to 'Edit List' with an item identifier of '10,672'.

W	Name	Changed	Sequence	Type	Web Template
<input checked="" type="checkbox"/>	Test WD	<input checked="" type="checkbox"/>	1	Edit List	Applet List (Base/Ed

Add the control to the template by editing the Edit List Template and dragging the 'Launch WD' icon onto the template

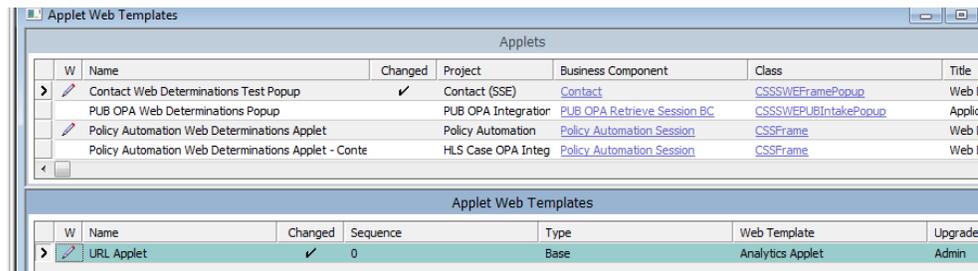
Combining Siebel IP 2016 and OPA v12 Interviews



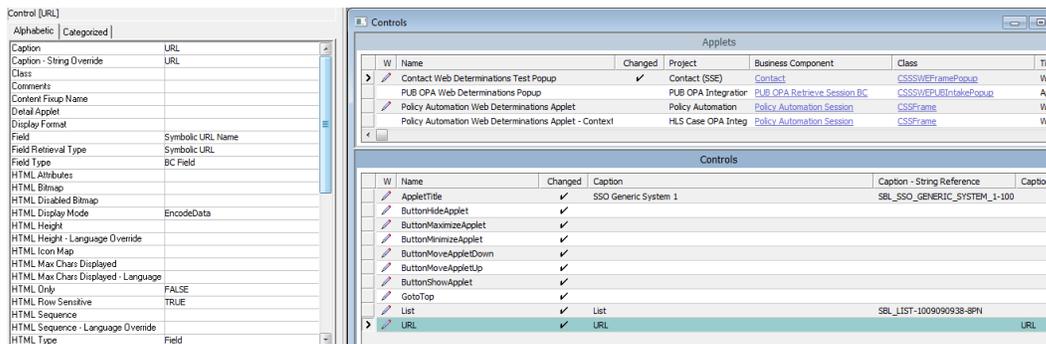
Compile the changes.

Add a new Applet to control the URL. Ours is named 'Contact Web Determinations Test Popu' and copied from an existing applet 'Policy Automation Web Determinations Popu'. Base on the Contact Business Component.

Under Applet Web Template add the following:



Under Applet Control add the following control



Additional control properties

Properties

Control [URL]

Alphabetic | Categorized

HTML Width	
HTML Width - Language Override	
Inactive	FALSE
MVG Applet	
Method Invoked	
Module	
Name	URL
Parent Name	Contact Web Determinations Test Popu
Pick Applet	
Prompt	FALSE
Prompt Text	
Prompt Text - String Override	
Read Only	FALSE
Runtime	FALSE
Show Popup	FALSE
Sort	TRUE
Target View Frame	
Text Alignment	Left
Text Alignment - Language Override	
Text Alignment-Label	Right
Text Alignment-Label - Language Overrid	
Tooltip Text - String Override	
Visible	TRUE
Visible - Language Override	

Under Applet Web Template Items add the following

Properties

Applet Web Template Item [URL]

Alphabetic | Categorized

Column Span	
Column Span - Language Override	
Comments	
Control	URL
Expression	
Grid Property	
Inactive	FALSE
Item Identifier	501
Item Identifier - Language Override	
Mode	
Module	
Name	URL
Namespace	
Namespace - Language Override	
Parent Name	URL Applet
Row Span	
Row Span - Language Override	
Type	Control

Applet Web Template Item List | Workflow Process - OPA Submit F...

Applet Web Template Items

W	Name	Changed	Sequence	Type	Web Template	Upgrad
>	URL Applet	✓	0	Base	Analytics Applet	Admin

Applet Web Template Items

W	Name	Changed	Control	Expression	Item Identifier	Item Id
>	URL	✓	URL		501	

Under Applet List Column add the following

Properties

List Column [Symbolic URL Name]

Alphabetic | Categorized

Available	TRUE
Available - Language Override	
Comments	
Content Fixup Name	
Detail Applet	
Display Format	
Display Name	
Display Name - String Override	
Field	Symbolic URL Name
Field Retrieval Type	Symbolic URL
HTML Attribute	
HTML Display Mode	EncodeData
HTML Height	
HTML Height - Language Override	
HTML Icon Map	
HTML List Edit	TRUE
HTML Max Chars Displayed	
HTML Max Chars Displayed - Language	
HTML Only	FALSE
HTML Row Sensitive	TRUE
HTML Sequence	
HTML Sequence - Language Override	
HTML Type	Text
HTML Width	

List Column List | Workflow Process - OPA Submit F...

List Columns

W	Name	Changed	HTML Hierarchy Bitmap	HTML Multi-Row	HTML Multi-Row :	Total Displayed	Total F
>	List	✓					

List Columns

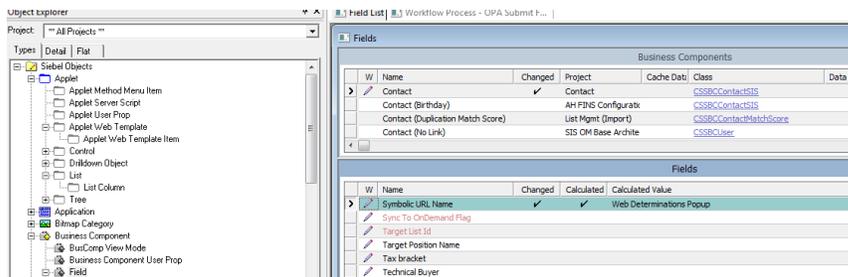
Module	W	Name	Field	Changed	Available	Availabl
>		Symbolic URL Name	Symbolic URL Name	✓	✓	

Additional properties

properties	
List Column [Symbolic URL Name]	
Alphabetic Categorized	
HTML Sequence	
HTML Sequence - Language Override	
HTML Type	Text
HTML Width	
HTML Width - Language Override	
Inactive	FALSE
MVG Applet	
Module	
Name	Symbolic URL Name
Parent Name	List
Pick Applet	
Prompt Text	
Prompt Text - String Override	
Read Only	FALSE
Runtime	FALSE
Show In List	TRUE
Show In List - Language Override	
Show Popup	FALSE
Text Alignment	Left
Text Alignment - Language Override	
Text Alignment-Label	Left
Text Alignment-Label - Language Overrid	
Total Currency Code Expression	
Total Required	FALSE

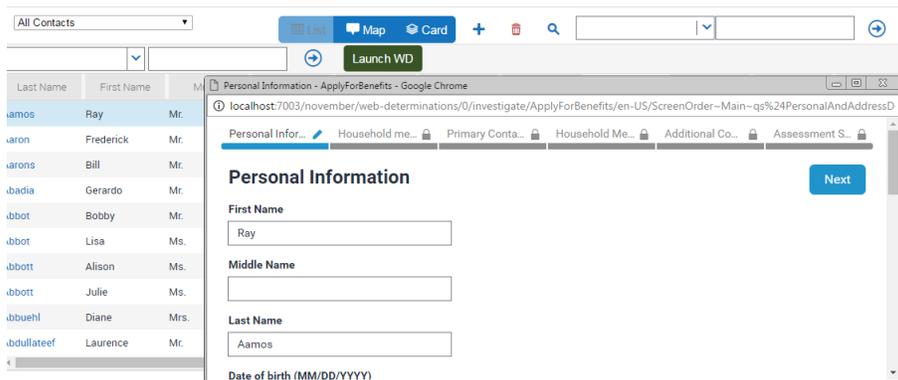
Compile the applet

On Business Component Contact, add a new field for the Symbolic URL



Compile all changes and put the new srf on the server and restart the services to test.

Test the button

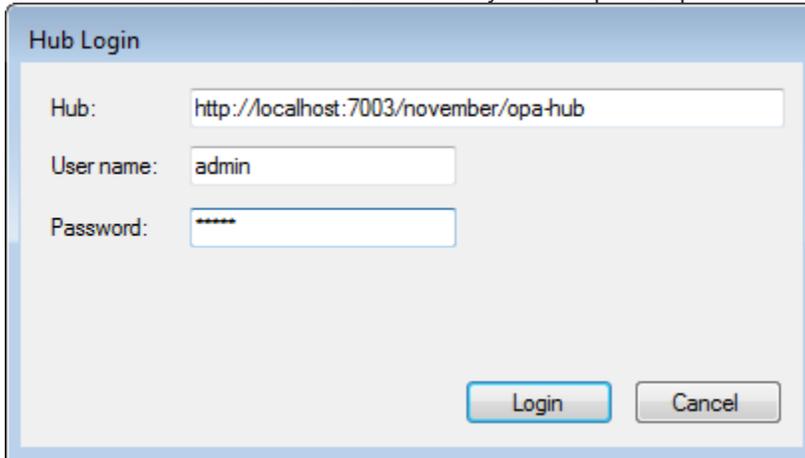


If you encounter any errors try looking in the Object Manager log files on the Siebel server.

Checkpoints

1. Create a new OPM project or choose one of Sample Projects.

The Interview should have a number of screens so you can validate the Checkpoints
On the Data Tab choose the Connection you set up in step 1

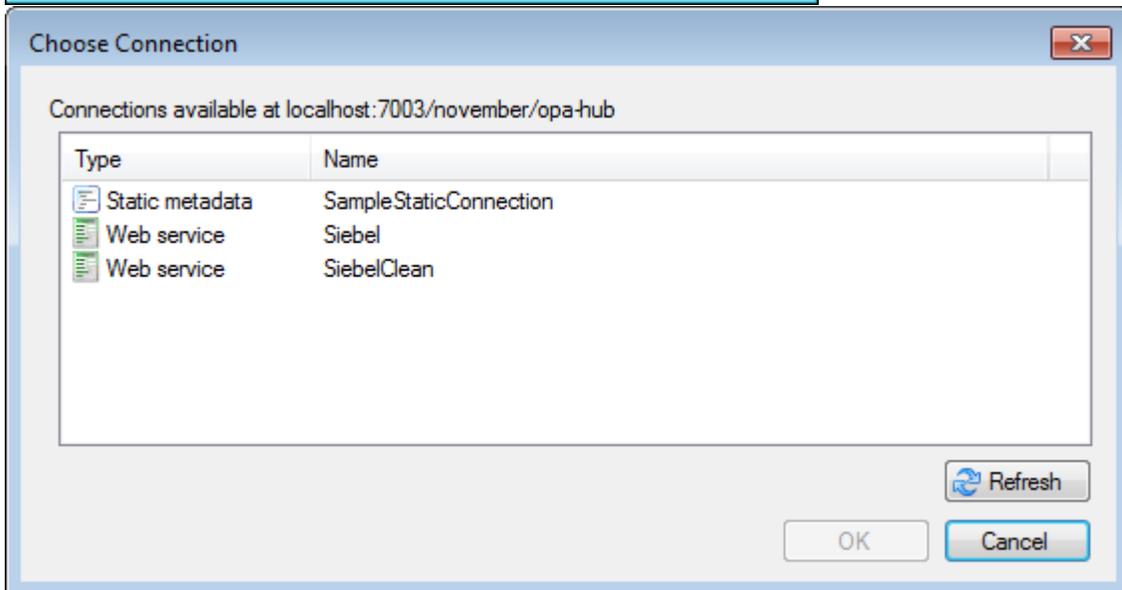


Hub Login

Hub:

User name:

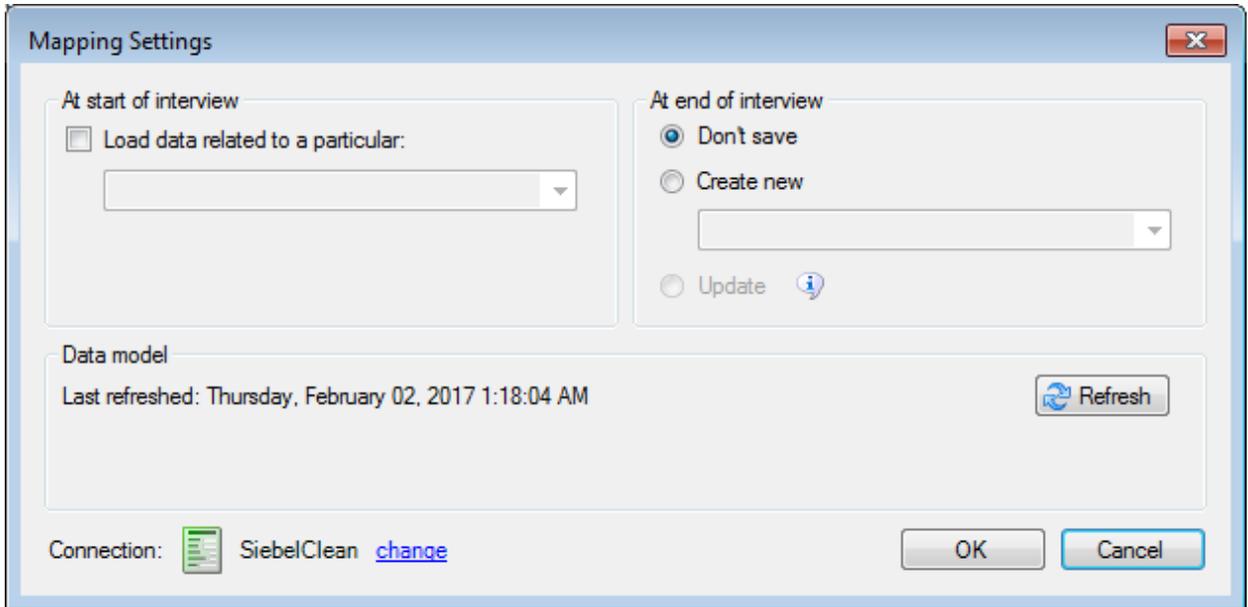
Password:



Choose Connection

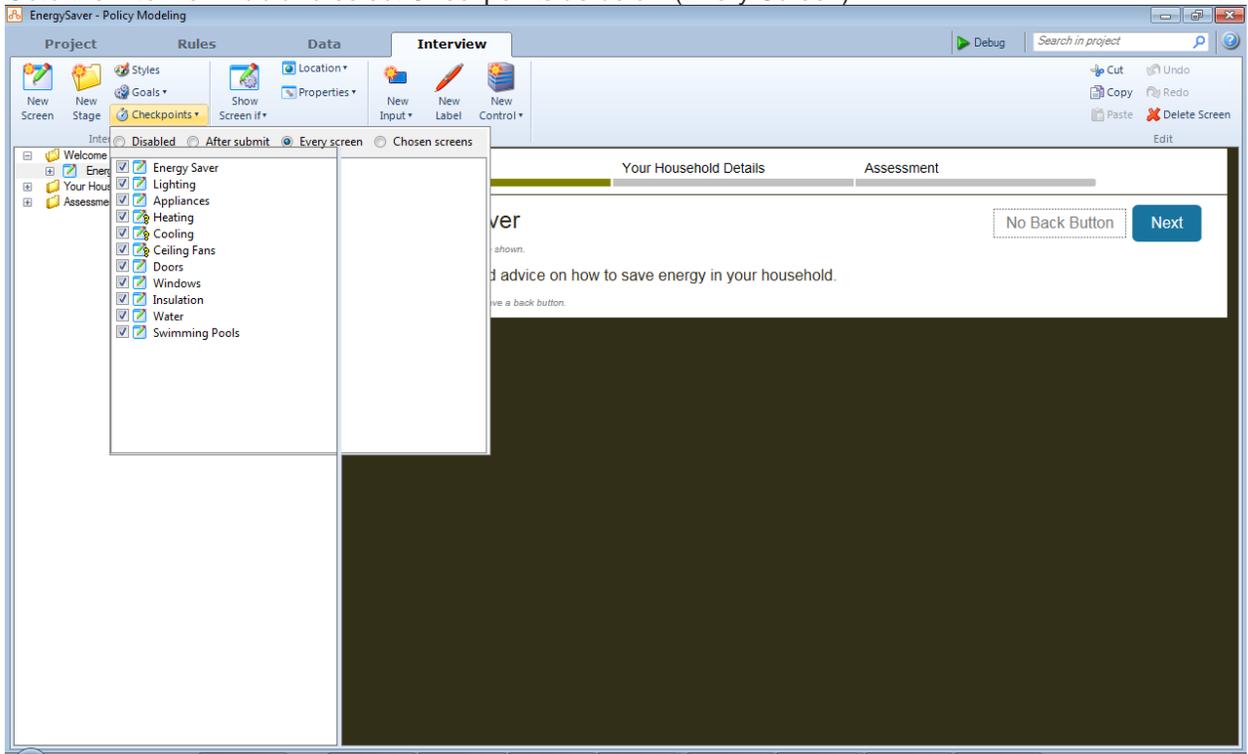
Connections available at localhost:7003/november/opa-hub

Type	Name
 Static metadata	SampleStaticConnection
 Web service	Siebel
 Web service	SiebelClean



Don't choose Load data or an action at the end of interview – First we will test checkpoints

Goto the Interview Tab and select Checkpoints as below (Every Screen)



Then Deploy the Interview

Deploy Snapshot

Hub:

Username:

Password:

Deployment Name:

Comments:

Activate Immediately

Interview:

Deploy Cancel

2. Test using the OPA interview

<http://localhost:7003/november/web-determinations/startsession/EnergySaver?id=1234>

<http://localhost:7003/november/web-determinations/startsession/CopyrightPermissions?id=12>

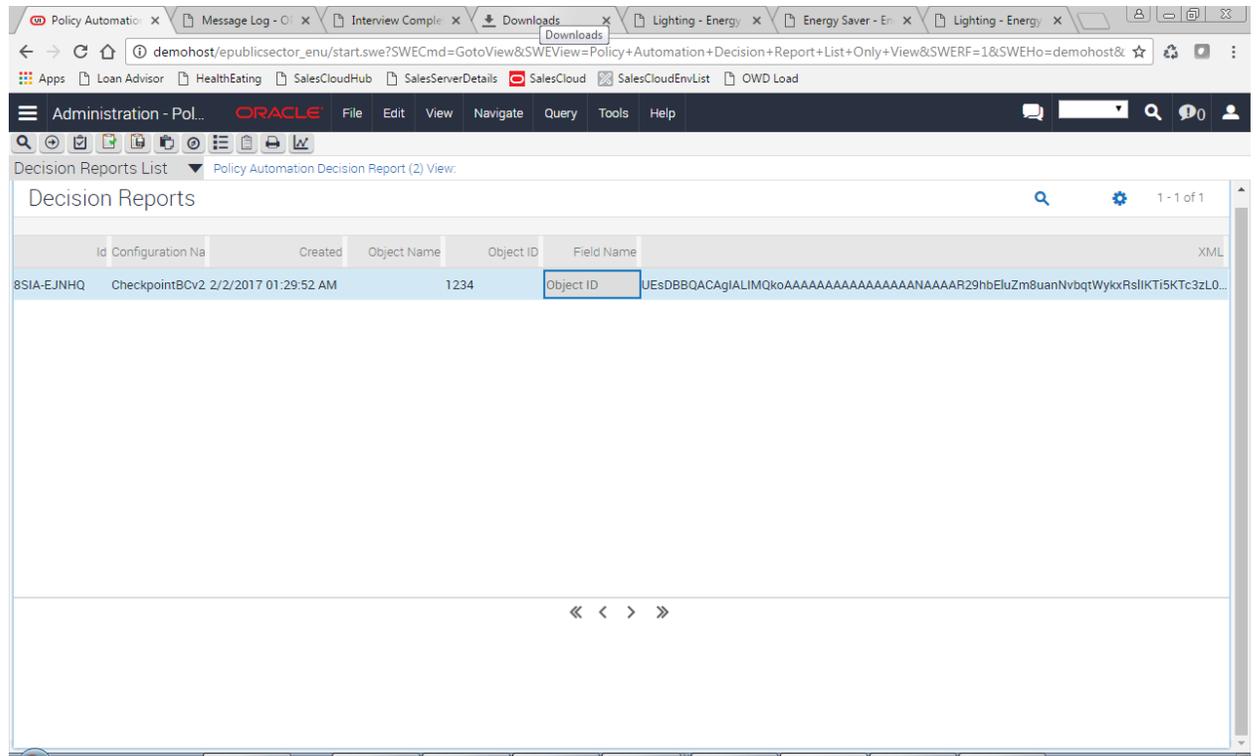
Note: EnergySaver is not a create sample as not clear that you have saved data

Note: The parameter name, id, is not important at the moment and number,1234, is just used to retrieve the correct checkpoint. This should be improved to identify checkpoint by user and Deployment Name

Complete a few screens with values so you can be sure checkpoint has retrieved

Check that a record has been create in Siebel – Administration Policy Automation – Decision report List

Combining Siebel IP 2016 and OPA v12 Interviews



<http://localhost:7003/november/web-determinations/resumesession/EnergySaver?id=1234>

<http://localhost:7003/november/web-determinations/resumesession/CopyrightPermissions?id=12>

You should open the session in a different browser to prove it is not caching

The screenshot shows a web browser window with several tabs open. The active tab is titled 'Initial Details - Co'. The browser's address bar shows a local URL: localhost:7003/november/web-determinations/11/investigate/CopyrightPermissions/en-US/ScreenOrder~Main~qs%243d8fcb6c-9be5-40f7-bf58-557e17fe7968%. The browser's taskbar at the bottom shows various applications like 'Apps', 'Loan Advisor', 'HealthEating', 'SalesCloudHub', 'SalesServerDetails', 'SalesCloud', 'SalesCloudEnvList', and 'OWD Load'. The main content area displays a form titled 'Initial Details' with a 'Next' button in the top right corner. The form contains the following fields and options:

- Name of person proposing to use the material:** A text input field containing 'Phil Test'.
- Gender:** Radio buttons for 'Male' and 'Female'. The 'Female' option is selected.
- Does the above person:** Two checkboxes: 'hold a licence to use the material?' (checked) and 'propose to use the material outside Australia?' (unchecked).
- Can the creator of the relevant material be identified?** Radio buttons for 'Yes' and 'No'. The 'Yes' option is selected.

The interview should resume where you previously left it. However it is possible for interviews to force where a resume interview will start.

Included is a SOAPUI project to allow testing

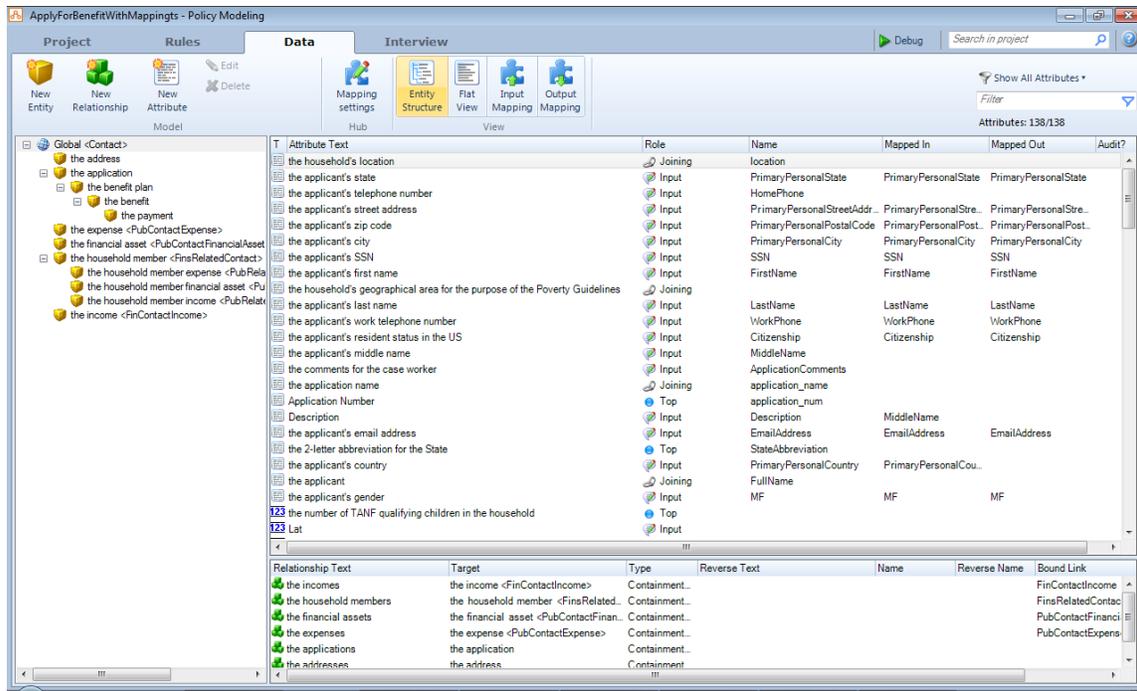
Appendix – How to use the OPA Answer Service

In the folder Answer Service Example is all that need to perform a workflow call to an OPA Answer Service. This reuses the ApplyForBenefits OPA Project.

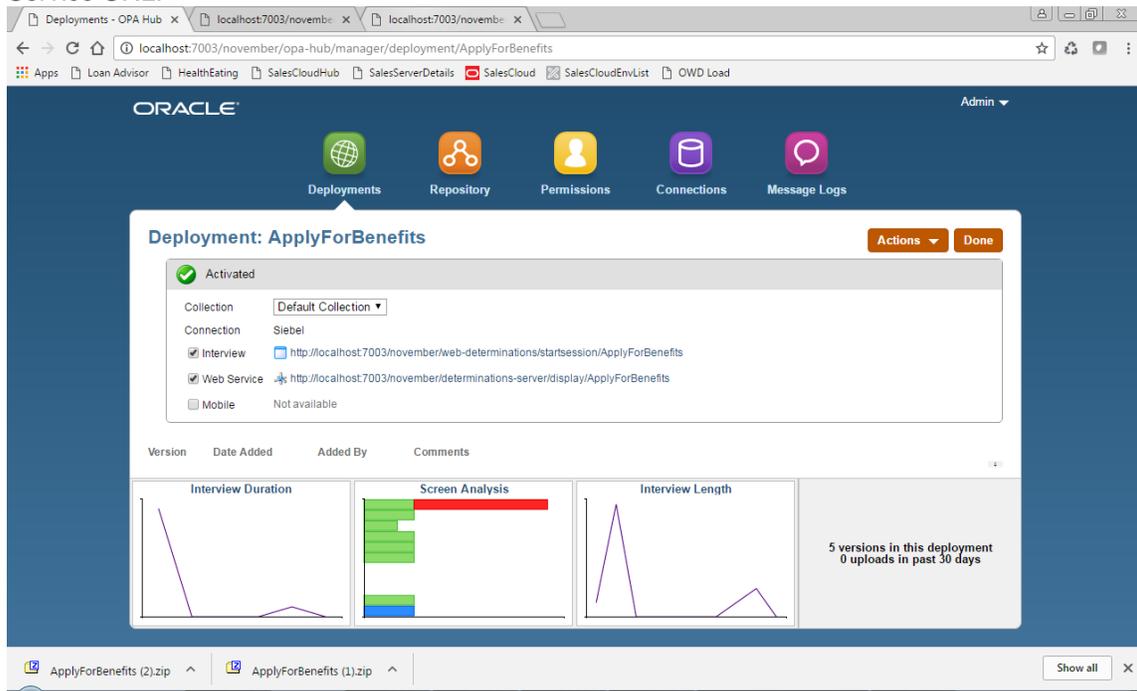
Deploy the OPA Policy Model

Open the ApplyForBenefits.zip file using Oracle Policy Modeling. Check that correct Connection is chosen in the Data tab and then deploy to your OPA Hub.

Combining Siebel IP 2016 and OPA v12 Interviews



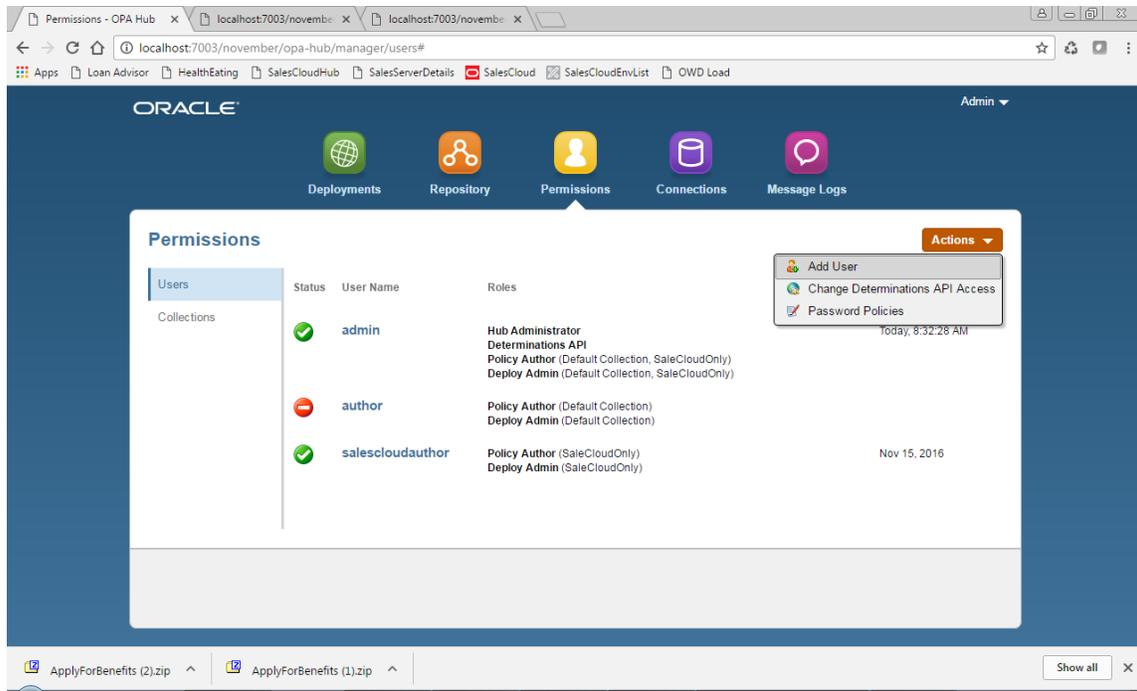
Check in the Hub that the deployment has been enabled for Web Services and has the Answer Service URL.



e.g <http://localhost:7001/november/determinations-server/answer/soap/12.2/ApplyForBenefits?wsdl>

Next ensure that anonymous access is enabled for Web Services.

Combining Siebel IP 2016 and OPA v12 Interviews

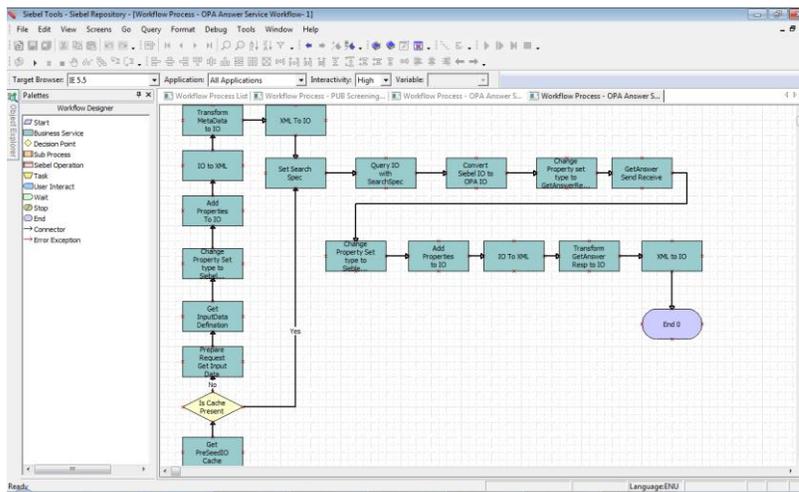


Goto Permissions -> Actions choose Change Determinations API Access and then select Allow anonymous Access

Deploy the Siebel Workflow

Import the OPA Answer Service Workflow Simulation.xml Workflow from the Assess Service Example folder into Siebel Tools – Work Processes

Note: This is a copy of a standard Siebel workflow with the Process Properties set to allow the workflow to be run in the workflow simulator. The workflow is normally called as a sub –process of the PUB*Screening* or PUB*Eligibility* workflows.



Check the process properties are suited to your local environment.

Process Properties		Process Metrics
Name	Default String	
Error Code		
Error Message		
HostURL		http://localhost:7003/november/determinations-...
InputDataIOName		PUB Sample Intake Contact
InputDataIOxmlTag		ListOPubSampleIntakeContact
InputDataPrimaryIC		Contact
InputDataPrimaryICxmlTag		Contact
Is Cache Present		
OPAResultIO		
Object Id		0-1
PreseedIO		
PreseedXML		
PreseedXSLPath		c:\\temp\\OPAMsgToSiebelMsgForPreseed.xml
Process Instance Id		
QueriedIO		
Rulebase		ApplyForBenefits
Search Spec		
SearchSpecIO		
Siebel Operation Object Id		
SubmitDataXML		
SubmitXSLPath		c:\\temp\\OPAMsgToSiebelMsgForSubmit.xml
TransformedIOwithSearchSpec		

Included are the two required XSL transform files that will need to be deployed to C:\temp\ if you keep the process properties as above.

The HostURL should be checked to ensure that it matches the deployment tested in the OPA section.

Once you have set-up this example you will be able to stimulate the workflow from Siebel Tools and then once happy with the IOs you can deploy and then call the workflow as part of your main Siebel to OPA decision making process.

Conclusion

It is possible to reuse the Siebel components supplied with IP2016 to allow users to create a web service connection from OPA that allow mapping of Siebel objects in Oracle Policy Modeling and then native OPA interviews are able to support Save, Load, GetCheckpoint and SaveCheckpoint.

If further help is required, then Oracle Consulting has the expertise to assist you with any aspect contained in this document and with your project implementation.



Leveraging the Power of Oracle Engineered
Systems for Enterprise Policy Automation
May 2012
Author: Davin Fifield

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2012, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

SOFTWARE. HARDWARE. COMPLETE.