

Exercise 10

*Using a Registry to publish endpoint metadata and retrieving that endpoint data
Looking at wider Registry Governance issues*

Prior Knowledge

Using the ESB, creating services, deploying services
WSDL

Objectives

Deploy a governance registry, connect the services server to the governance registry, publish services into it. Find those services from the ESB.

Software Requirements

- Previous installations of ESB and AS from Exercises 8 and 9
- Java Development Kit 7
- WSO2 Governance Registry 4.6.0

1. Start the Governance Registry

```
cd ~/servers/wso2greg-4.6.0  
bin/wso2server.sh
```

2. Go to the administrators console: https://localhost:9445/
3. Login with admin/admin



4. You should see a screen like this:

The screenshot shows the WSO2 Governance Registry Management Console. The URL is `https://localhost:9445/carbon/admin/index.jsp?loginStatus=true`. The interface includes a left sidebar with icons for Home, Manage, Resources, and Metadata. Under Home, there are sections for List (APIs, Documents, Endpoints, Policies, Providers, Proxies, Schemas, Sequences, Services, URIs, WADLs, WSDLs) and Add (API, Document, Endpoint, Policy, Provider, Proxy, Schema). The main content area displays the WSO2 Governance Registry Home page with a welcome message and server statistics. The server details are as follows:

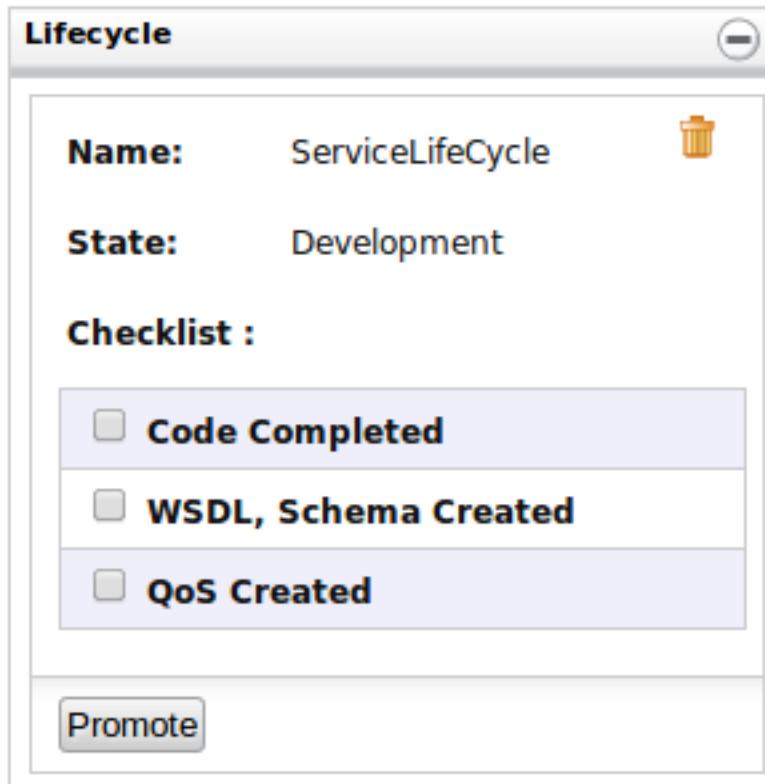
Server	
Host	172.16.40.136
Server URL	local://services/
Server Start Time	2013-11-26 20:41:40
System Up Time	0 day(s) 0 hr(s) 1 min(s) 8 sec(s)
Version	4.6.0
Repository Location	file:/home/ox-soa/servers/wso2greg-4.6.0/repository/deployment/server/

Below the server details are sections for Operating System and Java VM, each with a single row of data.

5. If you click on Services you should see an empty list (since we haven't registered any services with the Registry)
6. In a moment we are going to automatically add some services, but let's first do one the "hard way" – by hand. To make it easy we are going to be quite lazy.
7. Go to the **Add /Service** button on the left hand side.
8. This is a "top-down" way of defining a service. Fill in some information. Use the following information / leave other fields empty:
Name: OrderService
Namespace: me.freo.po
Version: 0.0.1
Description: Ordering Service.
Save it.



9. Find the Lifecycle section on the right hand side and expand it, if necessary. The service will have been given the “default service lifecycle”, and be in the first stage:



10. Now in order to move this Service out of Development we need to complete some tasks:
 - * Code Completed
 - * WSDL, Schema Created
 - * QoS Created
11. Pretend you've done all that. Select the checkboxes. Now click Promote. You will be prompted to enter a new Version Number. Type in 1.0.0
12. Now look at your Services list again. You will have another version of this service available, now in the Testing Lifecycle. Take a look at it.
13. If you want to understand the gory details of how to configure your own lifecycles and lifecycle actions, go to the left hand **Extensions** tab, click Configure/Lifecycles, then **View** the ServiceLifecycle configuration. You



will see an XML based on the SCXML standard.

```
<aspect name="ServiceLifeCycle" class="org.wso2.carbon.governance.registry.extensions.aspects.DefaultLifeCycle">
<configuration type="literal">
<lifecycle>
<scxml xmlns="http://www.w3.org/2005/07/scxml"
version="1.0"
initialstate="Development">
<state id="Development">
<datamodel>
<data name="checkItems">
<item name="Code Completed" forEvent="">
<!--<permissions>
<permission roles="" />
</permissions>
<validations>
<validation forEvent="" class="">
= = = = =
</validation>
</validations>
</item>
</data>
</datamodel>
</state>
</lifecycle>
</configuration>
</aspect>
```

14. Now lets get our AppServer Services published in the Registry. Our services are hosted in WSO2 Application Server, and this can automatically publish into the Registry using WS-Dynamic Discovery. To do this, we need to tell the AppServer where the Registry is.
[Please note this only currently works for Axis2 services deployed in AS as there is an incompatibility between CXF's WSDD and G-Reg's WSDD which we are looking into.]

15. To do this, edit `~/servers/wso2as-5.2.0/repository/conf/axis2/axis2.xml`

16. Add the following (bold) line in the correct place, which is next to any other parameter:
[You can find it at <http://freo.me/wsddgist>]

```
<parameter
name="DiscoveryProxy">https://localhost:9445/services/DiscoveryProxy</parameter>
```

17. Restart (or Start) the Application Server. Note that the Governance Registry must now be running whenever the AS is. You can do this from the Web Admin Console or from the command line.



18. Wait a bit and then check the Registry Services list:

The screenshot shows the WSO2 Governance Registry Management Console. The left sidebar has tabs for Home, Dashboard, Governance Dashboard, Manage (Carbon Applications, Resources, Metadata), Extensions, Configure, and Monitor. The main content area shows a 'Service List' with the following data:

Service Name	Service Version	Service Namespace	Actions
admin-RESTSample	1.0.0-SNAPSHOT	http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01	
echo	1.0.0-SNAPSHOT	http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01	
HelloService	1.0.0-SNAPSHOT	http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01	
StarbucksOutletService	1.0.0-SNAPSHOT	http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01	
Version	1.0.0-SNAPSHOT	http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01	
wso2carbon-sts	1.0.0-SNAPSHOT	http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01	

19. You can also find the WSDLs from these services in the WSDL list.

20. We can now “Link” the ESB to the Registry:

21. Start the ESB if it isn’t running.

22. Go to the ESB Admin Console (<https://localhost:9444>)

23. Change the left hand menu tab to Configure

24. Select WS-Discovery

25. Click Add Discovery Proxy



26. Fill in the details as follows:

Proxy Name: WSO2Registry

URL: <https://localhost:9445/services/DiscoveryProxy>

Security Policy: [Leave empty]

Home > Configure > WS-Discovery > Discovery Proxy

Configure WS-Discovery Proxy

Enter the name and the URL of a remote WS-Discovery proxy

Discovery Proxy Settings	
Proxy Name *	WSO2Registry
URL *	<input type="text" value="https://localhost:9445/services/DiscoveryProxy"/>
Security Policy	<input type="text"/> Browse
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

27. Click Save

28. It should now say that the server is Online

The screenshot shows the WSO2 Management Console interface. On the left, there's a navigation sidebar with sections like Main, Monitor, Configure, and Tools. Under the Configure section, 'WS-Discovery' is selected. The main content area is titled 'WS-Discovery Control Panel'. It contains a table with one row:

Proxy Name	URL	Server Status	Actions
WSO2Registry	https://localhost:9445/services/DiscoveryProxy	On-line	View Edit Delete

Below the table, there's a link to '+ Add Discovery Proxy'.

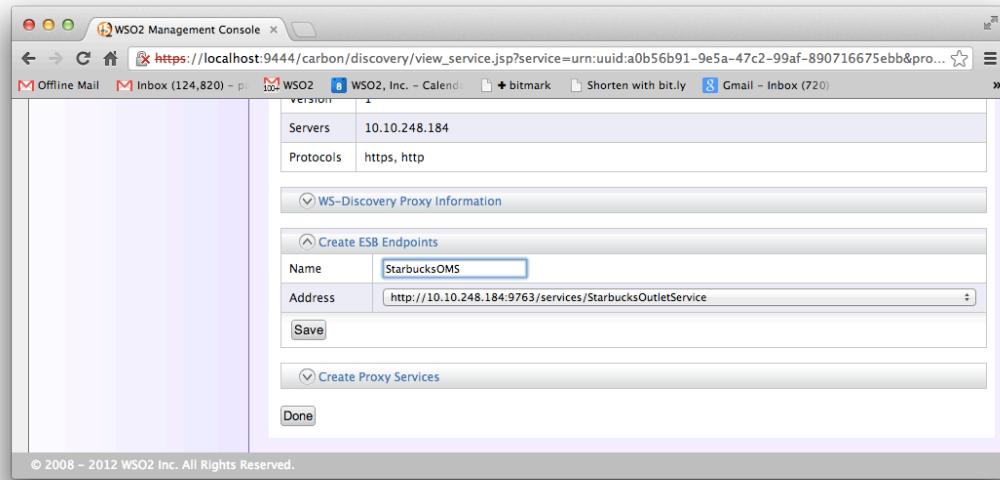
29. Click View

You will see the various endpoints from the services that are published in the Registry. Find the service with the “Starbucks endpoints” e.g.
<http://10.10.248.184:9763/services/StarbucksOutletService>

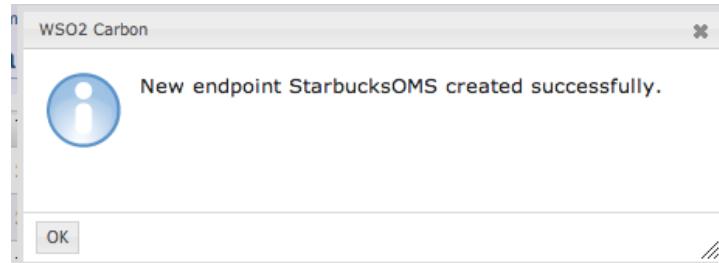
30. Click on the UUID link



31. Go down to the section “Create ESB Endpoints” and expand it.



32. Give the name StarbucksOMS and hit Save. You should see:



33. If you switch back to the Manage tab and look at the endpoints, you should now find it is there.

34. That's all folks!

