

WebSphere Lab Jam

Connectivity

WebSphere Service Registry & Repository

Lab Exercises



An IBM Proof of Technology

Catalog Number

Contents

LAB 1	EXPERIENCE PUBLISH AND FIND	6
	1.1 SOA NEEDS A SERVICE REGISTRY AND REPOSITORY	6
	1.2 IBM WEBSPHERE SERVICE REGISTRY AND REPOSITORY	7
	1.3 LAB INTRODUCTION	8
	1.4 START THE ENVIRONMENT	8
	1.5 EXPLORE WEBSPHERE SERVICE REGISTRY AND REPOSITORY	9
	1.6 PUBLISH A DOCUMENT	16
	1.7 FIND IN WEBSPHERE SERVICE REGISTRY AND REPOSITORY	22
	1.8 EXPERIENCE THE WEBSPHERE SERVICE REGISTRY AND REPOSITORY ECLIPSE PLUG-IN.....	29
	1.9 WORKING WITH METADATA	35
	1.10 SUBSCRIPTION & NOTIFICATION	37
	1.11 IMPACT ANALYSIS	40
	1.12 LAB REVIEW	42
LAB 2	CREATING AN ORGANIZATIONAL STRUCTURE	43
	2.1 LAB INTRODUCTION	43
	2.2 START THE ENVIRONMENT	43
	2.3 LAB REVIEW	47
LAB 3	GOVERNING A SCHEMA	48
	3.1 LAB INTRODUCTION	48
	3.2 CREATE THE SCHEMA SPECIFICATION	48
	3.3 PROPOSING THE SCHEMA SPECIFICATION	52
	3.4 APPROVING THE SCHEMA SPECIFICATION.....	54
	3.5 LAB REVIEW.....	56
	3.6 OPTIONAL EXPLORATIONS.....	56
LAB 4	GOVERNING A WSDL DEFINED SERVICE.....	58
	4.1 LAB INTRODUCTION	58
	4.2 CREATE A NEW BUSINESS SERVICE	60
	4.3 REVIEWING AND APPROVING THE BUSINESS SERVICE	65
	4.4 CREATING A SERVICE VERSION	67
	4.5 APPROVE THE SERVICE VERSION SCOPE.....	71
	4.6 LOAD THE WSDL DOCUMENT FOR THE ELIGIBILITY SERVICE.....	73
	4.7 CREATE A SERVICE INTERFACE SPECIFICATION AND LINK IT TO THE SERVICE VERSION	76
	4.8 APPROVING THE SERVICE VERSION	79
	4.9 CREATING A SERVICE LEVEL DEFINITION.....	81
	4.10 APPROVING THE SERVICE LEVEL DEFINITION	85
	4.11 DEPLOYING A SERVICE VERSION TO PRODUCTION	88
	4.12 AUDIT THE HISTORY OF THE SERVICE	93
	4.13 LAB REVIEW	94
LAB 5	EXAMINING THE GOVERNANCE MODEL IN WEBSPHERE SERVICE REGISTRY AND REPOSITORY WITH WEBSPHERE SERVICE REGISTRY AND REPOSITORY STUDIO	95
	5.1 INTRODUCTION TO CONFIGURATION PROFILES	95
	5.2 INTRODUCTION TO THE WEBSPHERE SERVICE REGISTRY AND REPOSITORY STUDIO	97
	5.3 START WEBSPHERE SERVICE REGISTRY AND REPOSITORY STUDIO	98
	5.4 LAB REVIEW	104
LAB 6	DEFINING A WEBSPHERE MQ APPLICATION TO WEBSPHERE SERVICE REGISTRY AND REPOSITORY.....	106
	6.1 LAB INTRODUCTION	106
	6.2 WEBSPHERE MQ SERVICE DEFINITION WIZARD	107
	6.3 LAB REVIEW	117
LAB 7	VERSION CONTROL WITH WEBSPHERE SERVICE REGISTRY AND REPOSITORY AND WEBSPHERE MESSAGE BROKER.....	118
	7.1 INTRODUCTION TO INTEGRATING WEBSPHERE MESSAGE BROKER WITH WEBSPHERE SERVICE REGISTRY AND REPOSITORY.....	118
	7.2 LAB SCENARIO	118
	7.3 LAB – VERSION CONTROL WITH WEBSPHERE SERVICE REGISTRY AND REPOSITORY AND WEBSPHERE MESSAGE BROKER	120
	7.4 LAB REVIEW.....	132

	7.5	OPTIONAL EXPLORATIONS.....	132
LAB 8		EXPLORING THE WEBSHERE SERVICE REGISTRY AND REPOSITORY BUSINESS SPACE	133
	8.1	INTRODUCTION TO WEBSHERE SERVICE REGISTRY AND REPOSITORY BUSINESS SPACE	133
	8.2	LAB SETUP	133
	8.3	EXPLORE THE SERVICE REGISTRY BUSINESS SPACE.....	134
	8.4	LAB REVIEW.....	142
APPENDIX A.		NOTICES	143
APPENDIX B.		TRADEMARKS AND COPYRIGHTS	145

THIS PAGE INTENTIONALLY LEFT BLANK

Lab 1 Experience Publish and Find

1.1 SOA needs a service registry and repository

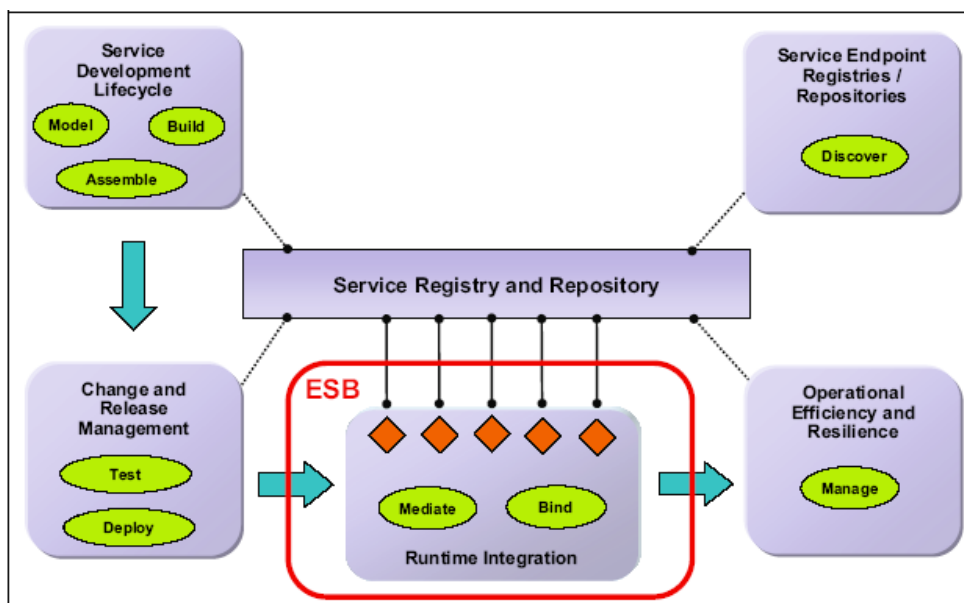
A primary goal of service-oriented architecture is to align the business world with the world of Information Technology (IT) in a way that makes both more effective. Service-oriented architecture (SOA) is about the business results that can be achieved through effectively enhancing this alignment.

To maximize this alignment, we need to employ a number of capabilities:

- A common language for capturing the business design
- A methodology / process for translating the business design into a set of information system artifacts to implement that design
- An infrastructure for hosting those implementation artifacts that allows the business to be flexible to changes in the marketplace
- A place for retaining the correlation between the business design and the information systems that can be used to identify and then fix or improve alignment with the goals and constraints of the business design
- A means by which to manage the system to ensure those goals are met

A service registry and repository provides a central place to organize, manage, find, and govern the service descriptions. In heterogeneous environments, the service registry and repository provides a standard, interoperable means to access, query, and manipulate the service descriptions.

A service registry and repository plays a central role throughout the SOA lifecycle. Therefore, it must flexibly integrate with applications from the service development lifecycle, change and release management, service runtimes, service management (operational efficiency and resilience) and interoperate with other service registries and repositories.

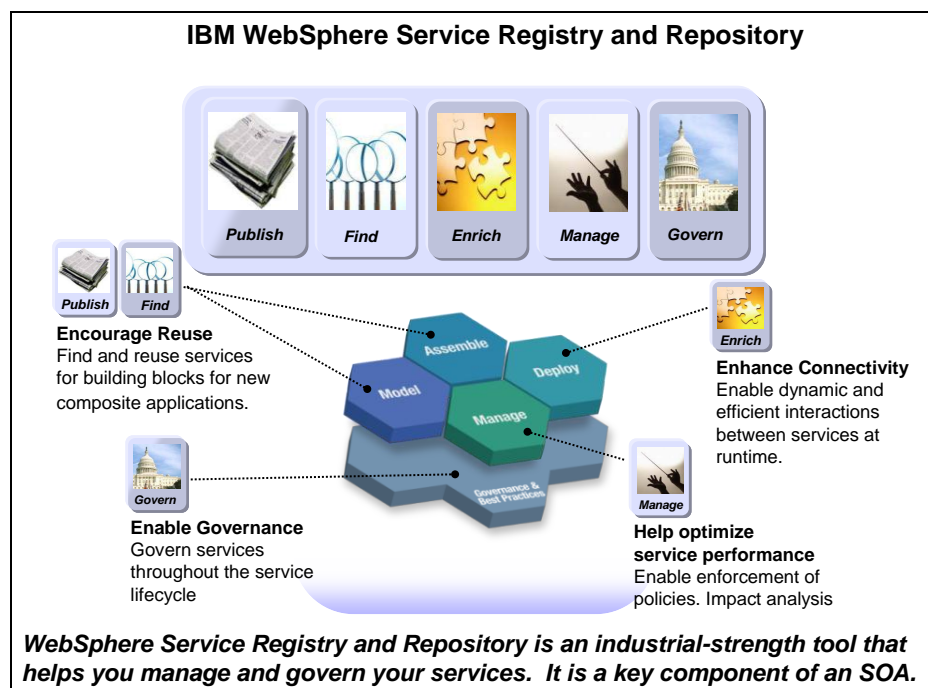


1.2 IBM WebSphere Service Registry and Repository

IBM® WebSphere® Service Registry and Repository is the master metadata repository for service interaction endpoint descriptions. It establishes a central point for finding and managing service metadata. Once service metadata is placed in WebSphere Service Registry and Repository, visibility is controlled, versions are managed, proposed changes are analyzed and communicated, and usage is monitored.

WebSphere Service Registry and Repository incorporates:

- A service registry that contains information about services, such as the service interfaces, its operations, and parameters
- A metadata repository that has the robust framework and extensibility to suit the diverse nature of service usage



The WebSphere Service Registry and Repository provides five key capabilities - Publish, Find, Enrich, Manage, and Govern.

- **Publish** refers to the capability to load or add services and services related information in the registry
- **Find** refers to finding these services using this information
- **Enrich** refers to enhancing the capabilities of the runtime environment e.g. an enterprise service bus (ESB), by providing the ability to dynamically select endpoints at runtime based on metadata
- **Manage** refers to management of this information across the service lifecycle
- **Govern** refers to the governance of the service through a lifecycle

1.3 Lab introduction

In this lab you will familiarize yourself with the WebSphere Service Registry and Repository Web user interface (UI) and exercise the product's capabilities with regard to the publishing and retrieval of documents that define Web services.

During this lab, you will:

- Explore the WebSphere Service Registry and Repository Web UI and its navigation
- Explore the Configuration perspective through which WebSphere Service Registry and Repository is configured.
- Publish service documents (WSDL and XSD)
- Search using Query Wizard and Quick searches using the Web UI
- Retrieve assets using the Web UI
- Search and retrieve assets using the Eclipse plug-in
- Add property metadata to your assets
- Subscribe to a service endpoint.
- Perform impact analysis

The labs are based on a fictional company JKHLE. As you perform the labs you will play the roles a number of JKHLE employees:

Bob – Business Analyst

David – SOA Governance

Connie – Developer

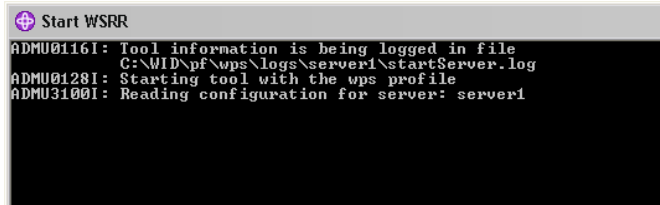
Jon - Operations

as well as the anonymous admin.

1.4 Start the environment

- __1. Start WebSphere Service Registry and Repository
 - __a. Open the **WSRR v7 PoT Shortcuts** folder on the desktop
 - __b. Click the shortcut **Start WSRR Server** to start WebSphere Service Registry and Repository Server that is hosted in WebSphere Application Server.

This will bring up a window. The window will display startup information and remain open after the startup has completed.



```

Start WSRR
ADMU0116I: Tool information is being logged in file
C:\MID\pf\wps\logs\server1\startServer.log
ADMU0128I: Starting tool with the wps profile
ADMU3100I: Reading configuration for server: server1

```

Read ahead until the window indicates the server has started.

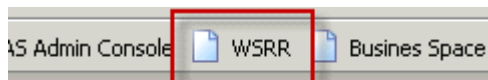
__c. This window will close automatically after startup has completed.

__2. Bring up the WebSphere Service Registry and Repository Graphical User Interface

__a. From the taskbar click the Mozilla Firefox icon to start a browser



__b. Click **WSRR** on the browser's taskbar.



Or type <http://localhost:9080/ServiceRegistry/> into a browser.

__c. Enter **admin** for the user name.

1.5 Explore WebSphere Service Registry and Repository

__1. The WebSphere Service Registry and Repository Web user interface (UI) enables you to use and manage all aspects of WebSphere Service Registry and Repository.

__2. To support an Enterprise's needs, this UI is customizable. This includes the look and feel, access control, roles, groups, action permissions, layout, etc.

__3. After completing this section, you will be more familiar with the configuration capabilities of WebSphere Service Registry and Repository.

1.5.1 Configuration Perspective

__1. Navigate to the Configuration Perspective

- __a. Click the perspective dropdown below the banner and notice that there are a number of different user perspectives defined for this profile.

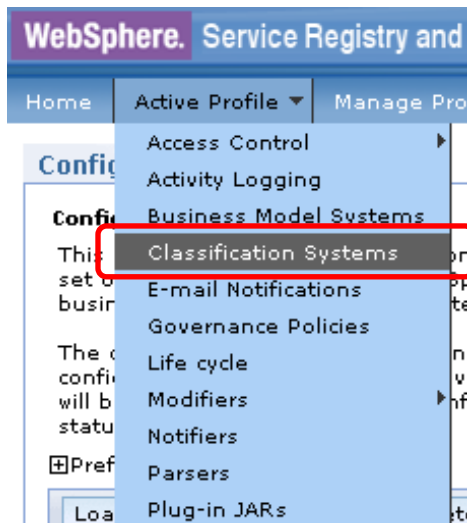


- __b. Select the configuration perspective.
- __2. Enterprises implementing governance of their services need to provide and enforce a common and effective way to name, organize, describe, manage, and share their services, service metadata, and other associated documents.
 - __3. The configuration perspective is only used to define and configure WebSphere Registry and Repository to meet an enterprises unique governance needs. It is not exposed to users of the system.

1.5.2 Classification Systems

- __1. Explore Classifications defined in WebSphere Service Registry and Repository
- __2. Good classification systems are a key factor for successful SOA governance. They make it possible to easily find, reuse, compose, and understand services and related information and artifacts.
- __3. WebSphere Service Registry and Repository enables Web Ontology Language (OWL)-based user-defined semantic classifications. Classification and socialization of high-value assets encourage reuse and interoperability.

- __a. Expand **Active Profile** and click **Classification Systems** in the navigation view.



- __b. In the work area, expand **Governance Profile Taxonomy** and then expand its collections.



- __c. The Business Domain has three children called Finance, Insurance, and Sales and Marketing defined. The Insurance class has two additional children. You can modify the classification systems through this interface, but it is often more convenient to use the WebSphere Service Registry and Repository Studio.



WebSphere Service Registry and Repository Studio

The WebSphere Service Registry and Repository Studio provides tooling for the creation and modification of configuration profiles, including business models, classification systems and life cycles. We will look at WebSphere Service Registry and Repository Studio in a later lab. WebSphere Service Registry and Repository Studio ships with WebSphere Service Registry and Repository

1.5.3 User Perspective

- __1. Switch to the Administrator user role perspective to explore WebSphere Service Registry and Repository.
 - __a. Select the perspective **Administrator** and click. Verify the following administrator homepage is displayed.



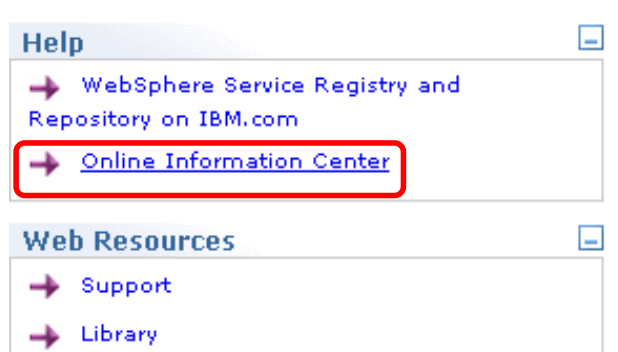
- __b. The layout of the Web User Interface (UI) consists of a banner at the top of the page, below that a set of navigation tabs and underneath that the work area.

Help and Web Resources

As you work through these labs, if you have access to the internet, you can use web based help to gain further understanding.

Click the link “Online Information Center” in the lower right part of the work area. A new tab will open in the browser giving you access to the WebSphere Service Registry and Repository v7.0 Information Center.

Click the original tab opened in the browser to return to the UI.



1.5.4 Service Documents

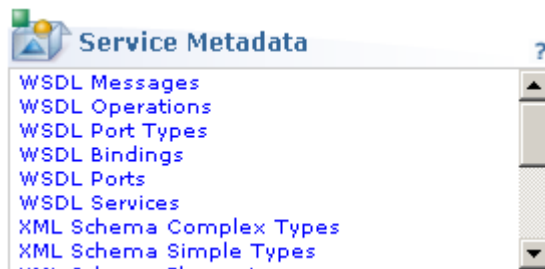
- __1. Explore the Physical Documents – Service Documents
- __2. The most basic building blocks for the WebSphere Service Registry and Repository content model are service metadata artifact documents (physical documents), such as XML Schema Definition (XSD) or Web Service Definition Language (WSDL) files.
- __3. These documents are stored and managed in WebSphere Service Registry and Repository. The coarse-grained model made up from registry objects that represent those documents is referred to as the physical model.
- __4. Documents can be versioned in WebSphere Service Registry and Repository content model, meaning that in addition to a Uniform Resource Identifier (URI), name, and description they also have a version property. Later we will see that WebSphere Service Registry and Repository has the concept of a Service Version which is the implementation of a Business Service and could reference one or more Service Documents.
- __5. Any service metadata artifact type can be stored in WebSphere Service Registry and Repository and receive the benefits of broader visibility and reuse, management, and governance.
 - __a. The collection of **Service Documents** is visible through a window in the workspace.



- __b. This collection of links point to all of the physical documents stored in the registry.
- __c. Below this window, Load Documents provides one of the ways that documents can be loaded into the system.
- __d. Included within Service Documents are **Other Documents** - Documents that are not of a standard type recognized by the registry. For example they may be of type PDF, Microsoft® Word, PowerPoint, text, Visio, Excel, and others.
- __e. Click various links and observe that there are no documents under management at this time. Use the browser's back button to return to the home page.

1.5.5 Service Metadata

- __1. Explore the Service Metadata collection.
- __2. When various types of physical service documents are loaded into the registry, they are parsed and a rich collection of information is generated, stored, and made visible as service metadata.
- __3. The service metadata links in this collection provide access to all of the derived logical objects stored in the registry.
 - __a. Scroll the **Service Metadata** window to view the derived metadata.



- __b. Later in the workshop, you will load documents that will populate these collections.
- __c. Click various links in the Service Metadata collection and observe that no derived information exists at this time.

1.5.6 SOA Governance – Component Models

- __1. Explore the component models of the governance enablement profile.
- __2. Enterprises implementing governance of their services need to provide and enforce a common way to describe their Service Oriented Architecture.
- __3. The component models in the governance enablement profile provide the entities that you need to define services in your SOA. They allow you to apply consistent and meaningful property and relationship definitions across the organization. You can view these entities in the Business Objects window.



In the other perspectives Business Objects are subdivided into:

- Business Governance
- Technical Governance
- SOA Governance
- Service Model

Users have access to different sets of entities according to their role.

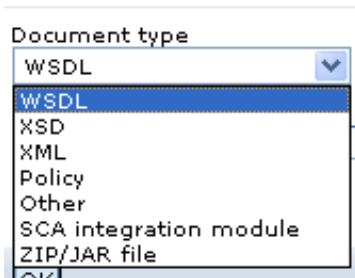
__a. Switch to the **SOA Governance** perspective. Here you have access to all of the entities.



__b. Examine the other perspectives finishing with the Development perspective.

1.6 Publish a document

- __1. In this section, you will publish physical documents into the WebSphere Service Registry and Repository. During this process information about these documents will be discovered automatically and will generate Service Metadata and Service Model entities.
- __2. After completing this section, you will be more familiar with the core “Publish” capabilities of WebSphere Service Registry and Repository.
- __3. Publish sample documents into WebSphere Service Registry and Repository.
- __4. There are a number of different document types that can be published.



- __5. In this lab, you are provided with two WSDL documents and an XML Schema Definition for a Cleanse Data Service.
- __6. Conceptually, Cleanse Data Service cleanses data of any confidential information.
- __7. You will be loading these documents individually. In real life, you are likely to have large collections of physical documents to be loaded e.g. WSDLs, XSDs, WS-Policy Documents etc. Loading zip files with large collections of documents is supported.
- __a. Click the **Home** tab to return to the home page and change perspective to **Development**.
- __b. Scroll to the **Load Documents** section of the work space.

Load Documents ?

This facility enables you to load one or more documents, with the option to save them as a group. Specify a file to load, select a document type and, optionally, enter a description and a version.

Path to the Document

Local file system

Specify path

Remote file location

Specify URL

Document type

WSDL

Enter document description:

Enter document version:

- __c. Browse to C:\student\WSRR\Lab1 and select the document titled **CleanseDataImport1_CleanseDataHttp_Service.wsdl**.
- __d. Click **Open**.
- __e. Enter the following information:

Table 1: CleanseDataImport1_CleanseDataHttp_Service.wsdl

Field	Entry
Enter document description	Cleanse Data http IT WSDL document
Version	1.0

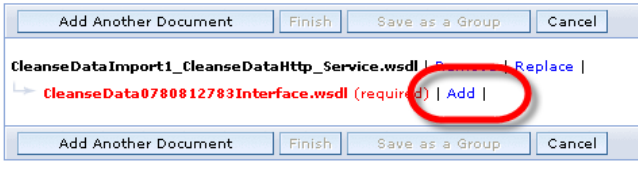
- __f. Click **OK**.
- __g. Dependency relationships between documents are automatically discovered and dependent document(s) are requested.
- __h. Click **Add** next to the required document name CleanseData0780812783Interface.wsdl

Documents to be Loaded

This view shows documents to be loaded and their dependency relationships. Some and these dependencies need to be resolved before you can complete the load:

- filename (required) indicates the document needs to be loaded, or selected from r

When all required documents are listed below select either 'Finish' to complete the l refer to these documents as a document group.

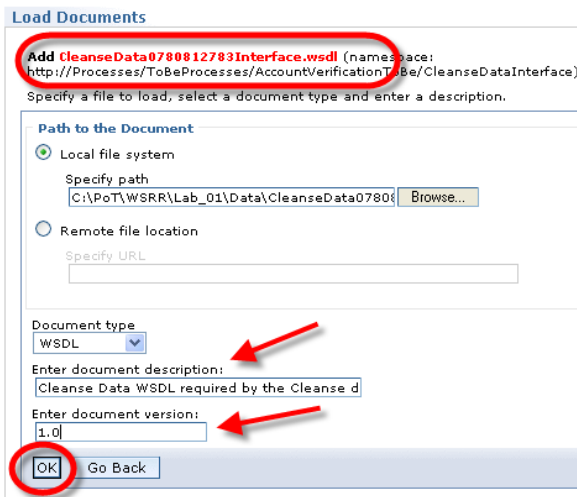


- __i. Click **Browse** and select the document **CleanseData0780812783Interface.wsdl** from the same location. Click **Open**.
- __j. Enter the following information:

Table 2: CleanseData0780812783Interface.wsdl

Field	Entry
Enter document description	Cleanse Data Interface WSDL
Version	1.0

- __k. Click **OK**



- __l. CleanseData0780812783Interface.wsdl requires Businessitems.xsd.

- __m. Add this document to the import collection as well and add the following information:

Table 3: Businessitems.xsd

Field	Entry
Enter document description	Cleanse Data Service Data Model
Version	1.0

- __n. These entries are examples of properties that can be defined as optional or required. In a subsequent lab, you will experience defining and associating additional properties to entities.
- __o. Click **OK**.
- __p. Click **Finish** to parse these documents for metadata placed in the registry and to load them into the repository.

Documents to be Loaded

When all required documents are listed below select either 'Finish' to complete the load or 'S refer to these documents as a document group.

CleanseDataImport1_CleanseDataHttp_Service.wsdl (ready to load) | [Remove](#) | [Replace](#) |

↳ **CleanseData0780812783Interface.wsdl** (ready to load) | [Remove](#) | [Replace](#) |

↳ **Businessitems.xsd** (ready to load) | [Remove](#) | [Replace](#) |



Save as a Group

Document groups allow you to group a number of service documents together for ease of use. This can be done from various collection views as well as during incremental publishing.

- __q. Documents Loaded Successfully will be displayed upon completion of the parsing and loading process.

Documents Loaded Successfully


The following documents have been loaded into the repository:

Name	Description	Namespace	Version
CleanseDataImport1_CleanseDataHttp_Service.wsdl	Cleanse Data http IT WSDL document	http://Processes/ToBeProcesses/AccountVerificationToBe/CleanseDataInterface/Binding	1.0
CleanseData0780812783Interface.wsdl	Cleanse Data WSDL required by the Cleanse data http service	http://Processes/ToBeProcesses/AccountVerificationToBe/CleanseDataInterface	1.0
Businessitems.xsd	Cleanse Data Service Data Model	http://Businessitems	1.0

- __r. Return to Home, and click on **WSDL Documents** and **XSD Documents**, in turn, to verify they have been loaded into the repository.
- __s. Return to the home page.
- __8. Examine the related service metadata information now available in registry.
 - __a. Click the **Services** link in the **Service Model** to examine a service derived from the loaded documents.



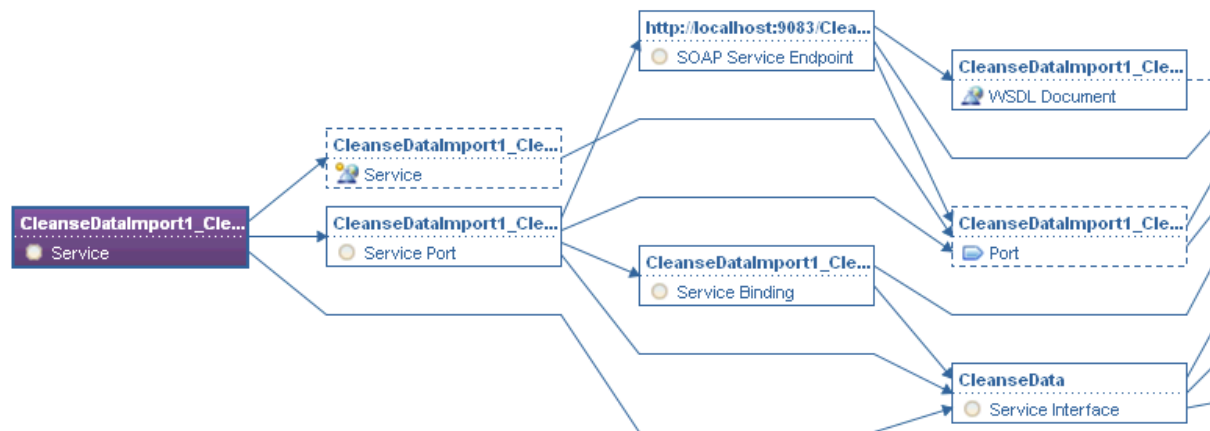
- __b. There is only one service listed, click the **Graph** Icon for **CleanseDataImport1_CleanseDataHttpService**.

Select	Name	Graph
<input type="checkbox"/>	CleanseDataImport1_CleanseDataHttpService	

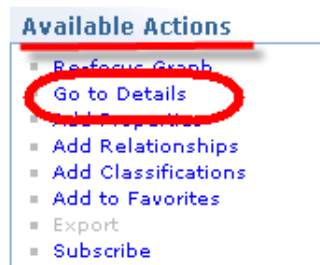
A red arrow points to the 'Graph' icon in the table row.

- __c. A graphical representation of the service concept derived from the WSDL document and its relationship to the other dependant objects and metadata will be generated.

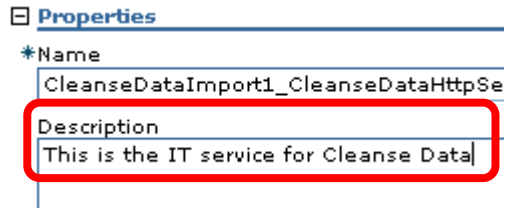
Graph for: CleanseDataImport1_CleanseDataHttpService



- __d. In the graphical view, Service Metadata, which have been derived from the Service documents are shown with dotted lines around them. Objects with solid lines are either Service Model entities such as our Service or Service documents such as the WSDL documents.
- __e. Locate the Viewing Window widget in the upper right corner of this work area.
- __f. Move the viewing area box to explore this service further.
- __g. With the focus still on the service CleanseDataImport1_CleanseDataHttpService, click **Go to Details** in the **Available Actions** box on the right.



- __h. Click **Edit Properties** and then add the following to the **Description** field – “**This is the IT service for Cleanse Data**”.



- __i. Click **OK**.
- __j. Return to Home.
- __k. Examine some of the other entities in the Service Model.

You have completed the publish section of this lab


1.7 Find in WebSphere Service Registry and Repository

- __1. In the previous section of this lab, you used Publish to place documents in the repository. During this portion of the lab you will:
- Search using Query Wizard and Keyword searches using the Web UI
 - Retrieve assets using the Web UI
 - Search using the Eclipse plug-in
 - Retrieve assets using the Eclipse plug-in
 - Published assets from an Eclipse plug-in enabled development platform
 - Added property metadata to your assets
 - Subscribed to assets
 - Performed Impact Analysis

- __2. After you complete this section, you will understand the main Find, and Publish capabilities of WebSphere Service Registry and Repository and be more familiar with it's the Web UI and the WebSphere Service Registry and Repository Eclipse plug-in.
- __3. There are two search mechanisms provided by the Web UI – Query Wizard and Quick search.

1.7.1 WebSphere Service Registry and Repository Query Wizard

- __1. The WebSphere Service Registry and Repository Web UI Query Wizard can search the registry for entities based on a number of criteria:
 - Different specific types of entities
 - All entity types
 - Entity names
 - Custom property name or value
 - Classifications (for example, Test, Development, Production)
 - All entity properties (for example, namespace)
- __2. It supports wild card (*) as well as complex AND and OR queries.



VMware Login after Time Out

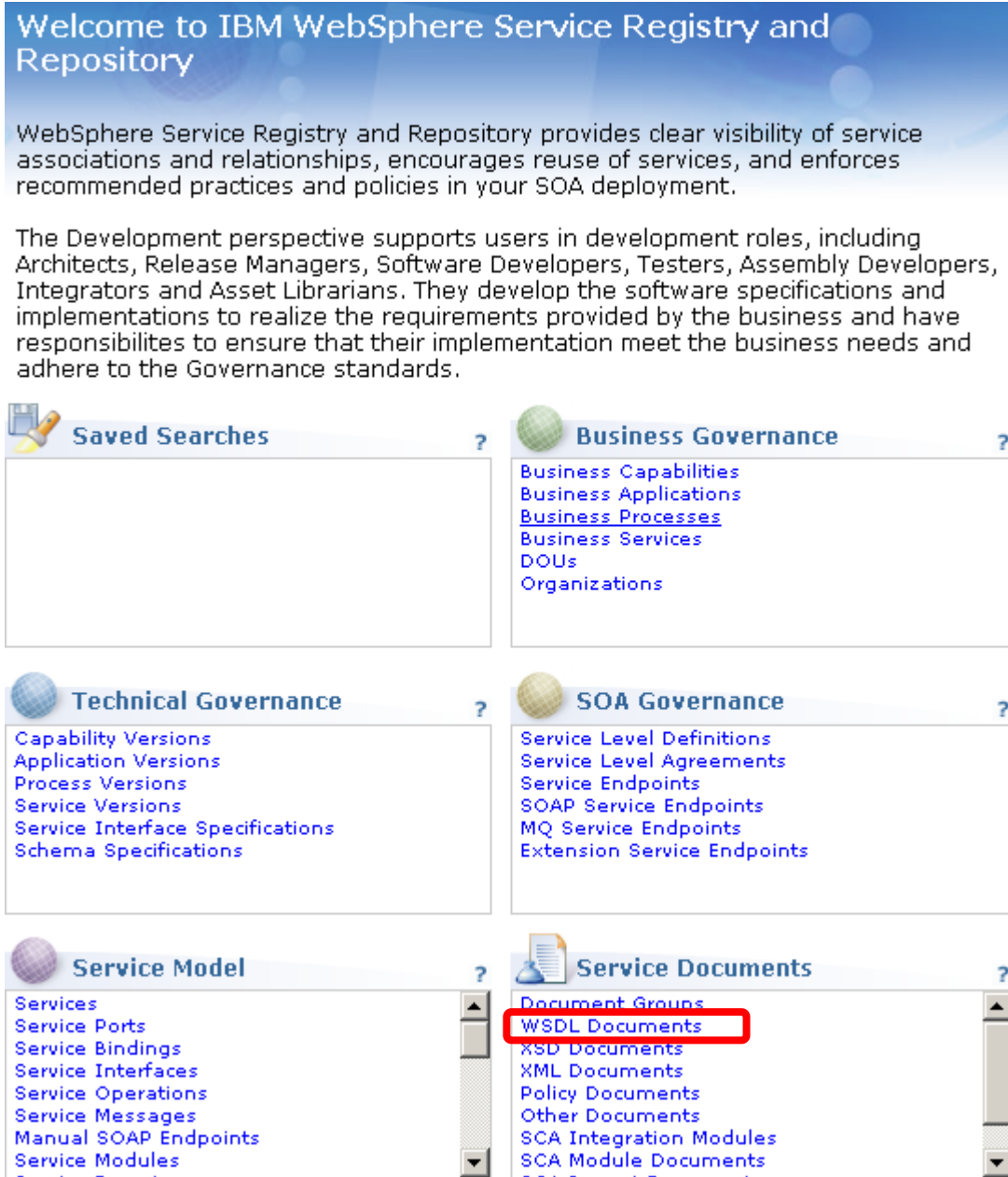
If your virtual machine has timed out you will need to log back on to the system.

Use CTR ALT Insert and then enter the password `passw0rd`.

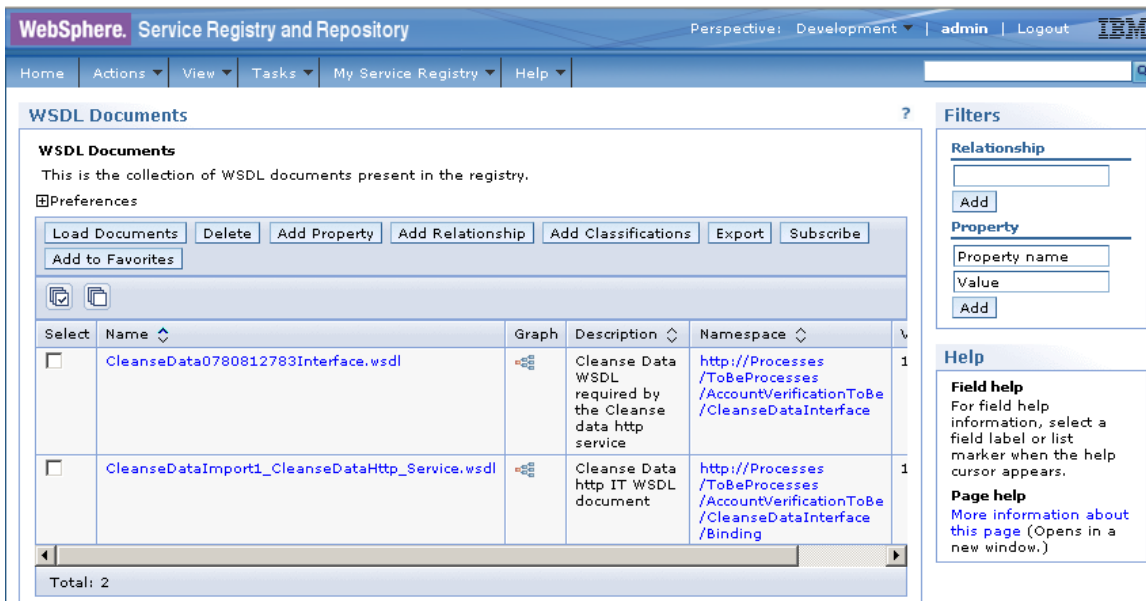
(CTRL ALT Insert will switch you to the host system.)

- __3. Switch to the Development perspective
 - __a. Select **Development** using the Perspective drop-down box.
 - __b. The web session may have expired. If so, log back in as user admin and then perform the previous step.

- __4. Find using the home page links
 - __a. The work area can be useful starting point for search.
 - __b. Locate the Service Documents window and click WSDL Documents.



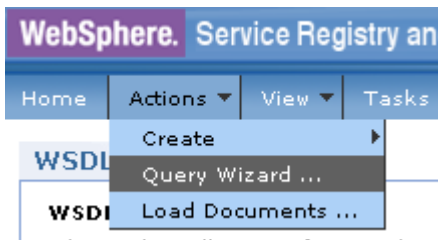
__c. This will return all WSDL documents in the registry – the two we loaded earlier!



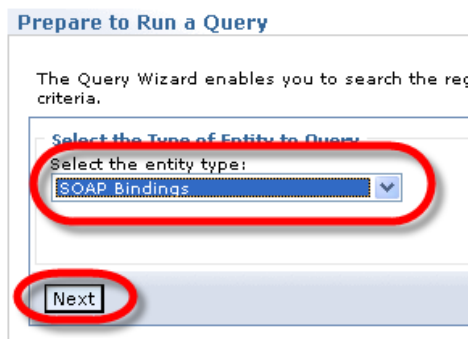
__d. To the right of the workspace, additional filters are also available to help locate assets and metadata.

__5. Find using the Query Wizard

__a. From the **Actions** drop down click **Query Wizard**.



__b. Use the dropdown box to select **SOAP Bindings**.



__c. Click **Next**.

- __d. Enter * (the wild card for all) in the Name field and click **Next** then click **Finish**.
- __e. Available SOAP bindings in the system are listed.
- __f. (There is only one!)
- __g. Return to the Home page.

You have successfully executed a queries to discover / locate assets using the WebSphere Service Registry and Repository Web UI query wizard.

Currently, the repository contains very little data. As this Proof of Technology progresses, it will contain far more objects. You might like to exercise the query wizard again later in the day.

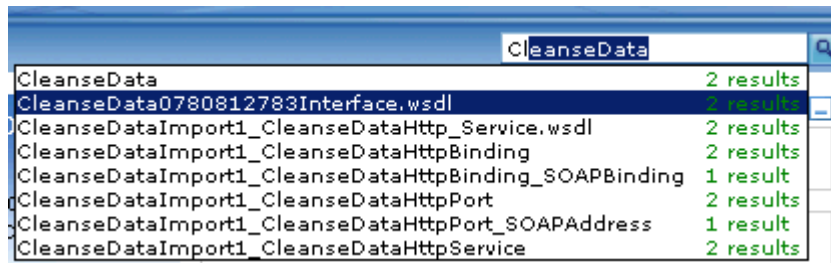
1.7.2 Quick Search

- __1. WebSphere Service Registry and Repository also has a quick search capability based on AJAX (Asynchronous JavaScript™ and XML) Web 2.0 technology. The quick search panel at the top right hand corner of all pages can be used to find any type of object stored in the registry.
- __2. The search is performed on the entire registry by using an exact-case match on the text given and will be matched against the name and description properties of any object. The text given will be treated as a partial match so, for example, a search for "Echo" might return "Echo.wsdl" as well as "MyEcho" as part of the results. It is also possible to specify an asterisk (*) in any part of the search text to denote that any number of other characters are permitted to occur at that position in the search results. This is equivalent to the regular expression ".*" term.
- __3. The WebSphere Service Registry and Repository Web UI provides an optional Auto-Suggest feature. If you enable this feature, then, when you enter text in the quick search field, you will be presented with a list of matching suggestions from which you can select your search string. The list consists of all the objects in WebSphere Service Registry and Repository with a name that begins with the text you have typed; the text matching is not case-sensitive. You can enable Auto-Suggest, and set a range of Auto-Suggest options, in My Service Registry > My Preferences.
- __4. Note: If Auto-Suggest is enabled, the search is performed by using an exact match on the search text, not a partial match.

For this Lab, Auto-Suggest is enabled. If you wish to switch it off: My Service Registry > My Preferences, un-check Enable Auto-Suggest, and click Apply.

- __5. Find using quick search
 - __a. In the upper right of the page, in the Search field, enter c1 (as if you were typing cleanse).

Notice that a list of possible items is displayed.



- __b. Select any item you are interested in from the suggestions provided. It will appear in a Query results window.

1.7.3 Retrieve a document

- __1. You now are able to search for and “Find” entities that are under the WebSphere Service Registry and Repository management.
- __2. Next, you will locate a WSDL document in this system of record, examine it, and then download it so that you can reuse it, theoretically, as an application developer implementing a SOA solution with maximum reusability.
- __3. Locate the WSDL document - **CleanseDataImport1_CleanseDataHttp_Service.wsdl**
- __a. Find this document using one of the methods you used previously.

You might use a search or possibly just go to the physical documents collection under Service Documents and open the WSDL Documents collection.

- __b. Click the **hyperlink** for this document.
- __c. Review the information on the **Details** tab and then click the **Content** tab.

WSDL Document

Query Results > CleanseDataImport1_CleanseDataHttp_Service.wsdl

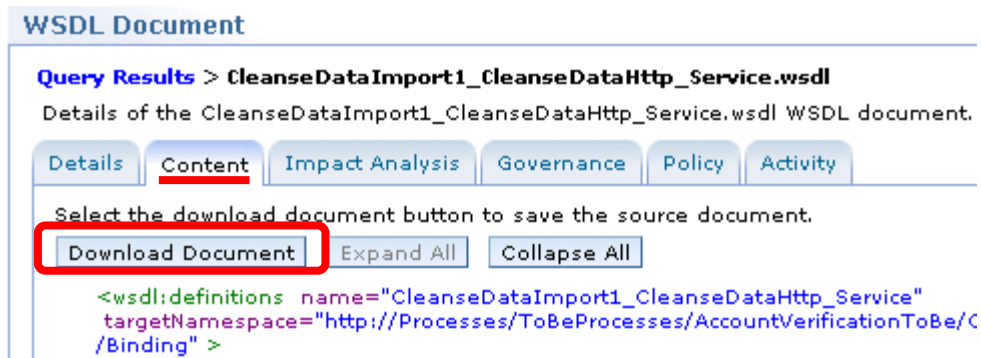
Details of the CleanseDataImport1_CleanseDataHttp_Service.wsdl WSDL doc

Details **Content** Impact Analysis Governance Policy Activity

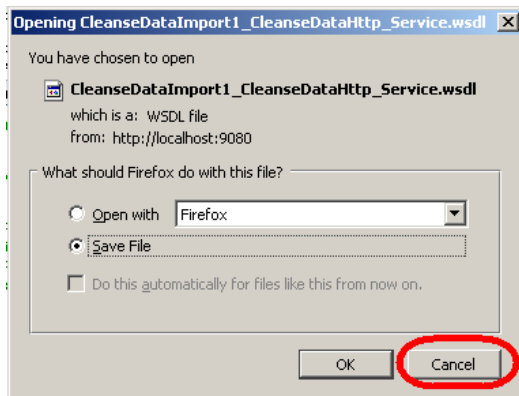
Edit Properties | Edit

General Properties	Links
Name CleanseDataImport1_CleanseDataHttp_Servi	<ul style="list-style-type: none"> Graphical View Applied Policy Applied Policy
Location CleanseDataImport1_CleanseDataHttp_Servi	
Description Cleanse Data http IT WSDL document	Service Metadata Port Types None

- __d. Examine the document to determine it is the one you, the Development user, really want.
 - __e. You might also want to view this document graphically to understand it and evaluate it for reuse more closely.
 - __f. Click the Details tab and then click the Graphical View link.
- __4. Download the document.
- __a. Click the browser back button or the **Go to Details** link to return to the Details tab.
 - __b. Click the **Content** tab and click **Download Document**.



- __c. The File Download pop-up window will appear.
- __d. With Save file selected, click **OK** to retrieve the file. Alternatively, click **Cancel**.



- __5. Thus far you have:
- Searched WebSphere Service Registry and Repository using your home page, Query Wizard, and Quick searches
 - Found and examined files of interest
 - Downloaded a WSDL document

- __a. Logout of WebSphere Service Registry and Repository
- __6. In the next section, you will be using the WebSphere Service Registry and Repository Eclipse plug-in. This plug-in will retrieve related documents as a collection. This capability addresses the needs of an IT participant and supports any tooling that is Eclipse based.



Microsoft Visual Studio support

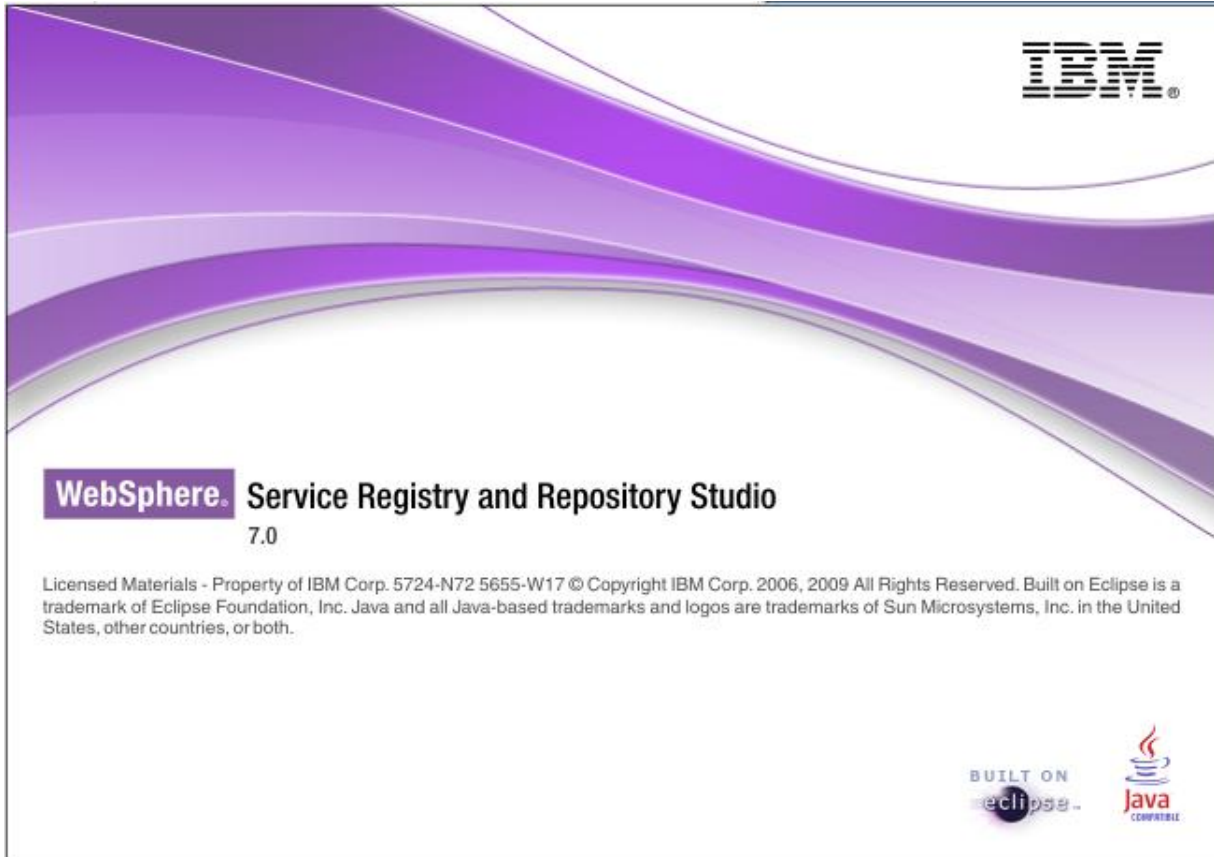
Microsoft Visual Studio is supported via the SA11 SupportPac.

As well as retrieving documents from WebSphere Service Registry and Repository, documents that have been created within a Microsoft Visual Studio Solution project can be published into WebSphere Service Registry and Repository along with some associated metadata.

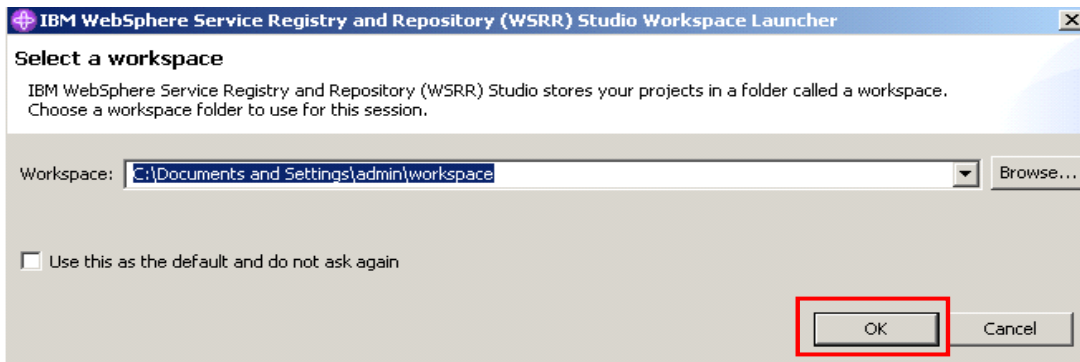
1.8 Experience the WebSphere Service Registry and Repository Eclipse plug-in

- __1. The WebSphere Service Registry and Repository includes an Eclipse plug-in. This adds a new view, the Service Registry view, to any Eclipse workbench environment (for example, IBM Rational® Software Architect, WebSphere Integration Developer, etc.).
- __2. With this Service Registry view you can access, search for, find, and download objects that are maintained in WebSphere Service Registry and Repository – your SOA system of record. 😊 Additionally, you can also publish to WebSphere Service Registry and Repository new objects that you create in any Eclipse workbench that has this plug-in installed.
- __3. In this section you will be using the Eclipse environment provided by WebSphere Service Registry and Repository Studio
- __4. Start WebSphere Service Registry and Repository Studio
 - __a. Open the WebSphere Service Registry and Repository PoT Shortcuts folder on the desktop
 - __b. Click the shortcut **WSRR Studio** to start the studio.

The splash screen will appear.

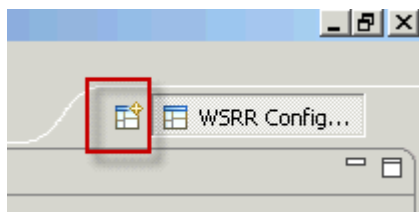


__c. Click **OK** to accept the default workspace

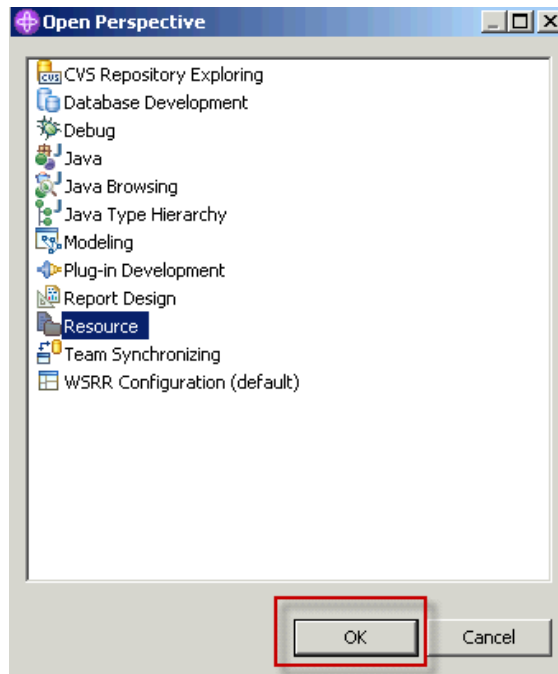


__5. Switch to the Resource perspective.

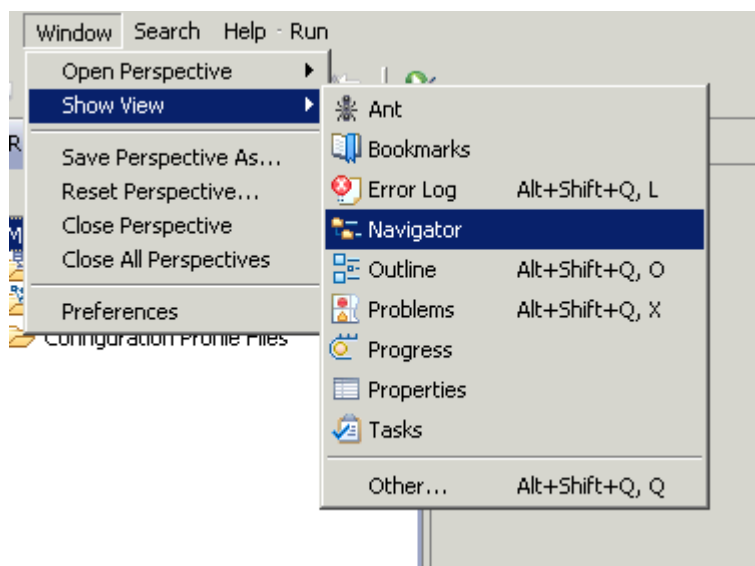
__a. Click the Perspective icon



- __b. Select **other**
- __c. When the Open Perspective dialogue appears, select **Resource** and click **OK**.

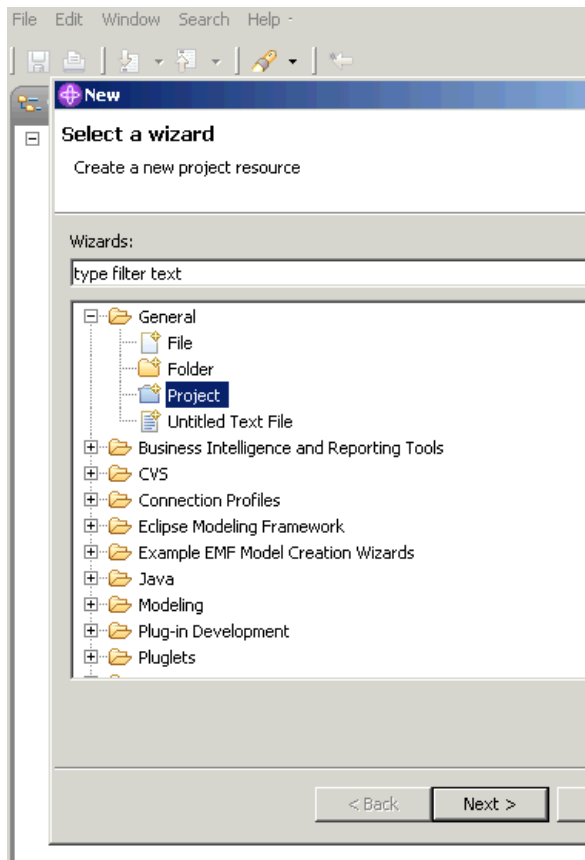


- __6. Open a Navigator view
 - __a. Click **Window -> Show View -> Navigator**



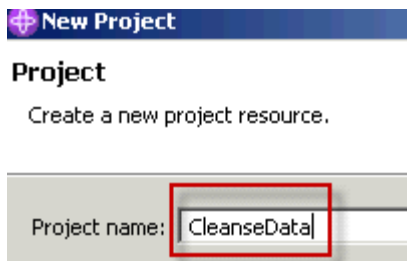
__7. Create a new project.

__a. Click **File -> New -> Other -> General -> Project**



__b. Click **Next**

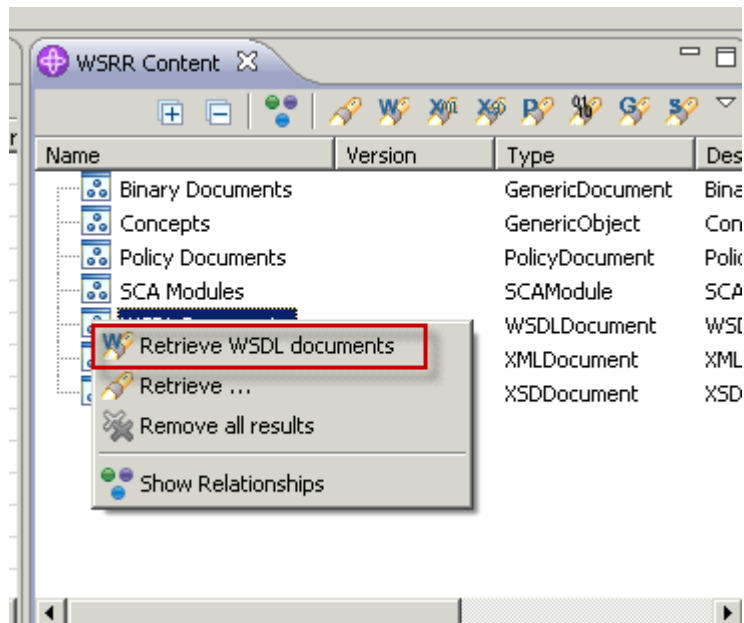
__c. Enter **CleanseData** for the Project Name.



__d. Click **Finish**.

__8. Load a WSDL document

- __a. In the WebSphere Service Registry and Repository Content view, right-click **WSDL documents**, then click **Retrieve WSDL Documents**.

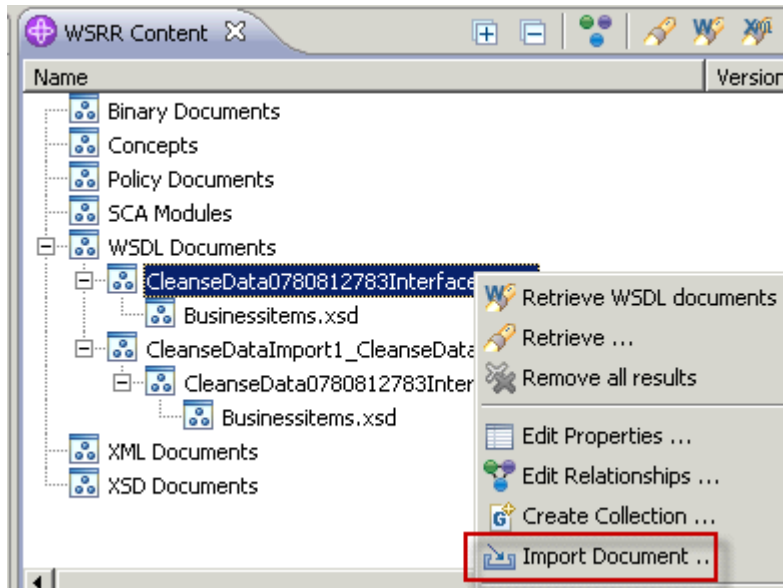


- __b. Expand **WSDL Documents** and then expand each sub-collection.

WSDL Documents		WSDLDocument
CleanseData0780e 1.0		WSDLDocument
Businessitems. 1.0		XSDDocument
CleanseDataImpor 1.0		WSDLDocument
CleanseData0: 1.0		WSDLDocument
Businessite 1.0		XSDDocument

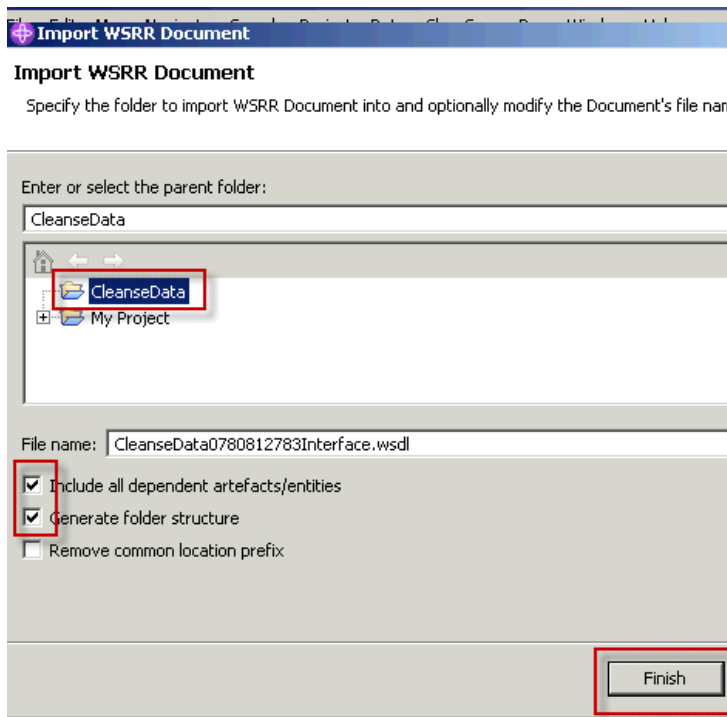
- __c. The plug-in has retrieved information about all dependant artifacts associated with the WSDL files you selected for retrieval.

- d. Right click **CleanseData0780812783Interface.wsdl** and select **Import Document**. (You may need to expand the Name column.)



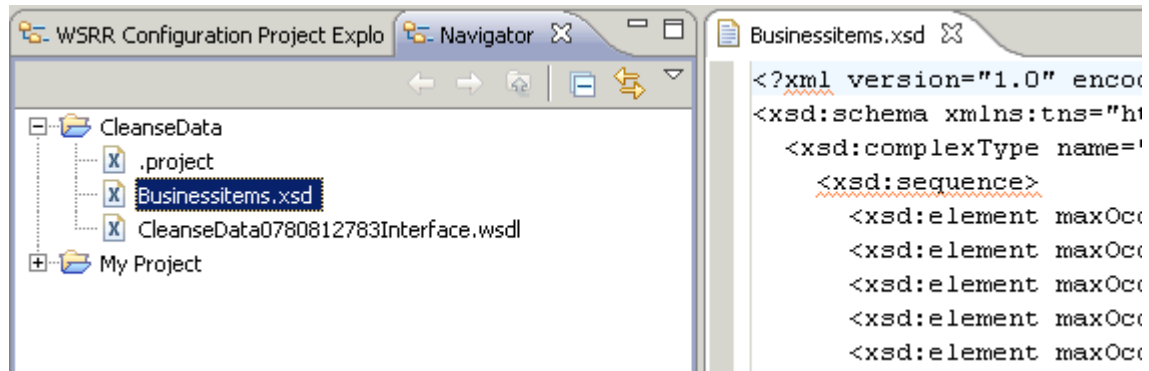
This will begin the actual import of these documents into the WebSphere Integration Developer workspace.

- e. Select the **CleanseData** folder; select the option boxes for **Include all dependent artifacts / entities** and **Generate folder structure**, and click **Finish**.



The WSDL and associated XSD documents are loaded.

- __f. In the Navigator view, expand the **CleanseData** project and right-click on **Businessitems.xsd**. Select **Open with -> Text editor**.



The contents of Businessitems.xsd are displayed.

- __g. You have successfully loaded a WSDL and associated XSD document into an Eclipse workspace.
- __h. It is also possible to publish to WebSphere Service Registry and Repository from Eclipse. Additionally, WebSphere Service Registry and Repository Studio provides a range of content management features for working with WebSphere Service Registry and Repository content.
- __9. Close WebSphere Service Registry and Repository Studio
- __a. Click **File -> Exit**

1.9 Working with Metadata

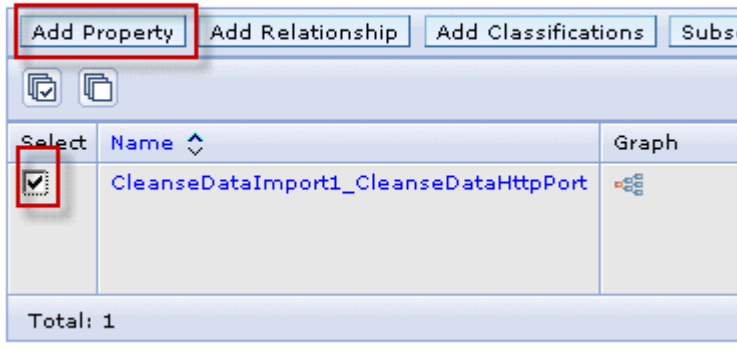
- __1. Metadata includes properties, relationships, and classifications. Authenticated users with appropriate permissions can use the Web UI to add, update, or delete metadata.
- __2. Metadata can be added to a collection of entities by using the Add Property, Add Relationship, or Add classifications buttons in the collection view.
- __3. Add metadata to your assets
- __4. Some of the properties like name, namespace, version etc. are predefined. Customers can also define their own properties. Properties are basically name/value pairs.
- __a. Log on to WebSphere Service Registry and Repository and change to the Administrator perspective.
- __b. In the Service Metadata window click **WSDLPorts**.

- __c. Select the check box for **CleanseDataImport1_CleanseDataHttpPort**, and click **Add Property**.

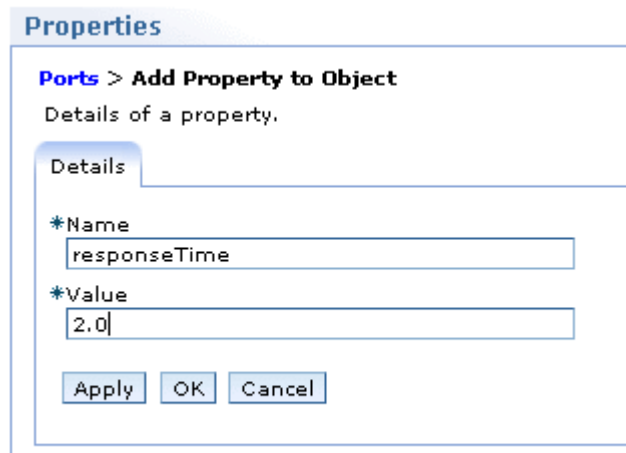
Ports

This is the collection of WSDL Ports present in the registry.

Preferences



- __d. Complete the panel as shown.



- __e. Click **OK**.

You just added a new user-defined property – responseTime. (This and other capabilities can be restricted as needed by customizing the UI access and permissions.)

- __f. To verify the addition of this property click **CleanseDataImport1_CleanseDataHttpPort**, and expand the **Additional Properties** collection in the work area.



- __g. Return to the Home page.

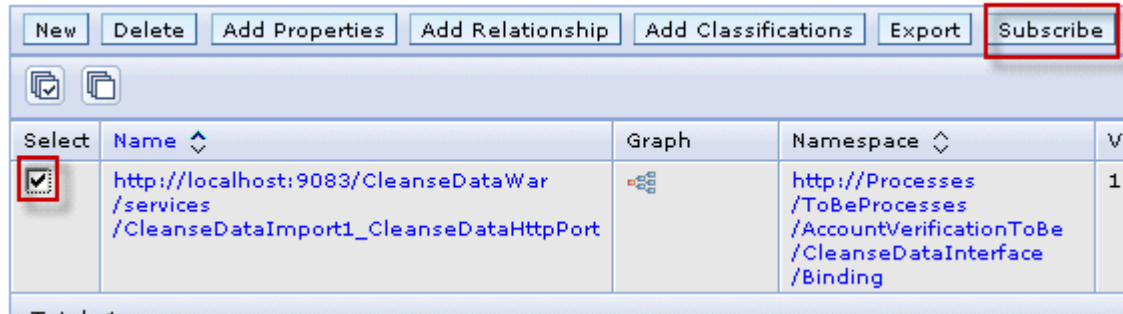
1.10 Subscription & Notification

- __1. WebSphere Service Registry and Repository provides an extensible notifications framework that allows events to generate notifications in a wide variety of ways. A built-in email notification mechanism is included as well.
- __2. Users can subscribe to entities they are interested in. They can then be notified of changes or proposed changes to these entities of interest.
- __3. Users often want to subscribe to certain events on certain artifacts.
- __4. For example, one could be notified if an entity is updated, deleted, or goes through a transition such as an endpoint changing from offline to online.
- __5. Notification supports the following operations:
 - create
 - update
 - delete
 - validate
 - make governable
 - remove governance
 - transition

Subscribe to a SOAP Service Endpoint.

- __6. You will create a subscription to the SOAP Service Endpoint object that was created when the CleanseDataImport1_CleanseDataHttp_Service.wsdl was loaded into WebSphere Service Registry and Repository, and configure the subscription to notify you via e-mail when an update is made to the concept.

- __a. Change to the Operations perspective.
- __b. In the **SOA Governance** window, click **SOAP Service Endpoints**.
- __c. Select the check box for **http://localhost:9083/CleanseDataWar/services/CleanseDataImport1_CleanseDataHttpPort** and click **Subscribe**.



- __d. Fill out the **Create a Subscription** form using the following information:

Article I.	Field	Article II.	Entry
Article III.	Name	Article IV.	CleanseData SOAP Service Endpoint
Article V.	Description	Article VI.	Subscribe to Online/Offline transitions
Article VII.	Owner	Article VIII.	(can't change this)
Article IX.	Destination address	Article X.	mailto:ac@JKHLEnterprises.com
Article XI.	Operations	Article XII.	Transition
Article XIII.	Transitions	Article XIV.	ApproveForUse
		Article XV.	RevokeFromUse

*Name
CleanseData SOAP Service Endpoint

Description
Subscribe to Online/Offline transitions

Owner
admin

Destination address type
Email

*Destination address
mailto:ac@JKHLEnterprises.com

Entity list
http://localhost:9083/CleanseDataWar/s<

Operations

Create

Update

Delete

Validate

Make governable

Remove governance

Transition

Transitions

Revise SLA Request

Revise Specification

Revise Staging Deployment

Revoke

Revoke From Use

Preferred notification message language
English

Apply OK Cancel

__e. Click **OK**.

An email server and account have not been setup, so you can't test this. If you could, an email would be sent whenever the endpoint transitioned from Online to Offline or in the reverse direction.

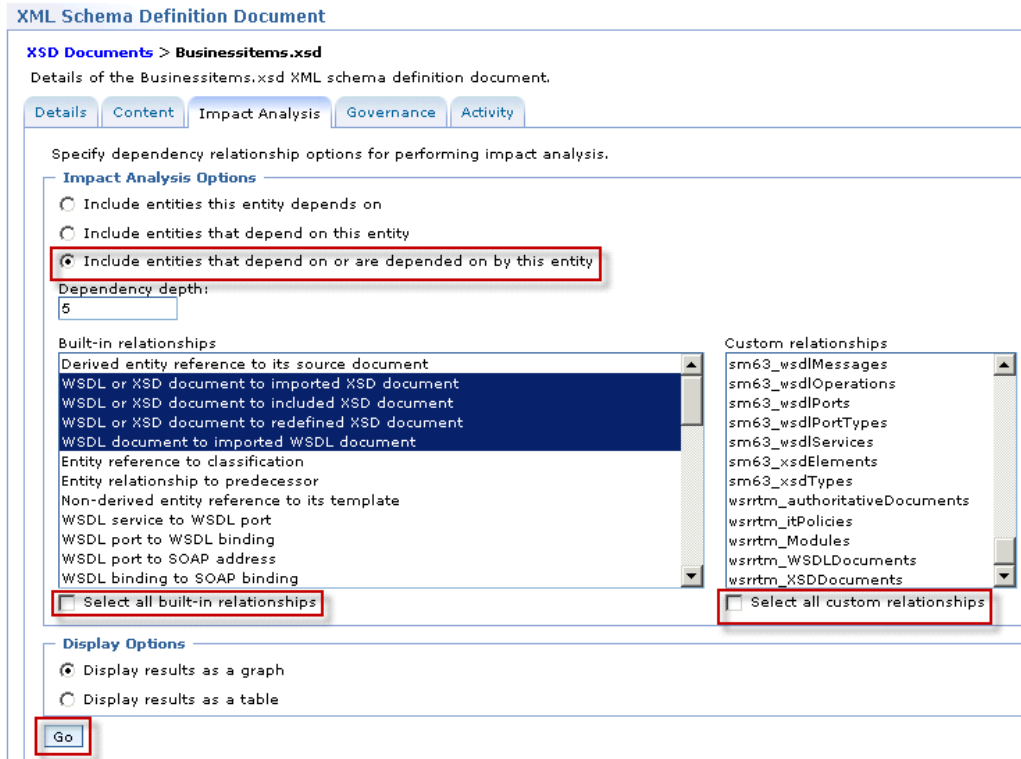
1.11 Impact Analysis

- __1. It is important to be able to identify the impact of a potential change.
- __2. WebSphere Service Registry and Repository supports this. Impact analysis can be used to identify entities that depend on a particular entity as well as entities that entity is dependant on. “Dependency depth” can be used to scope impact analysis appropriately.
- __3. Impact analysis uses relationships specified in the system. They can be built-in relationships or user-defined relationships. Impact analysis can be restricted to specific types of relationships as well.
- __4. For example, someone would like to change the data model, Businessitems.xsd. They would like to know which assets might be impacted, the owner of those assets, etc.
- __5. Perform Impact Analysis
 - __a. Select the Development perspective.
 - __b. In the **Service Documents** window click **XSD Documents**, and then click the link for **Businessitems.xsd**.
 - __c. Click the **Impact Analysis** tab and verify the radio button for **Include entities that depend on this entity or are depended on by this entity** is selected.

Unselect **Select all built-in relationships** and **Select all custom relationships**.

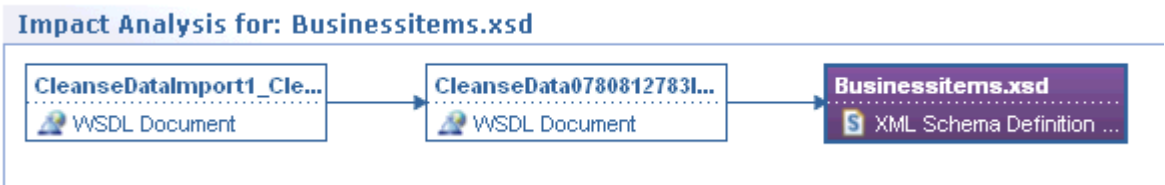
Select the following four Built-in relationships by holding down the ctrl key and selecting the following:

- WSDL or XSD document to imported XSD document
- WSDL or XSD document to included XSD document
- WSDL or XSD document to redefined XSD document
- WSDL document to imported WSDL document

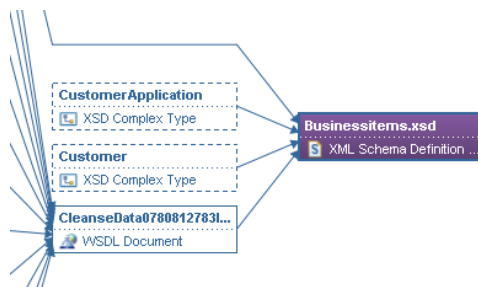


___f. Click Go

A graph is generated that indicates the dependencies and thus the potential impact of changes to this entity.



___d. Optionally use the browser back button to return to the Impact Analysis Options. All built-in and custom relationships are selected by default. Click **GO** to see the entire dependency tree.



You have completed your hands-on investigation of some of the main Publish and Find capabilities provided by WebSphere Service Registry and Repository.

__e. Logout of WebSphere Service Registry and Repository.

1.12 Lab Review

During this lab, you:

- Became familiar with the WebSphere Service Registry and Repository Web UI and its navigation
- Explored the Configuration perspective
- Published service documents (WSDL and XSD)
- Searched using Query Wizard and Quick searches using the Web UI
- Retrieved assets using the Web UI
- Searched using the Eclipse plug-in
- Retrieved assets using the Eclipse plug-in
- Added property metadata to your assets
- Subscribed to a service endpoint.
- Performed impact analysis

__1. In summary, WebSphere Service Registry and Repository provides, among others, a number of capabilities for publishing, finding, and managing services. It provides support for keyword as well as wizard based searches. It allows users to add the metadata i.e. properties, relationships, and classifications, to entities. It also provides support for impact analysis, subscription, and notification.



CONGRATULATIONS!

You have successfully completed Lab 1 - Publish and Find.

Lab 2 Creating an Organizational Structure

2.1 Lab introduction

In this lab you will create an organization structure to which we will associate services in later labs.

When multiple composite applications use a service, it is vital for effective governance to determine who is responsible for that service.

Often an enterprise organizes its staff reporting structure and finances around business operations. The department that is responsible for certain business operations might also be responsible for the development of services and the runtime IT for these operations. In a service-oriented architecture (SOA), however, another organization might be responsible for the development of the services and runtime IT for given operations. Therefore, services and composite applications in an SOA often do not follow an enterprise's strict hierarchical reporting and financial structure, creating gaps and overlap in IT responsibilities.

You can use an organizational structure defined in WebSphere Service Registry and Repository to assign ownership of the SOA services that are governed. WebSphere Service Registry and Repository also supports identification of those departments within the organization that subscribe to or use specific services.

We will create the following organizations:

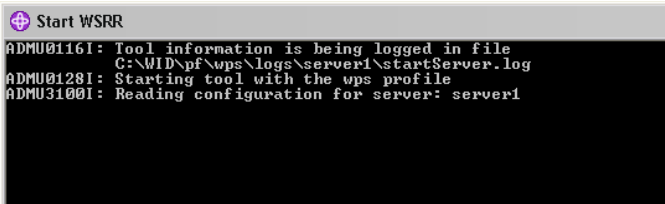
JKHL Enterprises, a top-level organization that represents the complete enterprise.

Common services, a child organization of JKHL Enterprises that represents the departmental team that is responsible for the development and delivery of services that are shared throughout the enterprise.

Commercial, an organization that represents the commercial line of business (LOB).

2.2 Start the environment

- ___1. If necessary, start WebSphere Service Registry and Repository
 - ___a. Open the **WSRR PoT Shortcuts** folder on the desktop
 - ___b. Click the shortcut **Start WSRR Server** to start WebSphere Service Registry and Repository Server that is hosted in WebSphere Application Server.
 - ___c. This will bring up a window. The window will display startup information and remain open after the startup has completed.



```
Start WSRR
ADMU0116I: Tool information is being logged in file
           G:\MID\pf\wps\logs\server1\startServer.log
ADMU0128I: Starting tool with the wps profile
ADMU3100I: Reading configuration for server: server1
```

Read ahead until the window indicates the server has started.

__d. After startup has completed, close this window with any key.

__2. Bring up the WebSphere Service Registry and Repository Graphical User Interface

__a. Click the Mozilla browser icon.

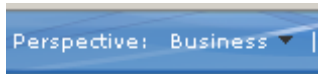


__b. Click the **WSRR Console** link on the Bookmarks Toolbar.

Or type <https://localhost:9080/ServiceRegistry/> into the address.

__c. Enter **Bob** for the user name.

__d. Select the **Business** perspective from the **Perspective** menu drop down list.



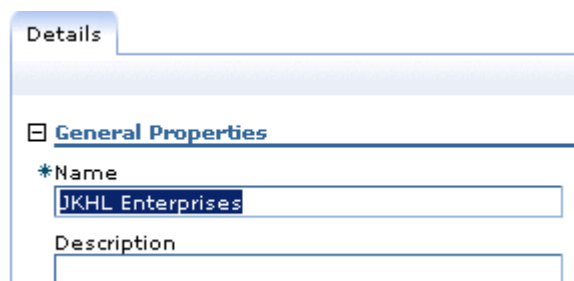
__3. Create an organization entity.

An Organization represents an organizational unit in a business to which ownership of services can be attributed. For example, Accounting, Procurement, Customer Service, etc.

__a. Click Organizations in the Business Governance window.

__b. Click **New**.

__c. In the Details tab, enter JKHL Enterprises in the Name field



You can complete the other fields if you wish.

__d. Click **Finish** to create this new organization.

__4. Adding child organizations

__a. Click **Edit Relationships**

Details | Impact Analysis | Governance | Policy | Activity

Edit Properties | **Edit Relationships** | Edit Classifications

Properties

- *Name: JKHL Enterprises

Links

- Graphical View
- Applied Policies
- Applied Policy Attachments

__b. Click **Add Organization**

Organization > JKHL Enterprises > Edit Relationships

The Organization 'JKHL Enterprises' has the following relationships and targets that will be applied.

Finish | Cancel

Targets
JKHL Enterprises Add Relationship

Child Organizations | **Add Organization**

__c. Click **Create** to create a new entity

Add Target

Use the options below

Search for Existing

Name: _____

Entity Type: Organization

Search | Add

Create New Entity

Entity type: Organization

Create

__d. Enter Common Services in the Name field, and click OK.



Common services is added as a child organization of JKHL Enterprises.

__e. Repeat from step b above to add the Commercial organization.

__f. Click **Finish** to save your changes.

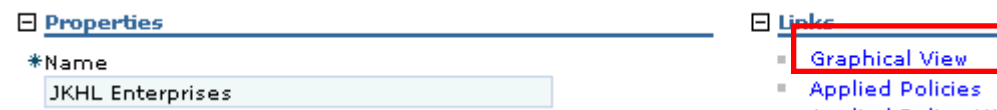


__5. View the JKHL Enterprises organization.

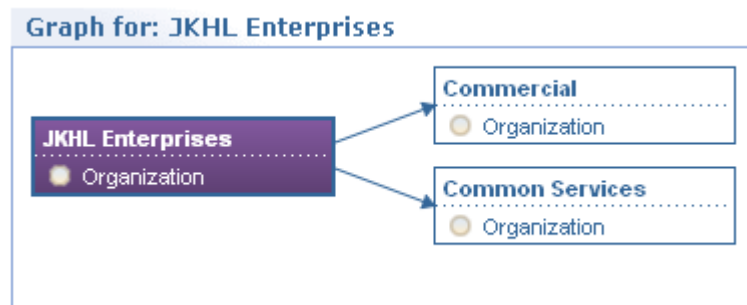
__a. Note that it is governed and has been classified



__b. Click on **Graphical View**



The organizational structure is displayed.



__c. Return to the home page.

You have completed the lab.

2.3 Lab Review

During this lab, you:

Created an organizational structure.



CONGRATULATIONS!

You have successfully completed Lab 2 - Getting Started with WebSphere Service Registry and Repository.

Lab 3 Governing a Schema

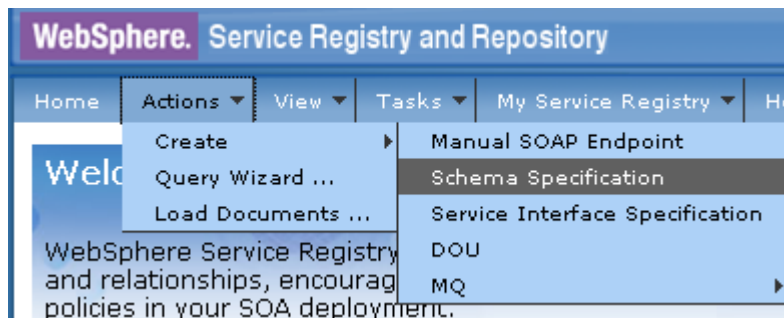
3.1 Lab introduction

At JKHL Enterprises (JKHLE), a schema specification is required to describe the business objects that define customer and account records. This schema is intended to be a re-usable asset. For example JKHLE wants to standardize the way customer information, including postal addresses, are defined. This standardization aids interoperability between services and operations. If a specific business operation requires more information than is already described in the schema, that operation can define its own schema that extends this schema, or the business can request a new version of the schema.

3.2 Create the schema specification

At JKHLE, creation of the schema specification is carried out by development.

- __1. Create the schema specification
 - __a. Log on to WebSphere Service Registry and Repository as **Connie**, and select the **Development** perspective.
 - __b. Click **Actions** → **Create** → **Schema Specification**



This brings up a New Schema Specification screen.

- __c. Enter the following property values:
 - Name: JKHLE global schema
 - Description: The schema definition for customer and account records.
 - Namespace: <http://jkhle.itso.ibm.com/Account>
 - Version: 1.0
 - Requirements Link: <http://requirements.jkhle.com/requirements.jsp&id=5682>

(The Requirements Link, in this case fictitious, references the relevant item in JKHLE's requirements tracking tool.)



Later in the lab, WebSphere Service Registry and Repository will check the contents of the Namespace field. If you wish, you can leave it blank for now and see what happens!

- __2. Assign an owning organization to the schema specification
- __a. Click **Add Organization** in the Targets columns to the right of the Owning Organization relationship

Targets	
JKHLE global schema Add Relationship	
↳ Depends On Schema	Add Schema Specification
↳ Specified Schemas	Add Schema
↳ Dependency	Add Asset
↳ Owning Organization	Add Organization

- __b. Enter C in the Name field, select **Common Services** from the auto suggest list, and click **Add**.

Add Target

Use the options below to select a new

Search for Existing Entity

Name

Commercial

Commercial 1 result

Common Services 2 result

Organization

Search Add

The Common services organization is added as a target of the Owning organization relationship.

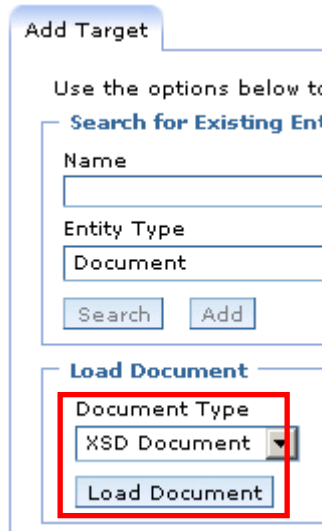
- __c. Click **Finish** to save the schema specification. The details page for the schema specification displays.

Note that the new schema specification is automatically governed and has been entered into the Asset life cycle with the initial state of Asset Identified.

- __3. Add the schema XSD file
 - __a. Click **Edit Relationships**
 - __b. Click **Add Document** to the right of the Artifacts relationship

JKHLE global schema Add Relationship	Targets
↳ Depends On Schema	Add Schema Specifici
↳ Specified Schemas	Add Schema
↳ Dependency	Add Asset
↳ Owning Organization	Common Services
↳ Artifacts	Add Document

- ___c. In the Load Document window, select **XSD Document** from the Document Type list, and click **Load Document**



The screenshot shows the 'Add Target' dialog box. It has two main sections: 'Search for Existing Ent' and 'Load Document'. The 'Load Document' section is highlighted with a red box. It contains a 'Document Type' dropdown menu with 'XSD Document' selected, and a 'Load Document' button below it.

- ___d. Click **Browse** and navigate to **C:\student\WSRR\Lab3** where the XSD file is located. Select the **JKHLEGlobalSchema.xsd** file, and click **Open**.
- ___e. Enter the document version as 1.0 and click **OK**.
- ___f. Click **Finish** to add the XSD. Then, click **Finish** again to save these changes to the schema specification.



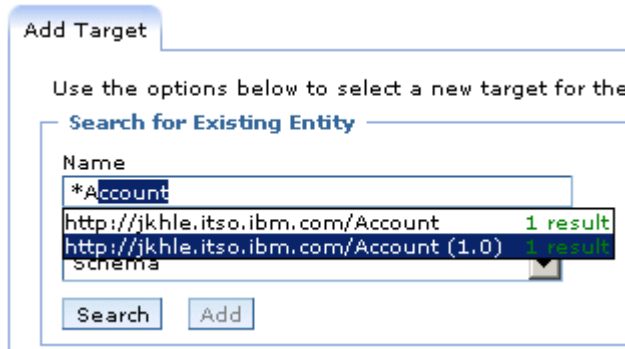
You are required to save your changes at this stage because the schema artifact is created only after the schema specification document is committed to the registry.

- ___g. Add the service schema to the schema specification



The service schema was automatically generated when the XSD file was loaded. It is an object in the service model and takes the namespace of the XSD file as its name.

- __h. Click **Edit Relationships**
- __i. Click **Add Schema**, and enter *Account* in the Name field.



Select <http://jkhle.itso.ibm.com/Account> (1.0) from the auto suggest list, and click **Add**

- __j. Click **Finish**.

3.3 Proposing the schema specification

- __1. At this stage of, the scope and requirements for the schema specification have been agreed, and an XML schema document describing the proposed schema has been loaded into WebSphere Service Registry and Repository.

- __2. Transition the schema specification to the Asset Scope Review governance state
- __a. Click **Propose** to transition the schema

Schema Specification

Schema Specifications > JKHLE global schema

Detail view for Schema Specification. A Schema Specification Asset defines shared schema documents (XSD) all governed independently.

Details | Impact Analysis | **Governance** | Policy | Activity

[Edit Properties](#) | [Edit Relationships](#) |

Properties

*Name
JKHLE global schema

Description
The schema definition for customer and account records

Namespace
<http://jkhle.itso.ibm.com/Account>

Version
1.0

Asset Web Link
<urn:serviceregistry>

Remote State

Owner Email

Additional Properties

[Back](#) [Propose](#)

Links

- [Graphical View](#)
- [Applied Policies](#)
- [Applied Policy Attachments](#)

Relationships

Specified Schemas
[http://jkhle.itso.ibm.com/Account \(1.0\)](http://jkhle.itso.ibm.com/Account (1.0))

Depends on Schema
None

Owning Organization
[Common Services](#)

Dependency
None

Artifacts
[JKHLEGlobalSchema.xsd](#)

Governance State
Asset Identified

Classifications
Asset Identified
Schema Specification

The governance state for the schema specification is now Asset Scope Review.

Governance State
Asset Scope Review

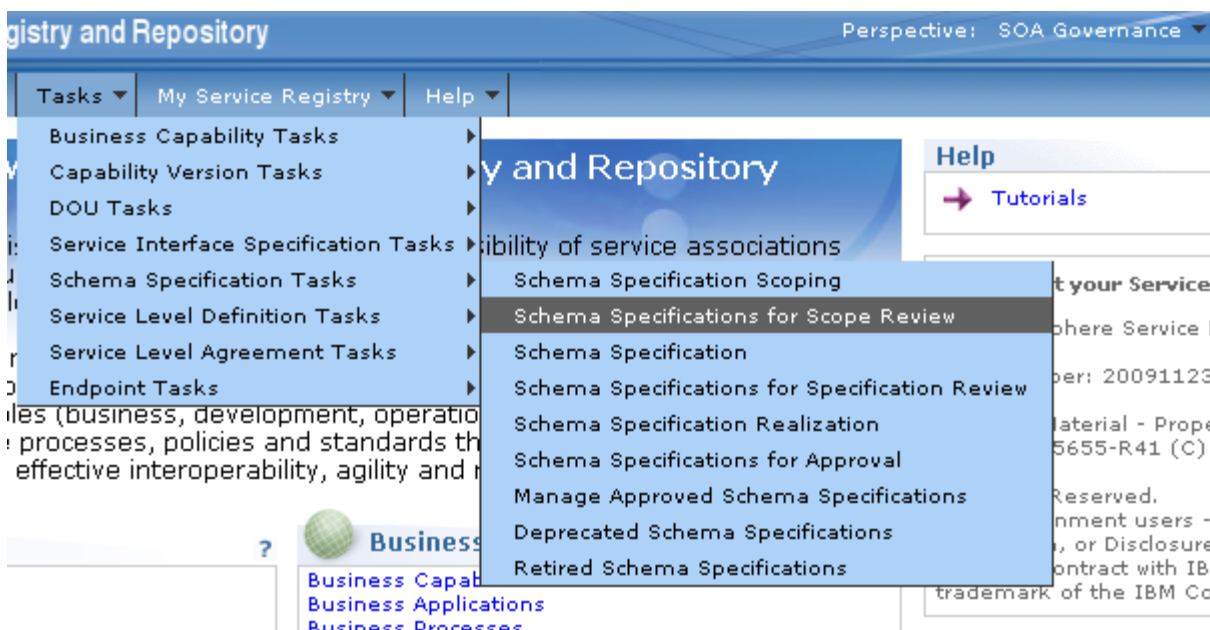
Classifications
Asset Scope Review
Schema Specification

- __b. Logout of WebSphere Service Registry and Repository

3.4 Approving the schema specification

In the Asset Scope Review state, the SOA governance team must now review the schema specification. If required, the SOA governance team can involve other stakeholders, such as the organizations that will consume the service, in this decision process. The SOA governance team review the specification to ensure that it meets their requirements and that the business objects that it utilizes are specified in a manner that promotes future reuse. After the team agrees that all aspects of the design meet the requirements, the schema is approved by a member of the SOA governance team.

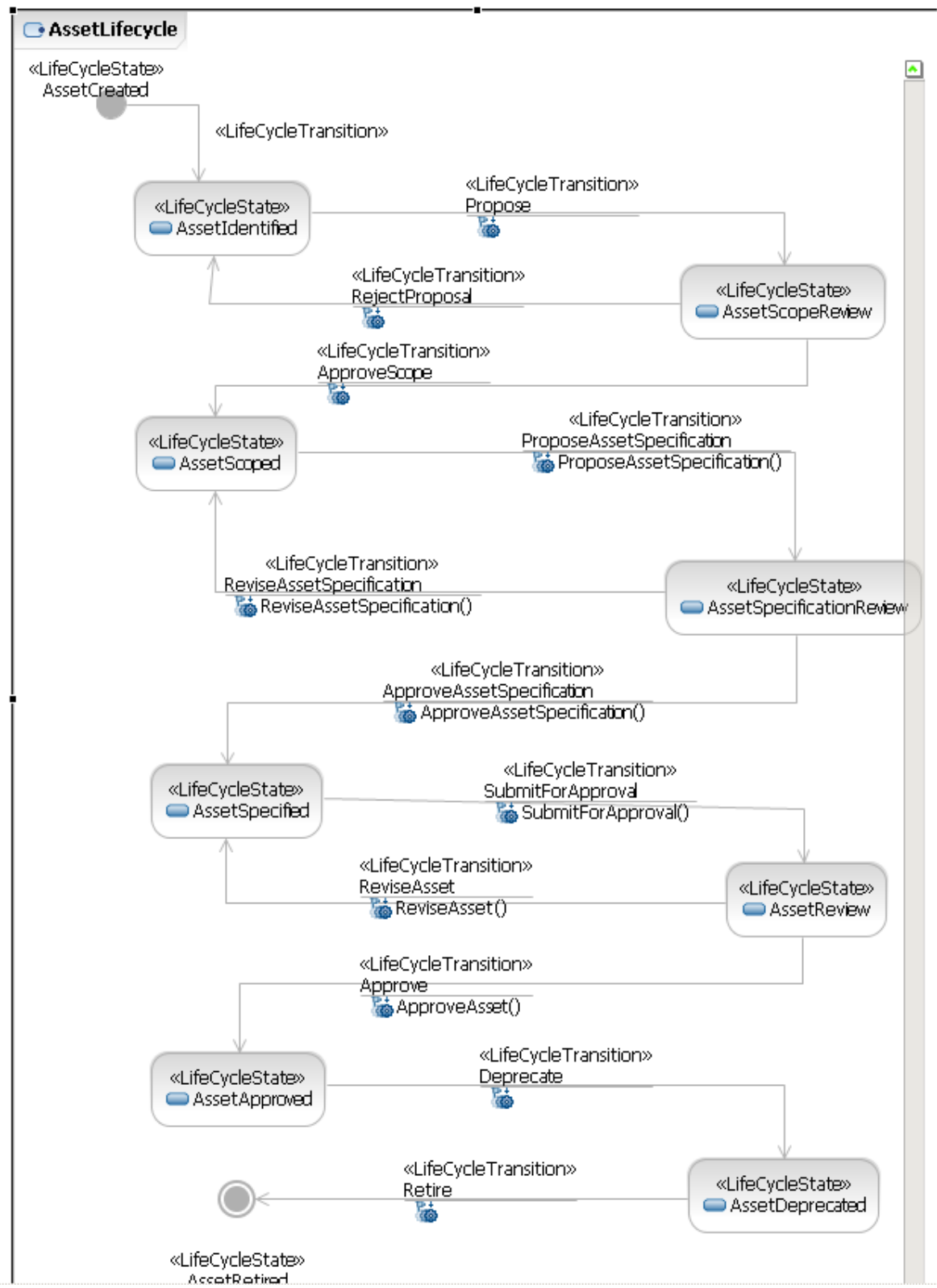
- __1. Transition the schema specification to the Approved state
 - __a. Log on to WebSphere Service Registry and Repository as **David** and select the **SOA Governance** perspective.
 - __b. Click **Tasks** → **Schema Specification Tasks** → **Schema Specifications for Scope Review**



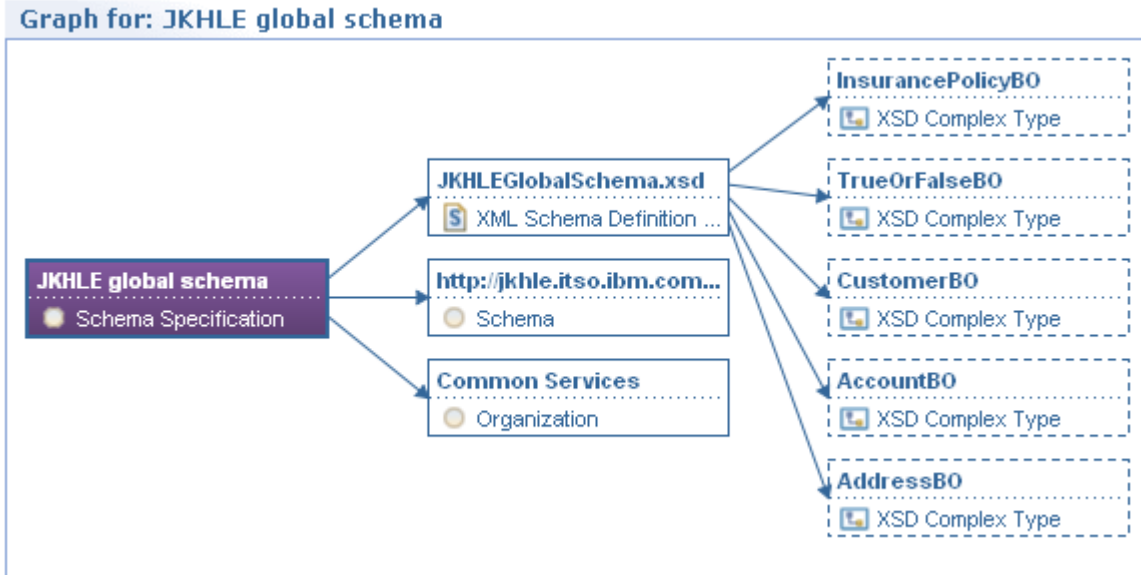
- __c. Click **JKHLE global schema** to display the schema specification details.
- __d. Click **Approve Proposal** and note that the new governance state is Asset Scoped

- Governance State**
 - Asset Scoped
- Classifications**
 - Asset Scoped
 - Schema Specification

Below you can see the full asset life cycle.



- __e. Click **Graphical View** to see the Schema Specification and its relationships.



- __f. Logout of WebSphere Service Registry and Repository

You have completed the lab.

3.5 Lab Review

During this lab, you:

- Created a schema specification
- Imported the related XSD document
- Transitioned the schema specification through part of its life cycle
- Saw that Development and SOA Governance have different roles to play in an asset's lifecycle

3.6 Optional explorations

If you have completed that lab ahead of schedule, you could explore the tutorials for the Governance enablement profile in the WebSphere Service Registry and Repository Information Center.

Read the tutorials at:

http://publib.boulder.ibm.com/infocenter/sr/v7r0/index.jsp?topic=/com.ibm.sr.doc/cwsr_gep_tutorials.html

The tutorial sample files are located at C:\student\WSRR\GEP_Tutorials.



CONGRATULATIONS!

You have successfully completed Lab 3 – Governing a schema

Lab 4 Governing a WSDL defined Service

4.1 Lab introduction

JKHLE, require to implement a new Eligibility service, which verifies that a customer satisfies the commercial credentials to be an account holder.

In this Lab you will create this new Business Service and progress it through most of its lifecycle into production. This process would in reality take days and weeks (– possibly months and years!) Hopefully, you will not take as long as that! The technical definition of the service is contained in a WSDL document.

In this lab you will:

- Create a new Business Service. This defines the required capability.
- Create a Service Version. This defines a version of the Business Service to be implemented, and relates to the technical definition of the service.
- Create a Service Level Definition. This defines non-functional requirements such as Average Response Time and Availability
- Deploy a Service Version to production.
- Audit the history of the service

(In practice, your service would transition through staging and pre-production states as well, but to keep the lab to a manageable size, we will skip these stages.)

Many people have a part to play in the lifecycle, and during the lab you will play the roles of:

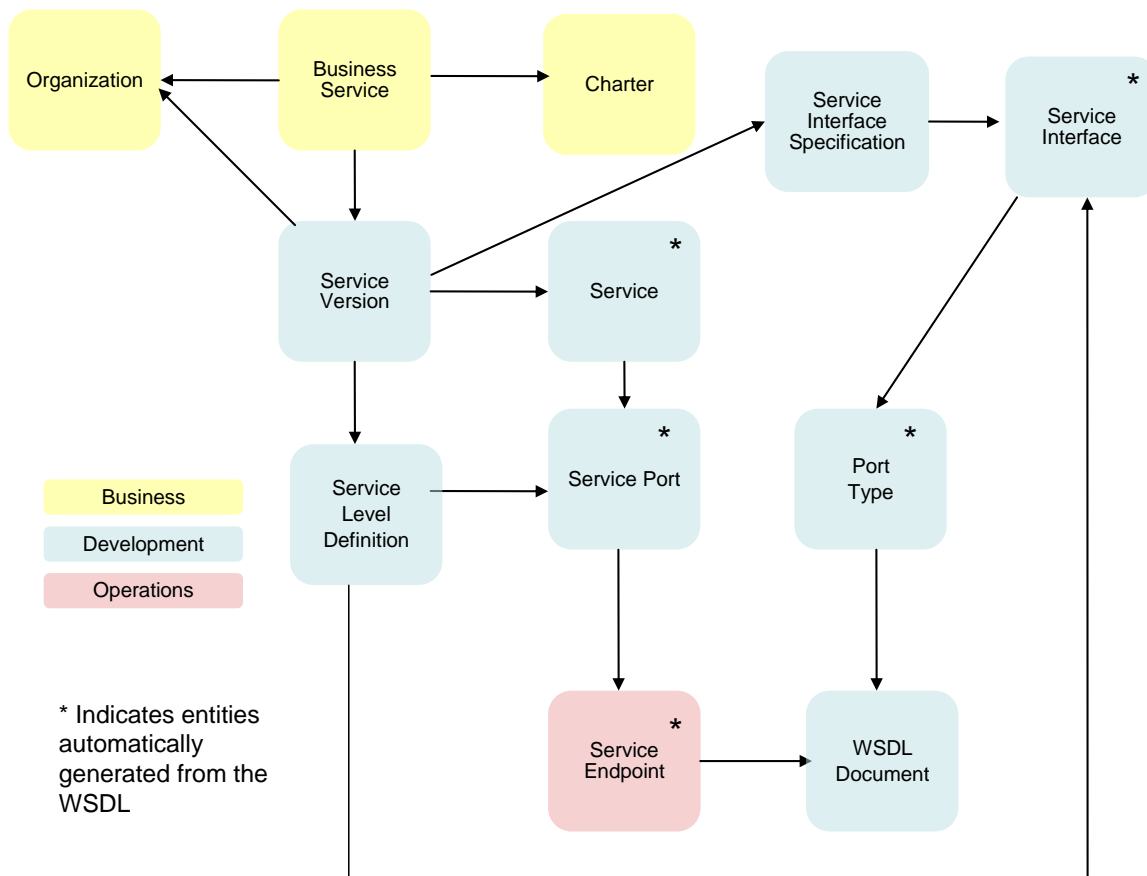
Bob – Business Analyst

David – SOA Governance

Connie – Developer

Jon - Operations

__1. As you work through the lab you will create the structure shown below.



Bob, the business analyst creates the Business Service, links it to its owning Organization, and to its Charter. The Charter is a specification document which defines the service.

Connie, the developer, creates the Service Version, which represents an implementation of the Business Service. Connie also loads the WSDL document that defines the service. When the WSDL document is loaded, it is parsed and objects are created to represent WSDL elements such as Service, Service Port and Port Type. In particular, the Service Version relates to a Service Interface Specification which relates, through a Service Interface to the WSDL Port Type. These are then linked to the Service Version. Development is also responsible for creating the Service Level Definition.

David, a member of the SOA Governance team is concerned with the overall governance and lifecycle of the service, and is responsible for executing many of the lifecycle transitions.

Jon in Operations, is concerned with the Service Endpoint lifecycle. He transitions it from Offline to Online.



Finish

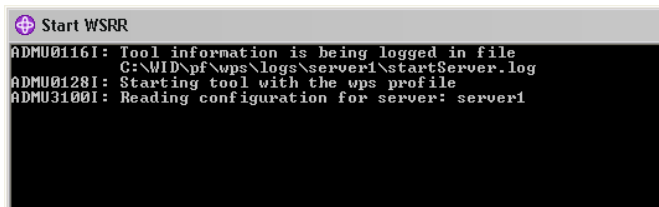
During this lab, there are several occasions when you must click **Finish** to save your changes. Look out for them!

4.2 Create a new Business Service

Bob, the business analyst for the commercial line of business at JKHLE, is tasked with defining the new Eligibility service.

Having used the search facilities in the WebSphere Service Registry and Repository Web UI to confirm that there is no suitable existing service, Bob creates a new Business Service object to identify the requirement for such a capability, and attaches to it a charter. The charter is a document that describes the required capability in detail, including all functional and non-functional requirements.

- __1. If necessary, start WebSphere Service Registry and Repository
 - __a. Open the **WSRR PoT Shortcuts** folder on the desktop
 - __b. Click the shortcut **Start WSRR Server** to start WebSphere Service Registry and Repository Server that is hosted in WebSphere Application Server.
 - __a. This will bring up a window. The window will display startup information and remain open after the startup has completed.



```

Start WSRR
ADMU0116I: Tool information is being logged in file
C:\MID\pf\wps\logs\server1\startServer.log
ADMU0128I: Starting tool with the wps profile
ADMU3100I: Reading configuration for server: server1
  
```

Read ahead until the window indicates the server has started.

- __c. After startup has completed, close this window with any key.
- __2. Log on to WebSphere Service Registry and Repository.

- __a. Click the Mozilla browser icon.



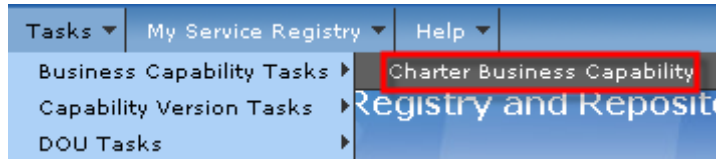
- __d. Click the **WSRR Console** link on the Bookmarks Toolbar.

Or type <http://localhost:9080/ServiceRegistry/> into the address.

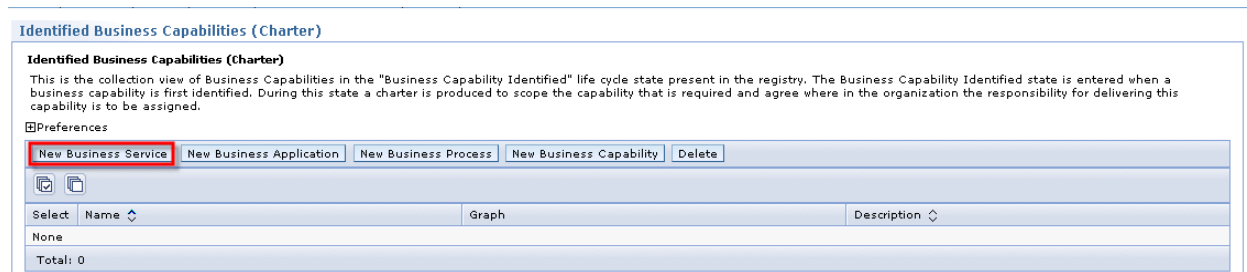
- __e. Logon as **Bob**, the Business Analyst and check that the **Business** perspective is selected.

__3. Charter a New Business Service

- __a. From the **Tasks** menu drop down list, select **Business Capability Tasks->Charter Business Capability**.



- __b. From the **Identified Business Capabilities (Charter)** list, click the **New Business Service** button.



- __c. Enter the following information

__b. Name: **Eligibility service**

__c. Description: **Validate customer information is complete and can be verified before opening an account**

 A screenshot of a 'Details' page in a web application. The page has a 'Details' tab at the top. Below it is a section titled 'General Properties' with a blue header. Under this section, there are two input fields. The first is labeled '*Name' and contains the text 'Eligibility service'. The second is labeled 'Description' and contains the text 'Validate customer information is complete and can be verified before opening an account'.

d. Click **Finish**. The details page for the new Business Service displays.

Eligibility service

Detail view for Business Service. A Business Service represents a business capability that is viewed as a se organization. Business Services have realizations (Service Versions) that are implemented using SOA tech to Service Interface Specifications.

The screenshot displays the 'Eligibility service' details page. At the top, there are tabs for 'Details', 'Impact Analysis', 'Governance', 'Policy', and 'Activity'. Below the tabs are links for 'Edit Properties' and 'Edit Relationships'. The main content is divided into several sections:

- Properties:**
 - *Name:** Eligibility service
 - Description:** Validate customer information is complete and can be verified before opening an account
 - Business Requirements Link:** urn:serviceregistry
 - Asset Web Link:** urn:serviceregistry
 - Remote State:** (empty field)
 - Owner Email:** (empty field)
- Additional Properties:**
 - Buttons: Back, Propose for Charter Review, Reject Capability
- Links:**
 - Graphical View
 - Applied Policies
 - Applied Policy Attachments
- Relationships:**
 - Charter: None
 - Service Interface Versions: None
 - Versions: None
 - Owning Organization: None
 - Dependency: None
 - Artifacts: None
- Dependent Entities:**
 - Dependent Assets: None
- Governance State:** Business Capability Identified (highlighted with a red box)
- Classifications:**
 - Business Capability Identified
 - Business Service

Notice that the Governance State of the new Business Service is Business Capability Identified, This state is the initial state in the capability life cycle; a new Business Service is entered into this life cycle automatically.

__4. Attach a charter

To attach a charter, load the document that contains the charter, and attach it to the Business Service by using the Charter relationship.

- __a. Click **Edit Relationships**
- __b. Click **Add Other Document** to the right of the Charter relationship

Edit Relationships

Eligibility service > Edit Relationships

The Business Service 'Eligibility service' has the following relationships and targets. Make to apply.

Finish Cancel

	Targets
Eligibility service Add Relationship	
↳ Service Interface Versions	Add Service Interface Specification
↳ Versions	Add Capability Version
↳ Charter	Add Other Document

- __c. Click **Load Document**

Edit Relationships

Eligibility service > Edit Relationships

The Business Service 'Eligibility service' has the foll to apply.

Finish Cancel

Eligibility service | Add Relationship

- ↳ Service Interface Versions
- ↳ Versions
- ↳ Charter
- ↳ Dependency
- ↳ Owning Organization
- ↳ Artifacts

Add Target

Use the options below to select a new target for

Search for Existing Entity

Name

Entity Type

Other Document

Search Add

Load Document

Document Type

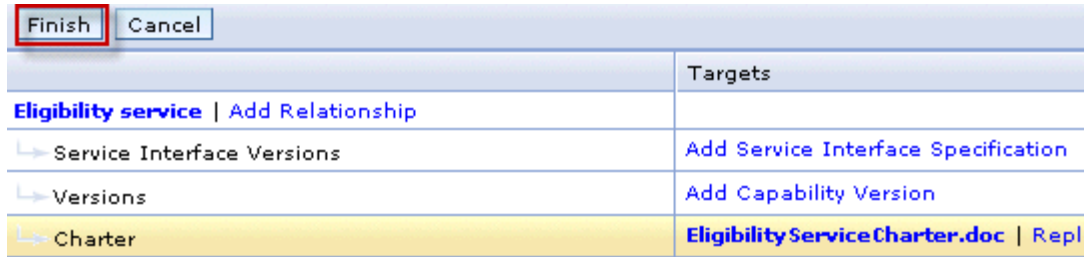
Other Document

Load Document

Cancel

- __d. Click **Browse**, and navigate to **C:\student\WSRR\Lab4**
- __e. Select EligibilityServiceCharter.doc, click **Open**, and then click **OK**.
- __f. Click **Finish** to load the charter document.

The charter document is added as a target of the Charter relationship.

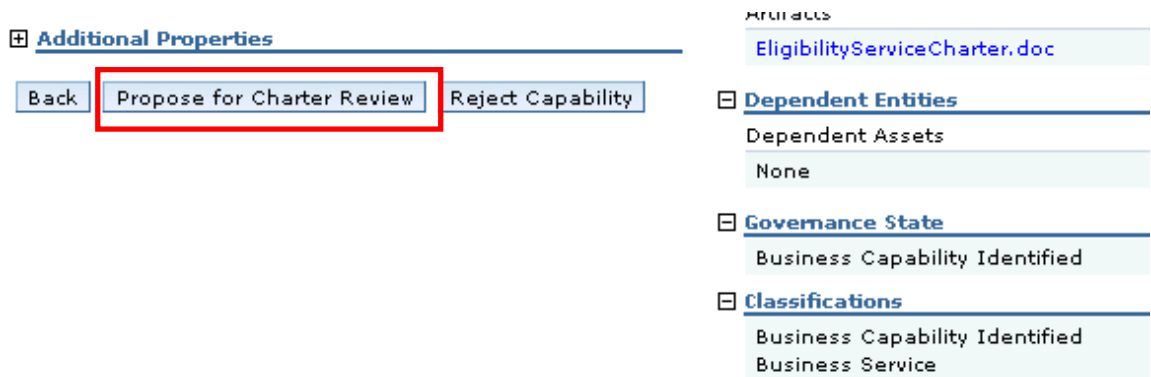


- __g. Click **Finish** to save your changes.

The charter document is loaded for the Eligibility Business Service

- __5. To indicate the service’s readiness for review and to make it available to the reviewers, the business service must be proposed.

- __a. To transition the service to the Charter Review state, click **Propose for Charter Review**.



The governance state changes to Charter Review.



- __b. Logout of WebSphere Service Registry and Repository

4.3 Reviewing and approving the Business Service

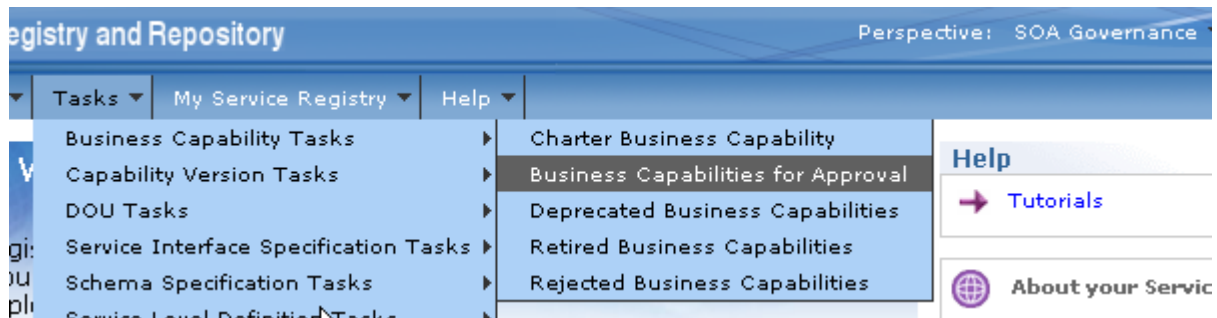
The SOA governance team ensures that governance processes are enforced. Before the Business Service can be transitioned to the next stage in the life cycle, the team must ensure that the proposed capability does not duplicate other services in the registry and that an owning organization is assigned that will be responsible for all versions of this capability and for managing requirements for this service. The next steps in the life cycle for our service are:

- Assign an organization
- Approve Capability

__1. Assigning an owning organization to the Business Service

__a. Log in to WebSphere Service Registry and Repository as **David**, and select the **SOA Governance perspective**

__b. Click **Tasks** → **Business Capability Tasks** → **Business Capabilities for Approval**



__c. Click **Eligibility service** to display the Business Service details.

__d. Click **Edit Relationships**.

__e. Click **Add Organization** to the right of the Owning Organization relationship.

__f. Enter C in the Name field. Then, select **Common Services** from the auto suggest list

__g. Click **Add**.

The Common Services organization is added as a target of the Owning organization relationship.

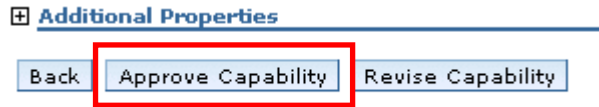
__h. Click **Finish** to save your changes.

__2. Approving the Business Service

Having ensured that all of the governance requirements are satisfied and having used the search facilities in the WebSphere Service Registry and Repository Web UI to confirm that the proposed capability does not duplicate other services in the registry, the SOA governance team member approves the Business Service.

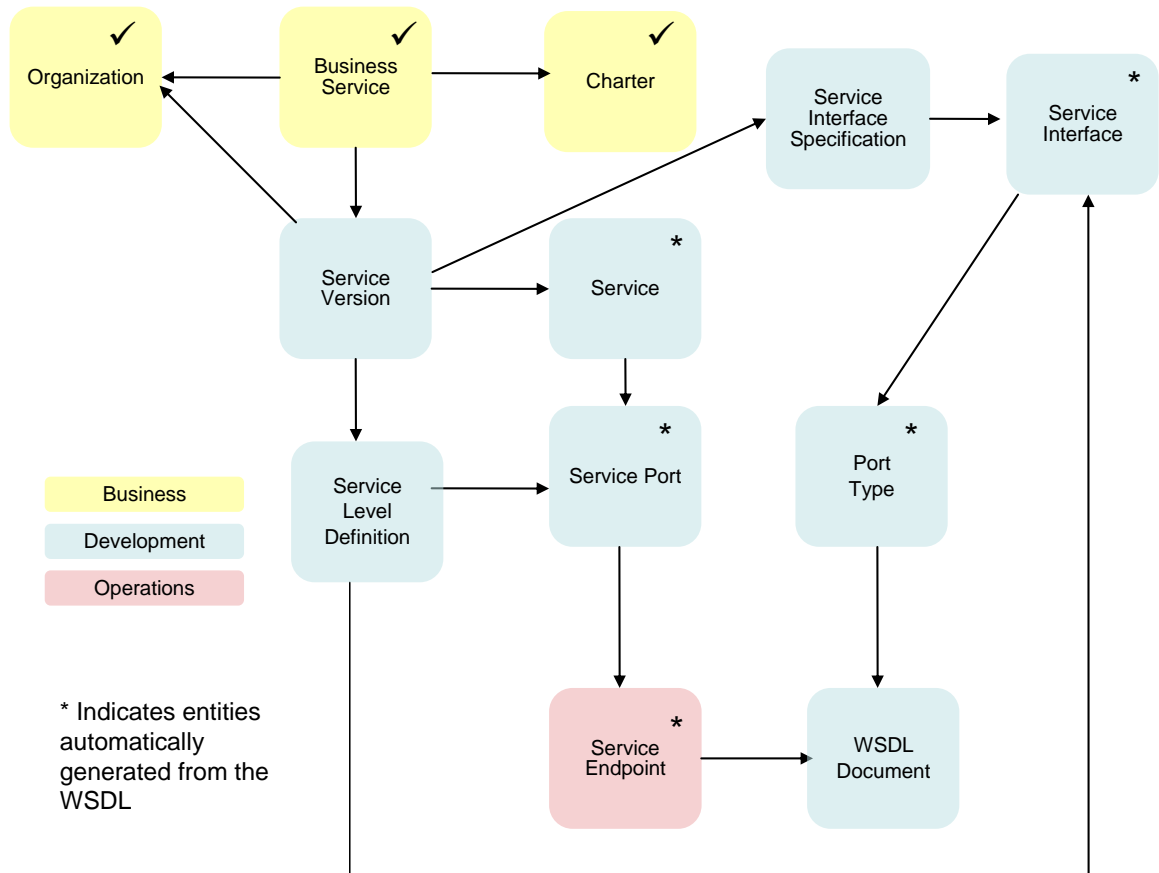
To transition the Business Service to the Business Capability Approved state,

__a. click Approve Capability

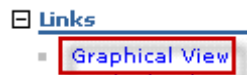


The new governance state is Business Capability Approved

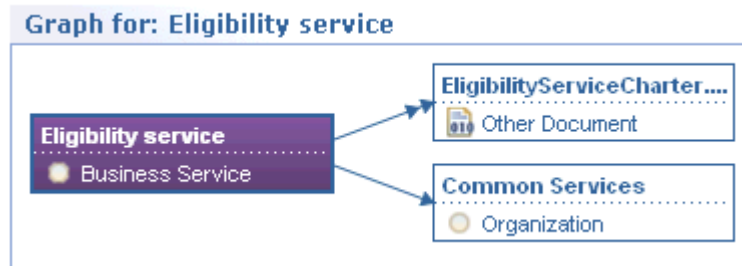
The diagram below shows the status of the WebSphere Service Registry and Repository entities, after the Business Service has been approved.



- __b. To confirm this, click **Graphical View**



- __c. A graph showing the Business Service linked to its Charter and owning Organization appears.



- __d. Logout from WebSphere Service Registry and Repository

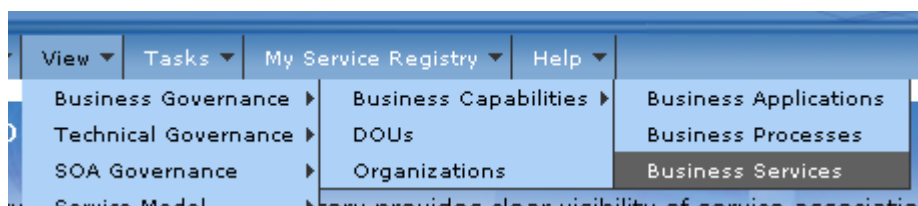
4.4 Creating a Service Version

A Service Version represents a specific version or release of a service, and includes a range of functional and non-functional specifications for that version of the service. Over time multiple versions of the same business capability may be created, as the service is enhanced and extended. The Service Version is governed separately from the Business Service and we will associate the WSDL technical artifacts in later steps to the Service Version.

- __1. Creating a new Service Version

After the business capability is defined, reviewed, and approved, a new Service Version is defined. It is the responsibility of the development organization to create an implementation of the service.

- __a. Log on to WebSphere Service Registry and Repository as **Connie**, and select the **Development** perspective
- __b. Click **View** → **Business Governance** → **Business Capabilities** → **Business Services**.



- __c. Click **Eligibility service** to display the Business Service details.

- d. Assign a new Service Version to the Business Service by clicking **New Capability Version**

Business Service

Messages
 The operation was completed successfully

Business Services > Eligibility service

Detail view for Business Service. A Business Service represents organization. Business Services have realizations (Service Version to Service Interface Specifications).

Details | Impact Analysis | Governance | Policy | Activity

Properties

*Name
 Eligibility service

Description
 Validate customer information is complete and can be verified before opening an account

Business Requirements Link
<urn:serviceregistry>

Asset Web Link
<urn:serviceregistry>

Remote State

Owner Email

Additional Properties

Back | **New Capability Version** | Deprecate Capability

A new Service Version is created for the Eligibility service, and is listed under the relationship called Versions

Relationships

Charter
[EligibilityServiceCharter.doc](#)

Service Interface Versions
 None

Versions
[Eligibility service](#)

Owning Organization
[Common Services](#)

__2. Edit the properties of the Service Version

__a. Click **Eligibility service**

Details of the new Service Version are displayed.

Service Version

Business Services > Eligibility service > Eligibility service

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides functional and non functional specifications that hold for that version of the service. The Service Version exposes service level definitions. It may also (in the case of a composite service) identify the services it depends on and Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details | Impact Analysis | Governance | Policy | Activity

[Edit Properties](#) | [Edit Relationships](#) | [Edit](#)

Properties	Links	Relationships	Dependent Entities
*Name Eligibility service	Graphical View Applied Policies Applied Policy Attachments	Interface Specifications None	Chartered Business Capability(s) Eligibility service
Description		Provided Web Services None	
Version 0.0.0		Provided SCA Modules None	
Consumer Identifier		Owning Organization Common Services	
Version Availability Date Wednesday, January 13, 2010		Dependency None	
Version Termination Date Wednesday, January 13, 2010		Artifacts None	
Version Requirements Link urn:serviceregistry		Provides None	
Asset Web Link urn:serviceregistry		Consumes None	
Remote State			
Owner Email			
Additional Properties			

The owning organization for the new Service Version is set automatically to Common Services, the same as the Business Service. You can change this owning organization if necessary.

The Chartered Business Capability relationship refers back to the Business Service.

__b. Click Edit Properties

- __c. Change the following property values:
 - Name: Eligibility service (1.0)
 - Description: Service to determine customer account eligibility.
 - Version: 1.0
 - Version Requirements Link: <http://requirements.jkhle.com/requirements.jsp?id=1210> This is a link, fictitious in this case, to the relevant item in JKHLE’s requirements tracking tool.

Properties

*Name
Eligibility service (1.0)

Description
Service to determine customer account eligibility

Version
1.0

Consumer Identifier

Version Availability Date
Wednesday, January 13, 2010

Version Termination Date
Wednesday, January 13, 2010

Version Requirements Link
<http://requirements.jkhle.com/requirements>

- __d. Click **OK** to save the changes.

The governance state of our Service Version is Identified. This state is the initial state in the model phase of the SOA life cycle; a new Service Version is entered into the SOA life cycle automatically.

__3. Proposing the Service Version scope

Now that the scope of the Service Version is defined, it must be circulated for review so that all potential consumers of the service can verify that the requirements are within the proposed scope from the development team.

- __a. Click **Propose Scope** to transition the Service Version to the Scope Review state.

Additional Properties

Back New Version **Propose Scope** New SLD

The new governance state is Scope Review.

- __b. Log out of WebSphere Service Registry and Repository.

4.5 Approve the Service Version scope

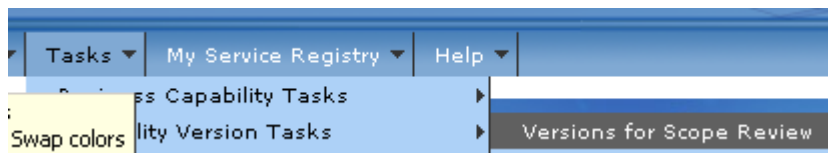
David from the SOA governance team reviews the Service Version requirements. He carries out the following verification activities:

- Checks that this Service Version is warranted throughout the organization
- Checks that the requirements and stakeholders are in agreement
- Checks that the owning organization that is responsible for delivering the requirements is identified and is assigned to the Service Version

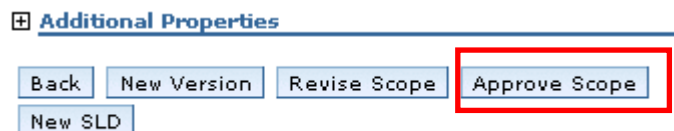
When the Service Version scope review is complete, the scope can be approved.

- __1. Transition the Service Version to the Scoped state.

- __a. Log on to WebSphere Service Registry and Repository as **David**, and select the SOA Governance perspective.
- __b. Click **Tasks** → **Capability Version Tasks** → **Versions for Scope Review**,

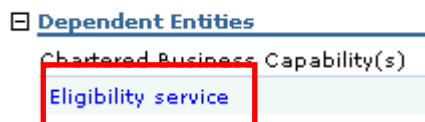


- __c. Click **Eligibility service (1.0)** to display the Service Version details..
- __d. Click **Approve Scope**.



The new governance state is Scoped

- __2. Let's view the relationships between the entities that we have created.
- __a. Under Dependant Entities, click **Eligibility service**

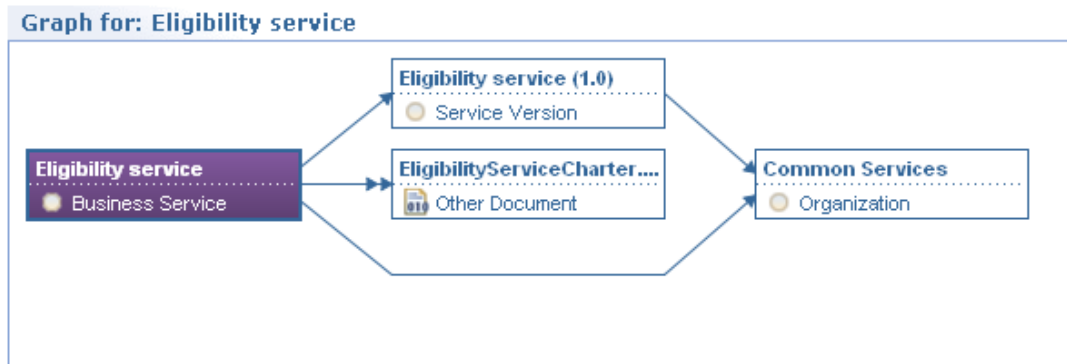


You are returned to the details view of the Business Service.

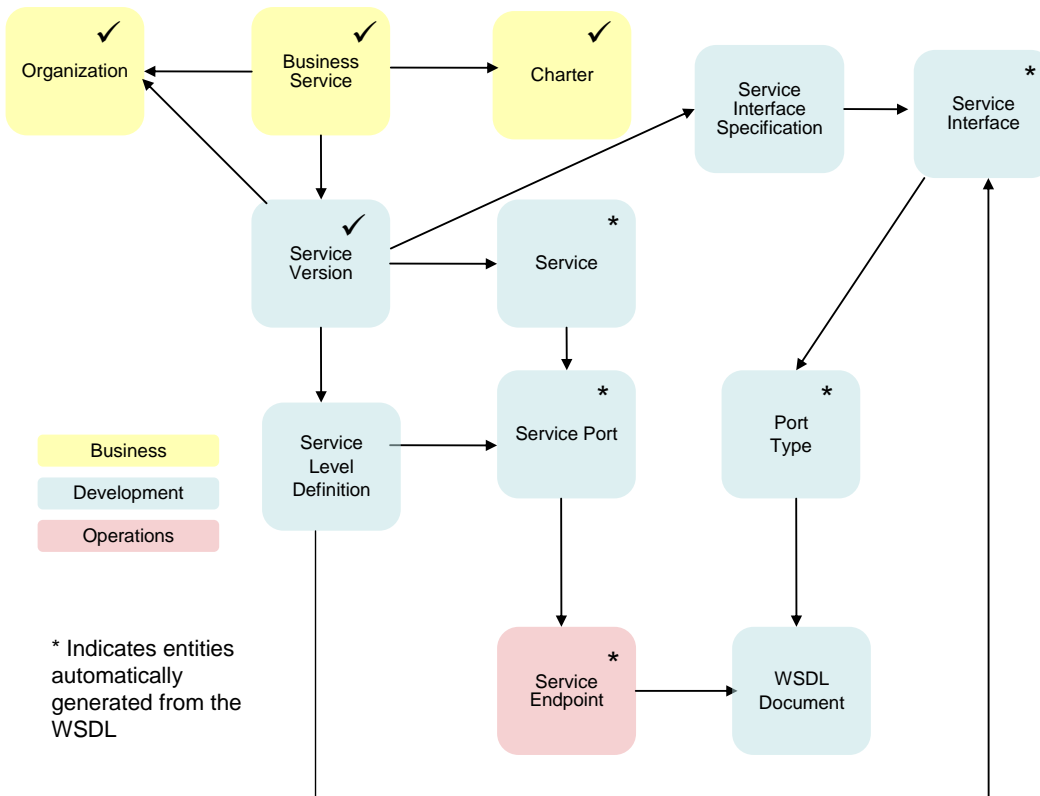
__b. Click **Graphical View**



The graph below is displayed.



__c. We have created a Business Service and linked it to its charter document and to a Service Version. Both the Business Service and the Service Version are related to the same organization.



__d. Log out of WebSphere Service Registry and Repository

4.6 Load the WSDL document for the Eligibility service.

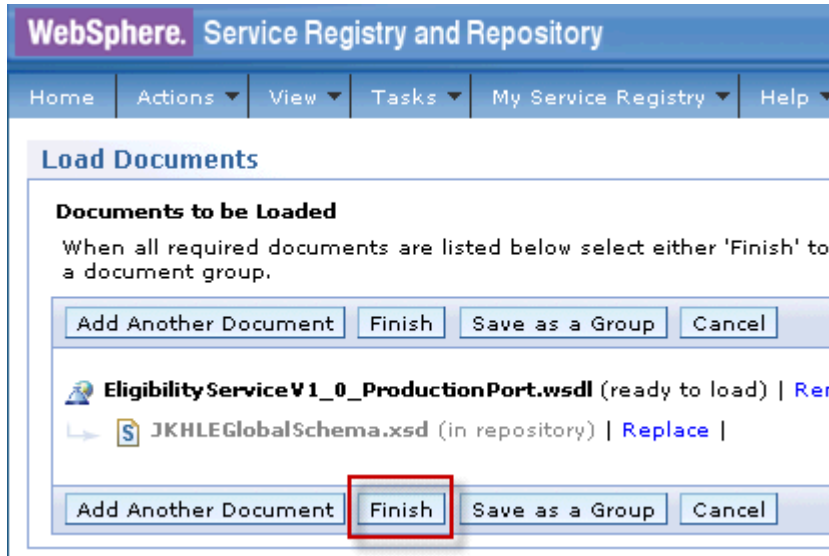
Development has now created the WSDL document for this service. We will load it and relate it to the Service Version through a service interface specification.


- __1. Loading the WSDL into registry is the responsibility of the development organization.
 - __a. Log on to WebSphere Service Registry and Repository as **Connie**, and select the Development perspective
 - __b. Scroll down to the load documents section on the Home page

- __c. Click **Browse**, and navigate to:
C:\student\WSRR\Lab4 where the WSDL document is located.
- __d. Select **EligibilityServiceV1_0_ProductionPort.wsdl**, and click **Open**.
- __e. Enter **1.0** in the document version field.
- __f. Ensure that the Document Type in the Load Document pane is set to **WSDL**.

- __g. Click **OK**.

__h. Click Finish to load the WSDL document.



 Note that the dependent XSDDocument JKHLEGlobalSchema is detected, identified as in repository, and is automatically linked to the WSDL being loaded. If you wished you could override this choice and load a new version of JKHLEGlobalSchema.xsd at the same time.

WebSphere Service Registry and Repository confirms that the WSDL document loaded successfully.

Load Documents

Documents Loaded Successfully

The following documents have been loaded into the repository:

Name	Description	Namespace	Version
EligibilityServiceV1_0_ProductionPort.wsdl		http://jkhle.itso.ibm.com/EligibilityV1	1.0

As the WSDL document was loaded, it was parsed to identify WSDL logical objects such as Service, Port, Port type and SOAP address. These are used to create correlator objects. The table below lists WSDL logical objects and the corresponding correlator objects.

WSDL logical object	Correlator object created	Correlation properties
Service	Service	The Service name, namespace, and version.
Port type	ServiceInterface	The Port type name, namespace, and version.
Port	ServicePort	The Port name. The name, namespace, and version of the Service from which the Port is derived.
Binding	ServiceBinding	The Binding name, namespace, and version.
Operation	ServiceOperation	The Operation name. The name, namespace, and version of the Port type from which the Operation is derived.
Message	ServiceMessage	The Message name, namespace, and version.
SOAP address	SOAPServiceEndpoint	The name, namespace, and version of the Service from which the SOAP Address is derived. The name of the Port from which the SOAP address is derived. The type of the Service endpoint from which the SOAP address is derived.
ExtensionLogicalObject	ExtensionServiceEndpoint	The name, namespace, and version of the Service from which the ExtensionLogicalObject is derived. The name of the Port from which the ExtensionLogicalObject is derived. The type of the Service endpoint from which the ExtensionLogicalObject is derived.

Correlator objects of particular interest to us are:

- Service
- Service Interface
- Service Port
- SOAP Service Endpoint



WebSphere Service Registry and Repository Correlator modifier plug-in

The correlator objects are created by a feature of WebSphere Service Registry and Repository called the correlator modifier plug-in. For more information about the correlator modifier, see the WebSphere Service Registry and Repository Information Center at:

http://publib.boulder.ibm.com/infocenter/sr/v7r0/topic/com.ibm.sr.doc/cwsr_correlator_modifier_R5.html

4.7 Create a Service Interface Specification and link it to the Service Version

You will now:

- Create a Service Interface Specification
- Relate the Service Version to the Service Interface Specification.

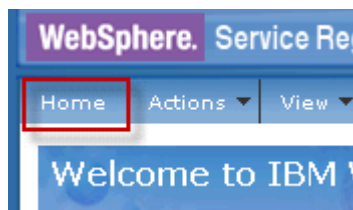
These tasks are carried out by development.

__1. Create a Service Interface Specification.

You now create the Service Interface Specification object and link it to the Service Interface object that was created when the WSDL document was loaded.

If necessary, log on as Connie and check that you are using the development perspective.

__a. Click the **Home** tab to take you to the home page.



__b. Under **Technical Governance**, click **Service Interface Specifications**



__c. Click **New**.

- __d. Enter **Name** and **Version** as shown and click on **Add Service Interface**.

Eligibility service (1.0) Add Relationship		Targets
↳ Depends On Schema		Add Schema Specification
↳ Specified Interfaces		Add Service Interface
↳ Dependency		Add Asset
↳ Owning Organization		Add Organization
↳ Artifacts		Add Document

Details

General Properties

*Name

Description

Version

- __e. Under Add Target, type E into the Name field and select **EligibilityV1_0**.

Add Target

Use the options below to select a new target for the

Search for Existing Entity

Name

EligibilityV1_0 1 result

Service Interface

- __f. Click **Add**.
- __g. Click **Finish**.

WebSphere Service Registry and Repository confirms that the Service Interface Specification has been created.



The Service Interface object that you selected is one of the correlator objects created when you loaded EligibilityServiceV1_0_ProductionPort.wsdl. It represents the Port type. You can confirm this by examining the content of the original WSDL document.

__2. Relate the Service Version to the interface specification

A relationship must now be created between the Service Version and the Interface specification.

- __a. If necessary, log on as Connie and check that you are using the development perspective.
- __b. On the home page, look under **Technical Governance** and click **Service Versions**.
- __c. Click **Eligibility service (1.0)**.
- __d. Click **Edit Relationships**.
- __e. Click **Add Service Interface Specification**.

Eligibility service (1.0) Add Relationship	
Interface Specifications	Add Service Interface Specification
Provides	Add Service Level Definition
Provided Web Services	Add Service
Consumes	Add Service Level Agreement
Provided SCA Modules	Add Service Module
Dependency	Add Asset

- __f. Enter E in the Name field, and select **Eligibility service (1.0)** from the auto suggest list.

Add Target

Use the options below to select a ne

Search for Existing Entity

Name

Eligibility service (1.0)

Eligibility service (1.0) 1 result

Service Interface Specification

Search Add

- __g. Click **Add**.
 - __a. The Eligibility Service Interface is added as a target of the interface specification.
- __h. Click **Finish** to save your changes.
- __i. Log out of WebSphere Service Registry and Repository

4.8 Approving the Service Version

After all parties have agreed the details in the plan and specification David from the SOA governance team can approve the Service Version, by progressing it through the following states:

- Plan Review
- Planned
- Specification Review
- Specified

- __1. Log on to WebSphere Service Registry and Repository as David, and select the SOA Governance perspective.
- __a. On the Home page under Technical Governance, click Service Versions



- __b. Click Eligibility service (1.0)
- __c. Click Propose Plan



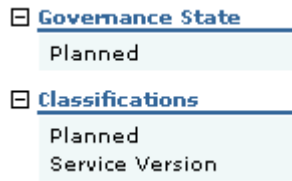
- __d. The governance state changes to Plan review



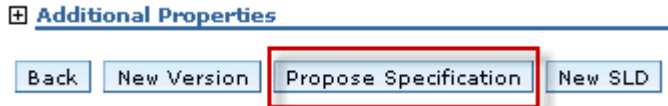
- __e. Click Approve Plan



The governance state changes to Planned



__f. Click Propose Specification.



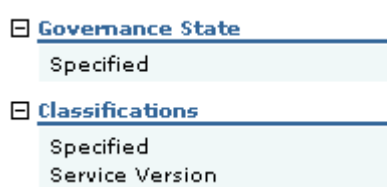
The governance state changes to Specification Review.



__g. Click Approve Specification



The governance state changes to Specified.



__h. Log out of WebSphere Service Registry and Repository

4.9 Creating a Service Level Definition

Now, the development team must define the Service Level Definition to which the Service Version will adhere. The Service Level Definition specifies non-functional or quality of service characteristics to be provided by the service. The new Service Level Definition is then related to the Service Interface Definition

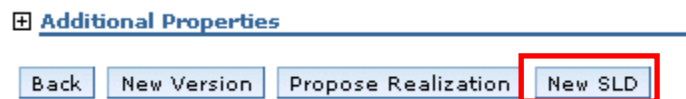
__1. Creating a Service Level Definition

Creating the Service Level Definition (SLD) is the responsibility of the development organization.

- __a. Log on to WebSphere Service Registry and Repository, as Connie and select the Development perspective.
- __b. On the Home page under Technical Governance, click **Service Versions**



- __c. Click **Eligibility service (1.0)**
- __d. Click **New SLD**



A new SLD object is created.

__e. Click **SLD - Eligibility service (1.0)** under Provides to display the SLD details.

[-] **Relationships**

Interface Specifications

Eligibility service (1.0)

Provided Web Services

None

Provided SCA Modules

None

Owning Organization

Common Services

Dependency

None

Artifacts

None

Provides

SLD - Eligibility service (1.0)

Consumes

None

__f. Click **Edit Properties**, and enter the following information

- Average Response Time: 100
- Availability: Working Hours Only

[-] **Properties**

*Name

SLD - Eligibility service (1.0)

Description

Average Response Time

100

Availability

24/7 High Availability

[-] Working Hours Only

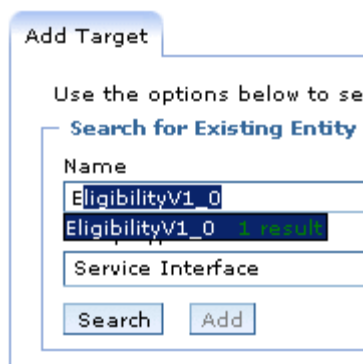
As Available (No gaurantee)

__g. Click **OK** to save your changes.

- __2. Adding the service interface
- __a. You now need to create a relationship between the Service Level Definition and the service interface.
 - __b. Click **Edit Relationships**.
 - __c. Click **Add Service Interface** to the right of the Service Interface relationship



- __d. Enter E in the Name field, select **EligibilityV1_0** from the auto suggest list, and click **Add**.



- __e. The development team define the Web service port that realizes this SLD; this specifies the interaction protocol that consumers will use. The service interface is added as a target of the service interface relationship.

To do this, you link the Service Level Definition to a Web Service Port.

- __f. Click **Add Service Port** to the right of the Bound Web Service Port

SLD - Eligibility service (1.0) Add Relationship	
↳ Compatible Service Level Definitions	Add Service Level Definition
↳ Service Interface	EligibilityV1_0 Replace R
↳ Available Endpoints	Add Service Endpoint
↳ Bound SCA Export	Add Service Export
↳ Bound Web Service Port	Add Service Port

- __g. Enter E in the Name field, select **EligibilityServiceV1_0_ProductionPort** from the auto suggest list, and click **Add**.
- __h. Click **Finish** to save the relationship changes.

Your new SLD is now related to a Service Interface and a Service Port

Relationships	
Service Interface	EligibilityV1_0
Available Endpoints	None
Bound SCA Export	None
Bound Web Service Port	EligibilityServiceV1_0_ProductionPort
Compatible Service Level Definitions	None

The SLD is ready for scope review.

- __i. Click **Propose Scope**

The new governance state is SLD Scope Review

Governance State	
SLD Scope Review	
Classifications	
Extended Service Level Definition	
SLD Scope Review	

- __j. Log out of WebSphere Service Registry and Repository

4.10 Approving the Service Level Definition

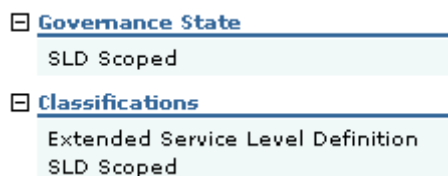
The SOA governance team confirm that the Service Level Definition meets the non-functional requirements. David from the SOA governance team can therefore approve the Service Level Definition through to its Subscribable state for use by operations. (For this Proof of Technology, we are combining several governance stages.)

- __1. Log on to WebSphere Service Registry and Repository as David, and select the SOA Governance perspective.
 - __a. Click **Tasks** → **Service Level Definition Tasks** → **Service Level Definition for Scope Review**



- __b. Click **SLD - Eligibility service (1.0)** to display the SLD details.
- __c. Click **Approve Scope**.

The new governance state is SLD Scoped.

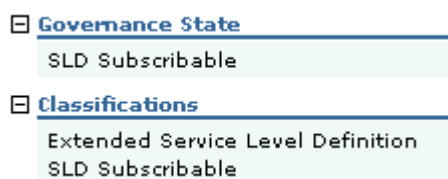


- __d. Continue to transition the Service Level Definition: click **Propose Specification**.

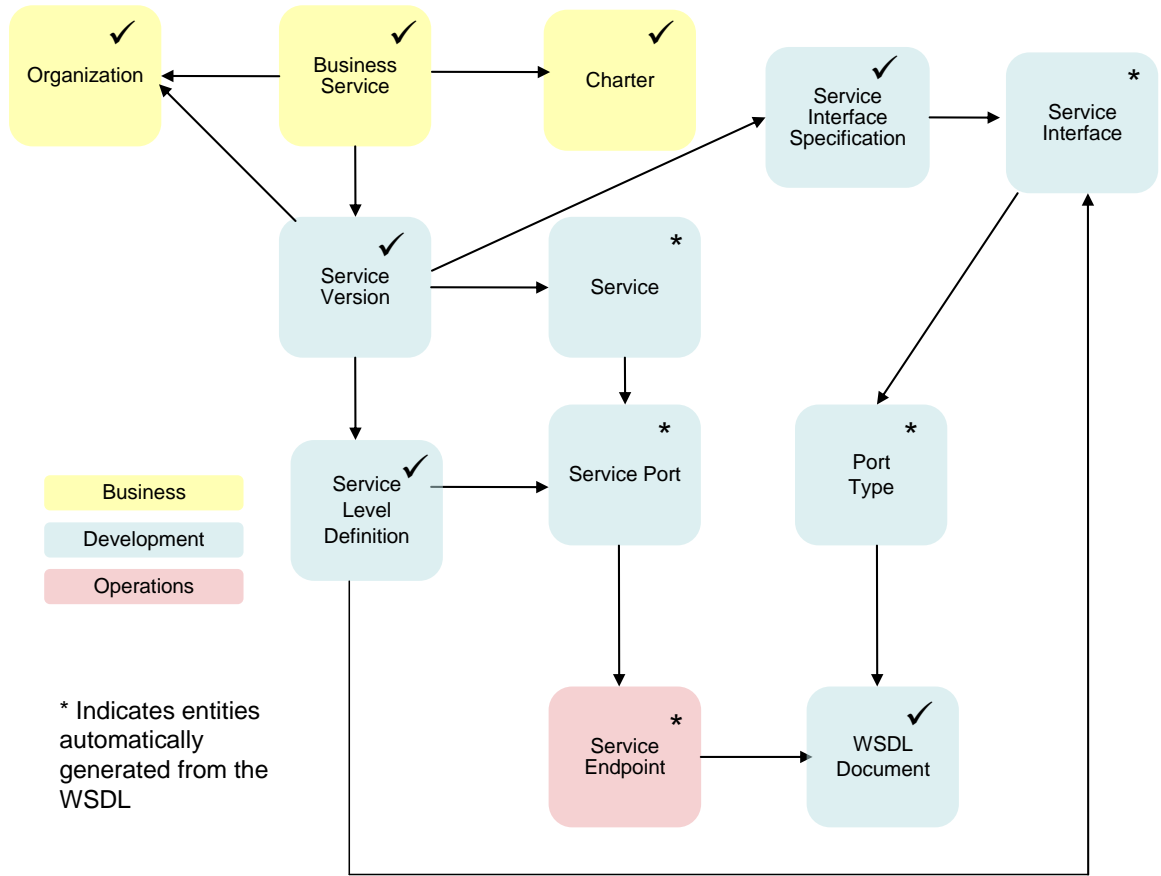
The new governance state is SLD Review.

- __e. Click **Approve Specification**.

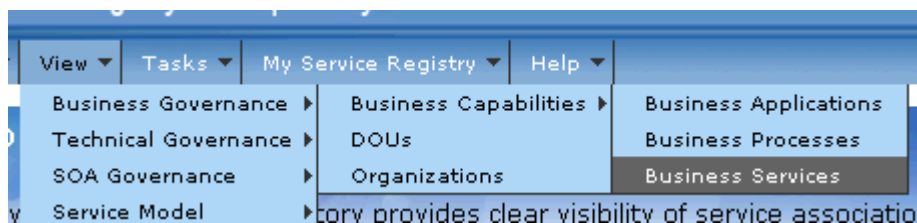
The new governance state is SLD Subscribable



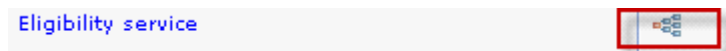
The diagram below shows the status of the WebSphere Service Registry and Repository entities after the new Service Level Definition is created.



- __2. Lets compare the diagram with the structure you have created in WebSphere Service Registry and Repository
 - __a. If necessary, log on to WebSphere Service Registry and Repository as David, and select the SOA Governance perspective.
 - __b. Click **View -> Business Governance -> Business Capabilities -> Business Services**



- __c. Click the Graph icon for Eligibility service



d. Set options to **All Objects**, **4 Levels**, and **Vertical**

Options

Object Display Mode

- All Objects
- Derived Objects
- External Relationships

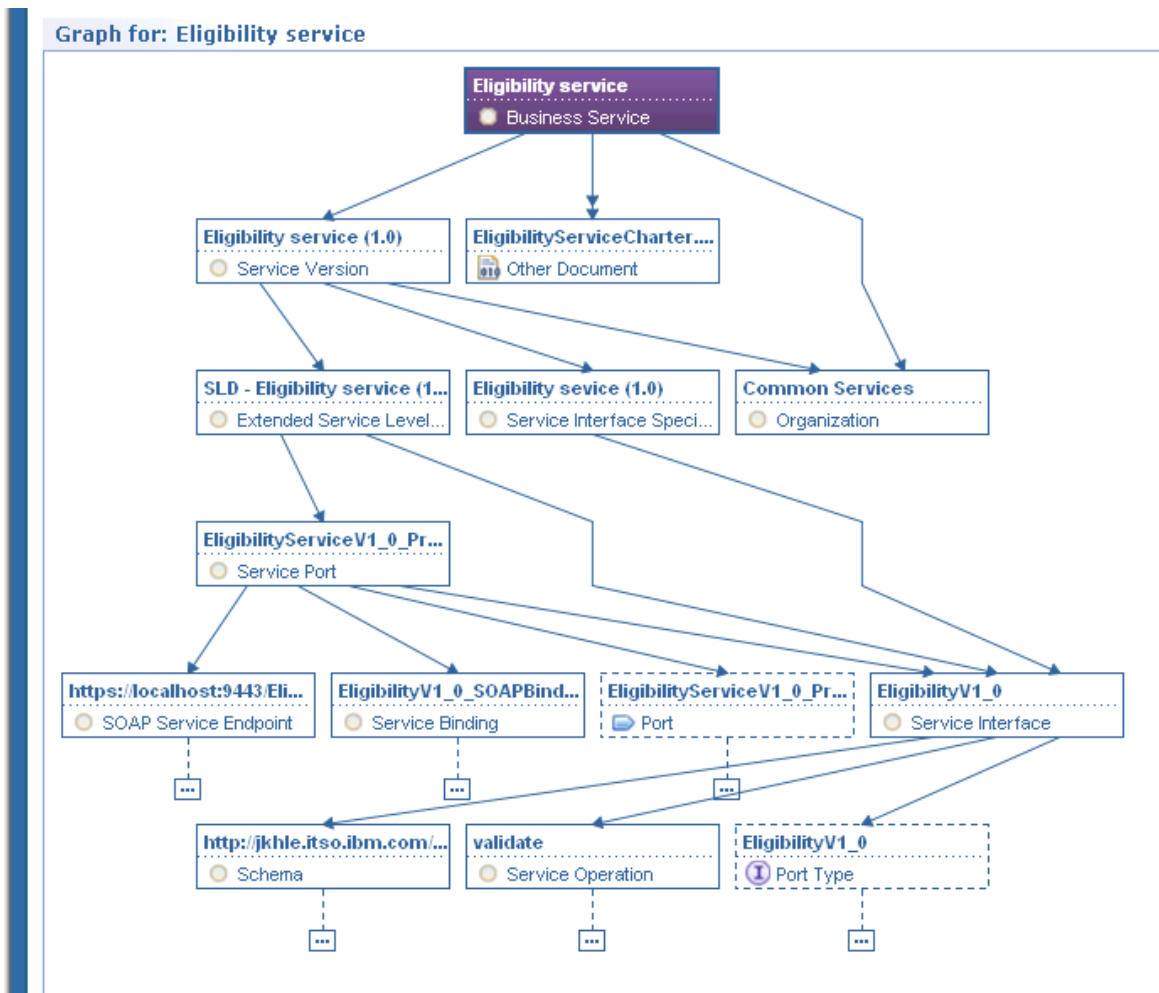
Graph Display Depth

- Automatic
- 1 Level
- 2 Levels
- 3 Levels
- 4 Levels
- Unlimited

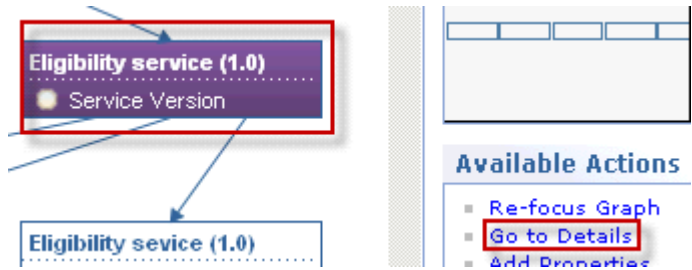
Orientation

- Horizontal
- Vertical

2. The resultant graph should be similar to:



- __3. Trace the various relationships and compare your graph with the one above and with the diagram of the structure we set out to create. If there are discrepancies, discuss them with your instructor.
- __4. If there are no discrepancies, your Service Version can be transitioned to the state of Realized, ready for deployment.
- __5. Transition the Service Version to Realized
 - __a. If you still have the graph displayed, click **Service Version** and **Go to Details**



Otherwise, from the Home page, under Technical Governance, click **Service Versions** and then **Eligibility service (1.0)**.

You should now see the details for the Service Version.

- __b. Click **Propose Realization**.
The new state is Realization Review
- __c. Click **Approve Realization**.
The new state is Realized.
- __d. Log out of WebSphere Service Registry and Repository

4.11 Deploying a Service Version to production

The Service Version now exists and must be taken through all the pre-production stages including appropriate testing before it is ready to be used in production. In this part of the lab, you will:

- Associate the endpoint with the Service Level Definition
- Classify the endpoint
- Transition the Service Version to Operational
- Transition the production endpoint to online.

- __1. Associate the production endpoint with the Service Level Definition.

The WSDL that we loaded previously contained the production endpoint. The next step in making the service consumable in the production environment is to associate the production endpoint with the

Service Level Definition. Jon from the Operations team is responsible for updating the Service Level Definition.

- __a. Log on to WebSphere Service Registry and Repository as Jon, and select the Operations perspective.
- __b. Click **Tasks** → **Service Level Definition Tasks** → **Manage Subscribable Service Level Definitions**.



- __c. Click **SLD - Eligibility service (1.0)**.
- __d. Click **Edit Relationships**.
- __e. Click **Add Service Endpoint** to the right of the Available Endpoints relationship.

SLD - Eligibility service (1.0) Add Relationship	
↳ Compatible Service Level Definitions	Add Service Level Definition
↳ Service Interface	EligibilityV1_0 Replace Remove
↳ Available Endpoints	Add Service Endpoint
↳ Bound SCA Export	Add Service Export
↳ Bound Web Service Port	Add Service Port
	EligibilityServiceV1_0_ProductionPort

- __f. Enter ***Prod** in the Name field,

Add Target

Use the options below to select a new target for the "Available Endpoints" relationship.

Search for Existing Entity

Name

https://localhost:9443/EligibilityV1_0/services/EligibilityServiceV1_0_ProductionPort 1 result

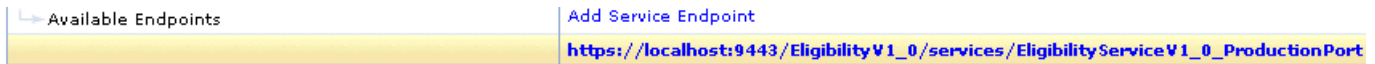
Service Endpoint ▼

select

https://localhost:9443/EligibilityV1_0/services/EligibilityServiceV1_0_ProductionPort

from the auto suggest list, and click **Add**.

- __g. The service endpoint is added as a target of the Available Endpoints relationship.

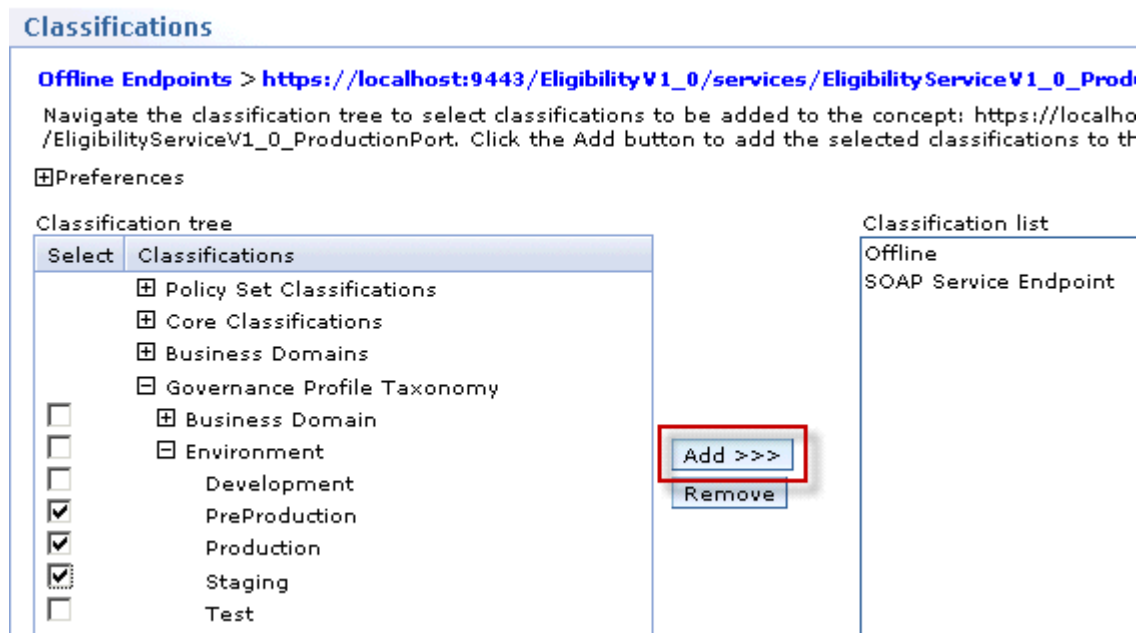


- __h. Click **Finish** to save your changes.

__2. Classify the endpoint

During the deploy phase, the Service Version is subjected to testing in staging and pre-production environments. Finally, it is moved to the production environment and made operational. For each of the environments, staging, pre-production and production, the Service Version must have an endpoint with the appropriate classification. For the purposes of this Proof of Technology, we have only one WSDL and one endpoint. To allow us to transition the Service Version through to production, we will classify our endpoint for all three environments.

- __a. If necessary, log on to WebSphere Service Registry and Repository as Jon, and select the Operations perspective.
- __b. Click **Tasks** → **Endpoints Tasks** → **Endpoints For Activation**.
- __c. Click the endpoint
https://localhost:9443/EligibilityV1_0/services/EligibilityServiceV1_0_ProductionPort.
- __d. Click **Edit Classifications**.
- __e. Expand **Governance Profile Taxonomy** → **Environment**.
- __f. Select **Pre-production**, **Production**, and **Staging** and click **Add**.



The Production classification is added to the Classification list.

__g. Click **OK** to assign the classification.

__3. Classify the WSDL

To classify the WSDL:

__a. Click the **EligibilityServiceV1_0_ProductionPort.wsdl** link under the Source Documents relationship.

__b. Click **Edit Classifications**.

__c. Expand **Governance Profile Taxonomy** → **Environment**.

__d. Select **Production**, and click **Add**.

The production classification is added to the Classification list.

__e. Click **OK** to assign the classification.

__f. Log out of WebSphere Service Registry and Repository

__4. Transition the Service Version from Realized to Operational Review

__a. These few steps take us through all of the testing and pre-production phases. We will assume the role of SOA Governance, but in reality many people would be involved.

__b. Log on to WebSphere Service Registry and Repository as David, and select the SOA Governance perspective.

__g. Under Technical Governance, click **Service Versions**.

__h. Click **Eligibility service (1.0)**.

__i. Click **Propose Staging Deployment**

The state becomes Staging review

__j. Click **Approve Staging Deployment**

The state becomes Staging review

__k. Click **Propose Certification**

The state becomes Certification review

__l. Click **Approve Certification**

The state becomes Certified

__m. Click **Propose Production Deployment**

The state becomes Operational Review

After final testing and verification, the Service Version can now be moved into production.

__n. Logout from WebSphere Service Registry and Repository

__5. Approving production deployment

At this point, the service is ready to be released officially.

__a. Log on to WebSphere Service Registry and Repository as Jon, and select the Operations perspective.


__b. Click **Tasks** → **Capability Version Tasks** → **Versions for Operational Review**,

__c. Click **Eligibility service (1.0)** to display the Service Version details.

__d. Click **Approve Production Deployment**.

The new governance state is **Operational**





Promotion

WebSphere Service Registry and Repository provides the ability to automatically copy a subset of objects from one WebSphere Service Registry and Repository instance to another.

JHKLE have a separate production registry, and when a Service Version goes through the Approve Production Deployment transition it is promoted to the production registry. Similarly, the endpoint is also promoted whenever its state changes between Offline and Online.

__6. Activating the endpoint

Although the service is promoted to the production environment and approved for use, the endpoint needs to be activated. This will update its status to Online.

__a. Click **Tasks** → **Endpoint Tasks** → **Endpoints For Activation**.

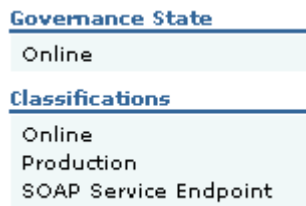
__b. Click the offline endpoint

https://localhost:9443/EligibilityV1_0/services/EligibilityServiceV1_0_ProductionPort.

- __c. Click **Approve For Use**.



The new governance state is Online.



The new Service Version of the Eligibility service is now available in the production environment.

- __d. Log out of WebSphere Service Registry and Repository

4.12 Audit the history of the service

David has been asked to conduct an audit on the deployment of this service. He can do this using the Activity Logging feature of WebSphere Service Registry and Repository

- __1. Examine the activity on the Business Service and the Service Version.
- __a. Log on to WebSphere Service Registry and Repository as David, and select the SOA Governance perspective.
- __b. On the Home page, under Business Governance, click **Business Services**.
- __c. Click **Eligibility service**.
- __d. Click the **Activity** tab.



A table showing a history of changes to the Business Service is displayed. If we had security enabled, the names of the persons who made the changes would be displayed.

- __e. Return to the Home page and examine the activity history for the Service Version and, if you wish, any of the other objects in the registry

You have completed the lab.

4.13 Lab Review

During this lab, you learned how to:

- Create a new Business Service. .
- Create a Service Version.
- Load the associated WSDL
- Deploy a Service Version to production.
- Audit the history of the service



CONGRATULATIONS!

You have successfully completed Lab 4 – Governing a WSDL defined service.

Lab 5 Examining the Governance Model in WebSphere Service Registry and Repository with WebSphere Service Registry and Repository Studio

5.1 Introduction to Configuration profiles

Overview

A configuration profile is a package of all the WebSphere Service Registry and Repository configuration artifacts, including the business model definitions, life cycle definitions, classification systems, governance policies, security roles and permission mappings, as well as custom plug-ins and user interface customizations.

WebSphere Service Registry and Repository ships with a standard profile, the governance enablement profile, which can be extended and customized to meet your specific requirements using WebSphere Service Registry and Repository Studio.

The governance enablement profile contains the following key components:

- **Models:** Entities that represent the objects in an organization's SOA environment
- **Roles:** A predefined set of roles into which users can be assigned according to their SOA tasks
- **Life cycles:** A predefined set of states and transitions that are applied to artifacts to indicate their progress through their life cycles.
- **Classification Systems:** These allow entities to be classified according to their business, technical, or other state. For example, an endpoint might be classified as Production, High Cost and Off-shore. Classification systems can be added and amended as required.

Roles in the governance enablement profile

The governance enablement profile provides the following roles:

- The Business role defines the governance processes, policies, and standards that are shared across the enterprise to ensure effective interoperability, agility, and robustness of the SOA solutions. In an organization, the Business role represents business managers, analysts, and subject matter experts who are interested in how the SOA services and processes contribute to the business.
- The SOA Governance role represents architects, architecture boards, and SOA centers of excellence. This role also includes individuals from other roles (business, development, and operations) who define the governance processes, policies, and standards that are shared throughout the organization to ensure effective interoperability, agility, and robustness of SOA solutions.
- The Development role represents software development practitioners, including architects, release managers, software developers, testers, assembly developers (who use tools such as WebSphere Integration Developer), integrators, and asset librarians. They develop the software specifications and implementations to realize the requirements that are provided by the business and ensure that the implementation meets the business needs and adheres to the governance standards. Development roles usually are associated with a specific organization or department, with responsibility for delivering implementations to support a particular area of the business.
- The Operations role represents operations managers, operations architects, system administrators, integration testers, and IT resource managers. They manage the IT infrastructure, and deploy, configure, and test the implementations produced by development. They are responsible for operational Quality of Service (QoS) and capacity planning, and although their main activities are later in the life cycle, they are involved in all specification activities and reviews to ensure that the planned capabilities can be successfully delivered. Operations roles can be centralized or arranged by line of business or organization, according to individual company preferences.

Life cycles in the governance enablement profile

Entities from the various models in the governance enablement profile are governed by moving them through states in a life cycle. The life cycle state of an entity indicates its progress in the development process. The table below describes the life cycles that are defined in the governance enablement profile and the entities that pass through them.

Name	Usage	Governed entity
Asset life cycle	Govern an entity from initial identification to retired	DOU, service interface specification, and schema specification
Capability life cycle	Govern a capability from initial identification to retired	Business capability, business application, business process, and business service
SOA life cycle	Govern a capability version from initial identification to retired	Capability version, application version, process version, and service version
SLD life cycle	Govern an SLD from initial identification to retired	SLD
SLA life cycle	Govern an SLA from initial identification to retired	SLA
Endpoint life cycle	Govern an endpoint from being approved for use to retired	Service endpoint, MQ service endpoint, SOAP service endpoint, and extension service endpoint

5.2 Introduction to the WebSphere Service Registry and Repository Studio

WebSphere Service Registry and Repository Studio is a stand-alone Eclipse application that you can use to complete the following tasks:

- Create and edit WebSphere Service Registry and Repository configuration profiles
- Generate business models, classification systems, and life cycles from Unified Modeling Language (UML) diagrams
- Create and edit stored artifacts
- Generate custom reports

WebSphere Service Registry and Repository Studio is available for Microsoft Windows® and Linux® platforms.

Because WebSphere Service Registry and Repository Studio is based on Eclipse, the suite also includes the standard plug-ins that are shipped with the Eclipse framework, such as the ability to synchronize with a Concurrent Versions System (CVS) code control server. You can also install additional Eclipse plug-ins into the environment.

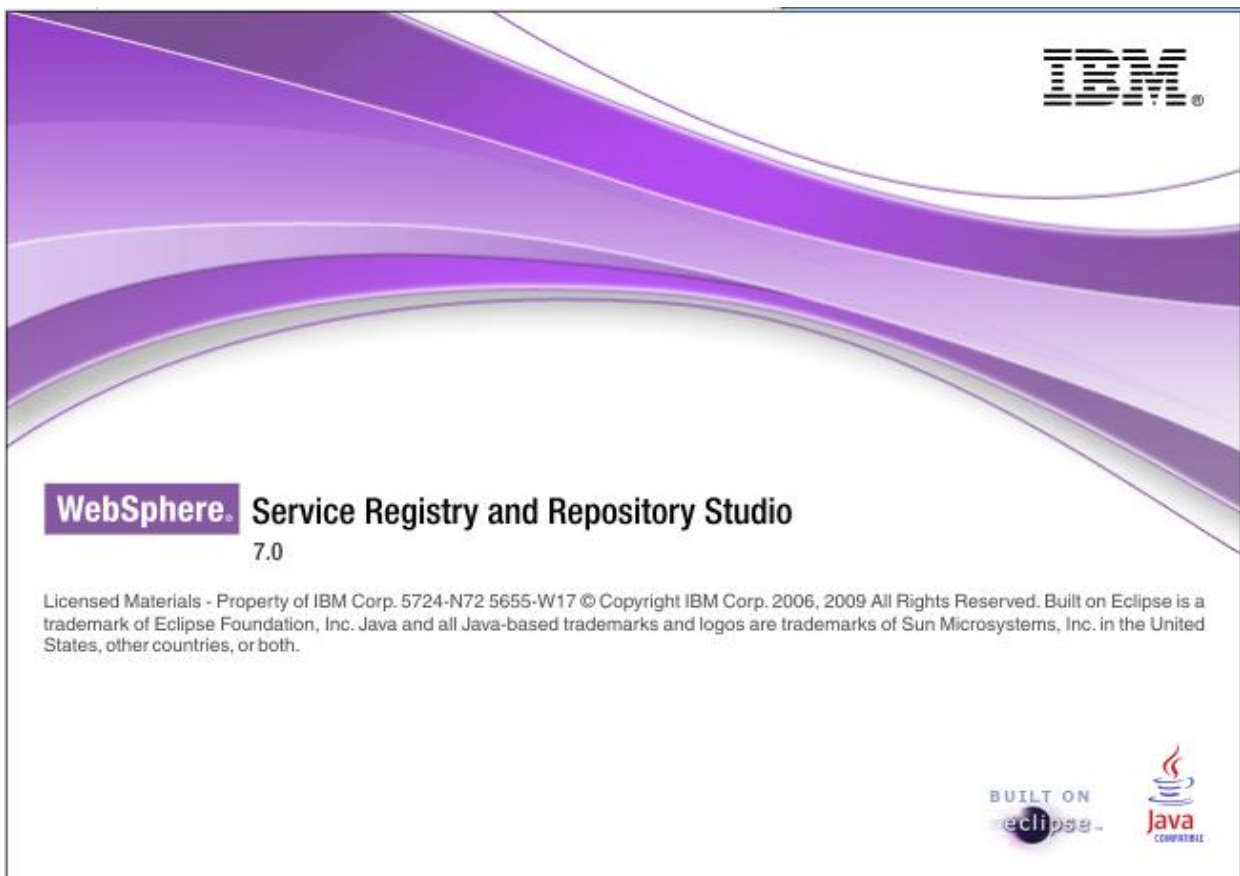
In this lab we use the modeling component of WebSphere Service Registry and Repository Studio to explore the Governance Model in WebSphere Service Registry and Repository. Specifically, you will explore:

- business models
- classification systems
- life cycles.

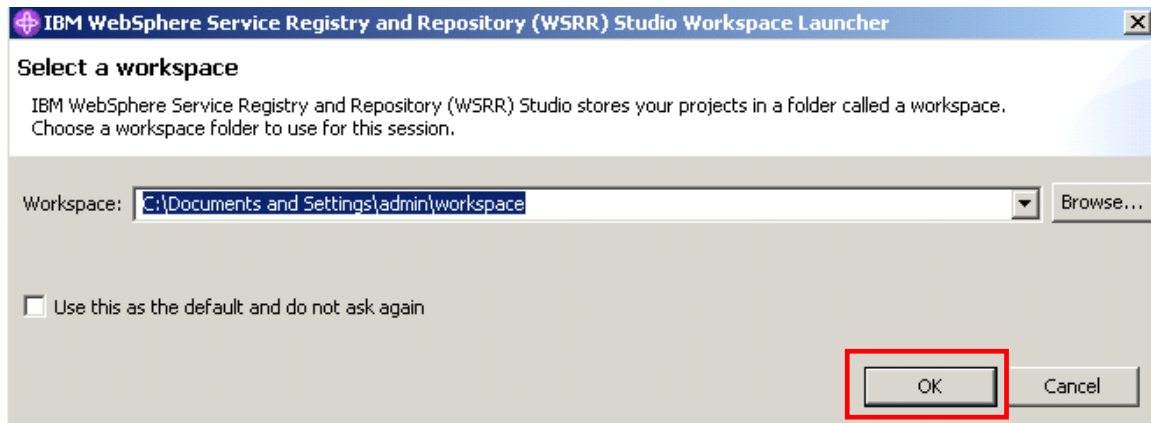
5.3 Start WebSphere Service Registry and Repository Studio

- __1. Start WebSphere Service Registry and Repository Studio
 - __a. Open the **WSRR PoT Shortcuts** folder on the desktop
 - __b. Click the shortcut **WSRR Studio** to start the studio.

The splash screen will appear.

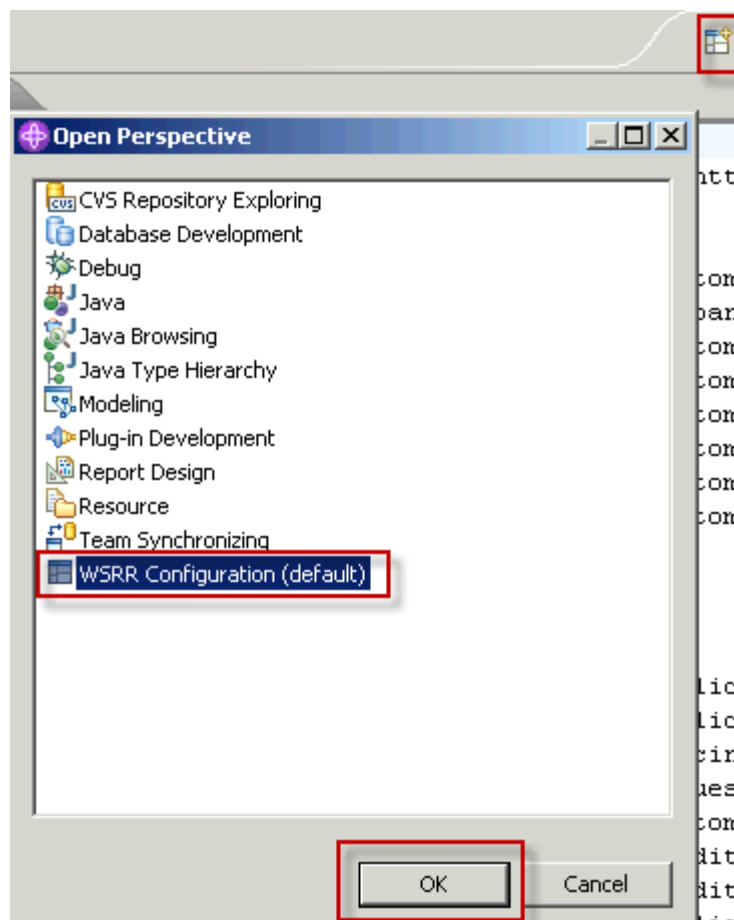


- __c. Click OK to accept the default workspace

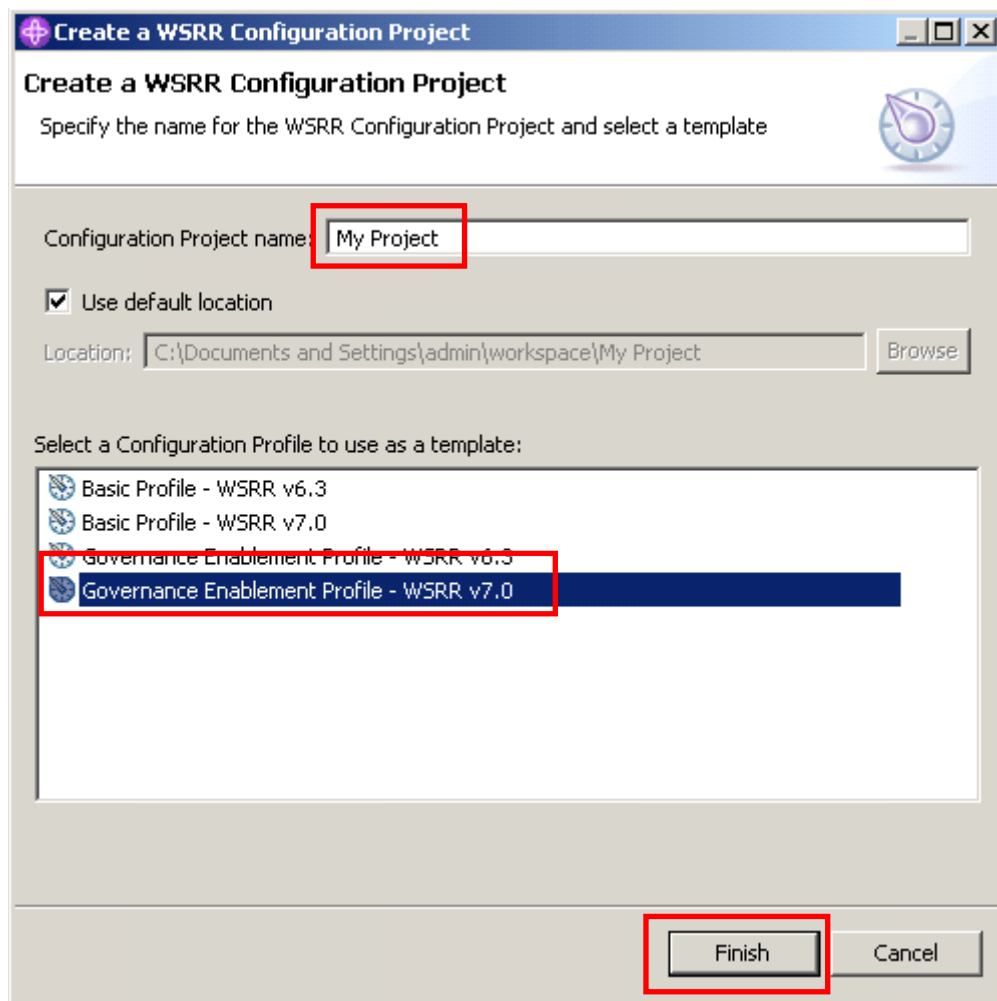


- __2. Open the WebSphere Service Registry and Repository Configuration perspective

- __a. Click the perspective icon, select other, select WebSphere Service Registry and Repository Configuration, and click OK.

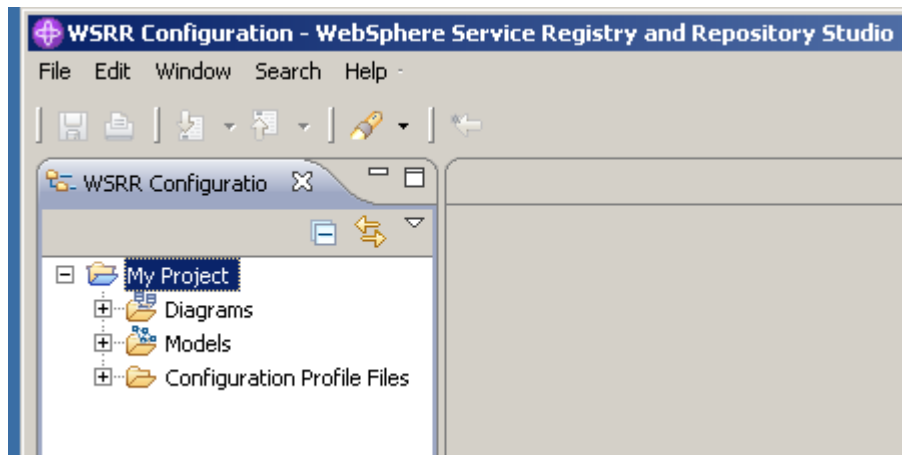


- __3. Create a new WebSphere Service Registry and Repository Configuration Project
 - __a. Click **File** → **New** → **WSRR Configuration Project**
 - __b. Enter a project name, select Governance Enablement Profile – WebSphere Service Registry and Repository v7.0, and click **Finish**



__4. Examine the contents of the Governance Enablement Profile

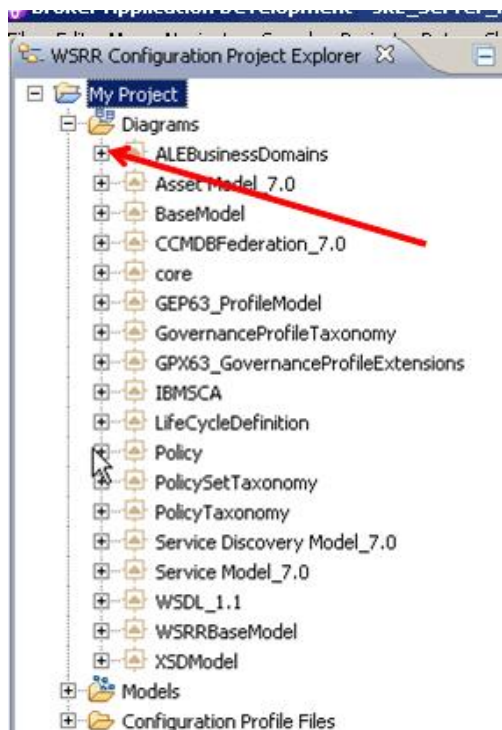
__a. Expand the project



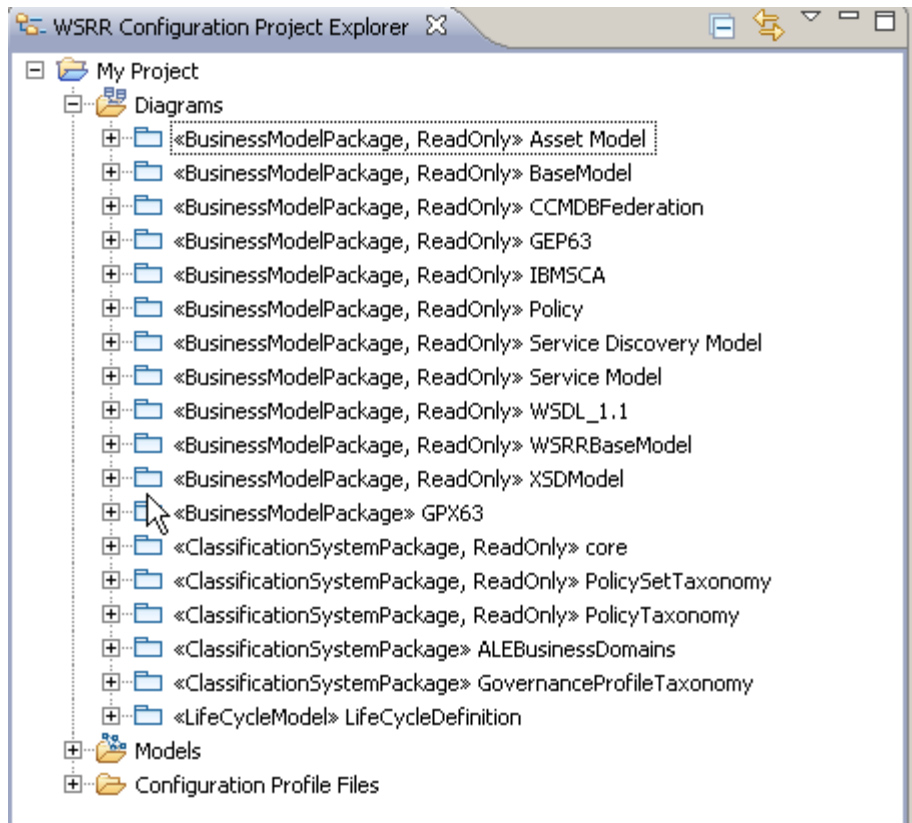
The configuration project consists of the following folders:

- Diagrams. Includes the UML Model of the WebSphere Service Registry and Repository business models, life cycles, and classification systems.
- Models Includes the same UML models as the diagrams folder as well as all the individual artifacts that make up the diagram.
- Configuration Profile Files Includes all the artifacts that make up a standard WebSphere Service Registry and Repository configuration profile.

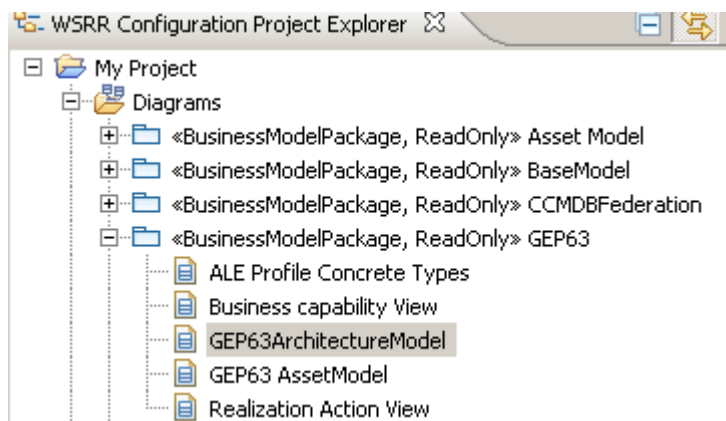
__b. Expand Diagrams and click each entry as shown



- ___c. As you do so each name is expanded to include its package name. Because the list is maintained in sorted order, this can be a little confusing. Do this for all items in the Diagrams folder.



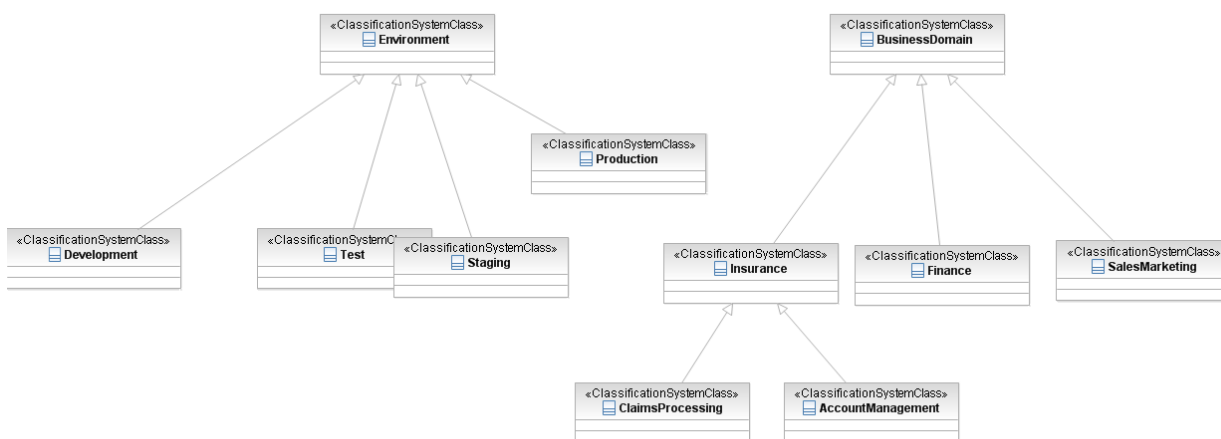
- ___d. You will see that there are several Models and Classification Systems, but a single Life Cycle Definition.
- ___e. Note that these are all read only. To edit the profile, you must explicitly write enable the packages that you wish to change.
- ___f. Expand <BusinessModelPackage, ReadOnly> GEP63 and double click on GEP63ArchitectureModel



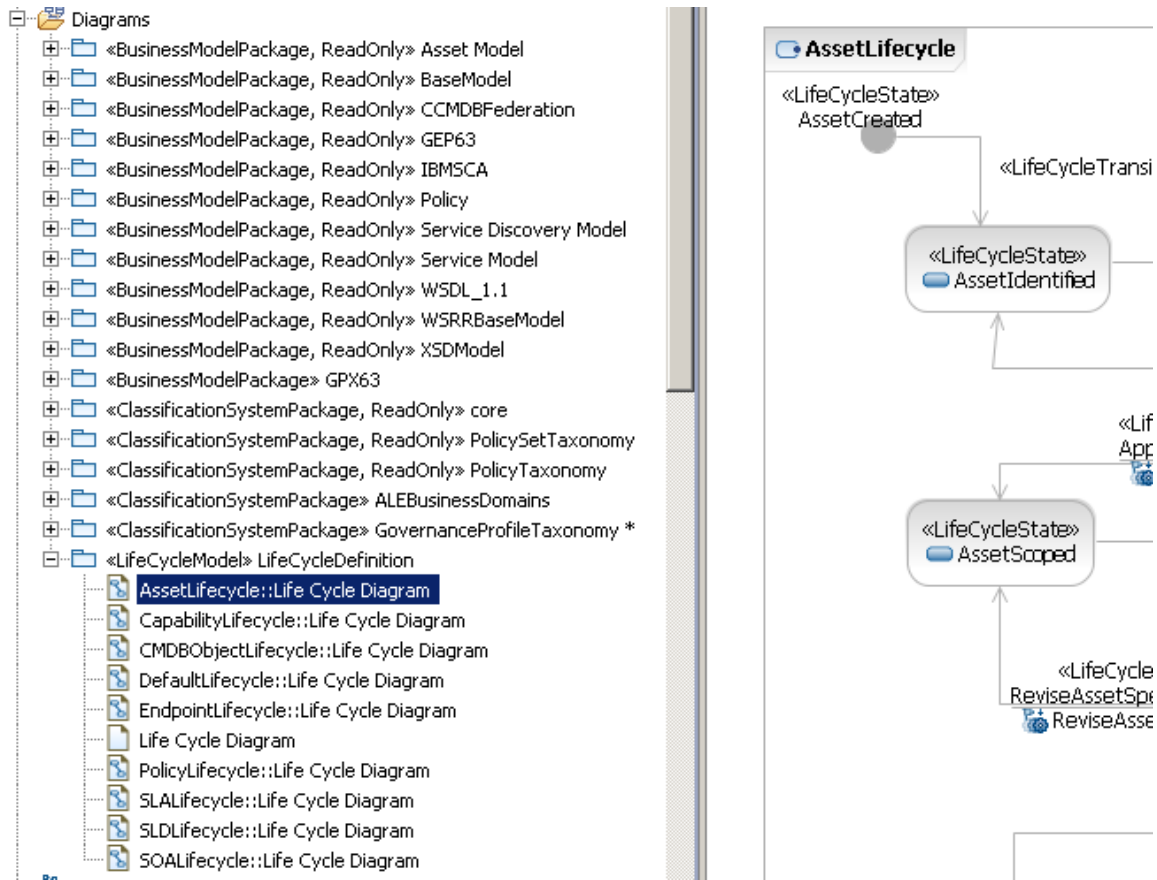
to open its UML diagram. (Version 6.3 of WebSphere Service Registry and Repository preceded the current version, and this model has been brought forward unchanged.) Note that the diagram is arranged into layers:

- Business Asset Governance Layer
- Technical Asset Governance Layer
- SOA Governance Layer
- Service Model Layer

__g. Expand <ClassificationSystemPackage>GovernanceProfileTaxonomy and click on ClassificationSystemDiagram



- __h. Expand <LifeCycleModel> LifeCycleDefinition and click AssetLifecycle::Life Cycle Diagram



- __i. Feel free to examine other parts of the governance enablement profile.
- __j. Close WebSphere Service Registry and Repository Studio (do not save any changes).
- __k. You have completed the lab.

5.4 Lab Review

During this lab, you:

- Were introduced to WebSphere Service Registry and Repository Studio
- Examined parts of the models, classification systems, and life cycles contained in the governance enablement profile.



CONGRATULATIONS!

You have successfully completed Lab 5 - Examining the Governance Model in WebSphere Service Registry and Repository with WebSphere Service Registry and Repository Studio.

Lab 6 Defining a WebSphere MQ Application to WebSphere Service Registry and Repository

6.1 Lab introduction

WebSphere MQ applications can be considered as services within a service-oriented architecture (SOA): they are loosely coupled and interoperate irrespective of the platforms on which they run.

Registering WebSphere MQ applications as services in WebSphere Service Registry and Repository brings benefits in the following areas.

- WebSphere MQ applications are catalogued as reusable assets.
- They become subject to SOA Governance
- You can perform impact analysis.
- Management and auditing.

WebSphere MQ applications can be described as services based on the WebSphere MQ Service Definition specification and the WebSphere MQ Internationalized Resource Identifier (IRI) specification. These can be downloaded from:

<http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24017518>

The WebSphere MQ Service Definition specification defines the standard for describing WebSphere MQ applications as services using WSDL and the WebSphere MQ IRI specification defines the standard for addressing WebSphere MQ message destinations as URIs in a service definition.

WebSphere MQ V7 provides a Service Definition Wizard in WebSphere MQ Explorer, which enables creation of WSDL for a WebSphere MQ application. This wizard allows easy creation of WebSphere MQ WSDL using WebSphere MQ Service Binding and generates the WSDL document based on WebSphere MQ Service Definition and WebSphere MQ IRI specifications.

In this lab you will:

- Use the WebSphere MQ Service Definition Wizard to define a WebSphere MQ application as a service and generate a WSDL document
- Load the document into WebSphere Service Registry and Repository
- Examine the WebSphere MQ entities created by WebSphere Service Registry and Repository
- Perform impact analysis on a WebSphere MQ Queue manager and explore the relationships between a WebSphere MQ service and the entities it depends on.

6.2 WebSphere MQ Service Definition Wizard

__1. Start WebSphere MQ Explorer

__a. Open the **WSRR PoT Shortcuts** folder on the desktop

__b. Click the shortcut WebSphere MQ Explorer.

The WebSphere MQ Explorer opens.

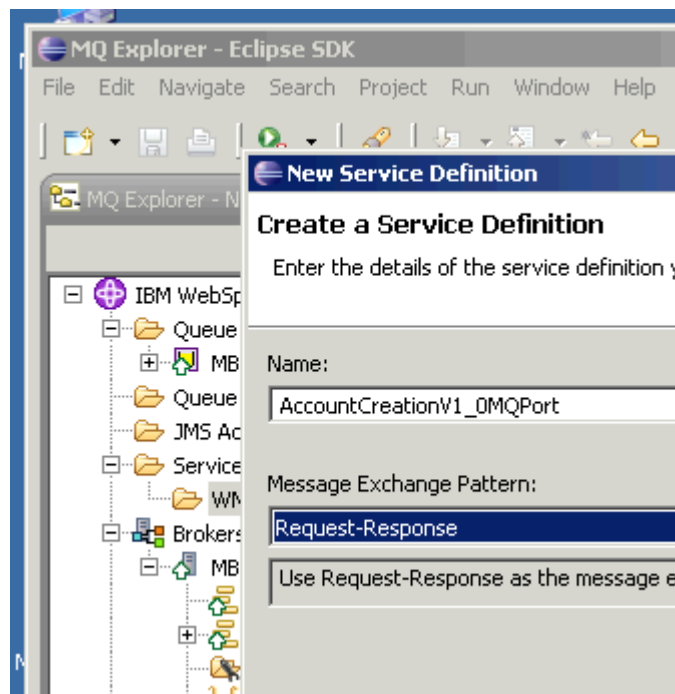
__2. Create a new WebSphere MQ Service Definition

__a. In the WebSphere MQ Explorer – Navigator window expand **Service Definition Repositories**.

__b. Right-click **WMQServices** and select **New -> Service Definition**

The New Service Definition window opens.

__c. Enter a name of **AccountCreationV1_OMQPort** for the new service and select the Request-Response message exchange pattern as shown below.



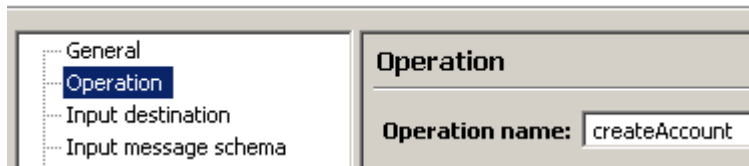
A Message Exchange Pattern describes the message pattern required by the WebSphere MQ application for which the WSDL document is being generated. WebSphere MQ Service Definition provides support for Request-Response and One-Way message exchange patterns.

__d. Click **Next**

__e. Click **Next** again to select a Binding Type of WebSphere MQ for the service.

WebSphere MQ Service Binding provides support for native WebSphere MQ and WebSphere MQ SOAP binding. The Binding Type should be selected, depending on whether the application expects a WebSphere MQ message or a SOAP message.

- __f. Select **Operation** and enter **createAccount** for the operation name

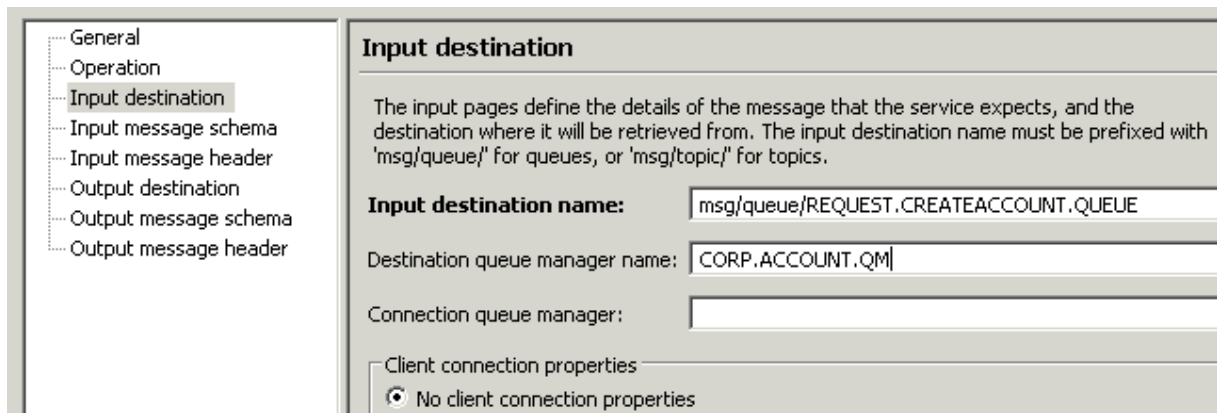


- __g. Select **Input destination** and provide:

input destination name: msg/queue/REQUEST.CREATEACCOUNT.QUEUE

destination queue manager name: CORP.ACCOUNT.QM

Leave Client connection properties set to **No client connection properties.**



__h. Select Input message schema and enter:

Inbound datatype: CustomerBO

Import schema file: AccountCreationSchema.xsd

Import namespace: http://www.jkhl.com/Account

To define the schema for the request message payload.

The screenshot shows the 'Input message schema' configuration window. On the left, a tree view lists various configuration options, with 'Input message schema' highlighted. The main area on the right is titled 'Input message schema' and contains three input fields: 'Inbound datatype' (CustomerBO), 'Import schema file' (AccountCreationSchema.xsd), and 'Import namespace' (http://www.jkhl.com/Account).

__i. Select Output destination and provide:

output destination name: msg/queue/RESPONSE.CREATEACCOUNT.QUEUE

destination queue manager name : CORP.ACCOUNT.QM

Leave connection properties unchanged.

The screenshot shows the 'Output destination' configuration window. On the left, a tree view lists various configuration options, with 'Output destination' highlighted. The main area on the right is titled 'Output destination' and contains three input fields: 'Output destination name' (msg/queue/RESPONSE.CREATEACCOUNT.QUEUE), 'Destination queue manager name' (CORP.ACCOUNT.QM), and 'Connection queue manager' (empty). Below these fields, there are two radio buttons for 'Client connection properties', with 'No client connection properties' selected.

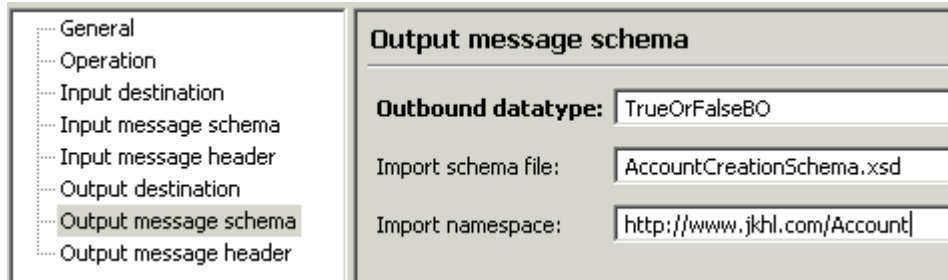
__j. Select Output message schema and enter:

Outbound datatype: TrueOrFalseBO

Import schema file: AccountCreationSchema.xsd

Import namespace: http://www.jkhl.com/Account

To define the schema for the response message payload.

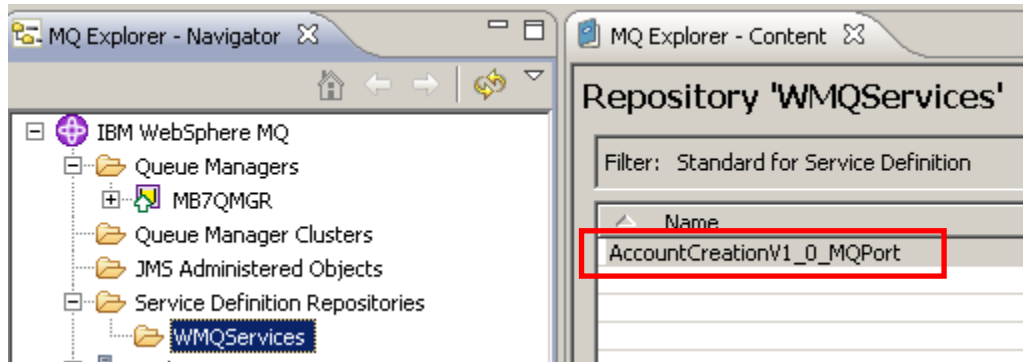


__k. Click Finish to create the service definition.

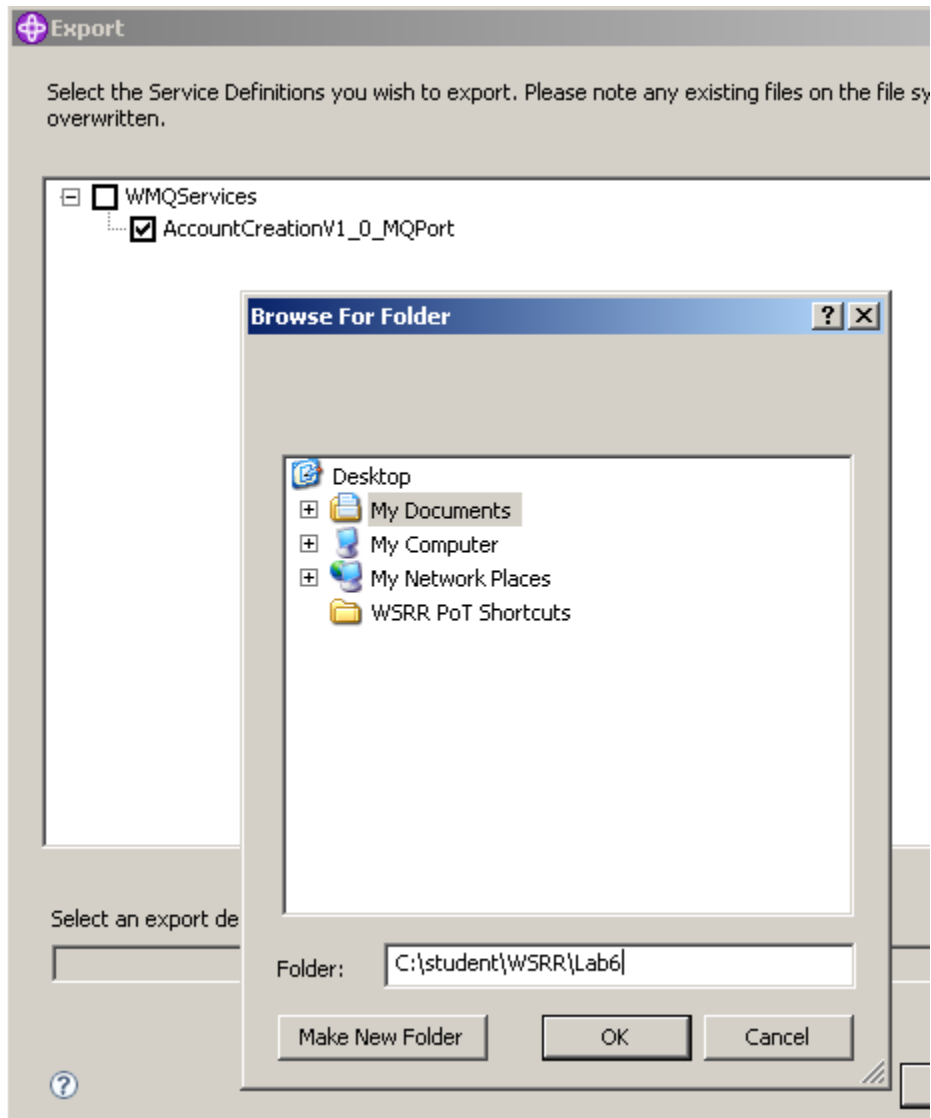
This process generates the AccountCreationV1_0_MQPort service definition.

__3. Export the WebSphere MQ Service definition from WebSphere MQ Explorer.

__a. Right-click **AccountCreationV1_0_MQPort**

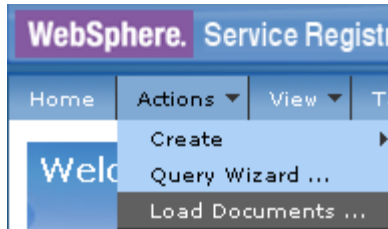


__b. and select **Export**



- __c. Browse for **C:\student\WSRR\Lab6**
- __d. Click **OK**
- __e. Click **OK** again.
- __f. This generates the AccountCreationV1_0_MQPort.wsd document in C:\student\WSRR\Lab6\WMQServices
- __g. Close WebSphere MQ Explorer

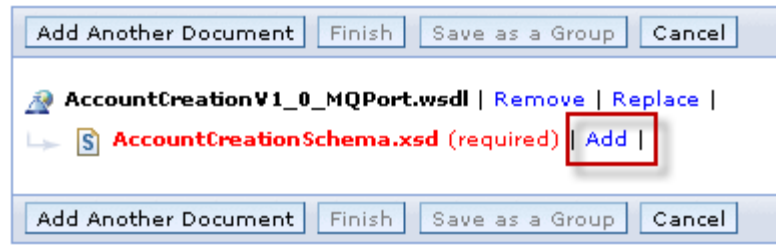
- __4. Loading the WebSphere MQ Service definition into WebSphere Service Registry and Repository
 - __a. Log on to WebSphere Service Registry and Repository as Connie, and select the Development perspective.
 - __b. Select **Actions -> Load Documents**



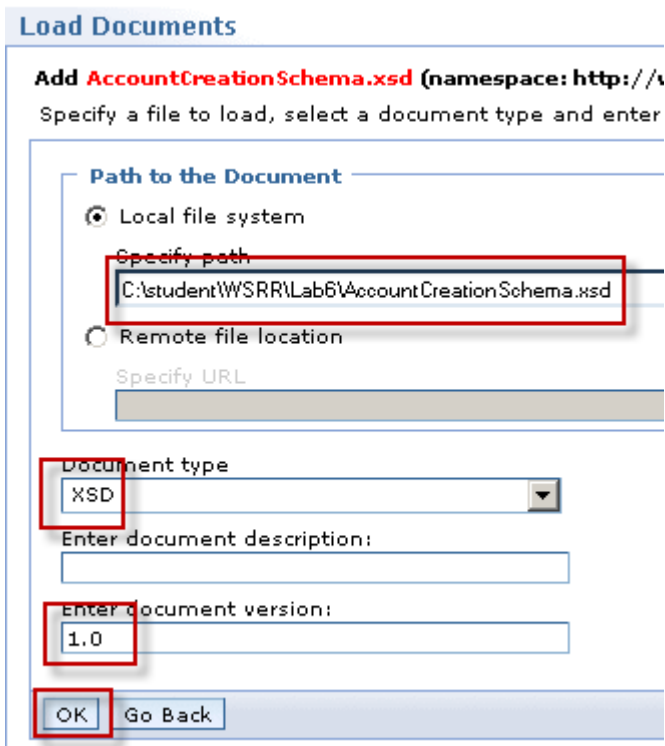
- __c. Browse to **C:\student\WSRR\Lab6\WMQServices** and select **AccountCreationV1_0_MQPort.wsdl**
- __d. Ensure document type is **WSDL** and enter a version of **1.0**.

A screenshot of the 'Load Documents' dialog box. The 'Local file system' radio button is selected. The 'Specify path' text box contains 'C:\student\WSRR\Lab6\WMQServices\AccountCreationV1_0_MQPort.wsdl'. The 'Document type' dropdown menu is set to 'WSDL'. The 'Enter document version:' text box contains '1.0'. The 'OK' button is highlighted with a red rectangular box.

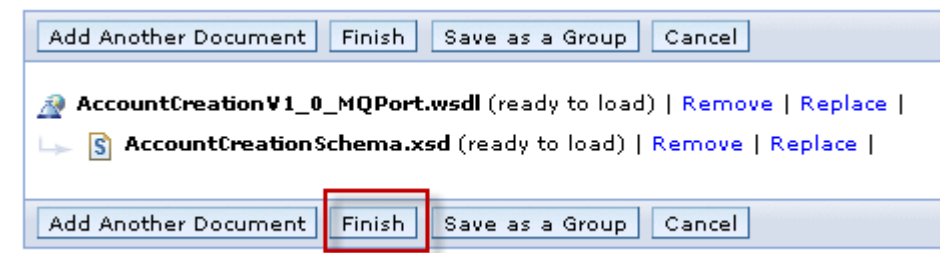
- __e. Click **OK**.
- __f. WebSphere Service Registry and Repository detects that the AccountCreationSchema.xsd is referenced by the WSDL.



- __g. Click **Add**.
- __h. Browse to **C:\student\WSRR\Lab6** and select AccountCreationSchema.xsd.
- __i. Ensure document type is **XSD** and enter a version of **1.0**.



- __j. Click **OK**.



__k. Click **Finish**.

WebSphere Service Registry and Repository confirms that the documents have been loaded

Documents Loaded Successfully

The following documents have been loaded into the repository:

Name	Description	Namespace	Version
AccountCreationV1_0_MQPort.wsdl		http://tempuri.org/AccountCreationV1_0_MQPort	1.0
AccountCreationSchema.xsd		http://www.jkhl.com/Account	1.0

__5. WebSphere MQ Objects in WebSphere Service Registry and Repository

When loading a WSDL document, WebSphere Service Registry and Repository automatically parses the document and builds WebSphere MQ entities from the service definition. These entities can be viewed from Development and SOA Governance Perspectives.

__a. Select **View** → **Service Model** → **MQ Services** → **Queue Managers**

The Queue Manager entity created by WebSphere Service Registry and Repository appear

Select	Name	Graph
<input type="checkbox"/>	CORP.ACCOUNT.QM	

Total: 1

__b. Click the WebSphere MQ Queue Manager **CORP.ACCOUNTS.QM** object to see its properties and dependent WebSphere MQ Queues

MQ Queue Manager

MQ Queue Managers > CORP.ACCOUNT.QM

Detail view for MQ Queue Manager. An MQ Queue Manager definition represents a WebSphere MQ Queue Mana

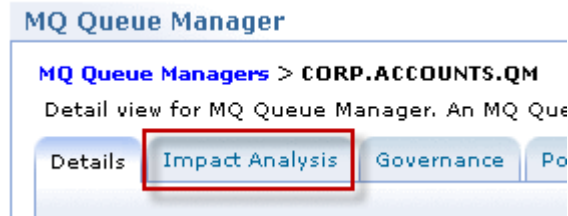
Details | **Impact Analysis** | Governance | Policy | Activity

[Edit Properties](#) | [Edit Relation](#)

<p>Properties</p> <p>*Name CORP.ACCOUNT.QM</p> <p>Description</p> <p>Namespace</p> <p>Owner UNAUTHENTICATED</p> <p>Version</p>	<p>Links</p> <ul style="list-style-type: none"> Graphical View Applied Policies Applied Policy Attachments <p>Dependent Entities</p> <p>Dependent MQ Queue(s)</p> <p>REQUEST.CREATEACCOUNT.QUEUE RESPONSE.CREATEACCOUNT.QUEUE</p> <p>Dependent Connection(s)</p> <p>None</p> <p>Classifications</p> <p>MQ Queue Manager</p>
---	---

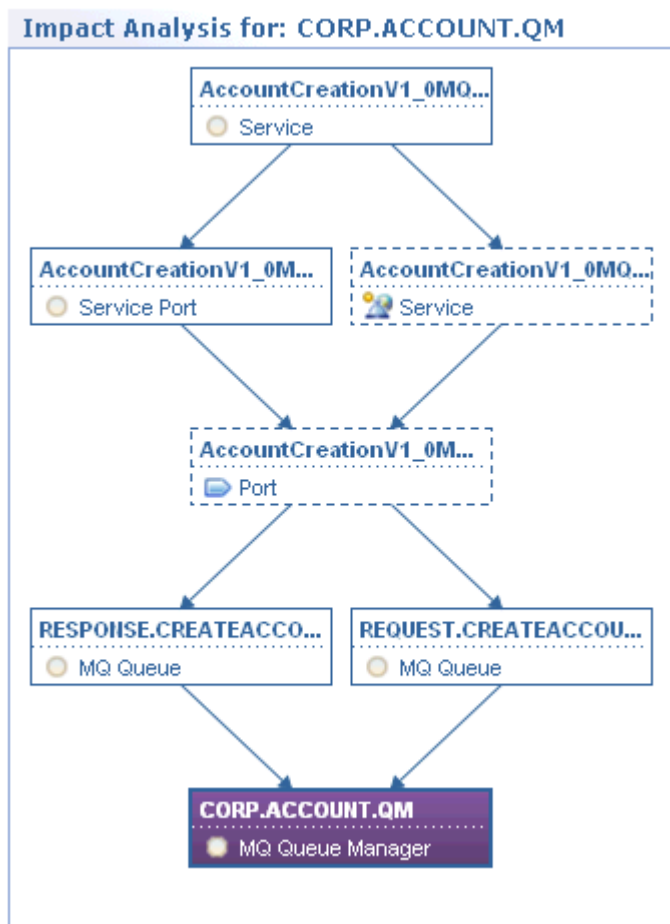
__6. Perform impact analysis on the Queue Manager entity CORP.ACCOUNTS.QM.

- __a. Click **Impact Analysis** on the Queue Manager properties window



- __b. On the next panel click **GO**.

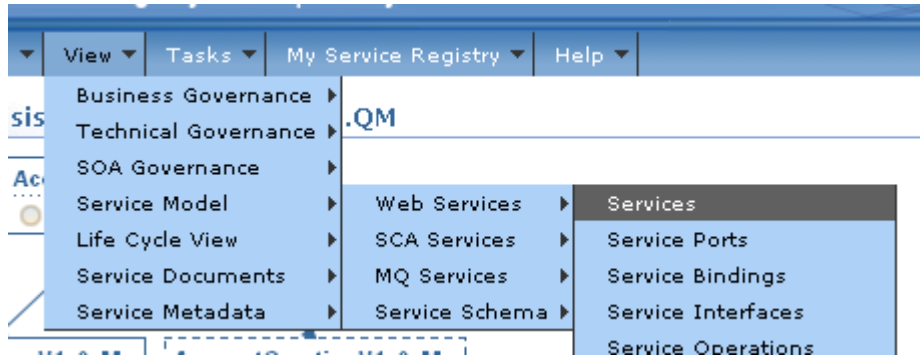
A graphical view of the Impact Analysis for the Queue Manager CORP.ACCOUNTS.QM entity will appear. This shows which services (currently only one) have a dependency on this Queue Manager. If you select **Vertical** orientation it will appear as shown below.





- __7. View the complete WebSphere MQ Service

It is also possible to see the complete WebSphere MQ Service and determine which operations, resources, and message exchange patterns are being used.

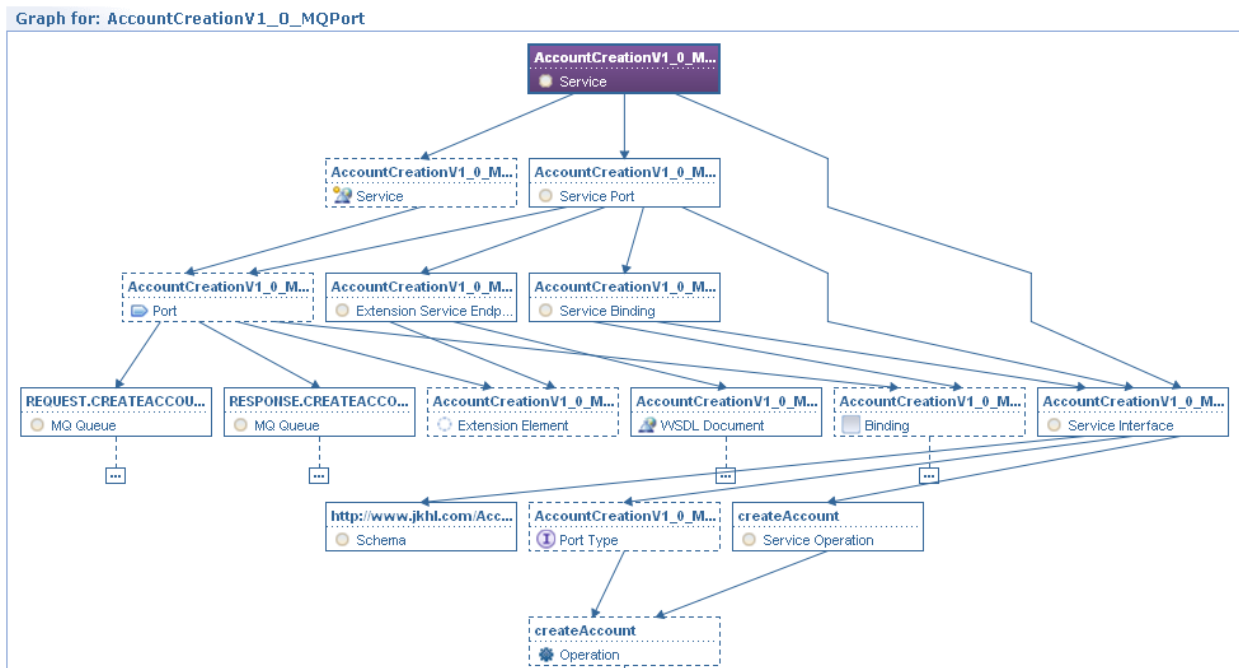
c. Select **View** → **Service Model** → **Web Services** → **Services**



d. Click the graph icon for the service AccountCreationV1_0_MQPort

Select	Name	Graph	Namespace	
<input type="checkbox"/>	AccountCreationV1_0_MQPort		http://tempuri.org/AccountCreationV1_0_MQPort	1
<input type="checkbox"/>	EligibilityV1_0		http://jkhle.itso.ibm.com/EligibilityV1	1
Total: 2				

The graphical view of the service AccountCreationV1_0_MQPort, showing the objects on which it is dependent appears.



e. Logout from WebSphere Service Registry and Repository.

You have completed the lab.

6.3 Lab Review

During this lab, you:

- Used the WebSphere MQ Service Definition Wizard to define a WebSphere MQ application as a service and generate a WSDL document
- Loaded the document into WebSphere Service Registry and Repository
- Examined the WebSphere MQ entities created by WebSphere Service Registry and Repository
- Performed impact analysis on a WebSphere MQ Queue manager and explored the relationships between a WebSphere MQ service and the entities it depends on.



CONGRATULATIONS!

You have successfully completed Lab 6 - Defining a WebSphere MQ Application to WebSphere Service Registry and Repository

Lab 7 **Version control with WebSphere Service Registry and Repository and WebSphere Message Broker**

7.1 **Introduction to integrating WebSphere Message Broker with WebSphere Service Registry and Repository**

WebSphere Message Broker can be used as an enterprise service bus for connectivity of enterprise applications over a wide range of protocols and message formats. It can access WebSphere Service Registry and Repository at run time to retrieve service end points, policy documents and other data. Together, WebSphere Service Registry and Repository and WebSphere Message Broker can dynamically manage the SOA infrastructure.

WebSphere Message Broker provides two built-in primitive nodes that can be used to dynamically retrieve service endpoints or other resources from WebSphere Service Registry and Repository. These are:

- EndpointLookup node
- RegistryLookup node

The EndpointLookup node is used to retrieve a service endpoint for a WSDL service from WebSphere Service Registry and Repository. The node inserts the retrieved service endpoint in the local environment tree for use by subsequent SOAP or HTTPRequest nodes to call the Web service.

The RegistryLookup node is used to retrieve any artifacts that are stored in WebSphere Service Registry and Repository, for example, WSDL, XML schema, XSLT, policy documents, and so on. This is a general purpose node that can be used to query and retrieve any document or meta data from WebSphere Service Registry and Repository. The node inserts the retrieved artifacts in the local environment tree for use in subsequent processing of the message flow.

In this lab we will work with an EndpointLookup node.

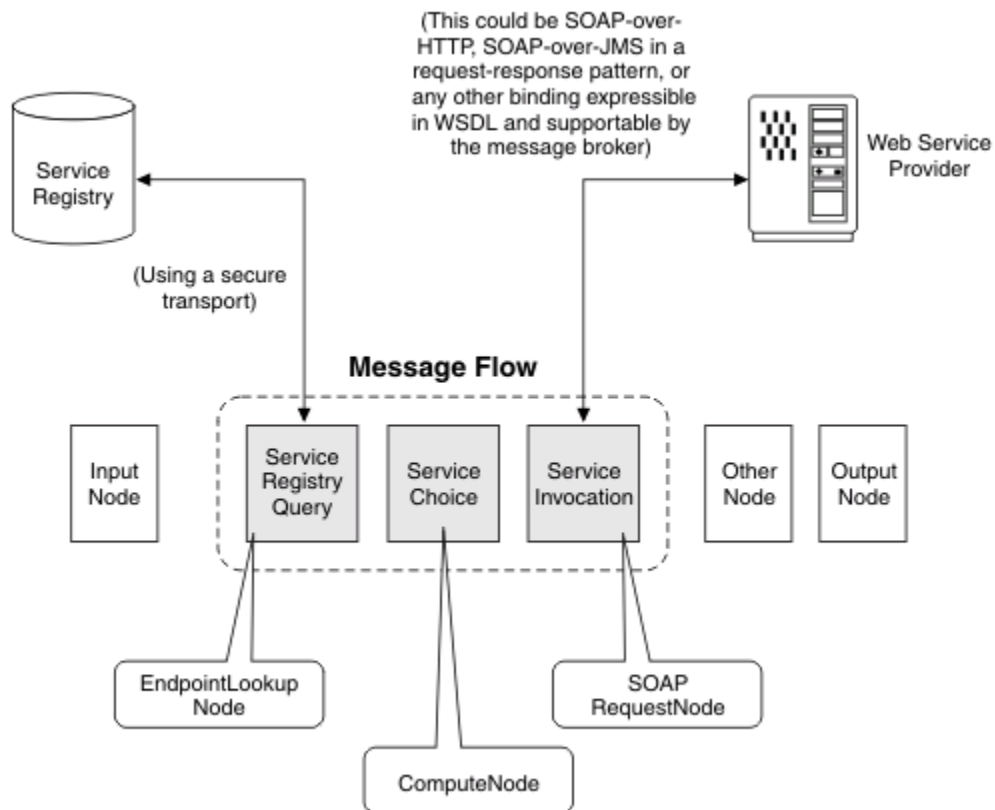
7.2 **Lab scenario**

This Lab demonstrates how an EndpointLookup node can be used to dynamically route a service request to an available service endpoint that is governed in WebSphere Service Registry and Repository.

JHKL has a Customer Update service running in their production environment that requires an upgrade. To protect applications from changes to the Customer Update service, JKHL utilizes an EndpointLookup node in a message flow to look up the online endpoint of the Customer Update service in WebSphere Service Registry and Repository.

The version that is currently running in production is Version 1_0. Once Version 2_0 is deployed and available, the endpoint for Version 1_0 will be classified as offline in WebSphere Service Registry and Repository. All service consumer requests for the Customer Update service will be directed to the endpoint for Version 2_0, enabling Version 1_0 to be deprecated with no impact on service consumers and with no interruption to the service.

2. The following diagram shows the architecture for accessing WebSphere Service Registry and Repository from WebSphere Message Broker.



In the lab we do not use a Compute Node to perform Service Choice, but go direct to Service Invocation using the endpoint retrieved from WebSphere Service Registry and Repository. The invoked web services are provided by two message flows in WebSphere Message Broker.

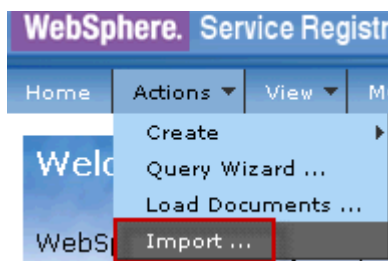
In this lab you will:

- Examine a Business Service with two Service Versions and related endpoints.
- Use a WebSphere Message Broker message flow to invoke the active version of the service.
- Make a new version active
- Take the original service off-line and observe that the new service version is selected.

7.3 Lab – Version control with WebSphere Service Registry and Repository and WebSphere Message Broker

During this lab you will be one of the operations team, but first you will set up the scenario by using the WebSphere Service Registry and Repository import facility.

- __1. Load the Customer Update business service, its service versions and WSDLs using the WebSphere Service Registry and Repository import facility.
 - __a. Log on to WebSphere Service Registry and Repository as **Admin**, with the Administrator perspective.
 - __b. Select **Actions -> Import**



- __c. Browse to **C:\student\WSRR\Lab7** and select **CustomerUpdateServiceImport.zip**
- __d. Click **OK**.

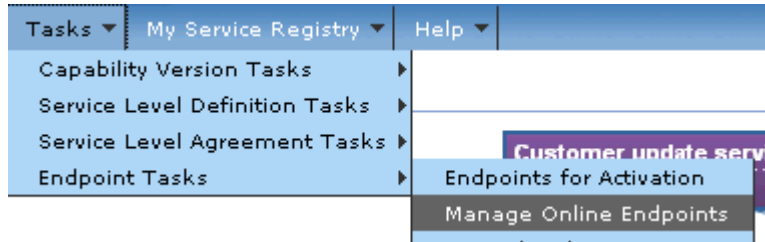
The Import Documents Summary displays.

Import Documents Summary		
The following items were successfully imported.		
Name	Description	Object Type
Common Services		concept
Customer		concept
Customer		concept
Customer		concept
Customer		concept
Customer update service	Updates customer information	concept
Customer update service (1.0)	Service for updating customer information	concept

- __e. Click **OK**.
- __f. Logout of WebSphere Service Registry and Repository.

Hover over the two Service Versions, and the Service Ports and SOAP Service Endpoints, and note that you have versions 1.0 and 2.0 of each.

- __d. Click **Tasks -> Endpoint Tasks -> Manage Online Endpoints.**



You can see that the version 1 endpoint is online.

Online Endpoints

This is the collection view of Endpoints in the "Online" life cycle candidate for routing to.

Preferences

Name	Graph
http://localhost:7800/DemoCustomer_v10/services/DemoCustomer	
Total: 1	

- __e. Click http://localhost:7800/DemoCustomer_v10/services/DemoCustomer

When its details appear, note that it is online and in production .Also note the Namespace and Port Name values. WebSphere Message Broker will use this information to retrieve the endpoint.

Namespace

Owner

Version

Last modified

***Endpoint Type**

***Port Name**

[CustomerUpdate_v1_0.wsdl](#)

WSDL Ports

[Customer](#)

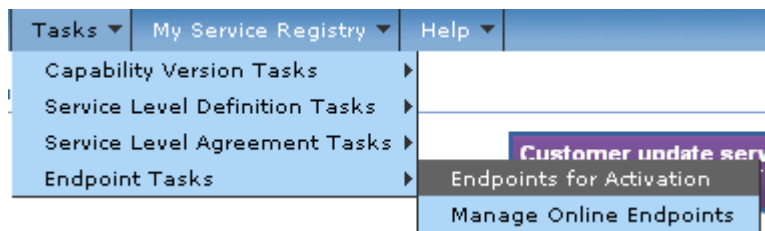
Governance State

Online

Classifications

Online
 Production
 SOAP Service Endpoint

- __f. Click **Tasks -> Endpoint Tasks -> Endpoints for Activation.**



You can see that the version 2 endpoint is offline.

Offline Endpoints

This is the collection view of Endpoints in the "Offline" life cycle state p by potential consumers but if protected by mediations that can access

Preferences

Select	Name	Graph
<input type="checkbox"/>	http://localhost:7800/DemoCustomer_v20/services/DemoCustomer	

- __g. Click http://localhost:7800/DemoCustomer_v20/services/DemoCustomer

When its details appear, note that it has the same Namespace and Port Name as the version 1 endpoint and is also classified as Production, but it is Offline.

Namespace http://jkhle.itso.ibm.com	CustomerUpdate_v2_0.wsdl
Owner UNAUTHENTICATED	WSDL Ports Customer
Version 2.0	<input type="checkbox"/> Governance State Offline
Last modified Wednesday, February 17, 2010 11:49:14 AM	<input type="checkbox"/> Classifications <input type="checkbox"/> Offline <input checked="" type="checkbox"/> Production SOAP Service Endpoint
*Endpoint Type SOAPAddress	
*Port Name Customer	

- __h. Log out of WebSphere Service Registry and Repository.

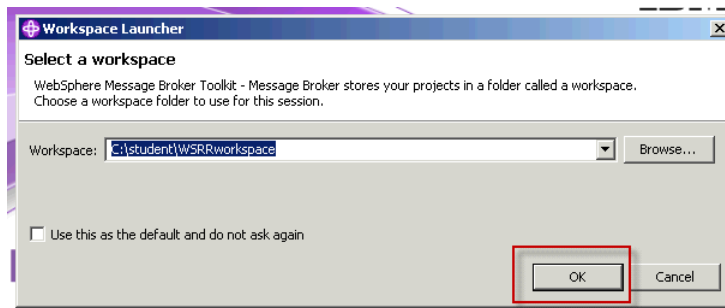
- __3. Start the WebSphere Message Broker Toolkit and examine the pre-configured message flow.

- __a. In the WebSphere Service Registry and Repository PoT shortcuts folder double click the icon for the WebSphere Message Broker Toolkit.



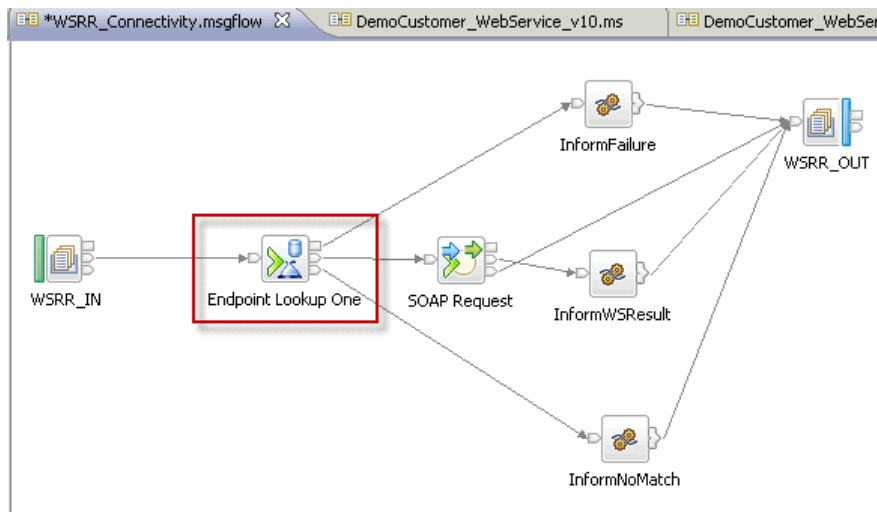
WebSphere Message
Broker Toolkit 7.0

- __b. Select the workspace **C:\student\WSRRworkspace** and click **OK**.

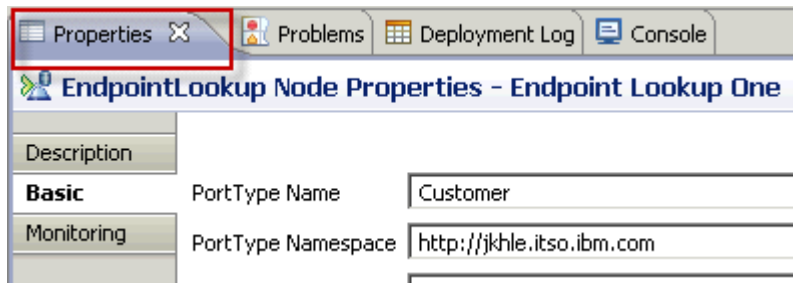


When the toolkit opens, the **WSRR_Connectivity.msgflow** will be visible. (If it is not, find it in the navigator panel on the left hand side and double click to open it.

- __c. Click on the **Endpoint Lookup One** icon to select it.



- __d. Click the **Properties** tab, and examine the properties of Endpoint Lookup One.



It has been configured to query WebSphere Service Registry and Repository based on:

- __i. PortType Name = Customer
- __ii. PortType Namespace = <http://jkhle.itso.ibm.com>

__iii. and a Classification:

Classification
<http://www.ibm.com/xmlns/prod/serviceregistry/lifecycle/v6r3/LifecycleDefinition#Online>

__iv.

Classification URI

Note that a fully qualified OWL URI is used here. To discover the URI for Online in WebSphere Service Registry and Repository:

1. Log on with the Configuration perspective.
2. Select **Active Profile -> Classification Systems**
3. Click **Governance Profile Lifecycle Classifications**
4. Click **Classes**
5. Expand **State**
6. Expand **Service Endpoint Lifecycle**
7. Click **Online**

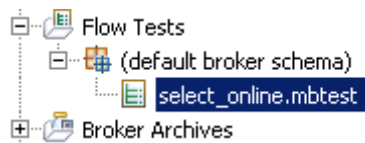
The URI is in the **Class ID** field.

__v. The Match policy is one so WebSphere Service Registry and Repository will return only one result.

__vi. Match Policy

__4. Run the message flow.

__a. In the navigation window, double click **select_online.mbttest**.



A new panel appears.

Message Flow Test Events

Enqueue

Dequeue

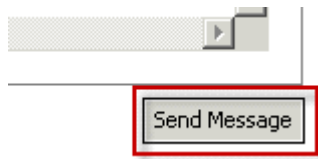
General Properties

Detailed Properties

Host:	<input type="text" value="localhost"/>
Port:	<input type="text" value="0"/>
Queue manager:	<input type="text" value="MB7QMGR"/>
Queue:	<input type="text" value="WSRR_IN"/>
Message	

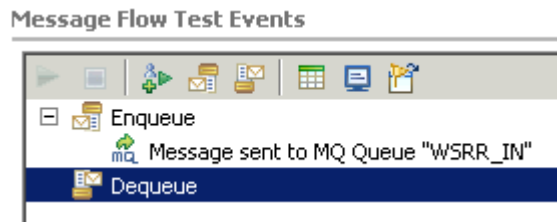
Note that **Enqueue** is already selected.

- __b. Click **Send Message**.



A message is sent to the WebSphere MQ Queue WSRR_IN.

- __c. Click **Dequeue** to select it.



- __d. Click **Get Message** to retrieve the response.



The retrieved message tells us that the version 1.0 Web service has been used.

Name	Value
NS2:updateCustomerResp	
xmlns:NS2	http://demo.sr.eis.ibm.com
updateCustomerReturn	Updated Customer via Webservice version 1.0

Possible Error Condition

Instead of the expected message, you may get the error message: **No more message available**

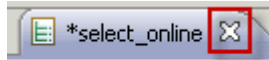
Exception message: **No more message available**

Exception trace: com.ibm.mq.MQException: MQJE001: Completion Code '2', Reason '2033'.

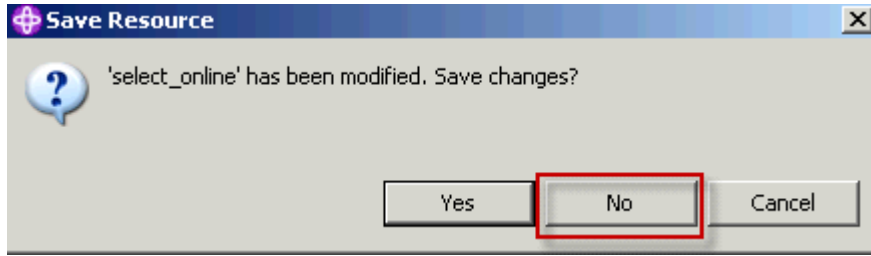
This means that the queue you are retrieving from is empty, but it may not be for long! Repeat steps **c** and **d** above until a message is received.

It is important not to ignore this message. Because we are using WebSphere MQ as the transport, you will always get a response to your request. If you send a new message before seeing the response to your original message, your requests and responses will get out of step. This can be very confusing.

- __e. Click the **X** in the **select_online** tab to close the Flow Test.



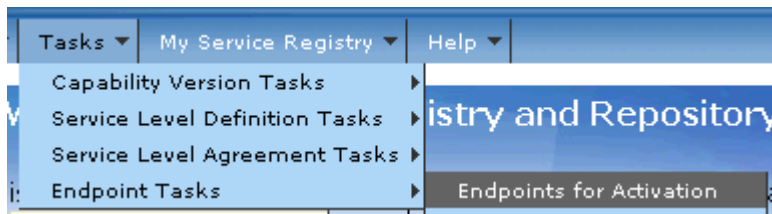
Do not save changes.



- __f. We will leave the WebSphere Message Broker Toolkit for now, but do not close it.

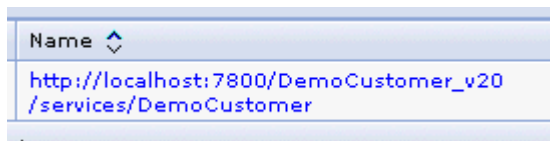
- __5. Transition the version 2.0 Endpoint to a state of Online.

- __a. Login to WebSphere Service Registry and Repository as Jon and select the **Operations** perspective.
- __b. Select **Tasks -> Endpoint Tasks -> Endpoints for Activation**.



A list of Offline Endpoints is displayed.

- __c. Click http://localhost:7800/DemoCustomer_v20/services/DemoCustomer



The details of this Endpoint are displayed.

__d. Click **Approve for Use**.

*Port Name

Service Version

*Service Namespace

*Service Name

Additional Properties

Back **Approve For Use** Retire From Use

The Endpoint is now Online.

Governance State
 Online

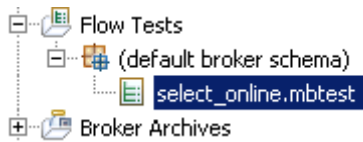
Classifications
 Online
 Production
 SOAP Service Endpoint

__e. Log out of WebSphere Service Registry and Repository.

__6. We now have two online endpoints. Let's see which one WebSphere Message Broker retrieves.

__a. Return to the WebSphere Message Broker Toolkit.

__b. In the navigation window, double click **select_online.mbstest**.



A new panel appears.

Message Flow Test Events

General Properties

Detailed Properties

Host:	localhost
Port:	0
Queue manager:	MB7QMGR
Queue:	WSRR_IN
Message	

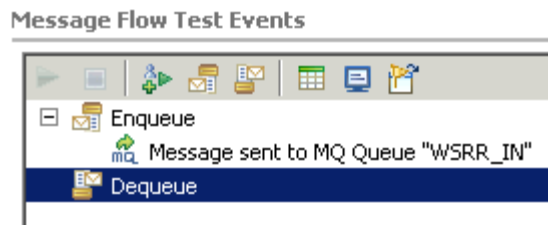
Note that **Enqueue** is already selected.

- __c. Click **Send Message**.



A message is sent to the WebSphere MQ Queue WSRR_IN.

- __d. Click **Dequeue** to select it.



- __e. Click **Get Message** to retrieve the response.



Note which version of the Web service has been used.

- __f. Close the Flow Test without saving it.
 __g. Repeat from step **_b.** a number of times.

You may get version 1.0 or version 2.0, but you cannot assume this.



If you need more certainty during transition, you could retrieve all matching endpoints and use a compute note to choose, say, the one with highest version number.

We will leave the WebSphere Message Broker Toolkit for now, but do not close it.

- __7. Take the version 1.0 endpoint offline
- __a. Log on to WebSphere Service Registry and Repository as Jon, and select the **Operations** perspective.
- __b. Select **Tasks -> Endpoint Tasks -> Manage Online Endpoints**.



The list of online endpoints is displayed.

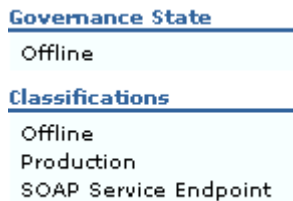
- __c. Click http://localhost:7800/DemoCustomer_v10/services/DemoCustomer

Details of the version 1.0 endpoint are displayed.

- __d. Click **Revoke from Use**.



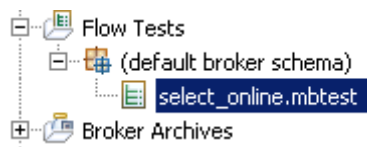
The endpoint is now Offline.



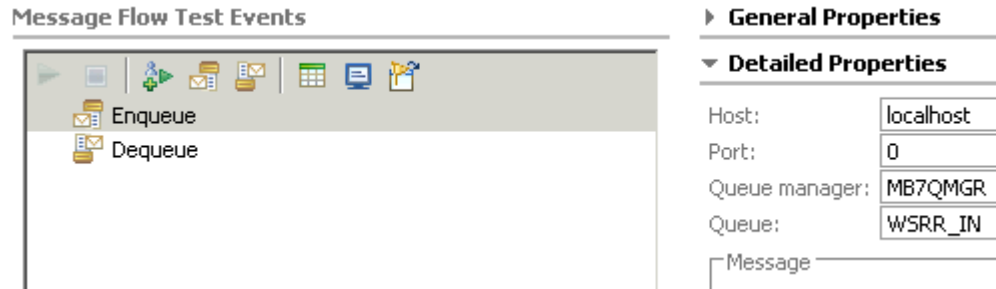
- __e. Logout from WebSphere Service Registry and Repository

__8. We now have a single online endpoint. Let's see which one WebSphere Message Broker retrieves.

- __a. Return to the WebSphere Message Broker Toolkit.
- __b. In the navigation window, double click **select_online.mbstest**.

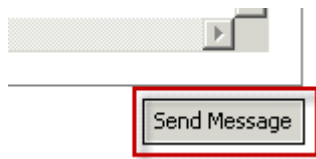


A new panel appears.



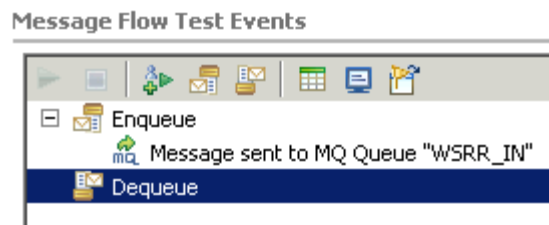
Note that **Enqueue** is already selected.

- __c. Click **Send Message**.



A message is sent to the WebSphere MQ Queue WSRR_IN.

- __d. Click **Dequeue** to select it.



- __e. Click **Get Message** to retrieve the response.



Note which version of the Web service has been used.

- __f. Close the Flow Test without saving it. Then repeat from step __b. a number of times. Version 2.0 should be selected every time.

You have successfully implemented version 2.0 without a service outage!

You have completed the lab.



Caching

So that the effects of making an endpoint online and offline are seen immediately, caching was disabled for this Lab

7.4 Lab Review

During this lab, you:

- Examined a Business Service with two Service Versions and related endpoints.
- Used a WebSphere Message Broker message flow to invoke the active version of the service.
- Made a new version active
- Took the original service off-line and observed that the new service version is selected.

7.5 Optional explorations

- __1. Revoke the version 2.0 service so that neither service is online, and run `select_online.mbttest`. You should get the message: No matching services.
- __2. Stop the WebSphere Service Registry and Repository server and run `select_online.mbttest`. (Only do this if you have plenty of time, because you will need to restart the server for the next lab.) You should get the message: Failure to retrieve Service from WebSphere Service Registry and Repository!
- __3. If you have WebSphere Message Broker skills you may like to experiment with the `RegistryLookup` node.



CONGRATULATIONS!

You have successfully completed Lab 7 - Version control with WebSphere Service Registry and Repository and WebSphere Message Broker

Lab 8 Exploring the WebSphere Service Registry and Repository Business Space

8.1 Introduction to WebSphere Service Registry and Repository Business Space

Business Space powered by WebSphere is a browser-based graphical user interface that lets business users interact with content from products in the WebSphere portfolio. WebSphere Service Registry and Repository Version 7.0 provides widgets that enable users to search, view details of items in the registry, perform key tasks, and view service reuse data. The spaces you create are collections of related content that can provide you with insight into your business and the capability to react to changes in it.

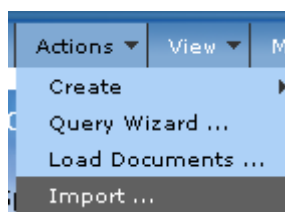
A space consists of pages that you define, or create based on pre-configured templates provided by each WebSphere product. WebSphere Service Registry and Repository provides two templates – a Service Registry and Repository template and a Policy Analytics template. Each page can contain one or more widgets that are configured to perform business tasks. You can have many spaces with each one having a different purpose. For example, a space with widgets from WebSphere Service Registry and Repository can be used to create and view items and data in the service registry. In addition, there are widgets that enable users to view data relating to the enforcement of governance validation policies, and WS-I compliance of WSDL documents.

In this lab you will explore a business space based on the Service Registry and Repository template. You will use the service registry search, collection, detail, and actions widgets to view and create data in WebSphere Service Registry and Repository. You will then explore Policy Analytics, particularly WS-I compliance.

8.2 Lab Setup

Here we will load some additional documents.

- __1. Login to WebSphere Service Registry and Repository as **admin**, and select the **Administrator** perspective.
 - __a. Select **Actions** -> **Import**



- __b. Browse to **C:\student\WSRR\Lab8** and select **AccountCreationService.zip** . Click **OK**.



The documents have been imported.

- __c. Click **OK**

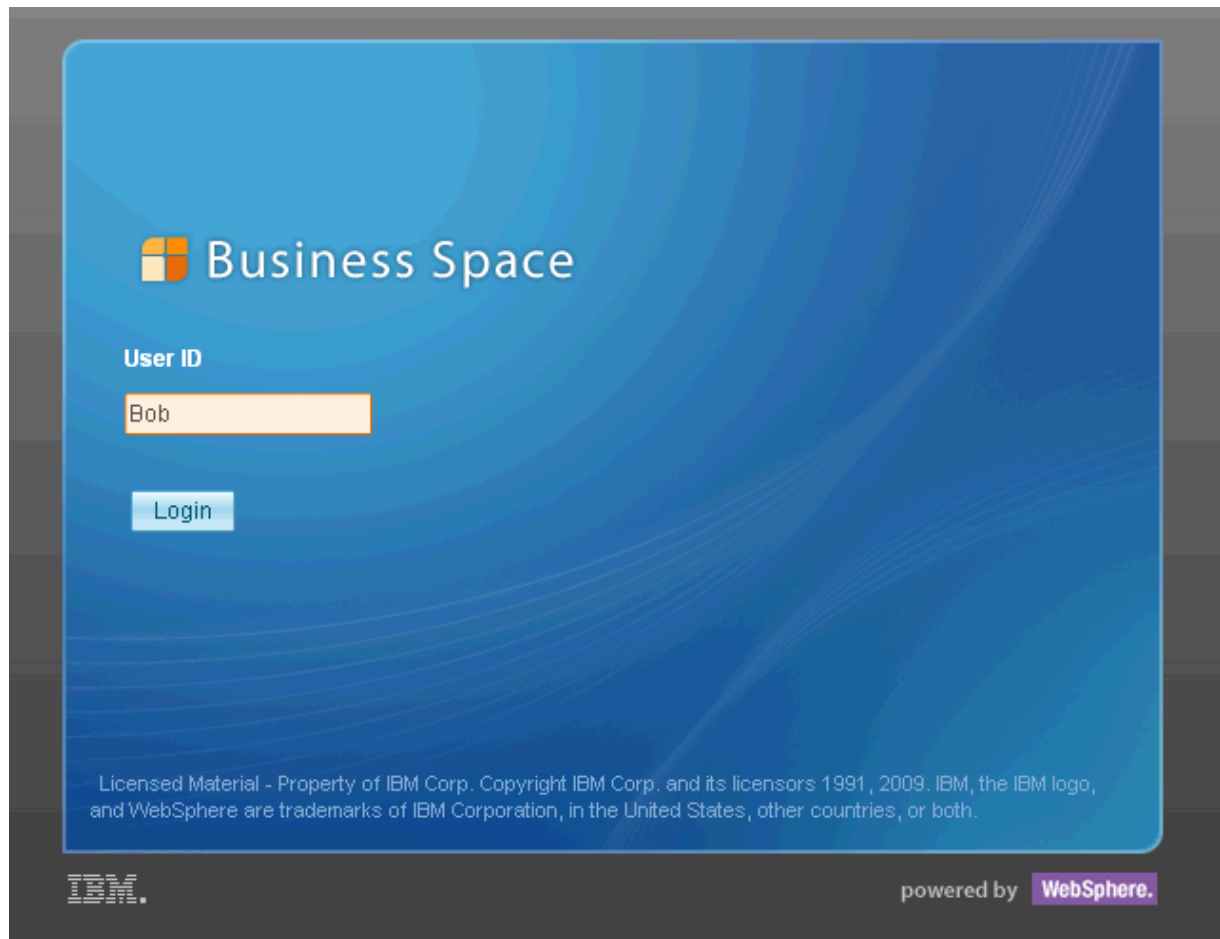


- __d. Logout of WebSphere Service Registry and Repository.

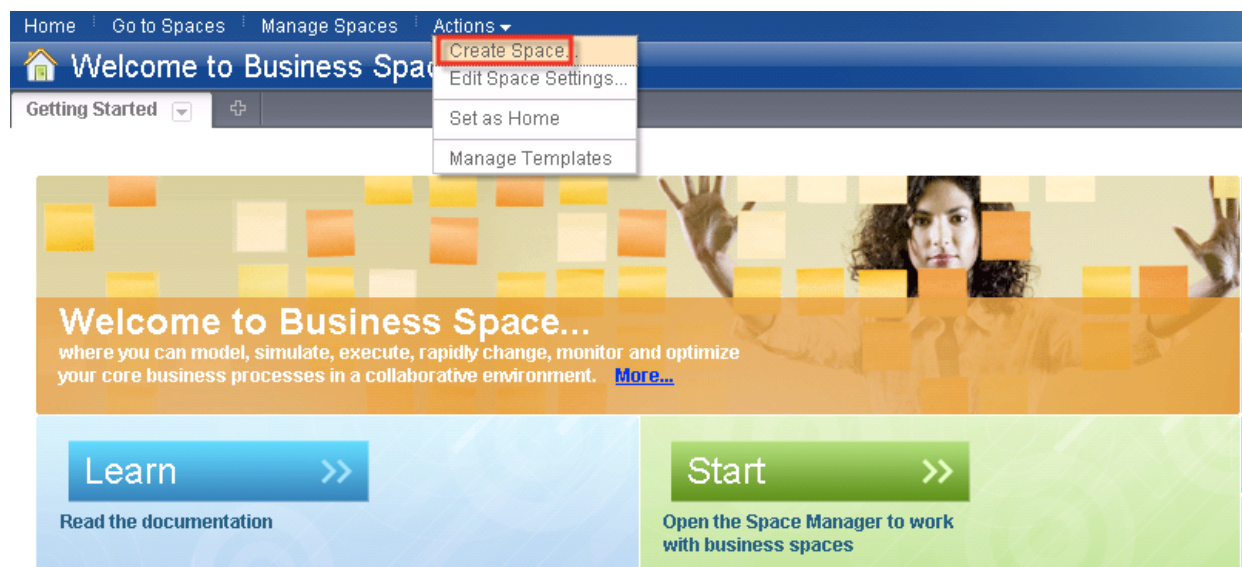
8.3 Explore the Service Registry Business Space

- __1. Work with the Service Registry Business Space and explore the Service Registry widgets.
 - __a. Open a browser window by clicking on the Mozilla Firefox shortcut.
 - __b. Connect to the Business Space powered by WebSphere by typing <https://localhost:9443/BusinessSpace/> or click on the Business Space dashboard shortcut

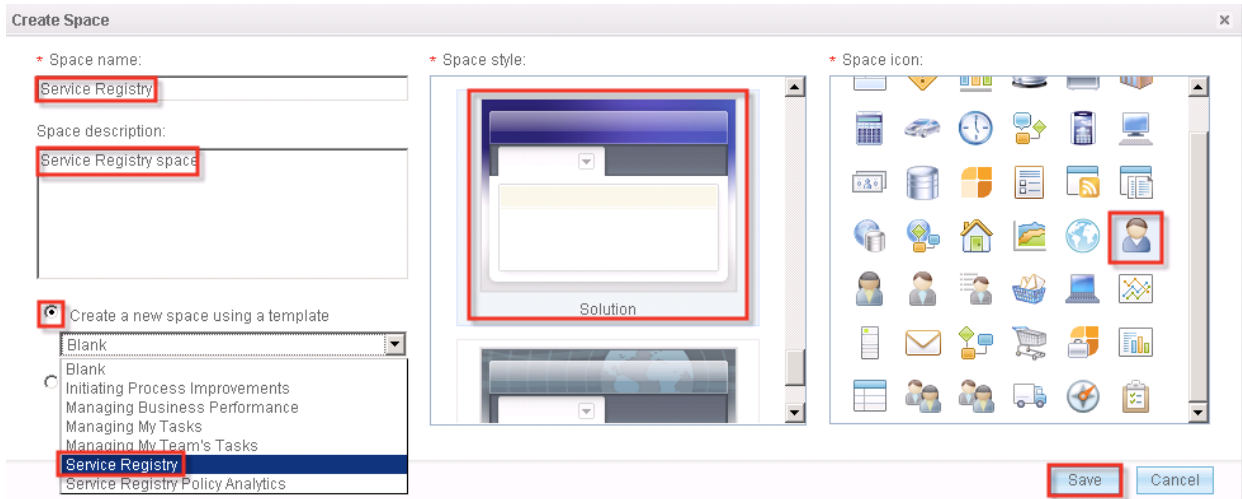
__c. Logon as Bob, the Business Analyst.



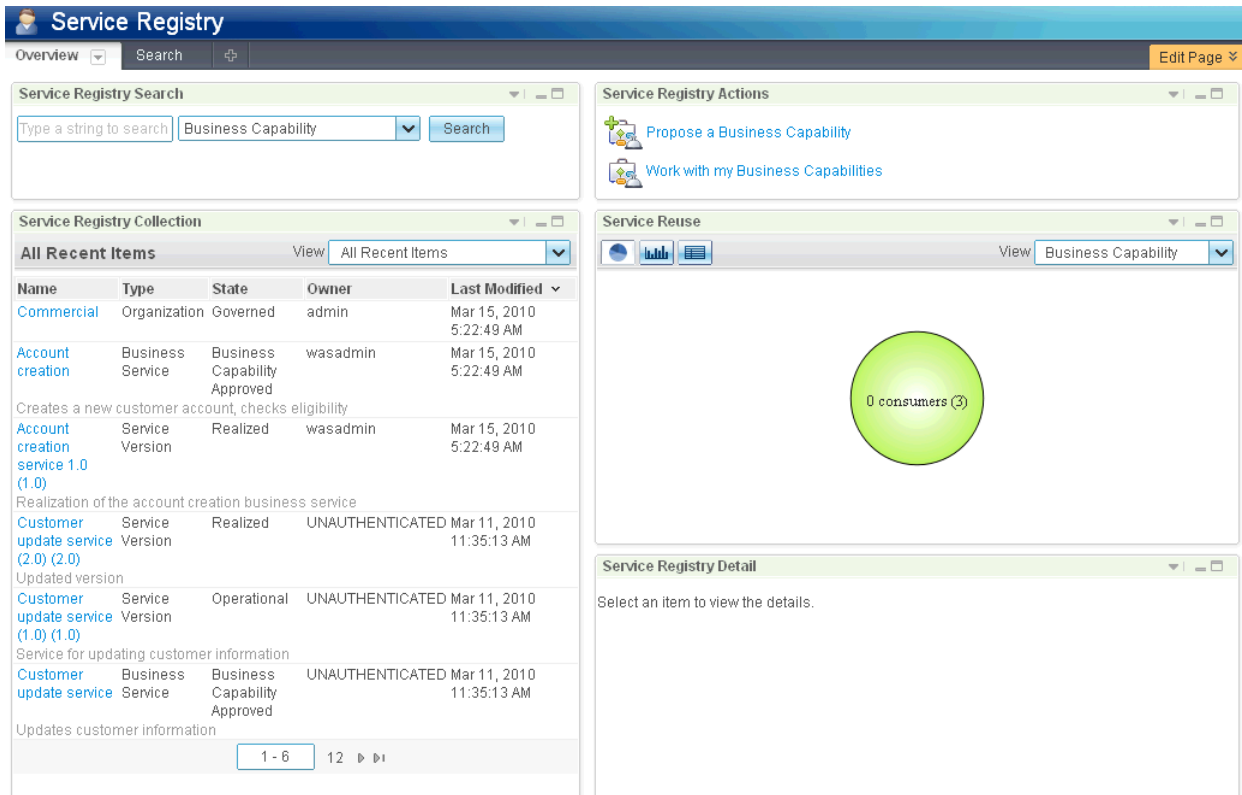
__d. From the Actions menu on the Welcome to Business Space page, select the Create Space link.



- e. On the Create Space page, enter a Space name of **ServiceRegistry**, a description of **Service Registry Space**, ensure Create a new space using a template is selected and select **Service Registry** as the template. Select a Space style and Space icon as shown. Click **Save**.



The Service Registry Business Space page appears.



Note, the items under All Recent Items in the Service Registry Collection widget, may differ from what is shown in the figure above.

Because you used the Service Registry template to create the Service Registry Business Space, you automatically have a page created with a set of Service Registry widgets. There are multiple widgets displayed on the Service Registry Business Space. A brief description of the widgets is provided below.

The **Service Registry Search widget** allows you to search for various types of artifacts within the service registry, including business services, business processes, and business applications. To search for all items of a specific type, select the type of item you wish to find in the drop down without entering anything in the search text box. Types include Business Capability, Business Application, Business Process, and Business Service. The widget provides an Auto-Suggest feature so when text is entered in the search field, you will be presented with a list of matching suggestions. The results of the search are displayed in the Service Registry Collection widget.

The **Service Registry Collection widget** displays a collection of service registry items. The collection of items displayed in this widget depends on the selection in the View dropdown menu

- All Recent Items shows the most recently changed 100 items in the registry.
- All My Items shows all the items in the registry, that you created.
- My Recent Items shows the most recently changed 100 items in the registry that were created by you.
- Search Results shows results of the last search.

Other widgets can be used to populate this widget with collections including the Service Registry Search widget, and Service Reuse widget.

The **Service Registry Reuse widget** measures reuse based on the total number of Service Level Agreements (SLAs) consuming Service Level Definitions (SLDs)

- A provider of a service version has a SLD in place to declare additional non-functional quality of service capabilities that the provider can currently support
- A consumer of a service version establishes a SLA to express functional and non-functional requirements
- 1 Consumers (1) means there is 1 business capability in the registry that has 1 active Service Level Agreement, which equates to 1 consumers

Clicking on a segment/bar/row of the reuse graph displays the relevant list of business services that are being reused in the Service Registry Collection widget.

The **Service Registry Detail widget** displays the details of a selected registry item. This widget displays the full details of items selected in other Service Registry widgets (e.g the Service Registry Collection widget). The display is divided into 3 major sections.

- **Properties** - displays the main and additional properties of the selected item.
- **Classifications** - displays the classifications that have been applied to the selected item.
- **Relationships** - displays the relationships and relationship targets for the selected item

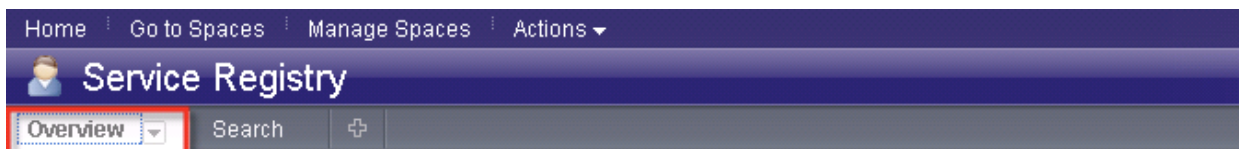
Click the name of a relationship target item to view the full details for that item.

The **Service Registry Actions widget** provides links to key service registry tasks for the business user. The **Propose a Business Capability** link links to Propose a Business Capability dialog to create a new Business Capability in the service registry, which can be a Business Service, Business Process, or Business Application. The **Work with my Business Capabilities** link populates the Service Registry Collection widget with Business Capabilities created by the current user. If the Service Registry Collection widget is not currently displayed on the same page as the Service Registry Actions widget then the 'Search' tab (mashup page) is launched.

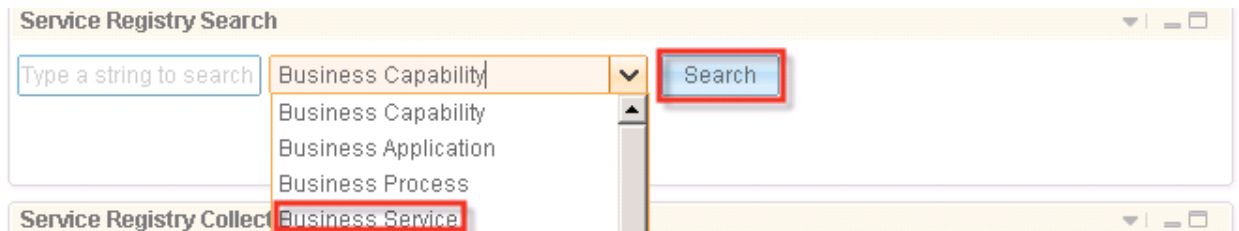
__2. Searching with Business Space

When the Service Registry Business Space is created, there are two pages created for the space, Overview and Search.

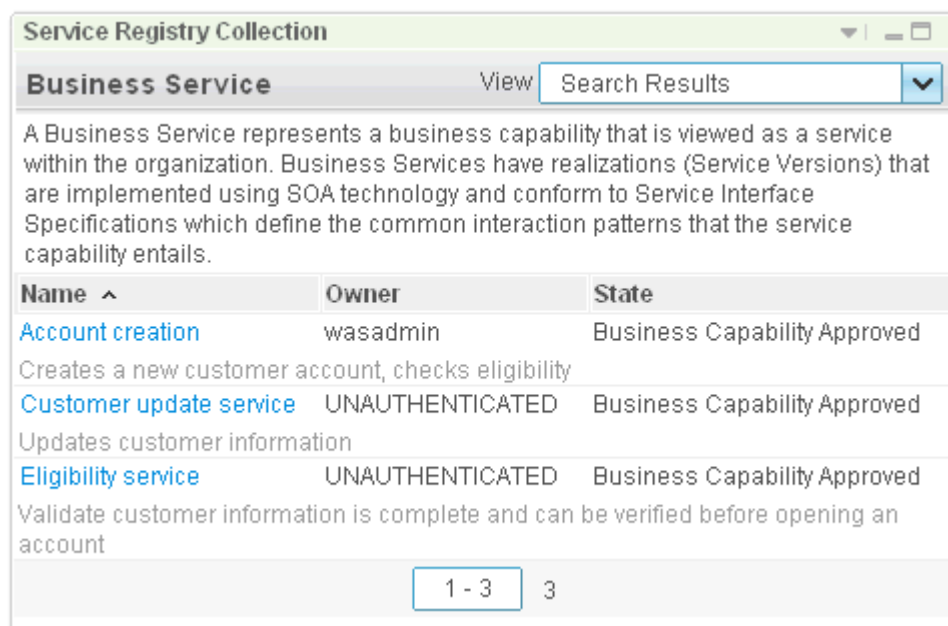
__a. Ensure the Overview page is selected by clicking on the Overview tab.



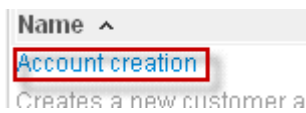
__b. From the Service Registry Search widget, select **Business Service** from the drop down list and click the **Search** button.



The list of Business Services is displayed in the Service Registry Collection widget.



- ___c. To view details for the Account creation business service, click the **Account creation** link in the Service Registry Collection widget.



The details for the Account creation business service are displayed in the Service Registry Detail widget.

Service Registry Detail	
Business Service Properties	
Name:	Account creation
Description:	Creates a new customer account, checks eligibility
Requirements Link:	urn:serviceregistry
Asset Web Link:	urn:serviceregistry
Remote State:	
Owner Email:	
Governance State:	Business Capability Approved
▶ Additional Properties	
▼ Classifications	
Business Service	
Business Capability Approved	
▼ Relationships	
▼ Owning Organization	
Name:	Commercial
Contact:	
Contact Email:	
▼ Charter	
Name:	AccountCreationServiceCharter.doc
Namespace:	

- ___d. Scroll down in the Service Registry Detail widget. To view details for the service version that realizes this business service, click the Account creation service 1.0 link in the Versions relationships section of the Service Registry Detail widget.

▼ Versions	
Name:	Account creation service 1.0
Governance State:	Realized

The details for the Account creation service 1.0 business service version are displayed in the Service Registry Detail widget.

Service Registry Detail	
Name:	Account creation service 1.0
Description:	Realization of the account creation business service
Version:	1.0
Consumer Identifier:	ACV100
Version Availability Date:	Friday, January 1, 2010
Version Termination Date:	Friday, December 31, 2010
Requirements Link:	urn:serviceregistry
Asset Web Link:	urn:serviceregistry
Remote State:	
Owner Email:	
Governance State:	Realized

▲ Additional Properties

__3. Publishing to WebSphere Service Registry and Repository through Business Space

Propose a new business capability.

You will now propose a new business capability that will eventually use the Account Creation service.

- __a. On the **Service Registry Actions** widget, click **Propose a Business Capability**.



The Propose a Business Capability panel appears.

- __b. Select a type of **Business Service** and enter the following information.

Name: Customer creation

Description: This business service capability will create a new customer.

Requirements Link: <http://requirements.jkhl.com/requirements.jsp&id=5510>

- __c. Browse to **C:\student\WSRR\Lab8** and select **CustomerCreationServiceCharter.doc**

Propose a Business Capability

Use this form to submit a proposal for a new business service.

Type
Business Service

*Name
Customer creation

Description
This business service capability will create a new customer

Requirements Link
/requirements.jkhl.com/requirements.jsp&id=5510

*Business Case (Charter)
C:\student\WSRR\Lab8\CustomerCreationServiceCharter.doc

Cancel OK

- __d. Click **OK**.

- __4. Use the Service Registry Search widget to retrieve your new business service.

- __a. Perform a search for business services as described in step 3.a previously, your new business service is listed.

Customer creation UNAUTHENTICATED Charter Review
This business service capability will create a new customer



Because we do not have security enabled and you are therefore not an authenticated user, Work with my Business Capabilities will not retrieve your Business capabilities.

- __b. Log out of Business Space

- __5. Explore Policy Analytics (Optional)

- __a. Log in to Business Space as **David** from SOA Governance.
- __b. Explore Policy Analytics, particularly WS-I compliance.
- __c. Once you have done enough exploring, log out of Business Space.
- __1. You have completed the lab.

8.4 Lab Review

During this lab, you:

- Created and explored a business space based on the Service Registry and Repository template.
- Used the service registry search, collection, detail, and actions widgets to view and create data in WebSphere Service Registry and Repository.
- Explored Policy Analytics, specifically WS-I compliance.



CONGRATULATIONS!

You have successfully completed Lab 8 - Exploring the WebSphere Service Registry and Repository Business Space

Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Appendix B. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
<i>System z</i>	<i>Tivoli</i>	<i>WebSphere</i>	<i>Workplace</i>	<i>System p</i>	

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. See Java Guidelines

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Other company, product and service names may be trademarks or service marks of others.



© Copyright IBM Corporation 2011.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

Other company, product and service names may be trademarks or service marks of others.



Please Recycle
