

WebSphere Service Registry and Repository integration with WebSphere Business Services Fabric

What this exercise is about	1
Lab requirements	1
What you should be able to do	2
Part 1: Exercise instructions	3
What you did in this exercise	16

What this exercise is about

The WebSphere Business Services Fabric 6.2 offers improved integration with WebSphere Service Registry and Repository. By completing this exercise, you will have configured a Fabric connection with WebSphere Service Registry and Repository and created a dynamic Fabric endpoint.

Lab requirements

Software needed:

WebSphere Integration Developer version 6.2 with the WebSphere Process Server unit test environment

WebSphere Tool Pack with WebSphere Business Fabric Server test environment version 6.2

WebSphere Registry and Repository version 6.2

Load WebSphere Service Registry and Repository with Web service interface documents from WSRR-wsdl.zip. These instructions are in the scope of this lab.

Ensure there is an existing Fabric project in this workspace. If not, run the Modeler-Fabric interaction lab.

These labs use username: admin and password: admin to login as administrator for any of the Web consoles including WebSphere Business Services Fabric and WebSphere Process Server. Your environment may use different credential details.

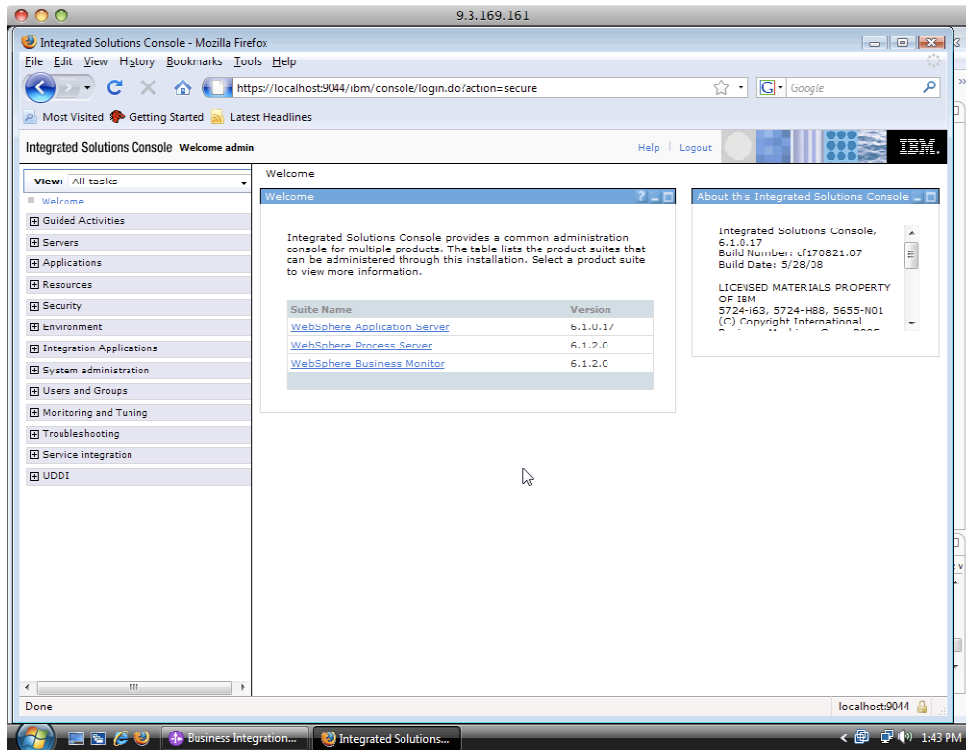
What you should be able to do

After performing this lab, you will have:

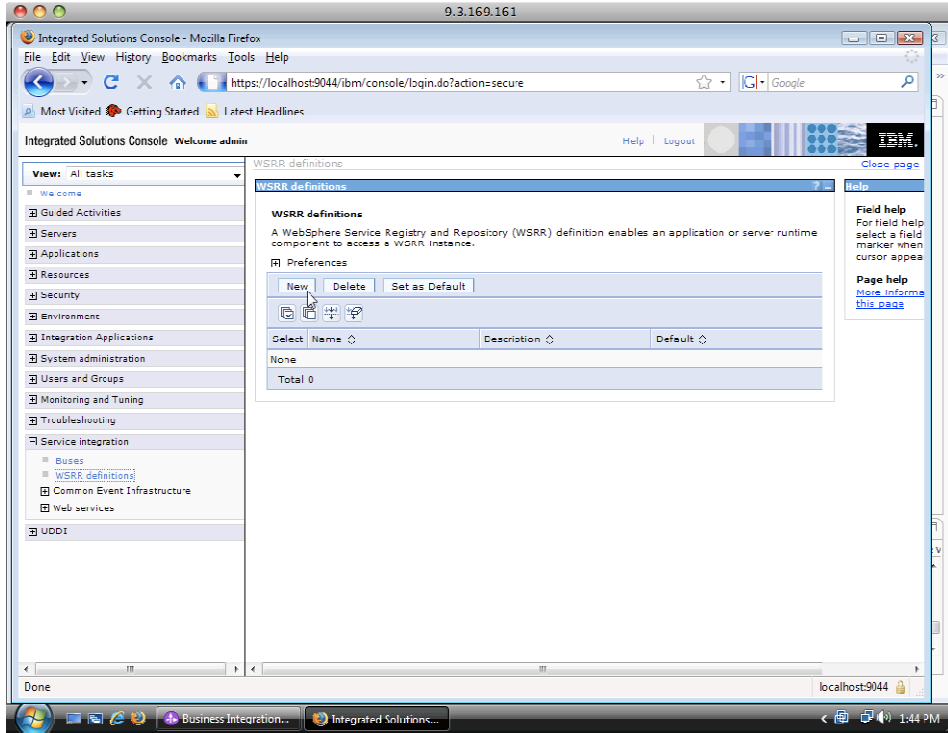
- Become more familiar with WebSphere Process Server, WebSphere Integration Developer, WebSphere Service Registry and Repository, and WebSphere Business Services Fabric
- Configure a Fabric Connection with WebSphere Service Registry and Repository
- Created a Dynamic Fabric endpoint

Part 1: Exercise instructions

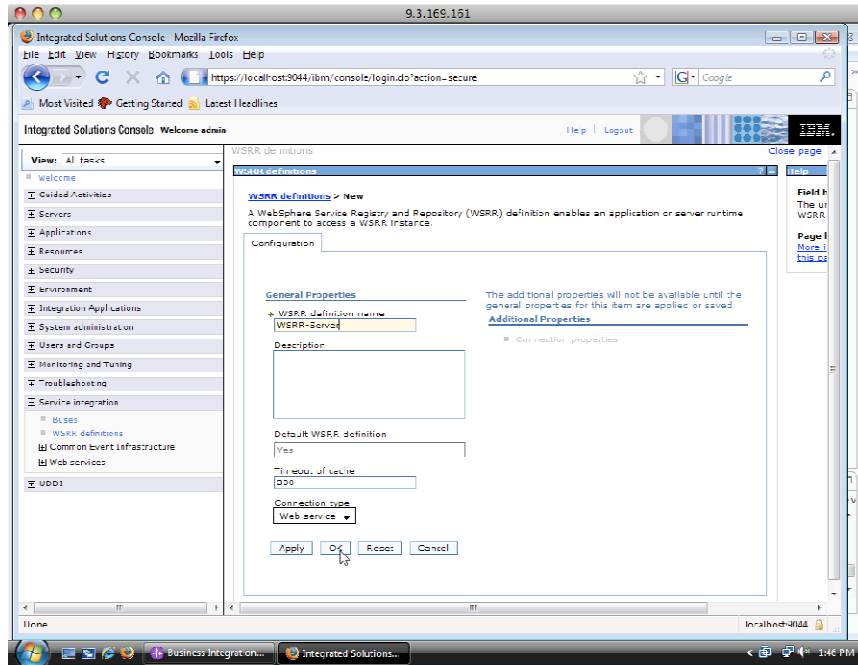
- __1. Start WebSphere Service Registry and Repository server
- __2. Log into the WebSphere Process Server administrative console as administrator on your Fabric server at <https://localhost:<port number>/ibm/console> using admin/admin.



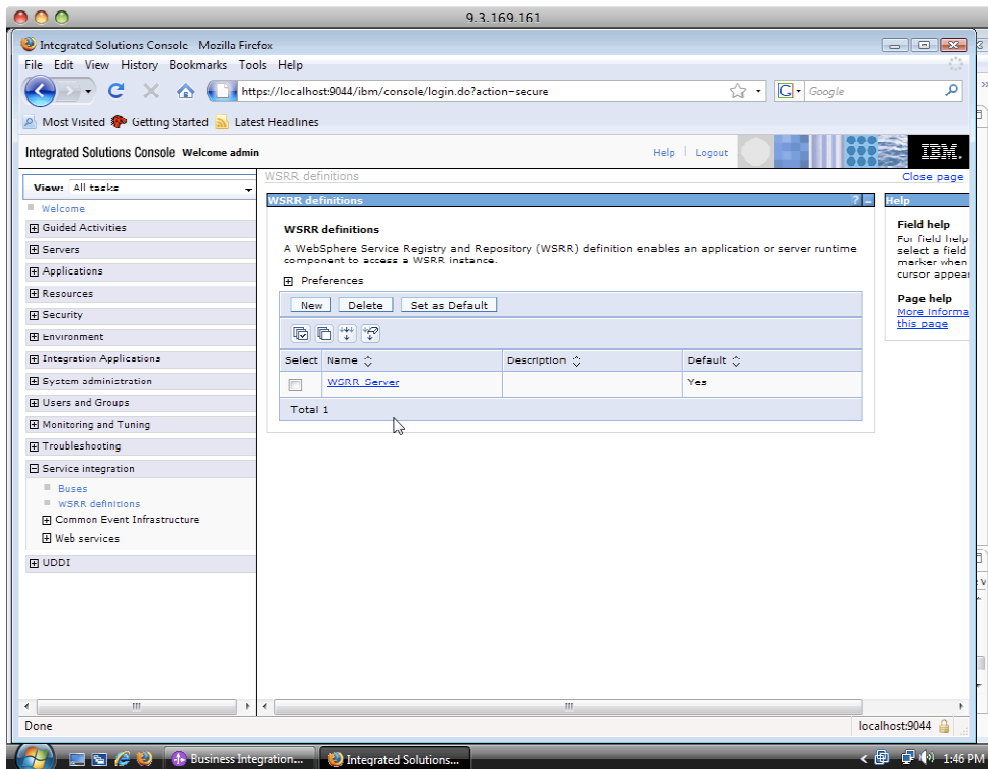
- __3. Click on **Service Integration > WSRR Definitions**
- __4. Click **New**



- 5. Enter **WSRR-Server** in the WSRR definition name field and click **OK**. **Save** changes to the master configuration



__6. In the WSRR definitions screen click on the link **WSRR-Server**



__7. Click **Connection properties** under **Additional Properties**

__8. Under the **Related Items** section click **JAAS-J2C authentication data** link

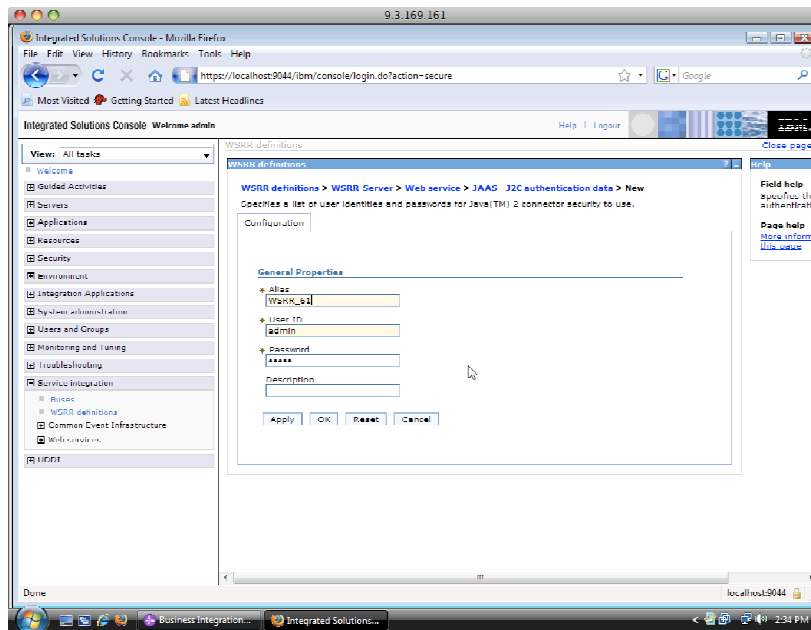
__9. Click **New**

__10. Enter the following details :

__a. Alias: wsrr62-auth

__b. Provide the WebSphere Services Registry and Repository server login credentials for the user ID and password.

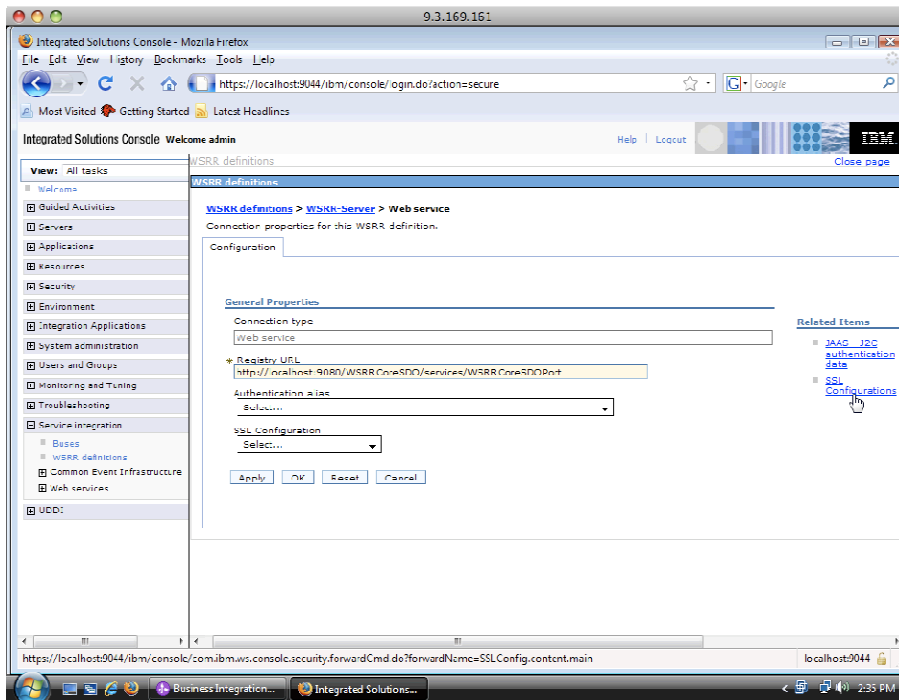
__c. Click **OK** and **Save** changes to the master configuration



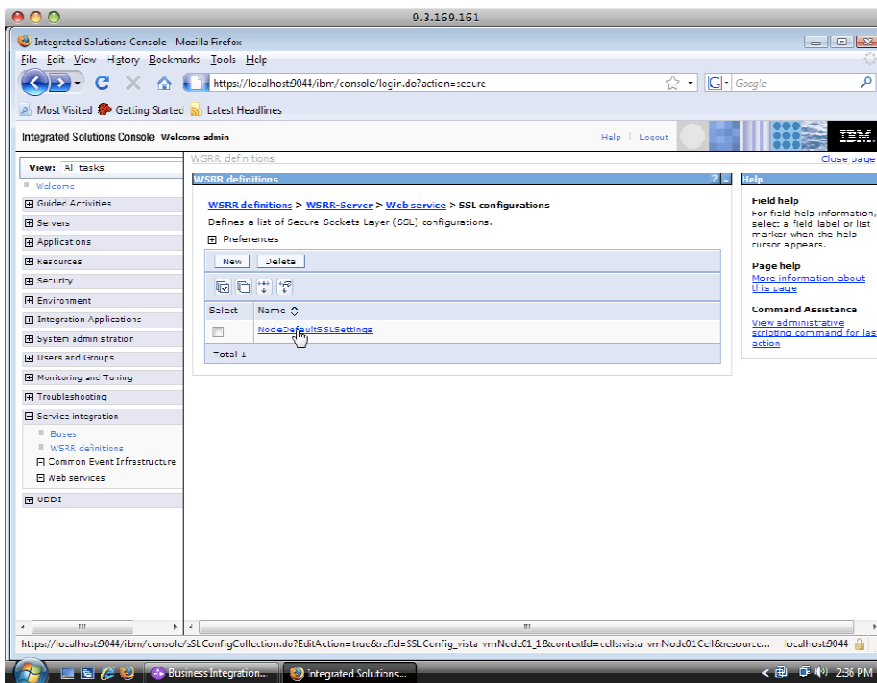
__11. Click **Web service** in the navigation tree.



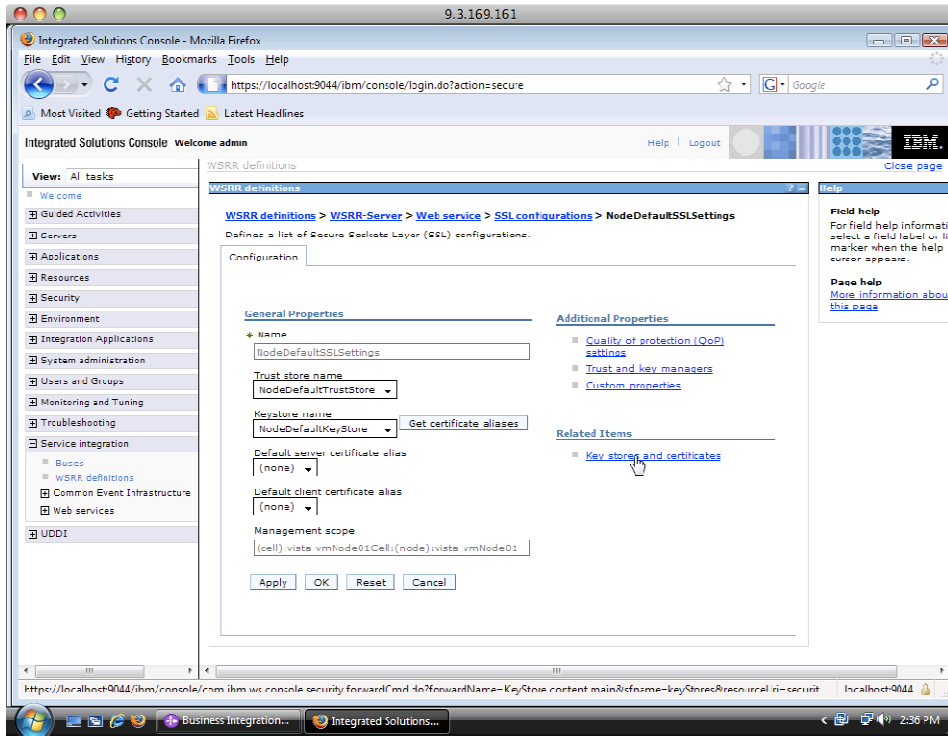
___12. Under the Related Items section click **SSL Configurations** link



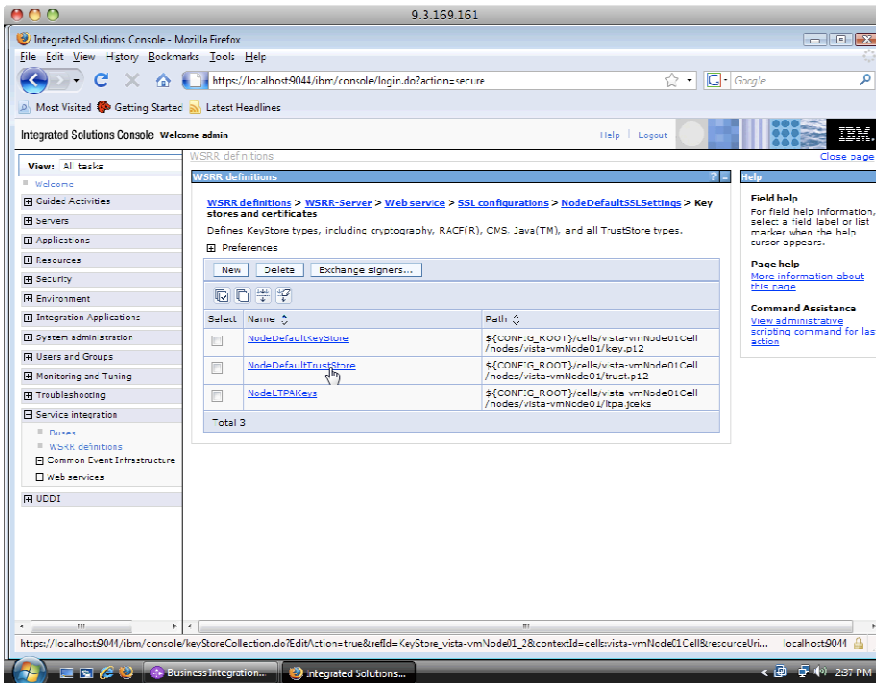
___13. Click **NodeDefaultSSLSettings** link



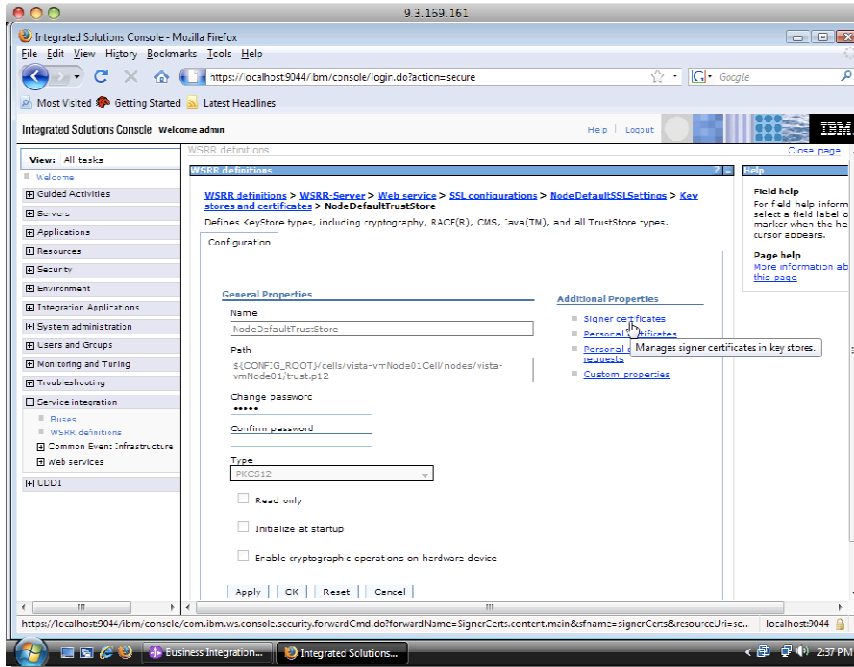
__14. Under related Items, click **Key stores and certificates**



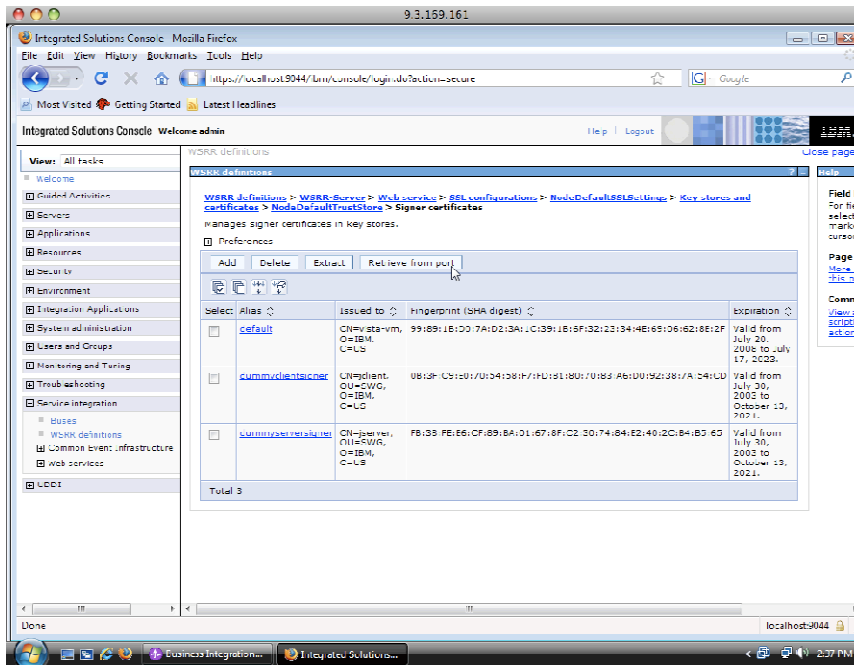
__15. Click **NodeDefaultTrustStore**



___16. Click **Signer Certificates** under the Additional Properties

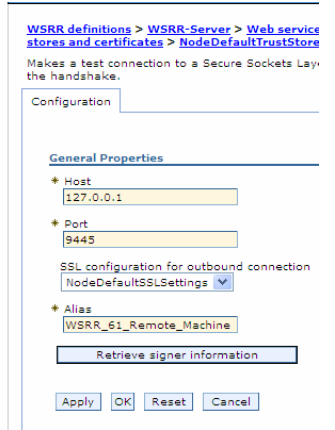


___17. Click the **Retrieve from port** button



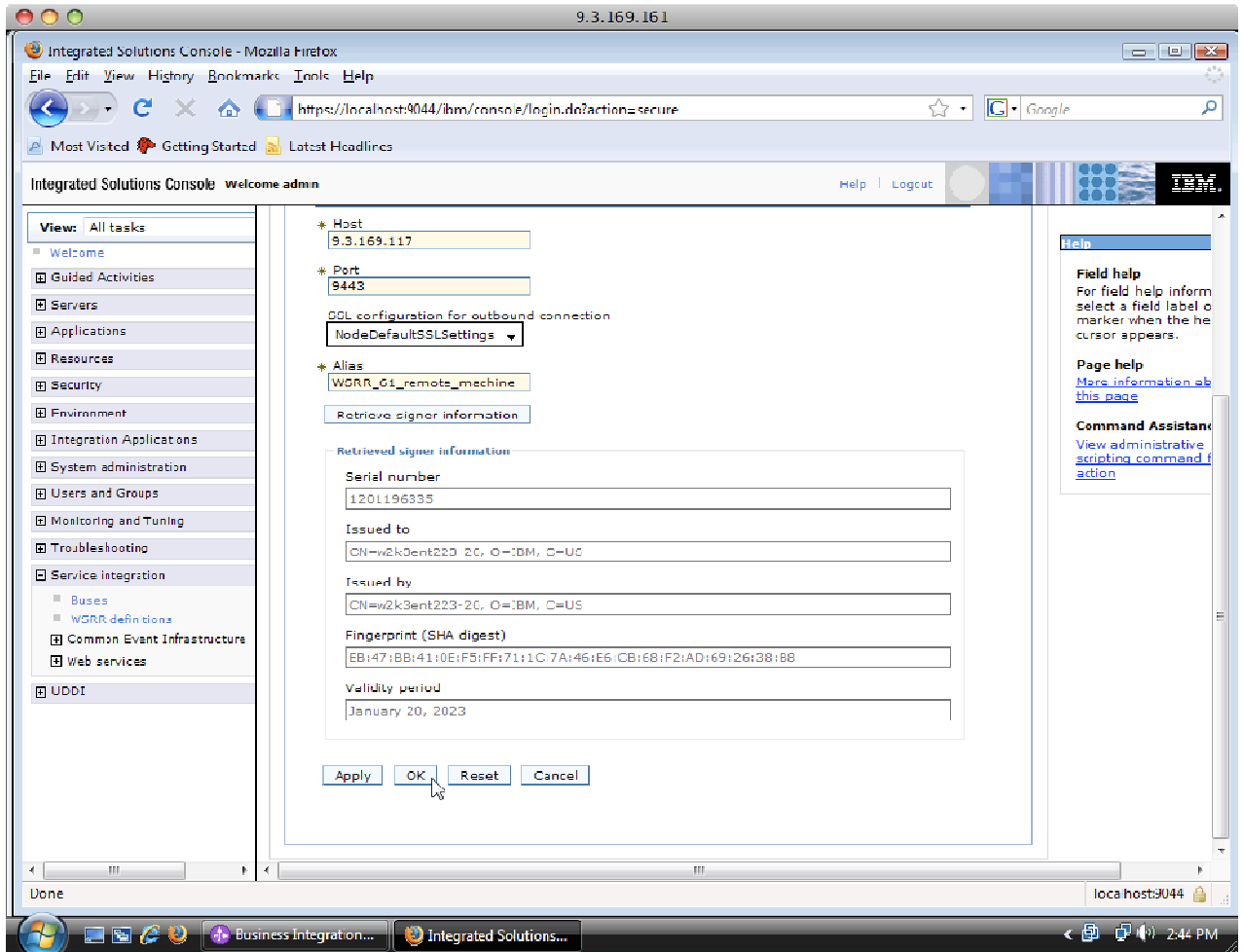
___18. Enter the following details :

Host	127.0.0.1 < change as per your environment >
Port	9445 < change as per your environment >
Alias	WSRR_62_Remote_Machine



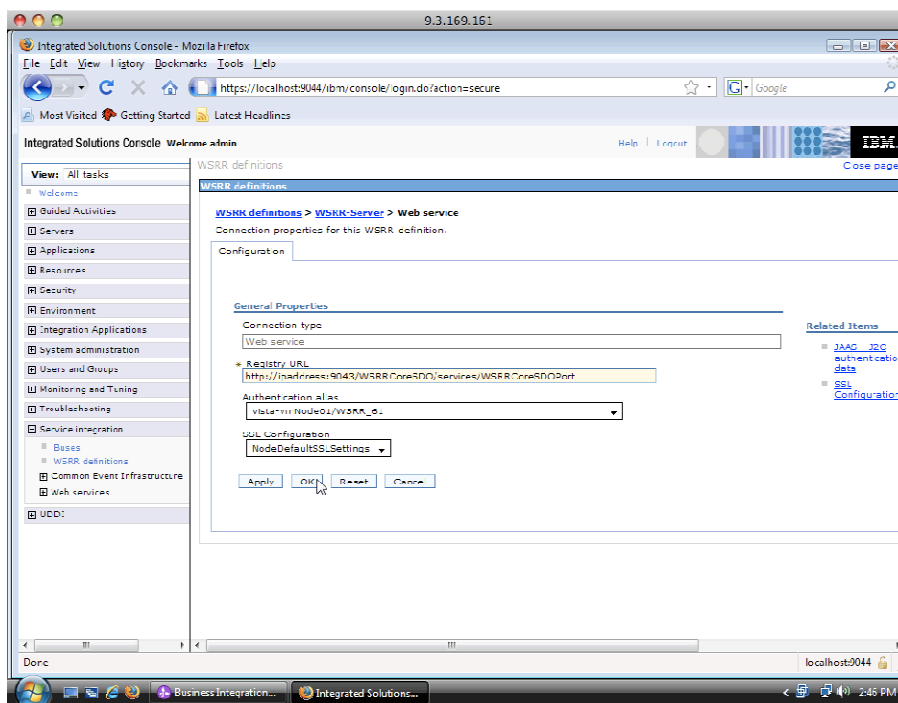
__19. Click Retrieve signer information

__20. If all the info you typed in is correct you will see the key and other information appear below the Retrieve signer information button



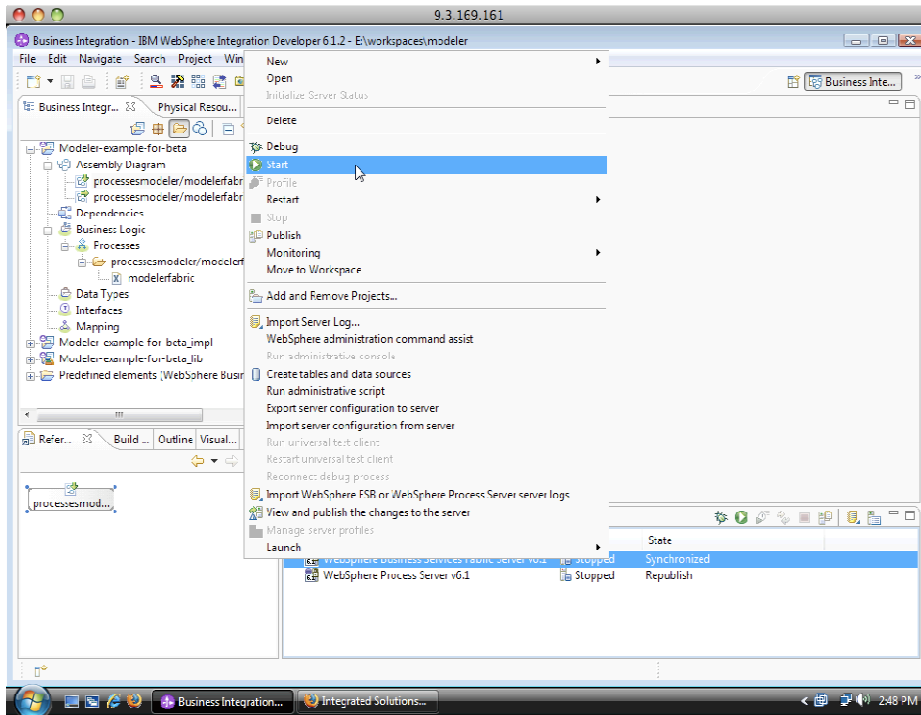
__21. Click OK.

- __22. **Save** the changes to the master configuration
- __23. In the Navigation tree click on **Web Service** (you should see WSRR definitions > WSRR-Server > Web service in your navigation tree in the main panel)
- __24. Change the **Registry URL** field to https://<host name: port number>/WSRRCoreSDO/services/WSRRCoreSDOPort
 - __a. Select **authentication alias** : **wsrr61-auth**
 - __b. Select **NodeDefaultSSLSettings** : **SSL Configuration**

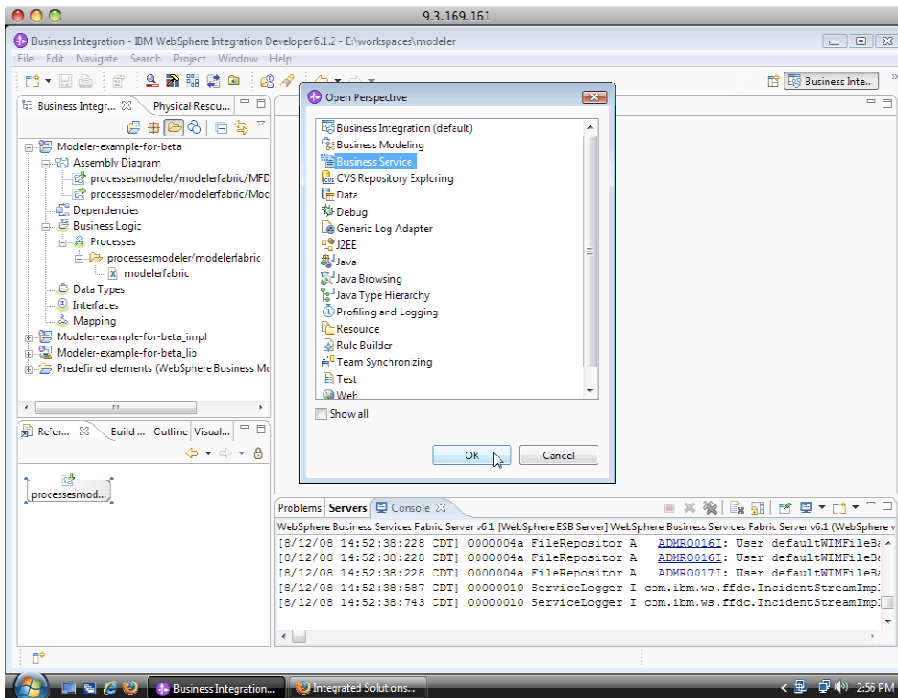


- __25. Click **OK**.
- __26. **Save** the changes to the master repository
- __27. Click **Logout** of WebSphere Process Server console and close the browser
- __28. Switch back to WebSphere Integration Developer in the Business Integration perspective

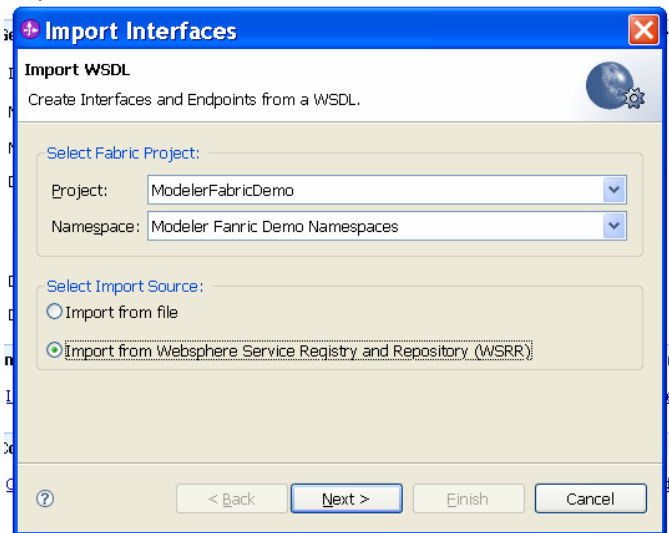
__29. Stop and Start the server WebSphere Business Services Fabric Server.



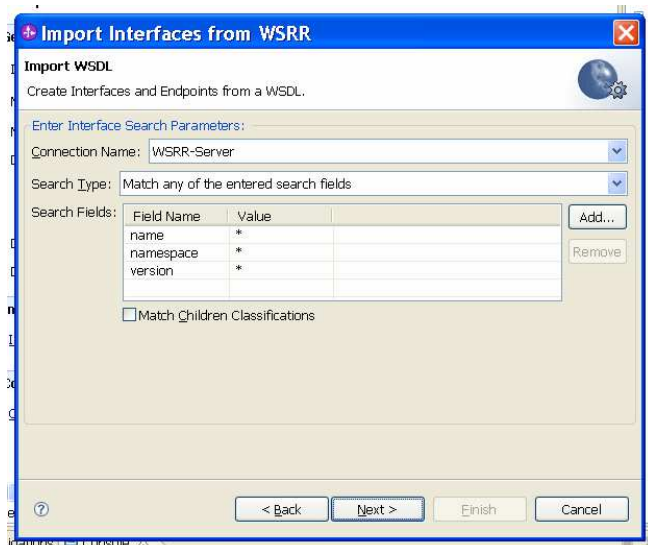
__30. Open the Business Service Perspective.



- __31. Ensure there is an existing Fabric project in this workspace. If not, run the Modeler –Fabric Interaction Lab.
- __32. Right click and select **New > Interface**
- __33. In the Import Interface window change the Import Source to **Import from WebSphere Registry and Repository (WSRR)** and click **Next**

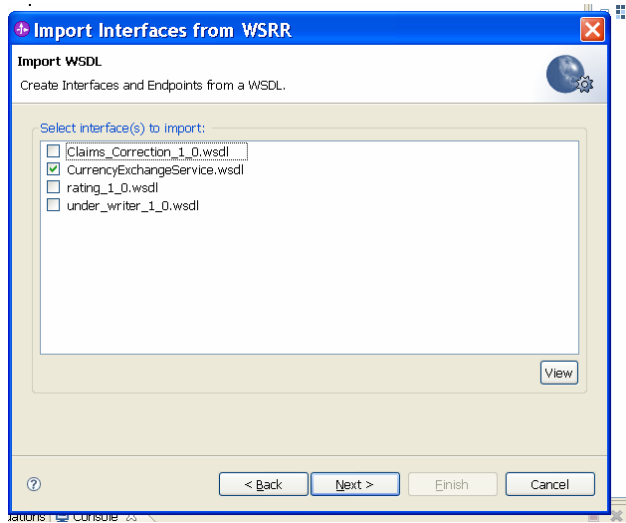


- __34. Click **Next**

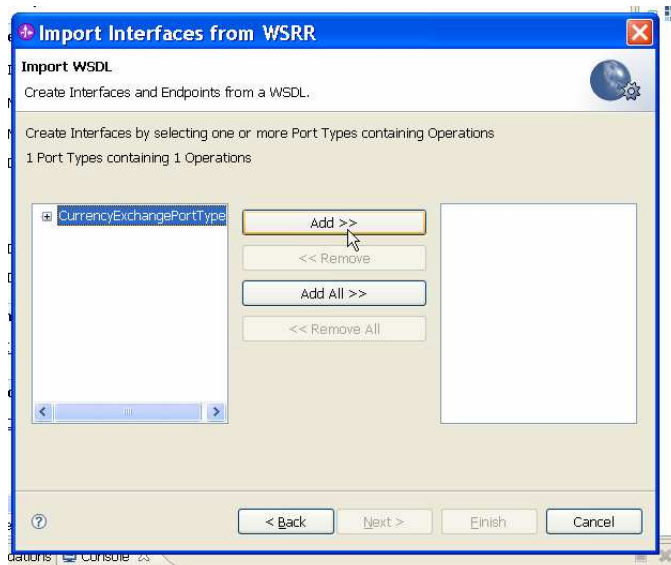


- __35. This will initiate a query into **WebSphere Services Registry and Repository** server that was registered earlier through the WebSphere Process Server administrative console.

__36. Select **CurrencyExchangeService.wsdl** and click **Next**.



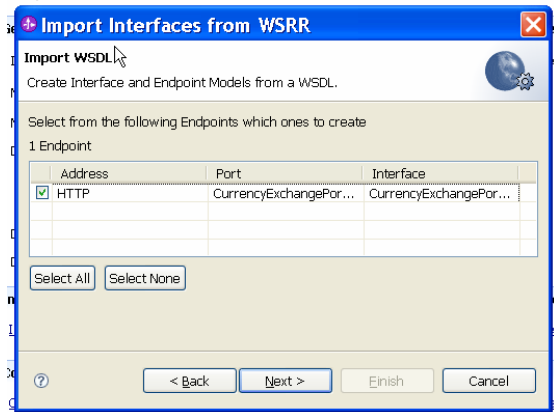
__37. Once the WSDL is parsed select **CurrencyExchangePortType** and click **Add**



__38. Click **Next**

__39. Verify the Port Type and Name and click **Next**

__40. Select the associated Endpoint and click **Next**



- __41. Verify the endpoint details and click **Finish**

- __42. Verify you see the interface and endpoint that were imported under ModelerFabricDemo > Interfaces and ModelerFabricDemo > Endpoint.

- __43. Stop the WebSphere Business Services Fabric server

- __44. Close WebSphere Integration Developer

What you did in this exercise

You have now successfully imported interfaces and associated endpoints from WebSphere Service Registry and Repository into WebSphere Business Services Fabric.