Generating Web Service From PI

Generating WSDL From PI

Requirement:

The Scenario covers SOAP to RFC Scenario, where the WSDL is generated by PI and is exposed to third party. Generally, when there are liabilities for the WSDL to be changed more often in future, it's convenient to generate the Web Service from PI itself rather than importing the WSDL again and again!!

It also depends upon the nature of the SAP System. For instance, here Non SAP system is Dot Net System, where the connectivity could be done either by HTTP or SOAP. WSDL is quiet compatible for SOAP Adapter.

Note: This can only be done when SOAP is at the sender end.

Steps:

1. Creation Of Data Type for Request SOAP Message
2. Creation of Data Type for Response SOAP Message
3. Creation Of Message Type for Request SOAP Message
4. Creation of Message Type for Response SOAP Message
5. Creation of Outbound Synchronous Interface
6. Creation of Request Message Mapping between SOAP and RFC
7. Creation of Response Message Mapping between RFC and SOAP
8. Creation of Interface Mapping(Request) between the Outbound Synchronous Interface and RFC
9. Creation of Interface Mapping(Response) between the RFC and the Outbound Synchronous Interface

Configuration:

Defining WS by PI

1. Go to Tools->Define Web Service in ID
2. Provide the accurate SOAP Channel URL: http://vessxid01:8001/XISOAPAdapter/MessageServlet?channel=:Sender
Business Service: Sender Soap Adapter
3. Specify the parameters like Sender Message interface, Namespace and Software Component Version
4. Define Sender Service, Interface Name and Namespace
5. Continue the wizard and WSDL is generated. It can be saved and exposed to third party.
6. All collaboration profiles like Receiver Determination, Interface Determination and Receiver Agreement are created. Note: Sender Agreement is not needed since WSDL is generated by PI itself!

We do have an alternative for generating WSDL.

After the configuration is complete, directly go to the sender agreement and click on Generate WSDL. The only difference is that with this WSDL we can use any communication channel while with the above approach, we cant change the sender communication channel.