CREATION AND CONFIGURATION OF WEB SERVICE FROM RFC AND DEPLOYMENT IN ANOTHER SYSTEM

Applies to:
SAP

Summary
The purpose of this document is to provide creation and configuration of web service from function module in one system and call it from other system.
Created on: 18 Sep 2012

Author Bio
Sachin Thadani Consultant working for SAP GDC
TABLE OF CONTENTS

INTRODUCTION ....................................................................................................................... 3
AUDIENCE .................................................................................................................................. 3
PURPOSE .................................................................................................................................... 3
CREATION OF WEB SERVICE PROVIDER (DEVELOPMENT SYSTEM) ............................................. 3
CREATION OF END POINT FOR WEB SERVICE PROVIDER (DEV SYSTEM): .......................... 6
CREATION OF WEB SERVICE CONSUMER (DEV SYSTEM) ..................................................... 9
GET URL PATH FOR LOGICAL PORT SETUP FOR CONSUMER PROXY ................................. 11
USER SETUP FOR WEB SERVICE EXECUTION ........................................................................ 11
CREATION OF LOGICAL PORT (TEST SYSTEM) ..................................................................... 12
DEPLOYMENT / INTEGRATING WEB SERVICE INTO PROGRAM ................................................ 14
INTRODUCTION
The purpose of this document is to create a web service from already existing function module in client and then deploy it in another system.

AUDIENCE

SAP

PURPOSE
The purpose of this document is to create a web service from a function module in test system (say ISX) and then deploy it in another system/production system (say ISV).

CREATION OF WEB SERVICE PROVIDER (DEVELOPMENT SYSTEM):

1. Create a RFC function module (remote enabled):
Creation and Configuration of Web Service from RFC and Deployment in another system

It will open the wizard. Give the service definition name prefix with Z or Y. Give description also.
Creation and Configuration of Web Service from RFC and Deployment in another system
CREATION OF END POINT FOR WEB SERVICE PROVIDER (DEV SYSTEM):

Execute transaction SOAManager and select Web Service Configuration:
Search for the service definition created earlier “ZE_GENCOND_REPLICATE”:

Click on Configurations Tab and click on “Create”. A pop-up box will appear where mention the service name, Description and New Binding name and click on Apply settings:
Consider the Provider Security – Transport Guarantee and Authentication Method settings and then Save it.

In the runtime configuration, you can configure service providers individually or together using profiles. Not all security settings are available when using profiles.

- **Transport Guarantee**
  - **HTTPS**
    
    HTTP communication that is secured with SSL (Secure Sockets Layer). You can use the Secure Sockets Layer (SSL) protocol to secure HTTP connections to and from the AS ABAP. When using SSL, the data being transferred between the two parties (client and server) is encrypted and the two partners can be authenticated. For example, if a user must transfer his or her account information, then you can use SSL to authenticate the user and encrypt the information during transfer.

- **Signature and Encryption**
  
  Messages are secured with an XML signature and XML encryption with asymmetrical keys. WS Security is a standard for securing SOAP messages. By using WS Security, you protect the SOAP messages that are exchanged between the Web service provider and the Web service client with digital XML signatures, XML encryption, time stamps, and security tokens. You can use symmetric or asymmetric encryption. The main difference between symmetric and asymmetric encryption is the type of signature.
Secure Conversation
Messages are secured with a pre-defined symmetrical key. The key is re-used in further calls. To use WS-SecureConversation, the following requirements must be met: No additional configuration is required.

To use WS-Security XML signatures and encryption with X.509 certificates, you need to enable the use of cryptographic functions for the AS ABAP system.

- The corresponding option is selected in the configuration (for example, in SOAMANAGER).
- Both the consumer and provider systems have an SSL trust relationship.

HTTP Authentication
The authentication information is found in the HTTP header.
- User Name/Password (Basic)
- X.509 Certificate
  Authentication with an X.509 certificate.
- Logon Ticket
  Authentication with an SAP Assertion Ticket.

Message Authentication
The authentication information is found in the SOAP header.
- User Name/Password (Basic)
  Authentication with WS Security UsernameToken
- X.509 Certificate
  Authentication with a signed SOAP message, user authentication by certificate
- SAML 1.1
  Authentication with a signed SAML 1.1 Assertion

You choose one of the predefined security settings during the runtime configuration for the service consumer.

**CREATION OF WEB SERVICE CONSUMER (DEV SYSTEM):**
Goto Se80 (Object Navigator) - > Enterprise Services -> Service Consumer -> Create
Execute the following transaction: SOAMANAGER
This will initiate a Web session and prompt you to log into the WebAS.
Once you have logged in, the next step is to locate the Enterprise Service that you created earlier.
GET URL PATH FOR LOGICAL PORT SETUP FOR CONSUMER PROXY

Open WSDL document for Selected Binding and find out the WSDL link to configure in the Service Consumer system.

USER SETUP FOR WEB SERVICE EXECUTION

Create Web Service Consumer Proxy in Dev System and the move it to another system through transport request.

Set up User of type ‘System’ in the Provider System for authentication and execution of Web Service. Assign Technical and Transactional profiles to the user.

For example here we are setting a system user “WEBSERVICE” with technical and transactional profile.
CREATION OF LOGICAL PORT (TEST SYSTEM):
Copy above URL for creation of Logical port for the Consumer proxy and execute transaction SOAMANAGER. Select Consumer Proxy ZCO_ZE_GENCOND_REPLICATE and click on “Apply selection”:

Go to Configurations Tab and Create Logical port for example LP_HTTP. As soon as we click on create button, it will open a popup where specify Logical port name and description. Choose configuration type “Manual Configuration” or any other configuration type based on system settings:
Click on “Apply Settings” and set “Logical Port is Default”, If you want to set this logical port as default, then check the checkbox.

In the consumer security, provide User name “WEBSERVICE” and password:

In the Transport settings tab, provide URL access path fetched above, Computer name of Access URL and Port Number and click on “SAVE”.
Once all settings done, click on Ping Web Service to check if Web Service provider in Dev System is getting pinged from Test System:

DEPLOYMENT / INTEGRATING WEB SERVICE INTO PROGRAM

From the consumer proxy created, get the generated method name from SE24 (Class Builder):
To call the web service, create an instance for the consumer proxy created above and call the method generated:

Create object with reference to Consumer proxy Logical port created through SOAMANAGER

DATA : lr_system_fault TYPE REF TO cx_ai_system_fault,
       lr_gencond_repl_except TYPE REF TO zcx_zvgencond_replicate_except,
       lr_appl_fault TYPE REF TO cx_ai_application_fault,
       lr_cons_proxy TYPE REF TO zco_ze_gencond_replicate.

TRY.
  CREATE OBJECT lr_cons_proxy
     EXPORTING
       logical_port_name = 'LP_HTTP' (Logical Port created).
  CATCH cx_ai_system_fault.
  lv_error_flag = 'X'.
  CATCH zcx_zvgencond_replicate_except INTO lr_gencond_repl_except.
  lv_error_flag = 'X'.
  CATCH cx_ai_application_fault INTO lr_appl_fault.
  lv_error_flag = 'X'.
ENDTRY.

Call Consumer proxy Web Service Method

TRY.
  CALL METHOD lr_cons_proxy->zvgencond_replicate (Consumer proxy method)
     EXPORTING
       input = lt_input
     IMPORTING
       output = lt_output.

Catch Exception if any raised

CATCH cx_ai_system_fault INTO lr_system_fault.
  lv_error_flag = 'X'.
CATCH zcx_zvgencond_replicate_except INTO lr_gencond_repl_except.
  lv_error_flag = 'X'.
CATCH cx_ai_application_fault INTO lr_appl_fault.
  lv_error_flag = 'X'.
ENDTRY.