

Consuming ABAP RFC in PAR using Deployable Webservice DC



Applies to:

This document applies to Portal Content Development, JAVA, SAP Net weaver ABAP.

Summary

This article details a solution steps for consuming an ABAP BAPI as webservice in a PAR application.

Author: Sumit Oberoi

Company: Infosys Technologies Limited

Created on: 31th August 2011

Author Bio



Sumit Oberoi is a Technology Analyst with Infosys Technologies Limited. His areas of expertise include Web Dynpro for JAVA , Web Dynpro ABAP, Enterprise Portal, SAP PI , CAF ,PDK development and portal administration.

Table of Contents

Overview	3
Preparing BAPI for use	3
Consuming ABAP webservice	3
Alternative Approach	8
Related Content	10
Disclaimer and Liability Notice	11

Overview

There have been various instances when I saw people looking out ways to use ABAP webservice in a Portal Application be it Abstract Portal Component or JSPDynpage. When I faced a similar situation I was unable to find a one stop guide to achieve the solution. After taking help from standard documentation and with a bit of trial and error I was able to call an ABAP webservice in a PAR file by means of deployable webservice DC. This article discuss below things in detail:

- Preparing BAPI for use
- Consuming ABAP webservice
- Alternative approach

Disclaimer

The below development steps are for NW2004s 7.0 EHP 1 SPS8 and are based on my personal experience.

Preparing BAPI for use

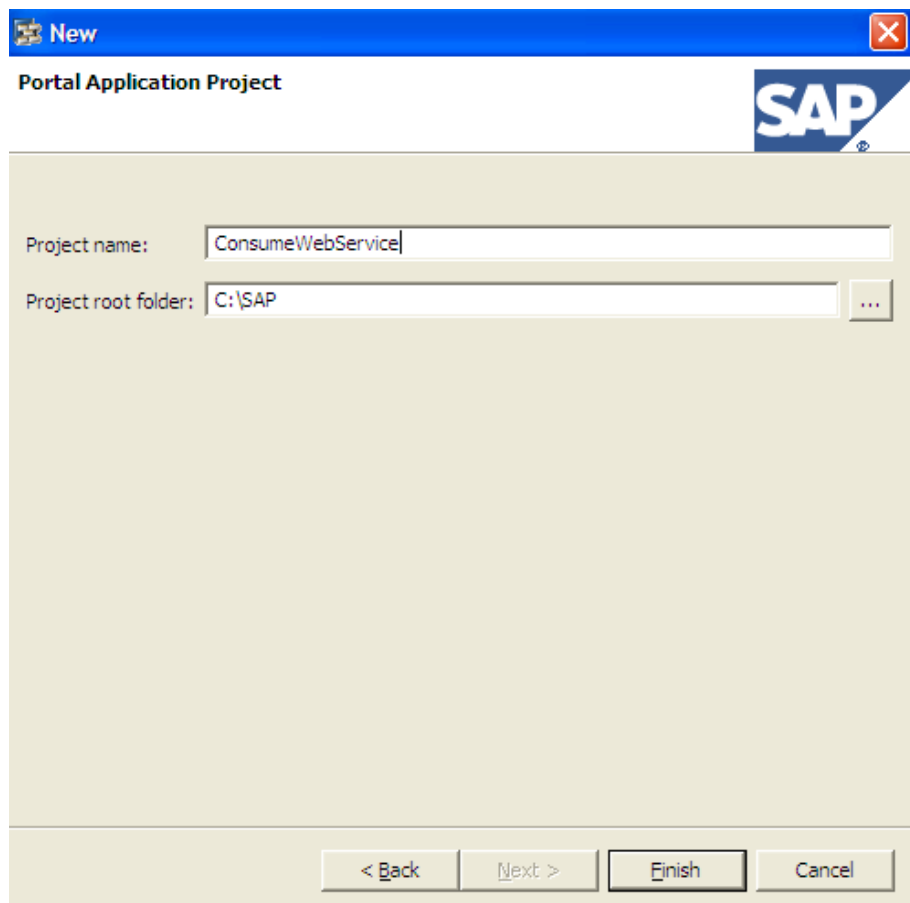
In order to use an ABAP BAPI in PAR file you need to make sure:

- A webservice is created from the BAPI along with its end-points in SOAMANAGER.
- You have successfully test the webservice using the Web service navigator.
- You have created a deployable webservice DC in NWDS using the WSDL.

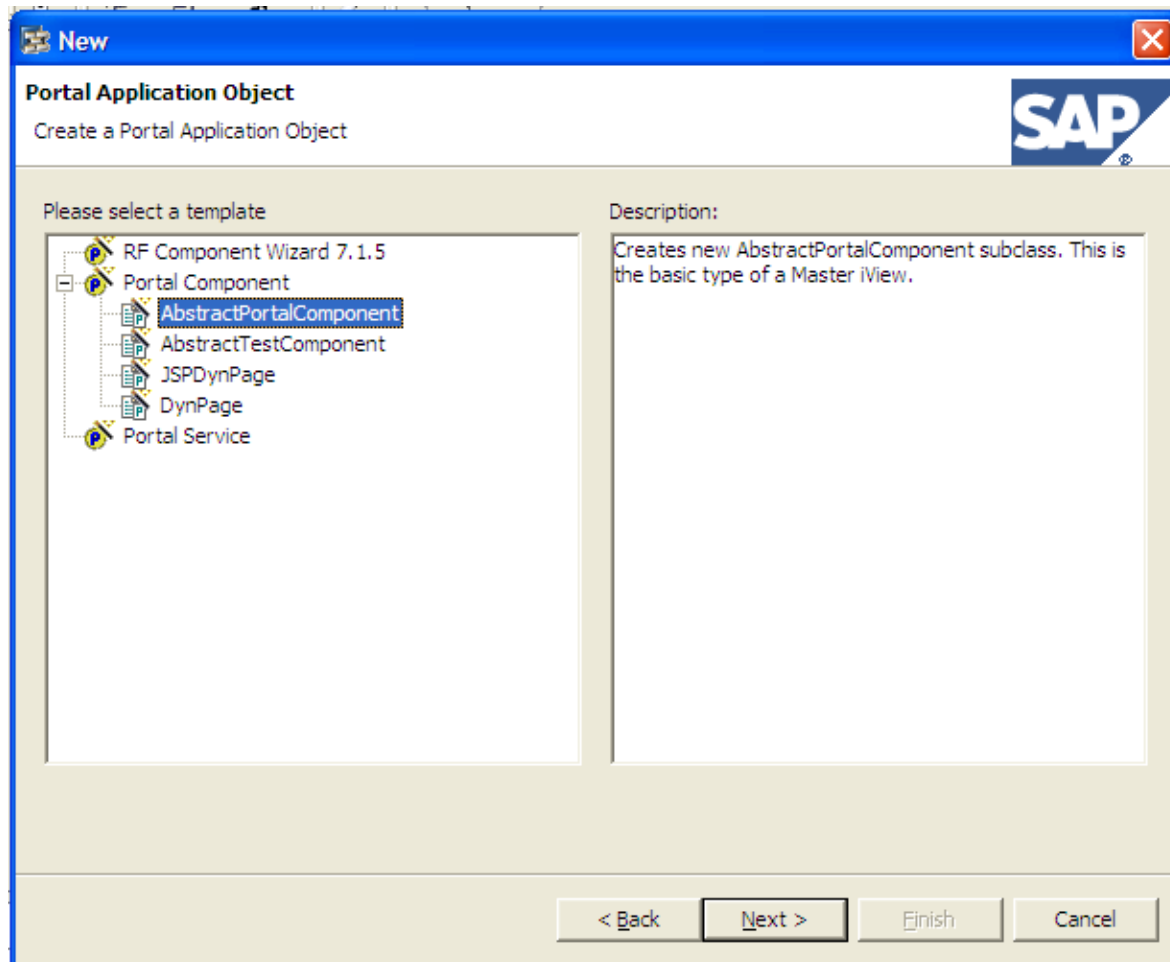
Consuming ABAP webservice

Below steps explain how to consume the webservice in a PAR project

- 1) Create a PAR project **ConsumeWebService**



2) Create an Abstract Portal Component **TestWebservice.java** in the PAR project.



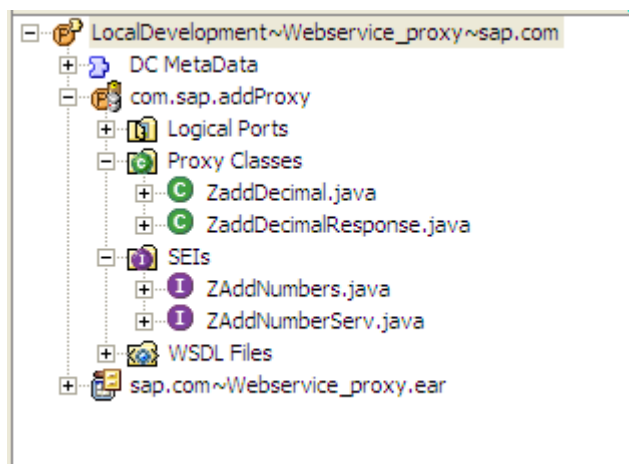
The Abstract Portal component after creation should look like this:



3) Add a sharing reference in the Portalapp.xml file for the Web Service J2EE dc.

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
  <application-config>
    <property name="SharingReference" value="SAPJ2EE::sap.com/Webservice_proxy"/>
  </application-config>
  <components>
    <component name="TestWebservice">
      <component-config>
        <property name="ClassName" value="com.sap.TestWebservice"/>
      </component-config>
      <component-profile/>
    </component>
  </components>
  <services/>
</application>
```

The Webservice DC should be deployed on Server and should look like this:



4) Write the below code to call the Webservice in the TestWebservice.java doContent method.

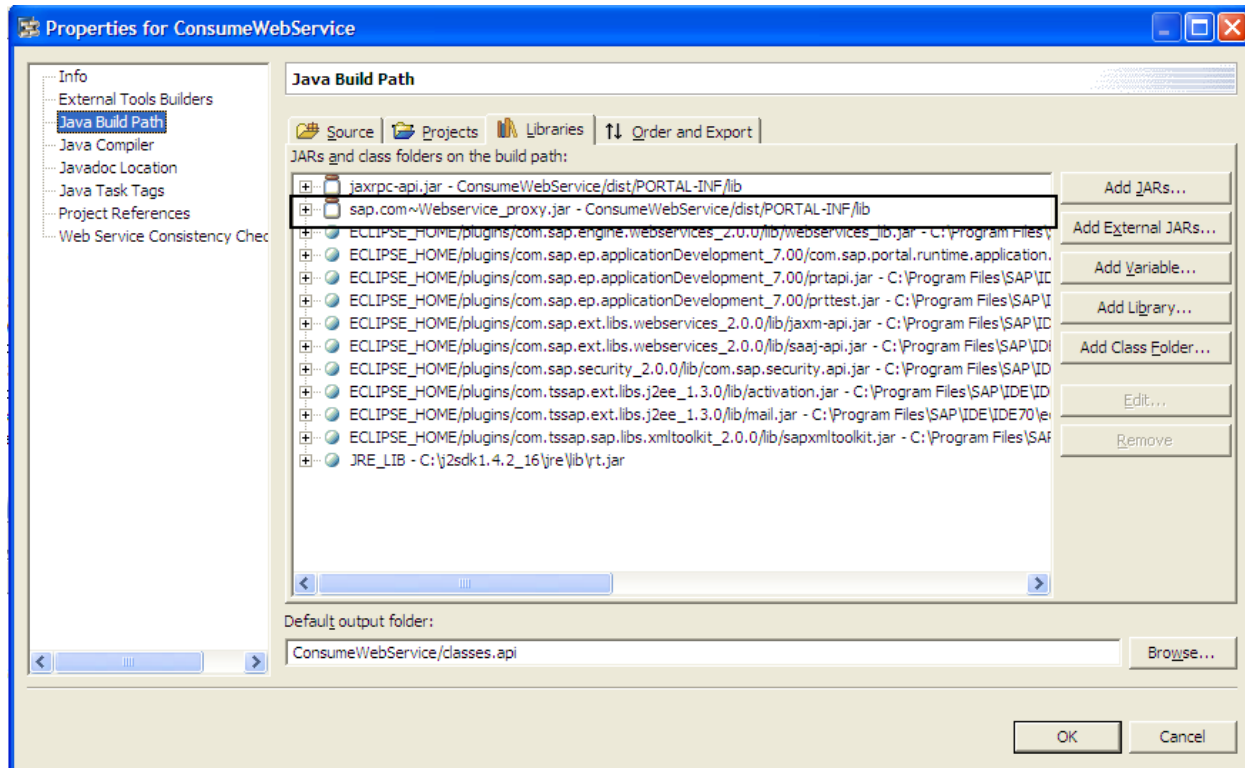
```
try{

//          Call Webservice to add two numbers
          InitialContext context = new InitialContext();
          Object obj =
context.lookup("/wsclients/proxies/sap.com/Webservice_proxy/com.sap.addProxy");
          ZAddNumberServ service = (ZAddNumberServ )obj;

          ZAddNumbers port = (
ZAddNumbers) service.getLogicalPort(ZAddNumbers.class);

          ZaddDecimalResponse resp = new ZaddDecimalResponse();
          resp.setTotal(port.zaddDecimal(23,24));
          response.write("The SUM =" +Integer.toString(resp.getTotal()));
}
catch(Exception ex){
          response.write(ex.getMessage());
}
}
```

- 5) To compile the code add the Webservice DC jar file into the LIB folder of PAR project, and then add it as a reference.



The file will be present at the below path on server

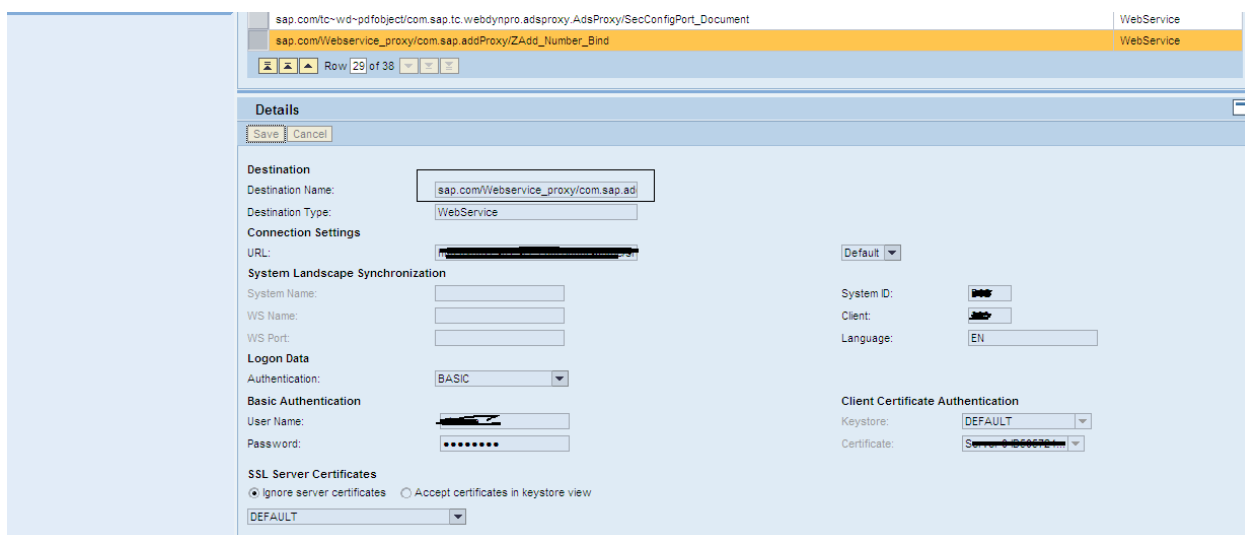
**\\j2ee\cluster\server0\apps\sap.com\
Webservice_proxy\webservices_container\wsclients\app_jars**

- 6) Provide authentication information for BAPI call in Netweaver Administrator

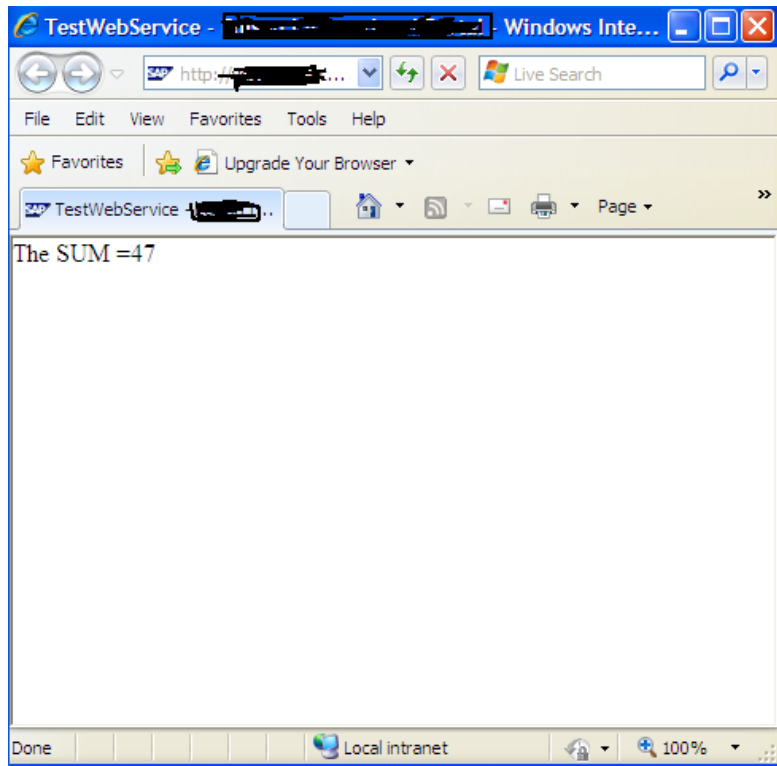
Go to <http://<Portal hostname>:<port>/nwa>

Go to configuration->Destinations and Search for your webservice

Click on Edit and enter system details and user details.

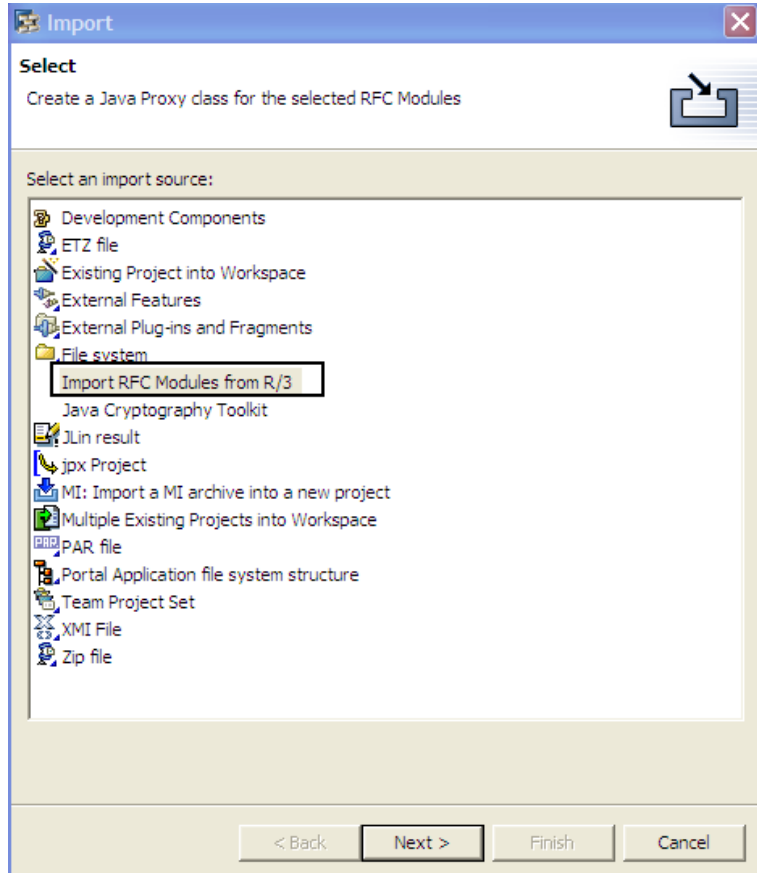


7) Create an iview form the par file and you should see below output:

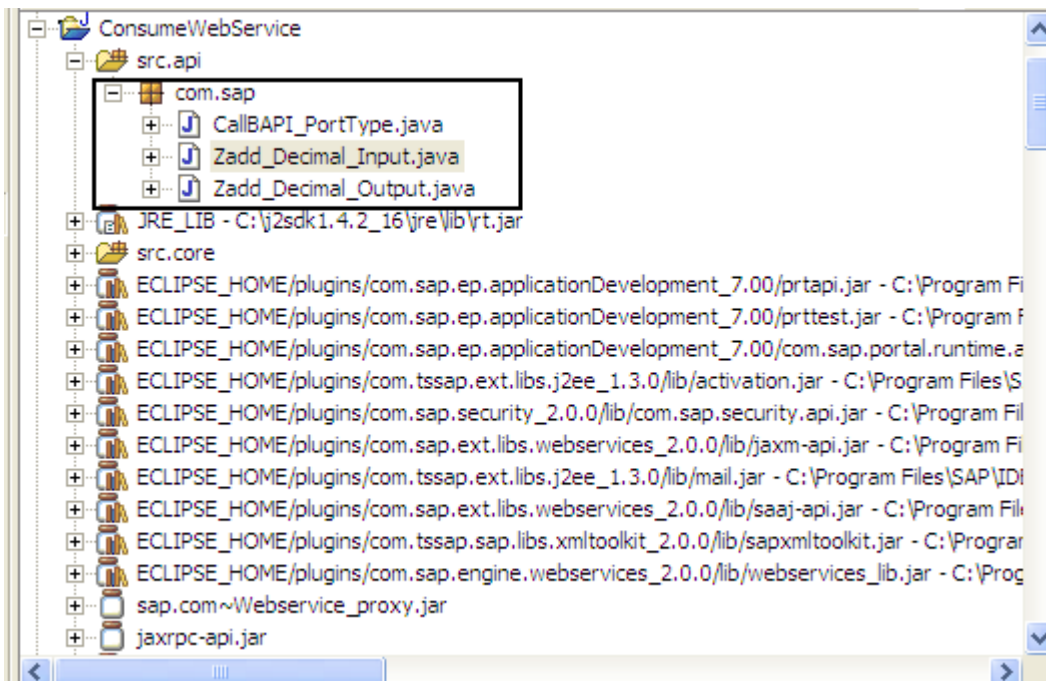


Alternative Approach

We can make an RFC call in the PAR; the framework itself creates the proxy classes.



After importing the RFC you see the below classes in your PAR project:



Please check the references to get more detail about generated proxies.

The major benefits in using the Webservice for BAPI consumption are:

- 1) We can reuse the Webservice in different projects.
- 2) No need to create JCOs in the code
- 3) Loosely coupled to the system as compared to generated proxies..

Related Content

[Example of using generated Proxies to call FM](#)

[Creating a Webservice deployable proxy DC](#)

[J2EEApplication Access from Portal Applications](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.