

BAPI Execution in offline Adobe Form



Applies to:

Adobe form, Web dynpro JAVA, SAP ECC. For more information, visit the [Web Dynpro Java homepage](#).

Summary

This article contains step by step description for executing BAPI in offline adobe forms starting with creation of web service in backend system followed by adobe form creation executing the web service.

Author: Namrata Dixit

Company: CapGemini

Created on: 15 March 2011

Author Bio

Namrata Dixit is currently working with CapGemini as a Netweaver Portal consultant. She has 4.8 years of experience in SAP Enterprise portal. Her skill sets include CE BPM , GP, MDM, Adobe Forms, KM, Web Dynpro JAVA and UWL.

Table of Contents

Executing BAPI in offline Adobe Form	3
Creating Webservice for BAPI in ECC.....	3
Creating Interactive form for Web service execution	8
Related Content.....	12
Disclaimer and Liability Notice.....	13

Executing BAPI in offline Adobe Form

One of an important scenario that can be implemented in offline adobe form is the execution of BAPI. In this scenario, an adobe form can be stored locally on file system, values can be entered and on click of a button BAPI can be executed. This approach offers advantages :

- a. Explicit login to SAP system not required for BAPI execution
- b. login to Portal not required for any custom application developed for BAPI execution
- c. Form can be saved on the local file system, printed, archived etc.
- d. No training required for this approach, if the user is aware of the request structure for BAPI execution.

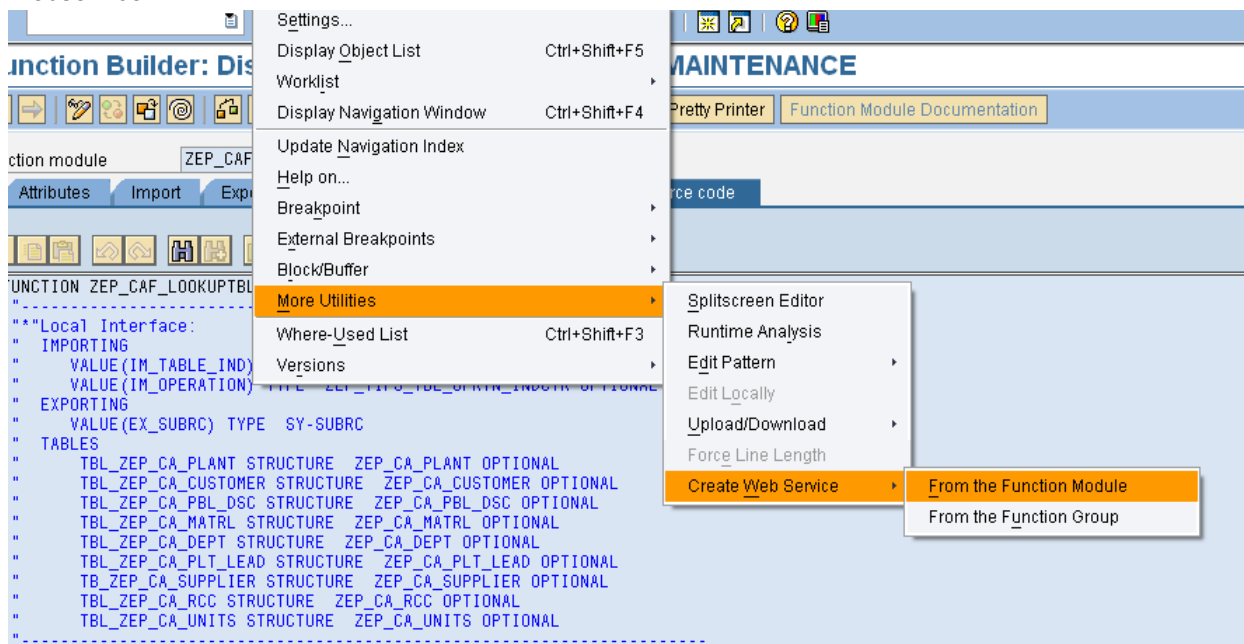
Below are the two steps required for executing BAPI on an offline adobe form

1. Creating Web Service for BAPI in ECC
2. Developing Interactive form which consumes this webservice.

This article will explain both the steps in details with relevant screenshot:-

Creating Webservice for BAPI in ECC

- a. Login to backend system with SAP UserID/Passwd.
- b. Run Transaction se37 and select the desired Function module for which you need to create Webservice.



c. Enter the Webservice name and description

Provide Service Definition details

Enter a name and a short description for the Web Service and choose an end-point type.

To change the Web Service, use the ABAP Workbench (transaction SE80).

Object Type

- Choose Endpoint
- Choose Operations
- Configure Service
- Enter Package/Request
- Complete

Service Definition: ZEP_TEST_LOOKUP_WS

Kurzbeschreibung: Test webservice for look up data

Endpoint Type: Function Module

Back Continue Cancel

d. The function module name will get populated automatically.

Choose Endpoint

Enter the name of the function module that is to be offered as a Web Service.

If you select the checkbox <ZKName Mapping, the existing descriptions of the end point elements are copied. Initial letters are capitalized and underscores removed.

Object Type

- Choose Endpoint
- Choose Operations
- Configure Service
- Enter Package/Request
- Complete

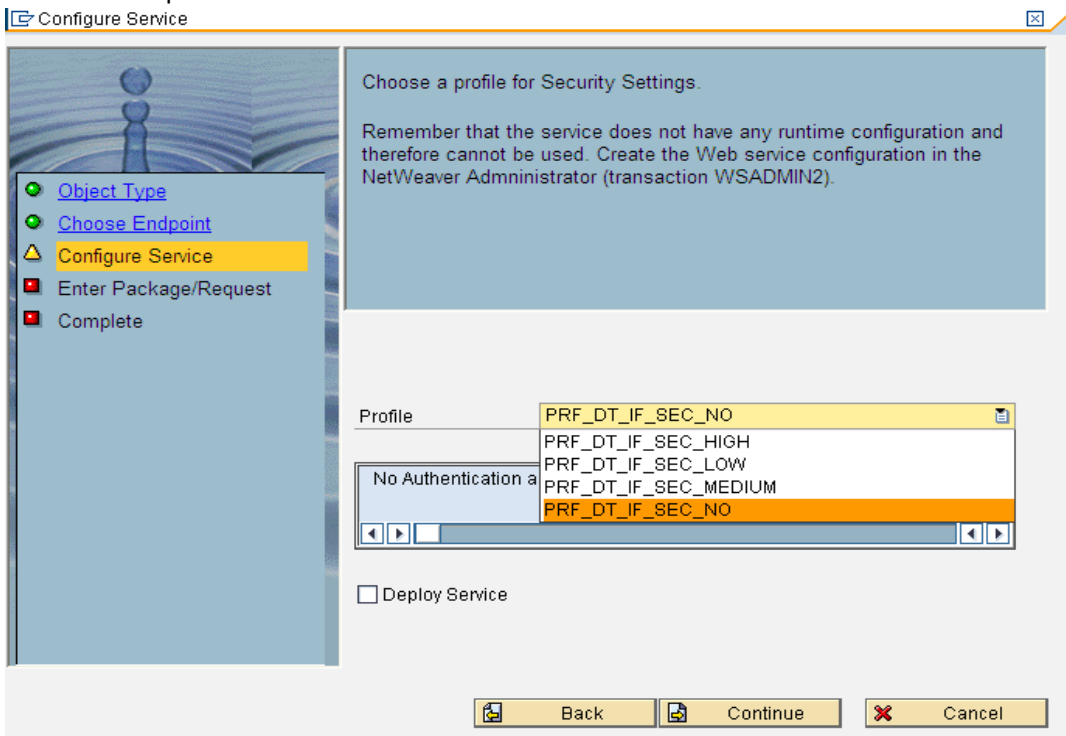
Function Module: ZEP_CAF_LOOKUPTBLS_MAINTENANCE

Mapping der Namen

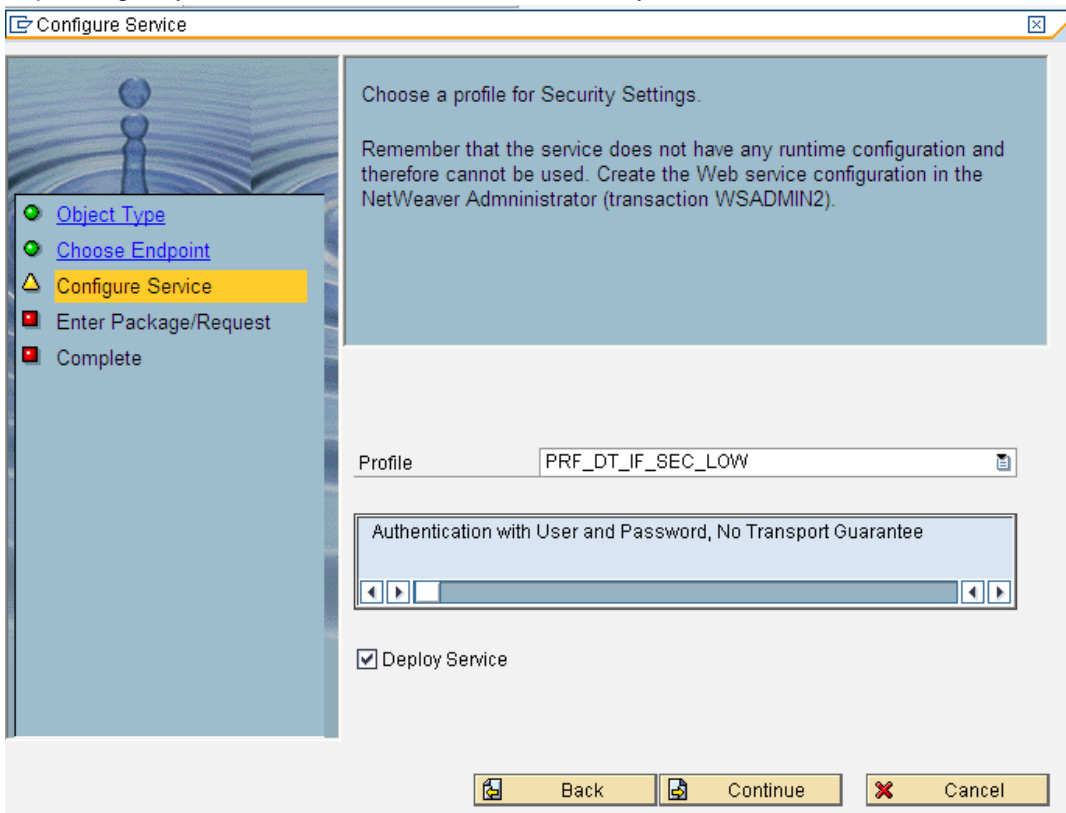
Back Continue Cancel

- e. In the next step we set the authorization profile for web service. Here in this case I have selected “authentication with user and password”. This suggests that while executing the web service from adobe form, it will ask user to enter UserID/Password for execution.

The different profiles that we can set for the web service are shown below.



Depending on your business scenario , set the security level for the web service.



- f. Enter the package name used for transport. Here I have set it as a local object

Enter Package/Request

Enter the name of the package and transport request.

Package/Request

Package: \$TMP

Transport Request:

Local Object

Back Continue Cancel

- g. Web Service creation is completed

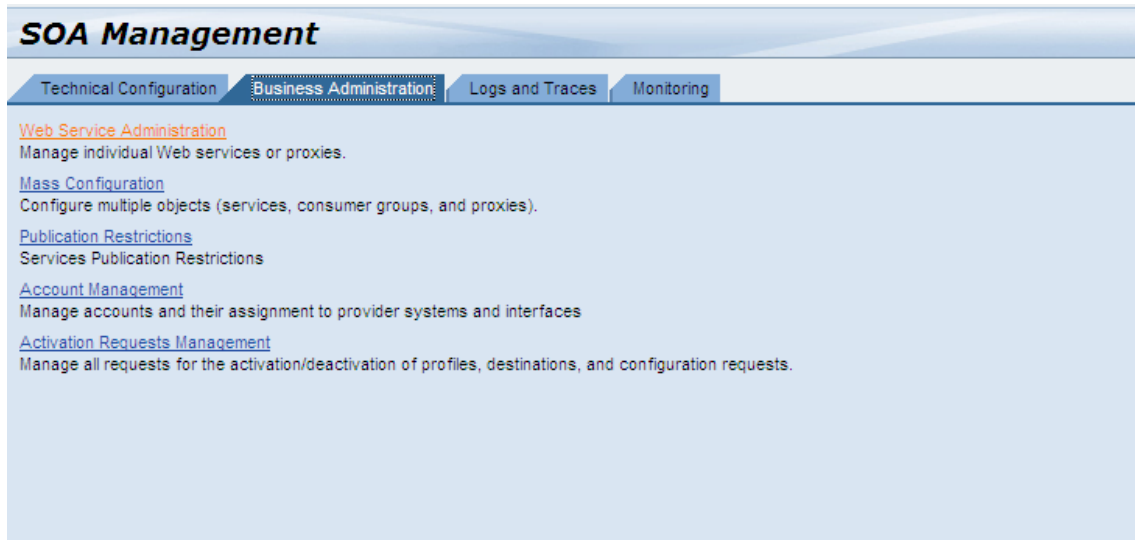
Complete

The Web Service ZEP_TEST_LOOKUP_WS will be created.

Transaction WSADMIN (Administration Web Services for SOAP Runtime) supports you with the UDDI registration of the Web Service.

Back Complete Cancel

- h. To check the web service and the endpoints, run transaction SOAMANAGER. Go to business Administration -> Web service administration



SOA Management

Technical Configuration | **Business Administration** | Logs and Traces | Monitoring

[Web Service Administration](#)
Manage individual Web services or proxies.

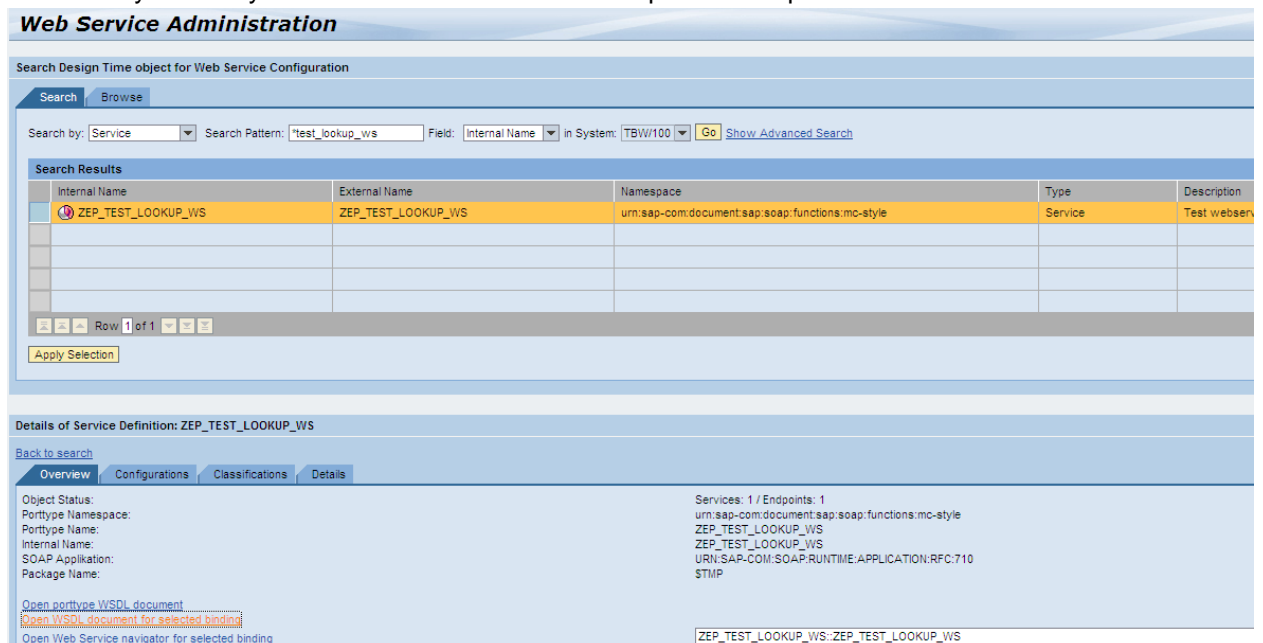
[Mass Configuration](#)
Configure multiple objects (services, consumer groups, and proxies).

[Publication Restrictions](#)
Services Publication Restrictions

[Account Management](#)
Manage accounts and their assignment to provider systems and interfaces

[Activation Requests Management](#)
Manage all requests for the activation/deactivation of profiles, destinations, and configuration requests.

- i. Search for your newly created web service with the help of search pattern.



Web Service Administration

Search Design Time object for Web Service Configuration

Search | Browse

Search by: Service Search Pattern: test_lookup_ws Field: Internal Name in System: TBW100 Go Show Advanced Search

Search Results

Internal Name	External Name	Namespace	Type	Description
ZEP_TEST_LOOKUP_WS	ZEP_TEST_LOOKUP_WS	urn:sap-com:document:sap:soap:functions:mc-style	Service	Test webser

Row 1 of 1

Apply Selection

Details of Service Definition: ZEP_TEST_LOOKUP_WS

Back to search

Overview | Configurations | Classifications | Details

Object Status: Services: 1 / Endpoints: 1
 Porttype Namespace: urn:sap-com:document:sap:soap:functions:mc-style
 Porttype Name: ZEP_TEST_LOOKUP_WS
 Internal Name: ZEP_TEST_LOOKUP_WS
 SOAP Application: URN:SAP-COM:SOAP:RUNTIME:APPLICATION:RFC:710
 Package Name: STMP

[Open porttype WSDL document](#)
[Open WSDL document for selected binding](#)
[Open Web Service navigator for selected binding](#)

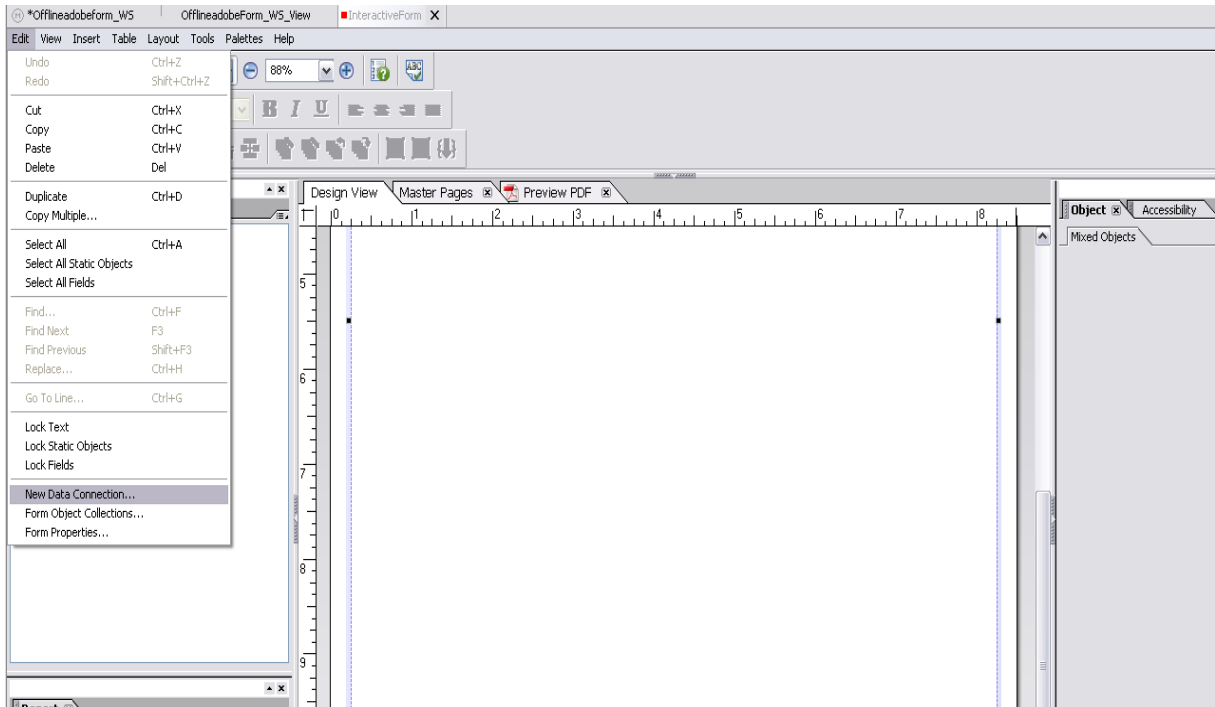
ZEP_TEST_LOOKUP_WS:ZEP_TEST_LOOKUP_WS

- j. Click on the link “Open WSDL Document for selected binding”. A new Internet explorer will open with the WSDL document binding file. This WSDL will be needed for creating the webservice in NWDS . You can also store the WSDL and import it while creating Data Connection in NWDS.

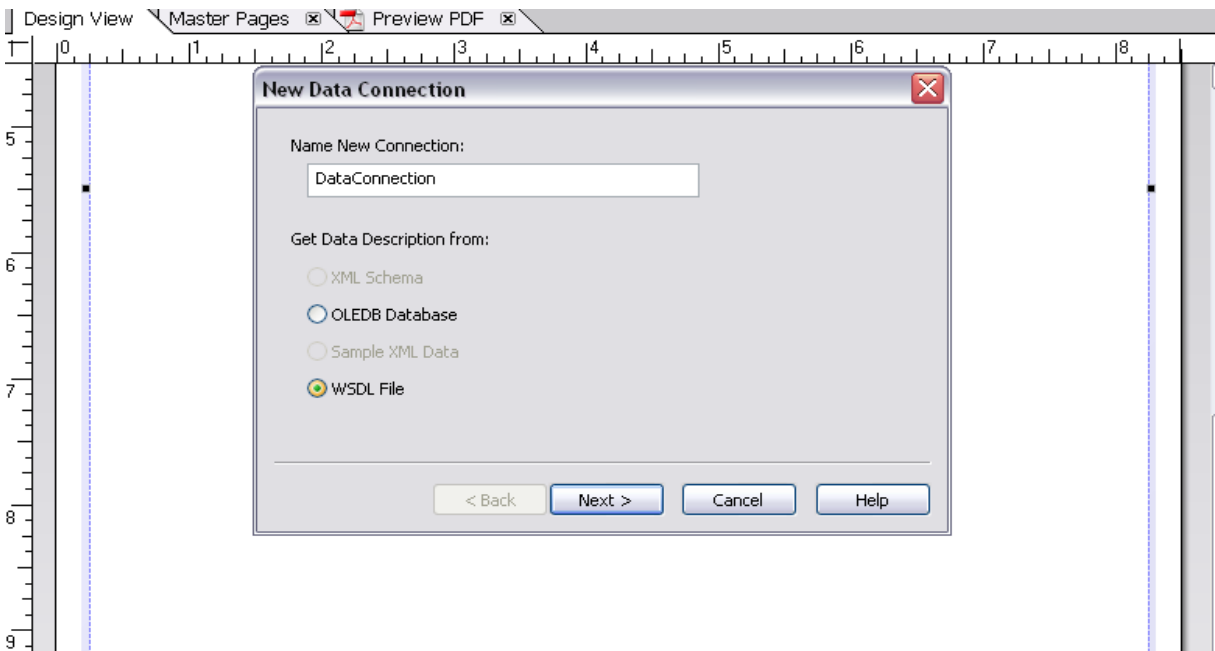
Till this step we are done with Webservice creation and we have WSDL needed to import in NWDS for its execution.

Creating Interactive form for Web service execution

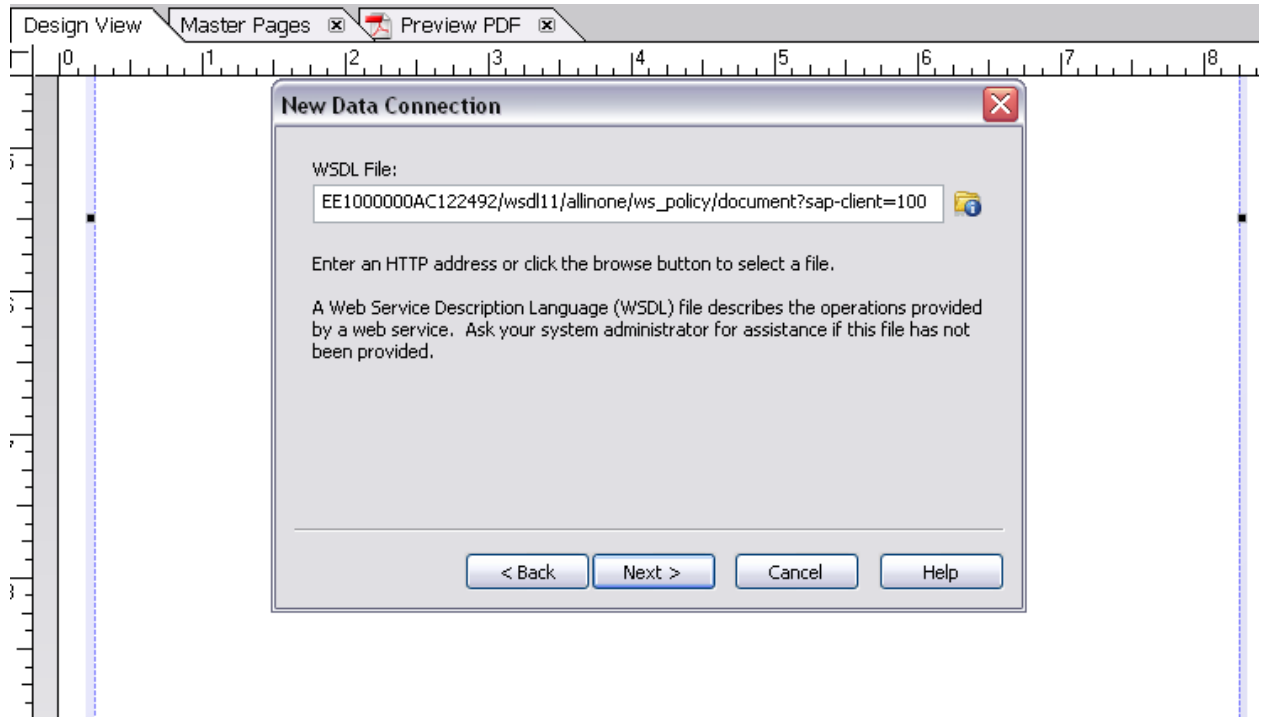
- Create Webdynpro project, embed a view and add Interactive Form as UI Element. Right click on Interactive Form UI element and click on “Edit”.
- The interactive form opens as shown below. We will now create new Data Connection that will import the WSDL for execution. Click on Edit -> New Data connection



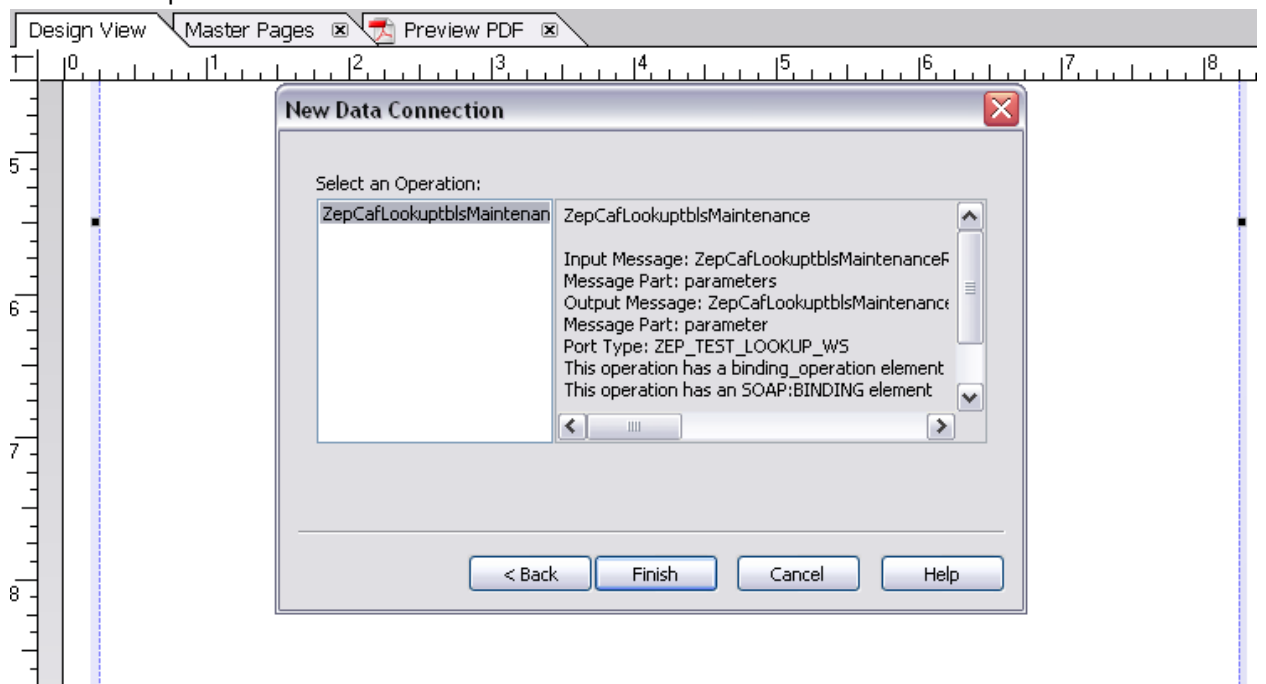
- Enter the desired Data Connection name.



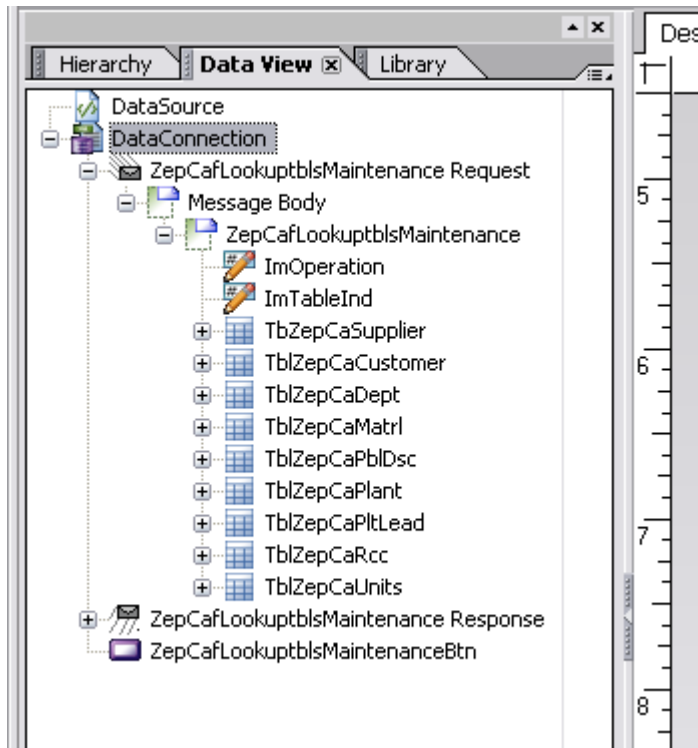
- Enter the WSDL URL mentioned in Step j of previous section.



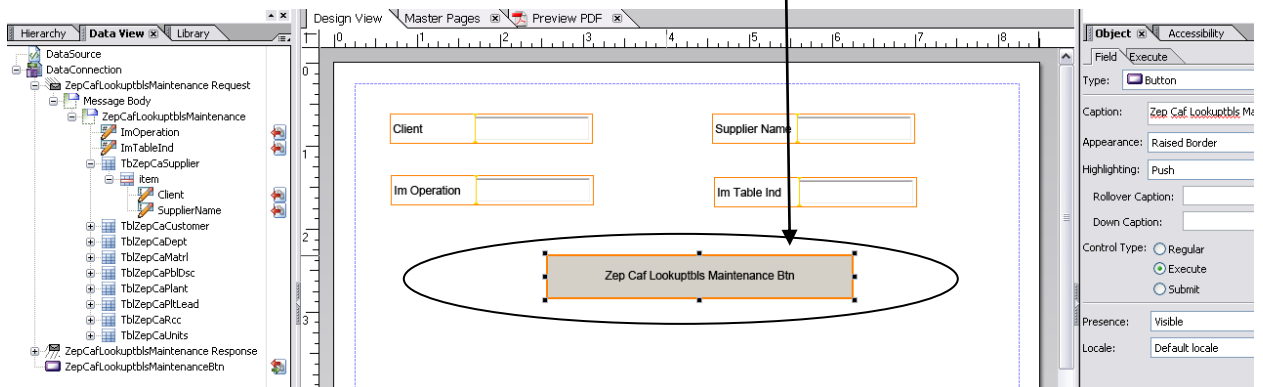
- e. Click on the operation name and finish.



- f. Once the WSDL is imported, you will view the BAPI structure in the Data view of the form as shown below



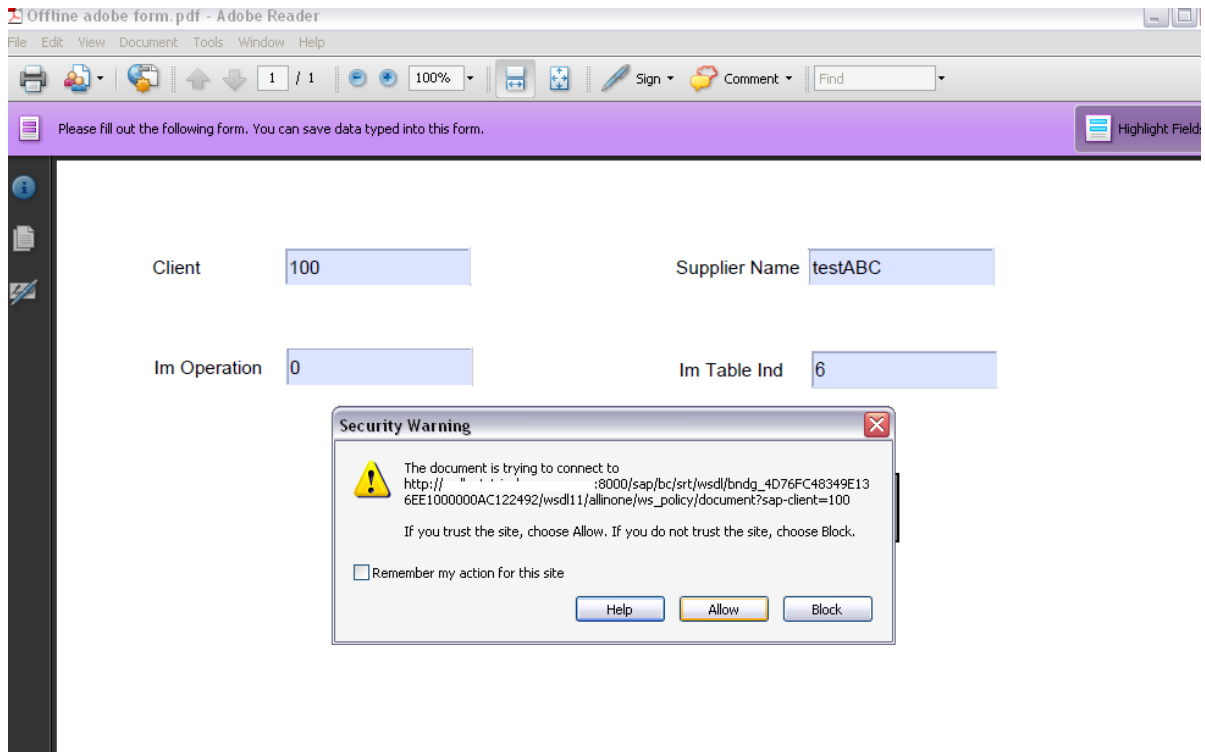
- g. Drag the fields from the Data view to the layout. Also drag the Button that will be execute the web service.



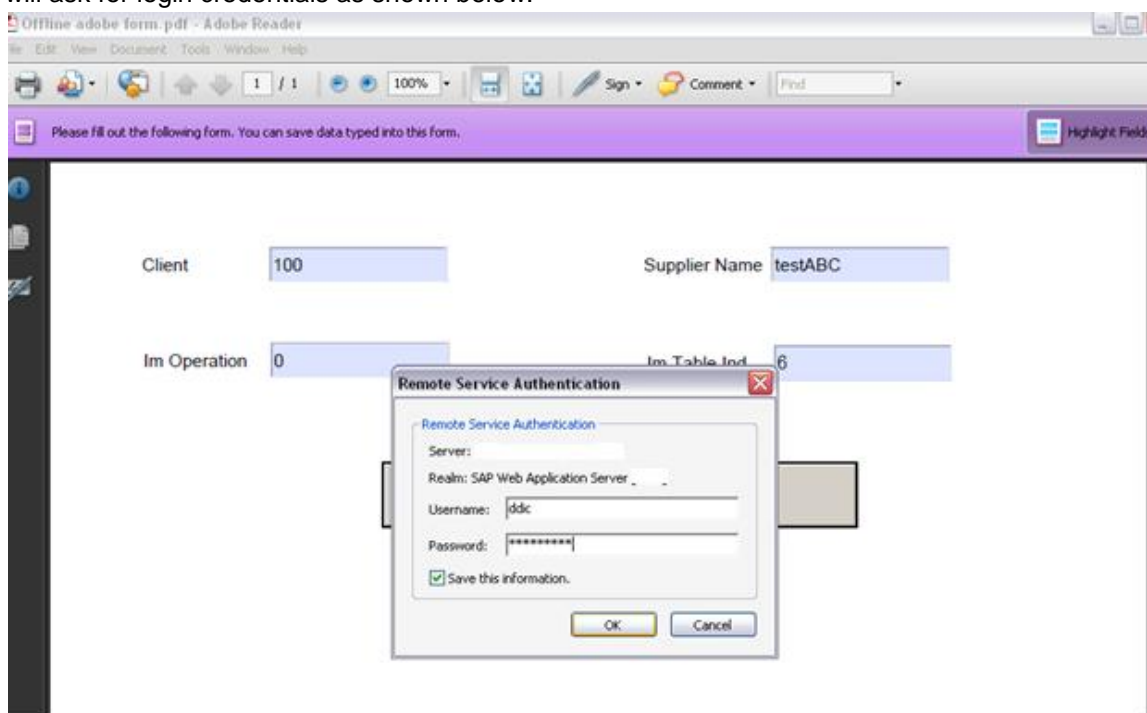
- h. On click event of the button on the form , write the below lines of code.

```
var cURL =
"http://xyz:8000/sap/bc/srt/wsd1/bndg_4D76FC48349E136EE1000000AC122492/wsd111/alli
none/ws_policy/document?sap-client=100";
var service = SOAP.connect(cURL);
xfa.connectionSet.DataConnection.execute(0);
```

- i. Save the form on the local file system. Enter all the details on the form and click on the web service button.



- j. During web service creation since we selected profile as “Authentication with user and password”, it will ask for login credentials as shown below.



- k. Here by this web service execution , a new record will be created in Supplier table with entered Supplier Name as “testABC”.

CLIENT	SUPPLIER_NAME
100	testABC

Related Content

[SAP Interactive forms by Adobe for off line processing in java](#)

[Adobe Document Services Configuration Guide](#)

For more information, visit the [Web Dynpro Java homepage](#).

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.