



## FLM Web Service Call

**Author:** Martin Cook  
**File name:** 271 Web Service Call  
**Version:** 1  
**Distribution:**

## Version History

Version	Author(s)	Reason for update
1	Martin Cook	Initial release of the document

## Table of Contents

1	Web Service.....	3
1.1	Function Module .....	3
1.2	Creating Web Service from Function Module .....	4
1.3	Configuring Web Service .....	7
1.4	Web Service Authentication .....	9
1.5	Testing The Web Service .....	11
1.6	Savine the Web Service Definition .....	15
1.7	Adding a Web Service to the Form .....	17

# 1 Web Service

## 1.1 Function Module

Create a Function Module name Z\_FLM\_HEARTBEAT with the below code

```

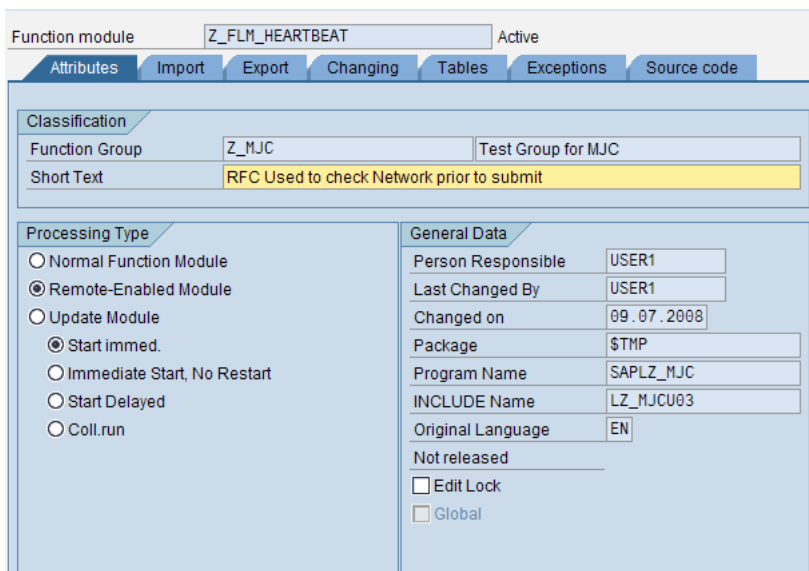
FUNCTION z_flm_heartbeat.
*-----
* " * "Local Interface:
* "   EXPORTING
* "       VALUE(HEARTBEAT) TYPE  STRING
*-----
* initial with Todays Date
  DATA : w_date_string TYPE string,
          w_time_string TYPE string.

  w_date_string = sy-datum.
  w_time_string = sy-uzeit.

  CONCATENATE w_date_string(4) '-' w_date_string+4(2) '-'
              w_date_string+6(2)
              '@' w_time_string(2) ':' w_time_string+2(2) ':'
              w_time_string+2(2) INTO heartbeat.
ENDFUNCTION.

```

Make sure the Remote-Enabled Flag is checked as shown below

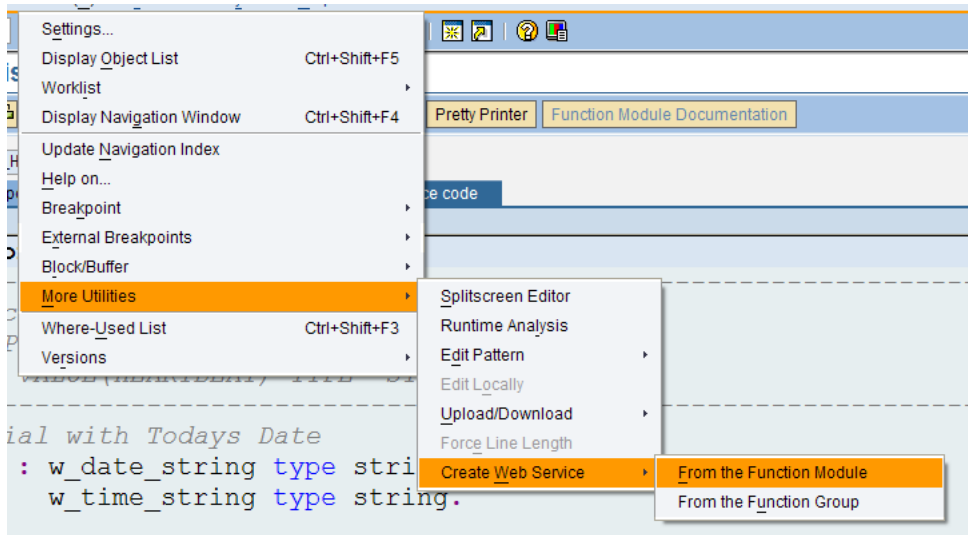


Now Save and Activate the Function Module.

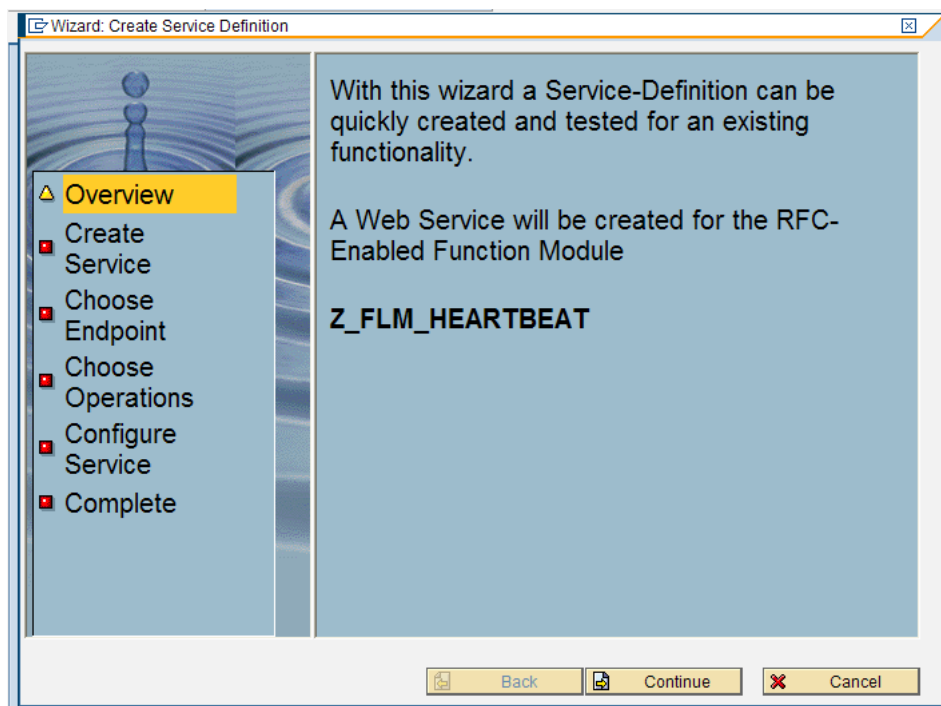
## 1.2 Creating Web Service from Function Module

With the function Module Z\_FLM\_HEARTBEAT in display mode or in change mode view the Source and then select the following Menu

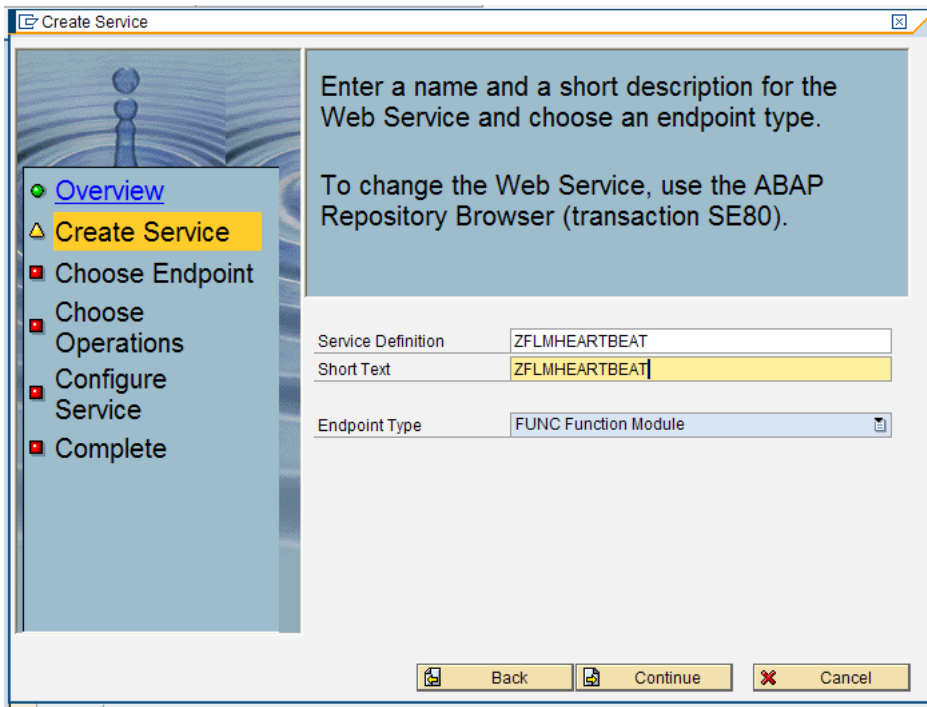
Utilities>More Utilities>Create Web Service>From the Function Module



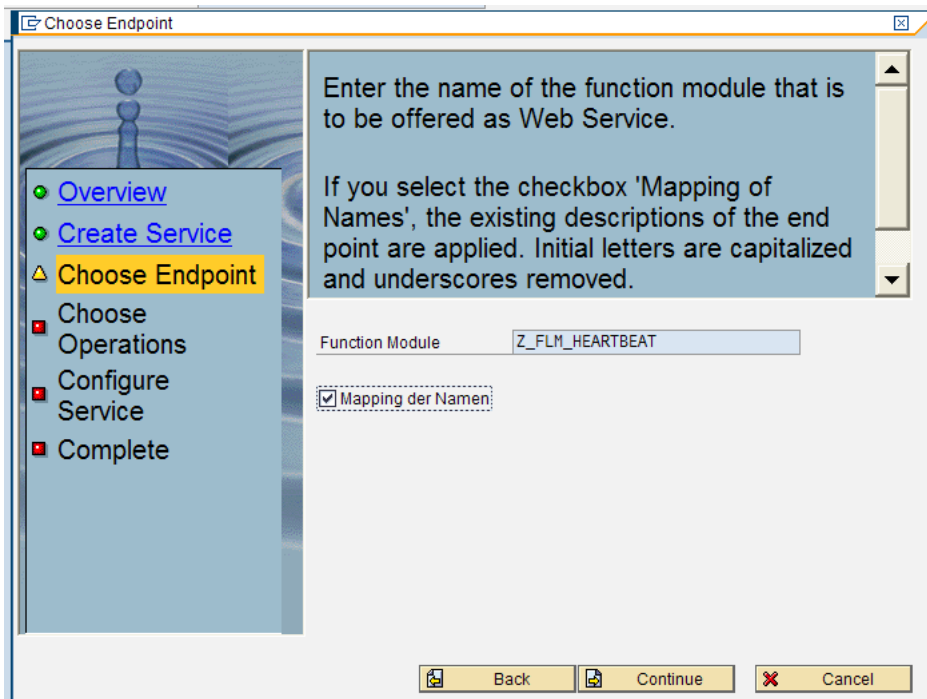
This will start the Web Service Wizard



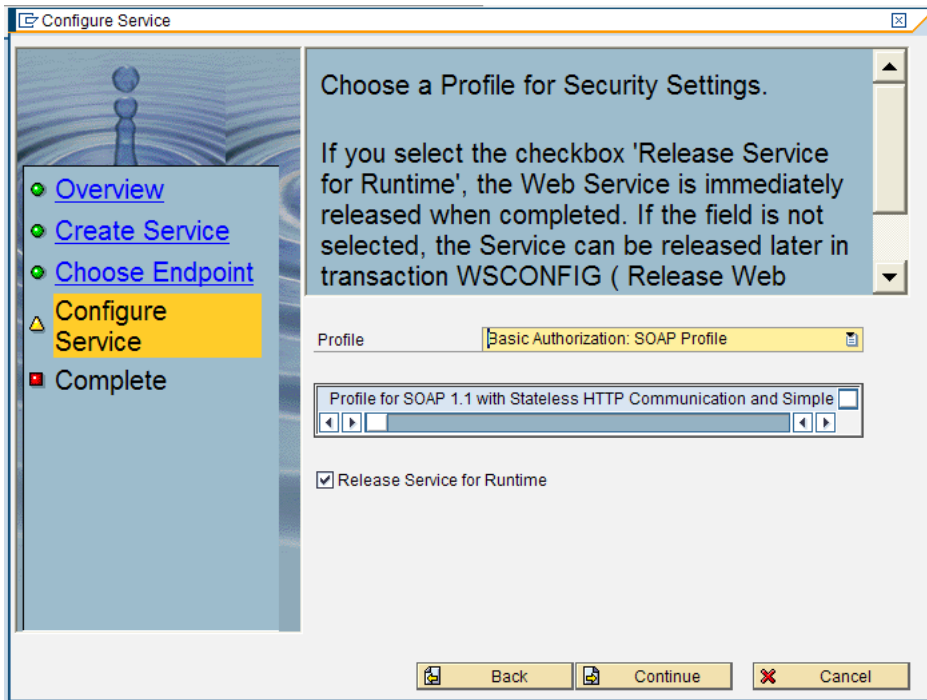
Click Continue



Enter ZFLMHEARTBEAT for the Service Definition and Short Text and click Continue



Make sure Mapping der Namen is checked and click Continue



Make sure **Basic Authorization : SOAP Profile** and **Release Service for Runtime** are selected and click **Continue**

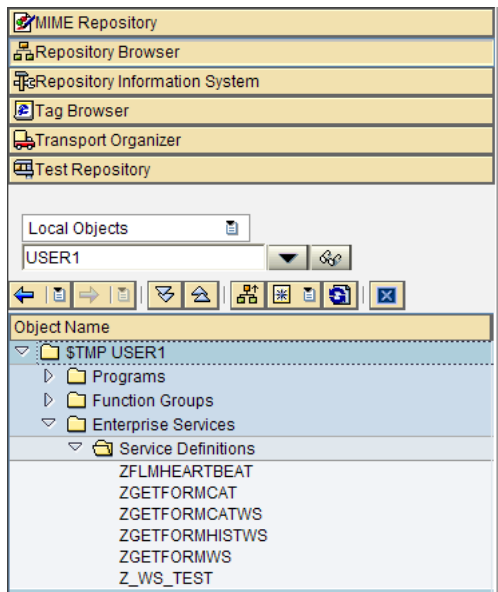


Click **Complete**, when prompted for a Package select **\$TMP**

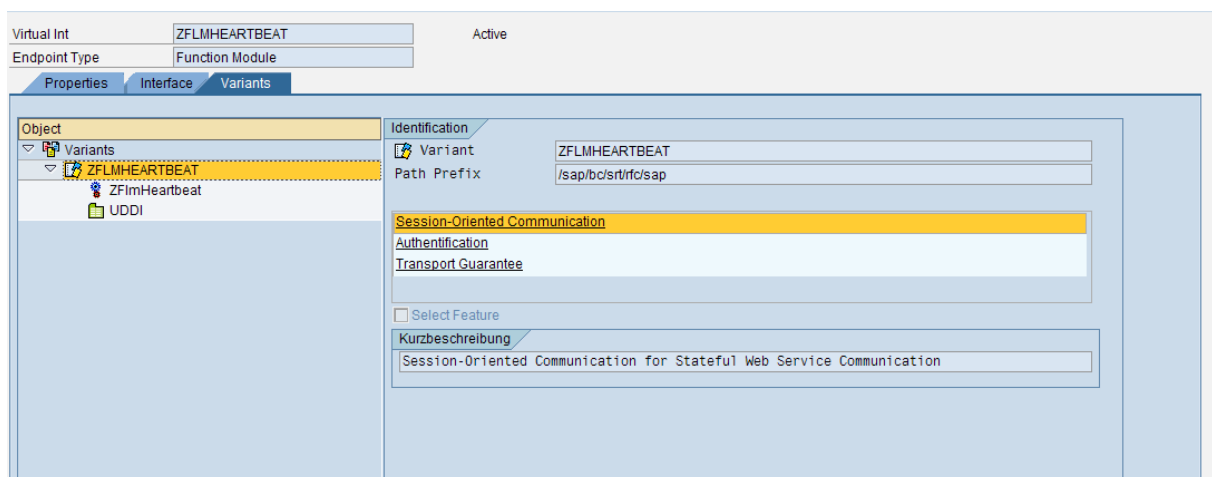
### 1.3 Configuring Web Service

Using SE80, Select Repository Browser and Select the Package you saved the Web Service in. Select Enterprise Services>Service Definition and you see ZFLMHEARTBEAT listed.

The below shows screen shots (NB this is for package \$TMP)



Double Click ZFLMHEARTBEAT and Select the Variants Tab, Select ZFLMHEARTBEAT by Double Clicking it, this will display the below screen



Select **Authentication** and if not in Edit Mode Select Edit and change the Radio Selection to **None**.

**Identification**

Variant

Path Prefix

**Session-Oriented Communication**

**Authentication**

Transport Guarantee

Select Feature

**Kurzbeschreibung**

Security: Authentication (Identity Check)

**Authentication Level**

Select Property

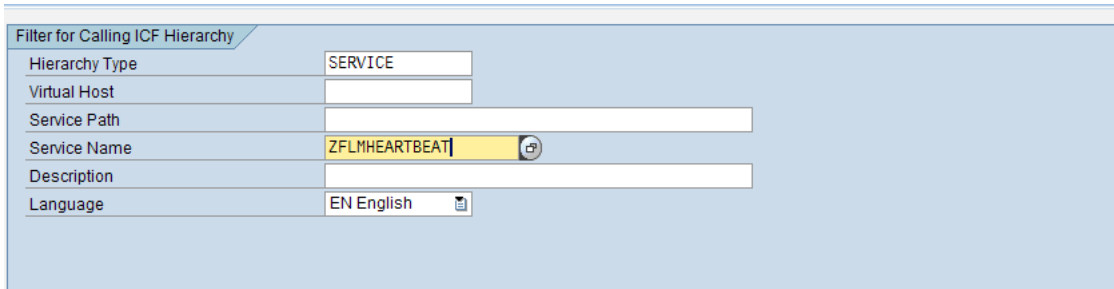
**Values**

<input checked="" type="radio"/> None	Default Value
<input type="radio"/> Basic	
<input type="radio"/> Strong	

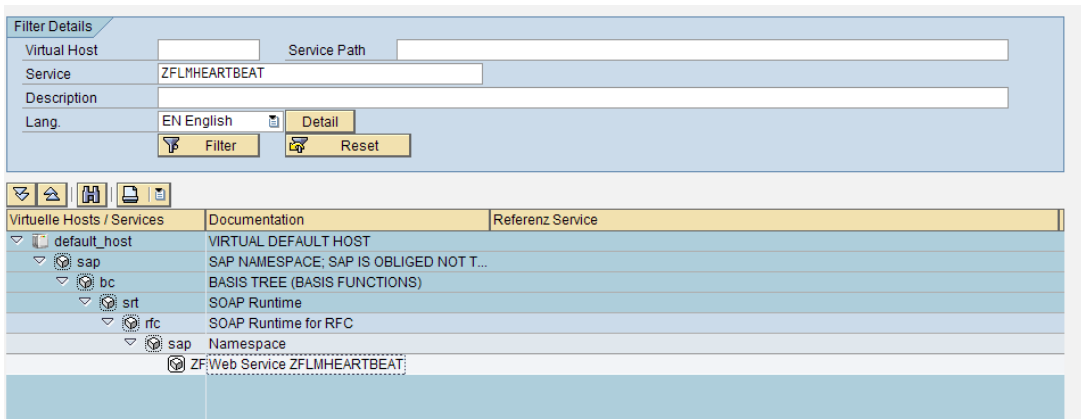


## 1.4 Web Service Authentication

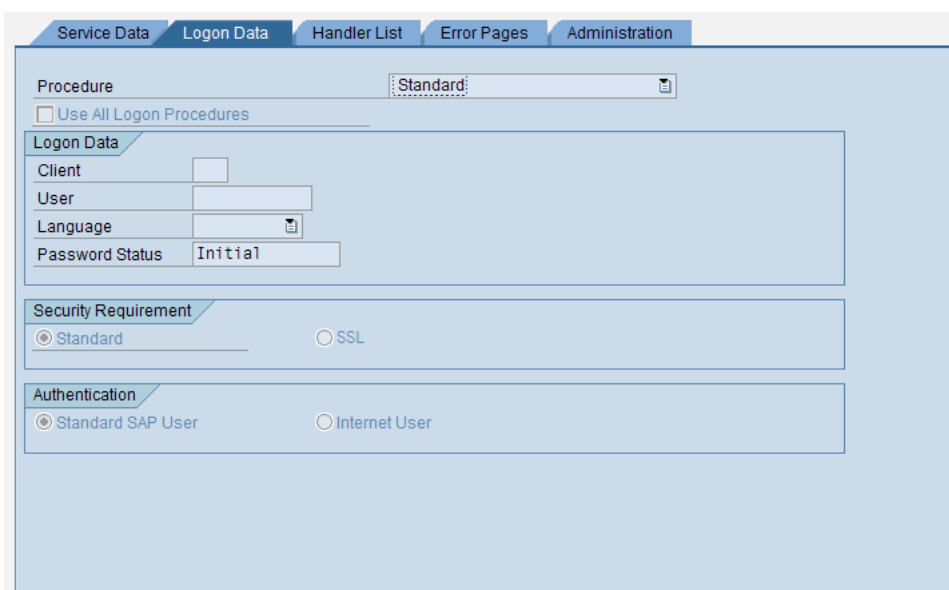
Using SICF Make sure Hierarchy Type Selected is Service and enter ZFLMHEARTBEAT in the Service Name and click Execute



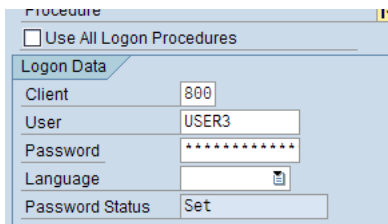
You should now see the below



Double click The Web Service ZFLMHEARTBEAT and Select the Logon Tab.



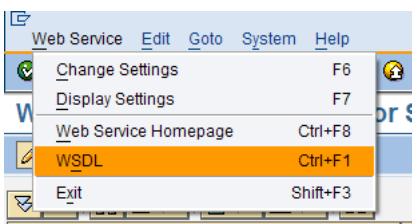
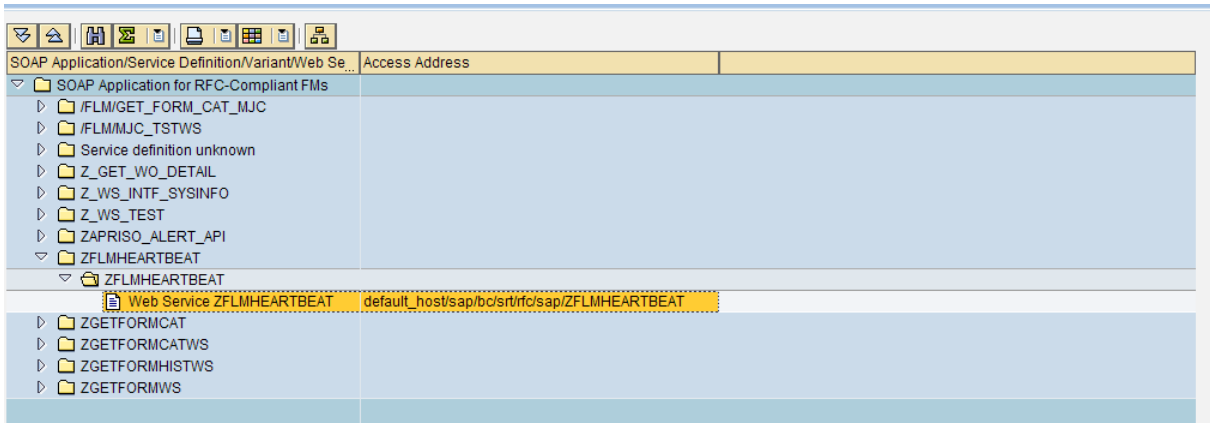
Enter Edit Mode and Set-up the Logon Data Correctly in our case we are using Client 800 and user USER3 for the RFC Call.

A screenshot of a software configuration window titled 'Procedure'. At the top, there is a checkbox labeled 'Use All Logon Procedures' which is currently unchecked. Below this is a section titled 'Logon Data' containing several input fields: 'Client' with the value '800', 'User' with the value 'USER3', 'Password' with a masked value of ten asterisks, 'Language' with a dropdown menu icon, and 'Password Status' with the value 'Set'.

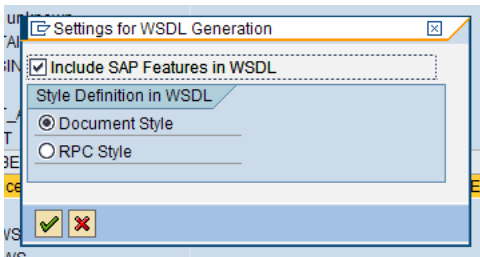
Now click Save

## 1.5 Testing The Web Service

Using WSADMIN Expand the Selection until you have the Web Service for ZFLMHEARTBEAT Selected



Now from the menu Select Web Service>WSDL



Select the Style Definition to be Document Style and also ensure that Include SAP Features in WSDL is selected, and click Continue

This will launch a new IE Browser session and will show the WSDL File

```

http://quattro.w4.local:8003/sap/bc/srt/rfc/sap/ZFLMHEARTBEAT?sap-client=800&wsdl=1.1&mode=sap_wsdl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites
Address http://quattro.w4.local:8003/sap/bc/srt/rfc/sap/ZFLMHEARTBEAT?sap-client=800&wsdl=1.1&mode=sap_wsdl Go Links
<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions targetNamespace="urn:sap-com:document:sap:soap:functions:mc-style" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:n0="http://www.sap.com/webas/630/soap/features/authentication/" xmlns:sap="http://www.sap.com/webas/630/wsdl/features"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="urn:sap-com:document:sap:soap:functions:mc-style"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <wsdl:types>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn:sap-com:document:sap:soap:functions:mc-style"
  targetNamespace="urn:sap-com:document:sap:soap:functions:mc-style" elementFormDefault="unqualified" attributeFormDefault="qualified">
- <xsd:element name="ZFImHeartbeat">
- <xsd:complexType>
- <xsd:sequence />
- </xsd:complexType>
- </xsd:element>
- <xsd:element name="ZFImHeartbeatResponse">
- <xsd:complexType>
- <xsd:sequence>
- <xsd:element name="Heartbeat" type="xsd:string" />
- </xsd:sequence>
- </xsd:complexType>
- </xsd:element>
- </xsd:schema>
- </wsdl:types>
- <wsdl:message name="ZFImHeartbeat">
- <wsdl:part name="parameters" element="tns:ZFImHeartbeat" />
- </wsdl:message>
- <wsdl:message name="ZFImHeartbeatResponse">
- <wsdl:part name="parameters" element="tns:ZFImHeartbeatResponse" />
- </wsdl:message>
- <sap:Feature name="design_0" uri="http://www.sap.com/webas/630/soap/features/authentication/">
- <sap:Property qname="n0:AuthenticationLevel">
- <sap:Option value="n0:None" />
- </sap:Property>
- </sap:Feature>
- <wsdl:portType name="ZFLMHEARTBEAT">
- <sap:useFeature feature="tns:design_0" />
- </wsdl:portType>
- <wsdl:operation name="ZFImHeartbeat">

```

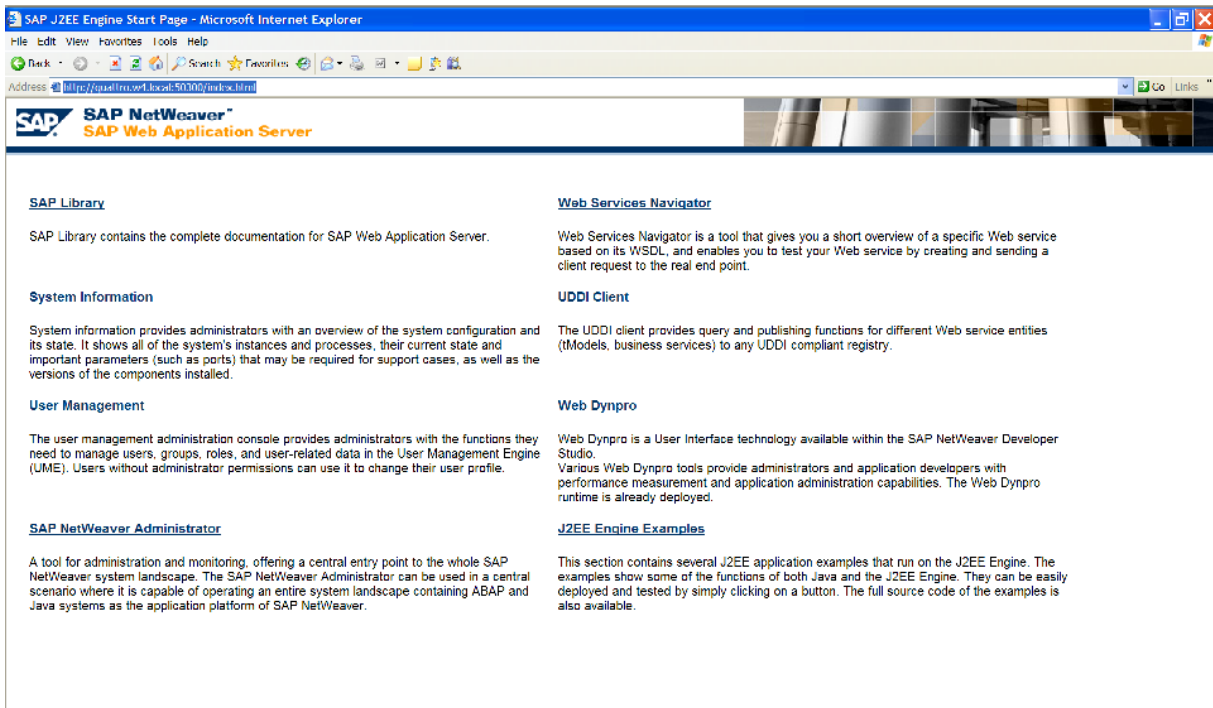
Select the WSDL URL and Launch a new Browser Session and Navigate to the your Main Java Admin Page

In this case our WSDL is

http://quattro.w4.local:8003/sap/bc/srt/rfc/sap/ZFLMHEARTBEAT?sap-client=800&wsdl=1.1&mode=sap\_wsdl

And our Main Java Admin Page url is

http://quattro.w4.local:50300/index.html



The screenshot shows the SAP NetWeaver SAP Web Application Server start page. The page has a blue header with the SAP logo and the text "SAP NetWeaver SAP Web Application Server". Below the header, there are several sections with links and descriptions:

- SAP Library**: SAP Library contains the complete documentation for SAP Web Application Server.
- System Information**: System information provides administrators with an overview of the system configuration and its state. It shows all of the system's instances and processes, their current state and important parameters (such as ports) that may be required for support cases, as well as the versions of the components installed.
- User Management**: The user management administration console provides administrators with the functions they need to manage users, groups, roles, and user-related data in the User Management Engine (UME). Users without administrator permissions can use it to change their user profile.
- SAP NetWeaver Administrator**: A tool for administration and monitoring, offering a central entry point to the whole SAP NetWeaver system landscape. The SAP NetWeaver Administrator can be used in a central scenario where it is capable of operating an entire system landscape containing ABAP and Java systems as the application platform of SAP NetWeaver.
- Web Services Navigator**: Web Services Navigator is a tool that gives you a short overview of a specific Web service based on its WSDL, and enables you to test your Web service by creating and sending a client request to the real end point.
- UDDI Client**: The UDDI client provides query and publishing functions for different Web service entities (tModels, business services) to any UDDI compliant registry.
- Web Dynpro**: Web Dynpro is a User Interface technology available within the SAP NetWeaver Developer Studio. Various Web Dynpro tools provide administrators and application developers with performance measurement and application administration capabilities. The Web Dynpro runtime is already deployed.
- J2EE Engine Examples**: This section contains several J2EE application examples that run on the J2EE Engine. The examples show some of the functions of both Java and the J2EE Engine. They can be easily deployed and tested by simply clicking on a button. The full source code of the examples is also available.

Select the Web Service Navigator and Paste the Web Service URL and click Next  
*NB You should not be prompted for logon details when testing the Web Service*

## Web Services Navigator

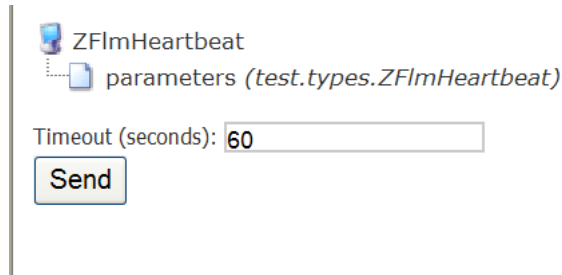
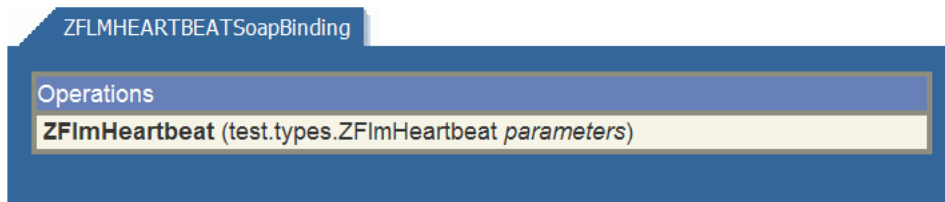
### Welcome to the Web Services Navigator

Enter the WSDL URL of the Web service:

Once the page is loaded, click the Test Link and then Select the Operation ZFImHeartbeat

## Web Services Navigator

### Test



Now click Send and you should be shown the screen

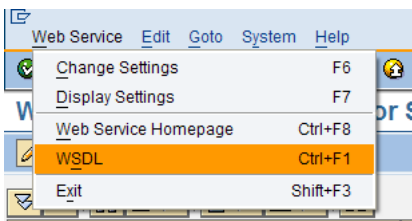
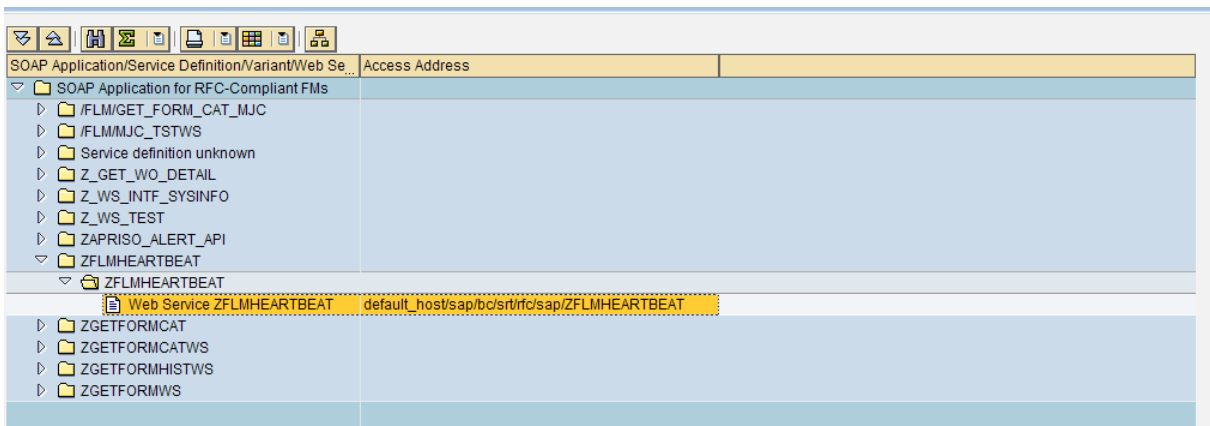




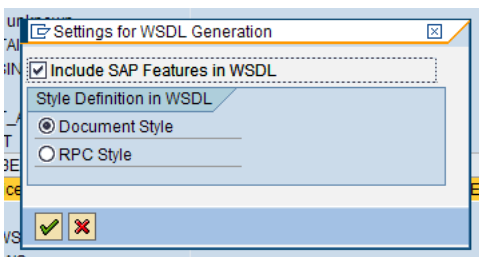
Note that the Response has a String Element called Heartbeat and that this has the current Date and Time displayed. If you see this then the Web Service is configured and running properly.

## 1.6 Savine the Web Service Definition

Using WSADMIN Expand the Selection until you have the Web Service for ZFLMHEARTBEAT Selected

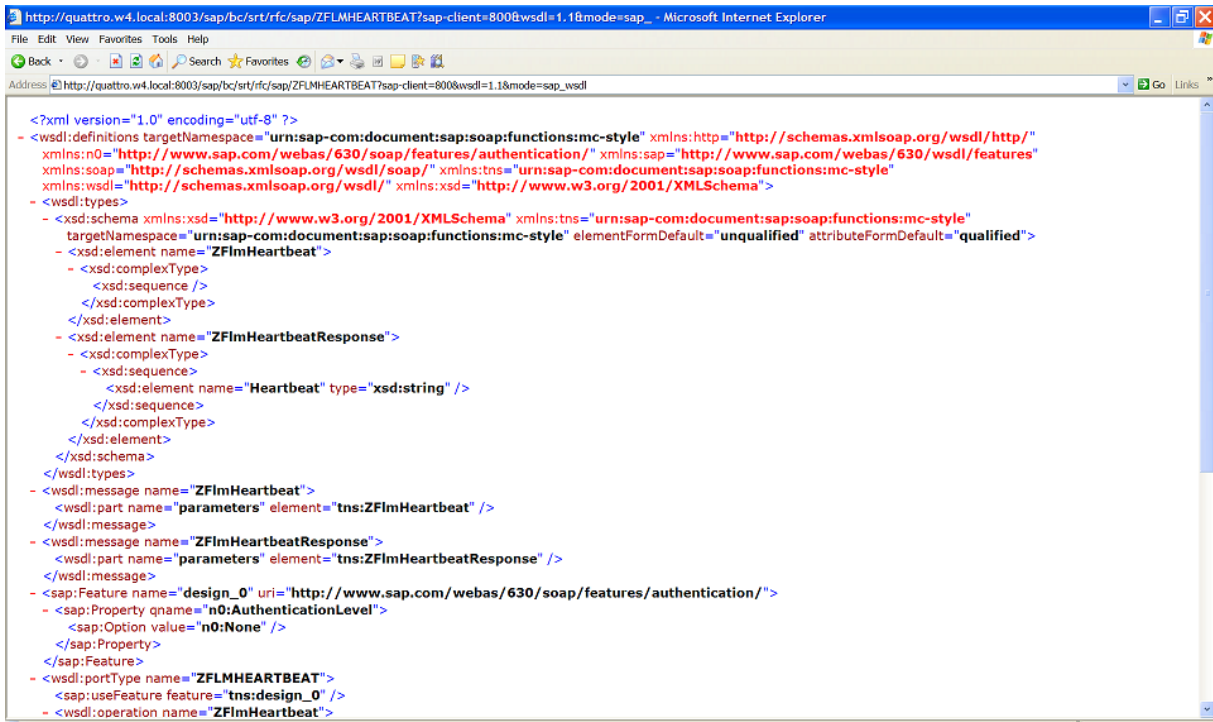


Now from the menu Select Web Service>WSDL

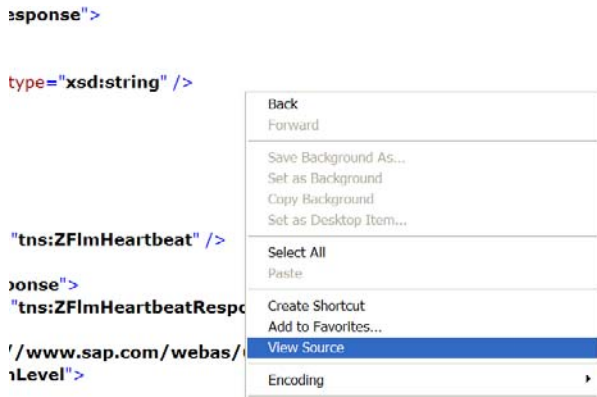


Select the Style Defintion to be **Document Style** and also enure that **Include SAP Features in WSDL** is selected, and click Continue

This will launch a new IE Browser session and will show the WSDL File



We now need to save the WSDL definition to a File, so right click on the page and Select View Source



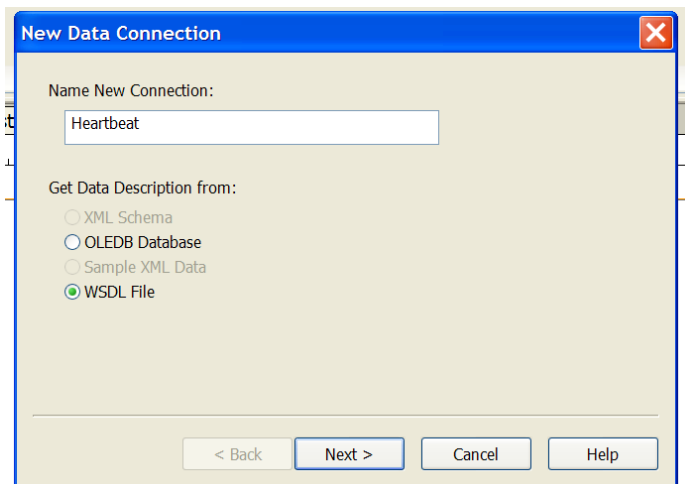
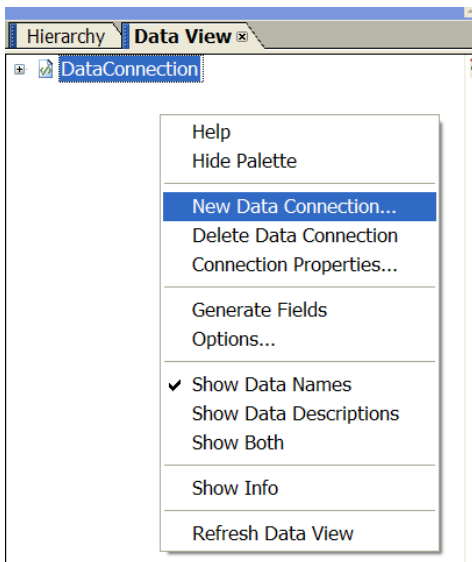
When the source is shown save this in C:\FLM\Form Templates as *ZFLMHEARTBEAT.wsdl* (Make sure that the extension is wsdl)



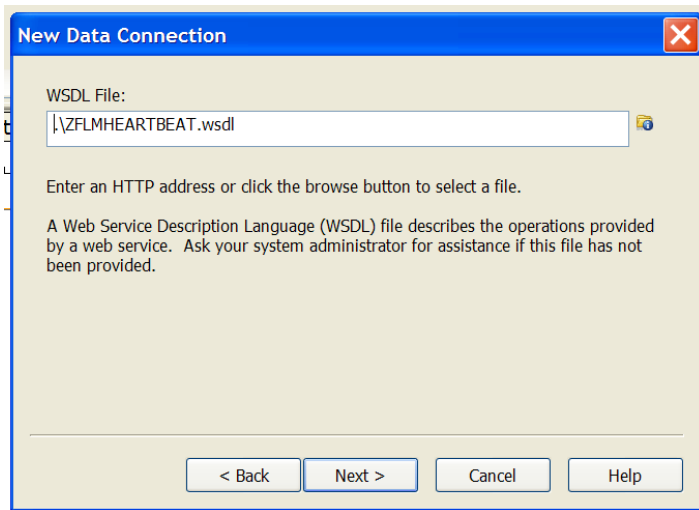
## 1.7 Adding a Web Service to the Form

Open the XDP in Adobe Form Designer and Select the **Data View** Tab.

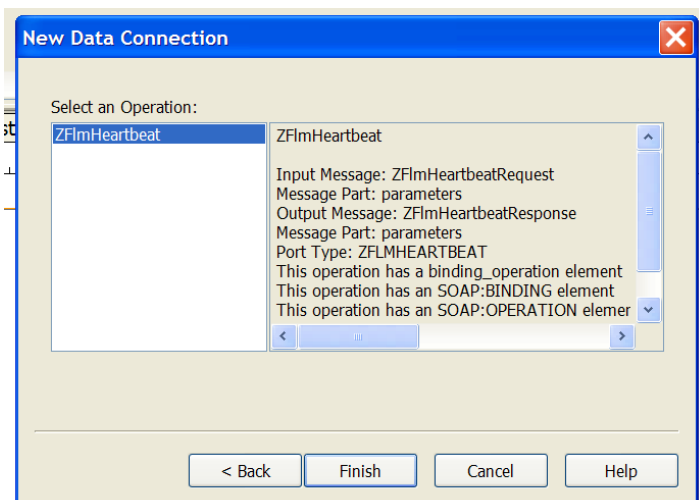
Now Right click in the Data View Tab and Select **New Data Connection...**



Name the Data Connection **Heartbeat** and Select the **WSDL File** Option, and click **Next**

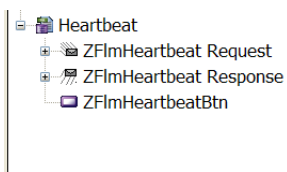


Now Select the **File Browser** Icon and Navigate to your locally saved WSDL File, click **Next**

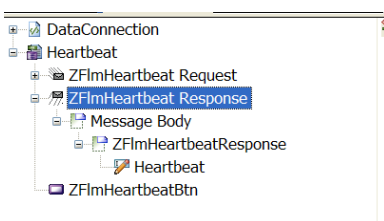


Select the Operation **ZflmHeartbeat** and click **Finish**

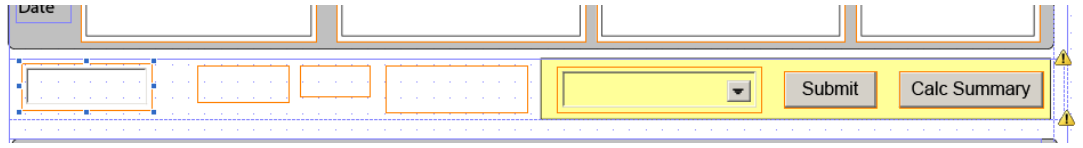
You should now see the below Data Connection in the Data View Tab



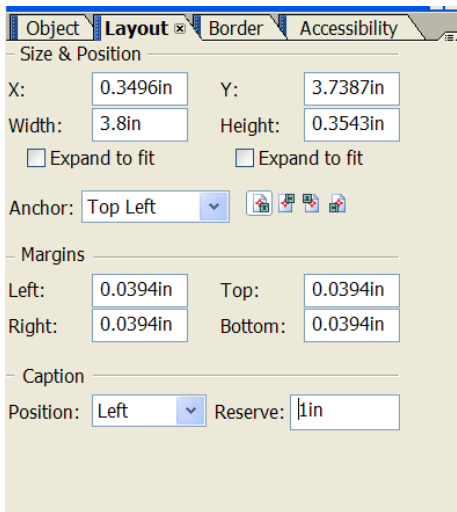
Expand the Response Tree as shown



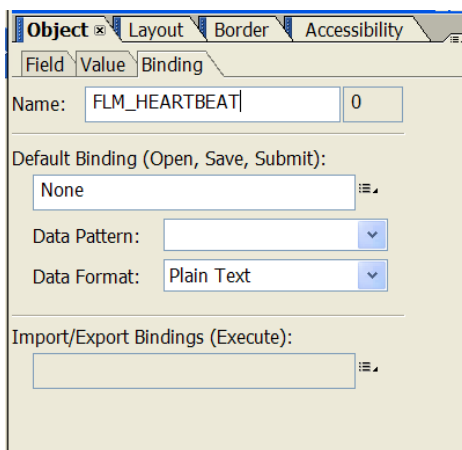
Now Add a Field onto the Form, and make sure it goes into the same subform (SF\_FLM) as the FLM\_SUBMIT Button.

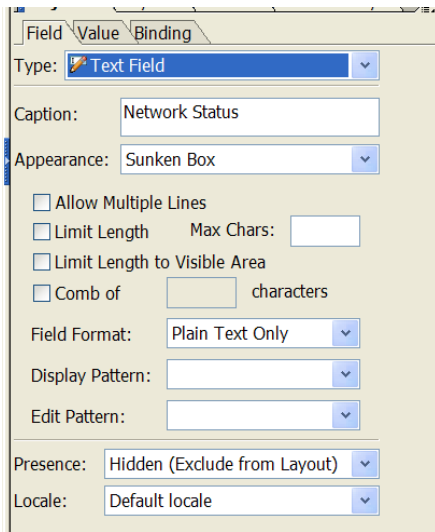


Using the Layout tab sets its Caption to None and its Width to 3.8 Inch, and the Caption position Left Reserving 1 Inch

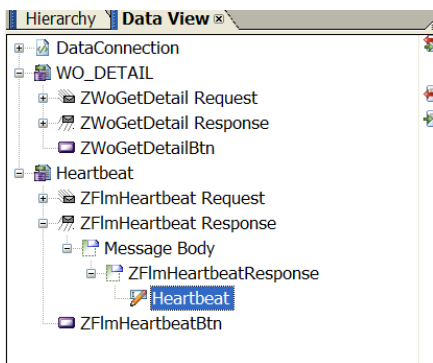


Using the Object tab set its name to FLM\_HEARTBEAT Caption to Network Status and and Object Visibility to Hidden

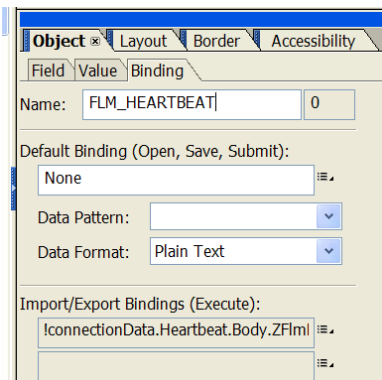
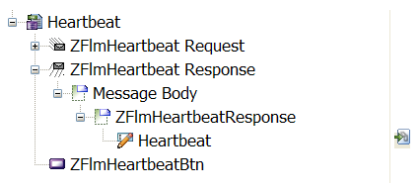




Now using the Data View Tab Select the Data Element Heartbeat (The one with a pencil icon) and drag this onto the form and drop onto your New Field FLM\_HEARTBEAT



You should now see the Data View update like below and the Bindings Tab on the field FLM\_HEARTBEAT showing a new Import/Export Binding



Now Select the field FLM\_SUBMIT, this will be in the same subform (SF\_FLM) and change the code in the click event to the below

```
//Make sure we reset the Web Service field before we can it
FLM_HEARTBEAT.presence = "visible";
FLM_HEARTBEAT.rawValue = "NO NETWORK";

//Now call the Web Service
xfa.connectionSet.Heartbeat.execute(false);

if (FLM_HEARTBEAT.rawValue == "NO NETWORK")
{
    xfa.host.messageBox("Unable to check the network, please save a copy of the PDF for you
records.", "Amey Work Order", 0, 0);
}
else
{
    FLM_ACTION.rawValue = "";

    soUtils.lockDownForm(xfa.resolveNode("DATA.Amey_Form"), true);

    if (FTRANSPORT.value == "1")
    {
        submitToSAP1.execEvent("click");
    }
    else
    {
        event.target.submitForm({cURL: "mailto:" + RET_EMAIL.value,
bEmpty: true, // Post all fields (true), or do Not post all
fields (false)
cSubmitAs: RET_FILE_TYPE.value});
    }
}
```

Now save the XDP and upload to FLM and test.