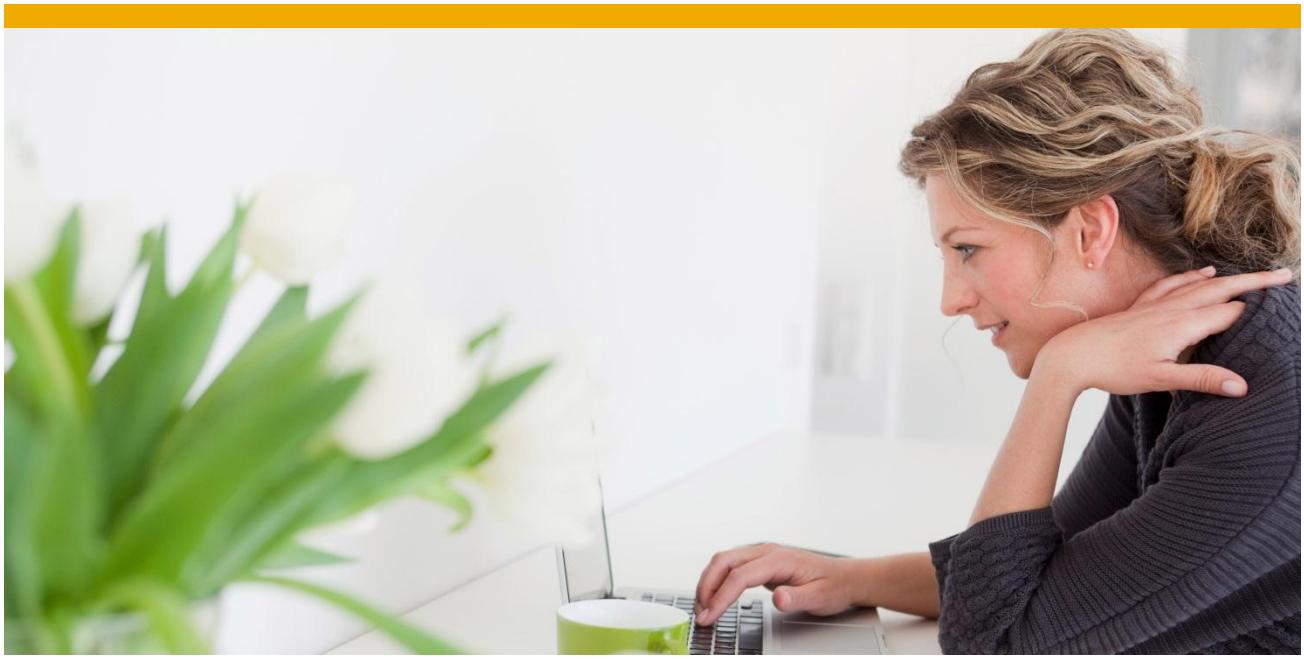


Async/Sync and Sync/Async Bridge Scenarios

Configuring Async/Sync Bridge and Sync/Async Bridge without BPM

TABLE OF CONTENTS

1 PREFACE

1.1 Constraints

The texts, references, and graphics contained in this manual have been compiled with utmost care; nevertheless, it is impossible to guarantee that they are fully without error. SAP cannot assume any responsibility for the correctness or completeness of the following documentation; the user alone is responsible for verifying the information contained therein.
SAP will only assume liability for damage arising from the use of this documentation – irrespective of the pertinent legal basis – in the case of intentional or active negligence, under no other circumstances will a warranty be made.

1.2 Definition

This manual describes simple application cases for the de-central adapter engine for process integration and all the configuration steps that are necessary to execute the application cases on the basis of SAP NetWeaver 7.31.

1.3 Intended Audience

This manual is intended to be used by both technology and application consultants.

1.4 Structure

The structure of this document follows the sequence of steps required to configure and run the use cases.
In this document, the scenarios are developed in swing client and Integrated Configuration Objects are created.The same scenarios can be developed using Integration iFlows (NWDS/eclipse/IntegrationTool)

2 INTRODUCTION AND PREREQUISITES

2.1 Introduction

The following use case variants are defined in this test:

Variant	Description
Variant 1: Async/Sync Bridge by means of module processor	This test case describes how to connect an asynchronous system to a synchronous system by means of an async/sync bridge without using BPM
Variant 2: Sync/Async Bridge by means of module processor	This test case describes how to connect a synchronous system to an asynchronous system by means of a sync/async bridge without using BPM

2.2 Prerequisites

The web service tool used in this test case is SOAP workbench (from Apache) for testing. Any external SOAP Request/Response tool which supports web service can be used for testing. Eg, SOAP UI. Refer http://www.soapui.org/Getting-Started/web-service-sample-project.html for steps to use this tool.

3 VARIANT 1: ASYN/SYNC BRIDGE BY MEANS OF MODULE PROCESSOR

3.1 Introduction

This test case describes how to connect an asynchronous system to a synchronous system by means of an async/sync bridge without using BPM. This is required if you have a sender system that only supports asynchronous message processing to communicate with a purely synchronous receiver application. The async/sync bridge handles the conversion from an asynchronous message sent from the sender to the receiver into a synchronous message, and the other way round for the synchronous reply.

3.2 Async/Sync Bridge

The asynchronous request and response messages are mapped to a synchronous call by means of the module processor. The overall communication sequence looks like below diagram: an asynchronous request message is converted to a synchronous request in the module processor. The synchronous system sends a response which is converted to an asynchronous response message in the module processor which is then passed to receiver.

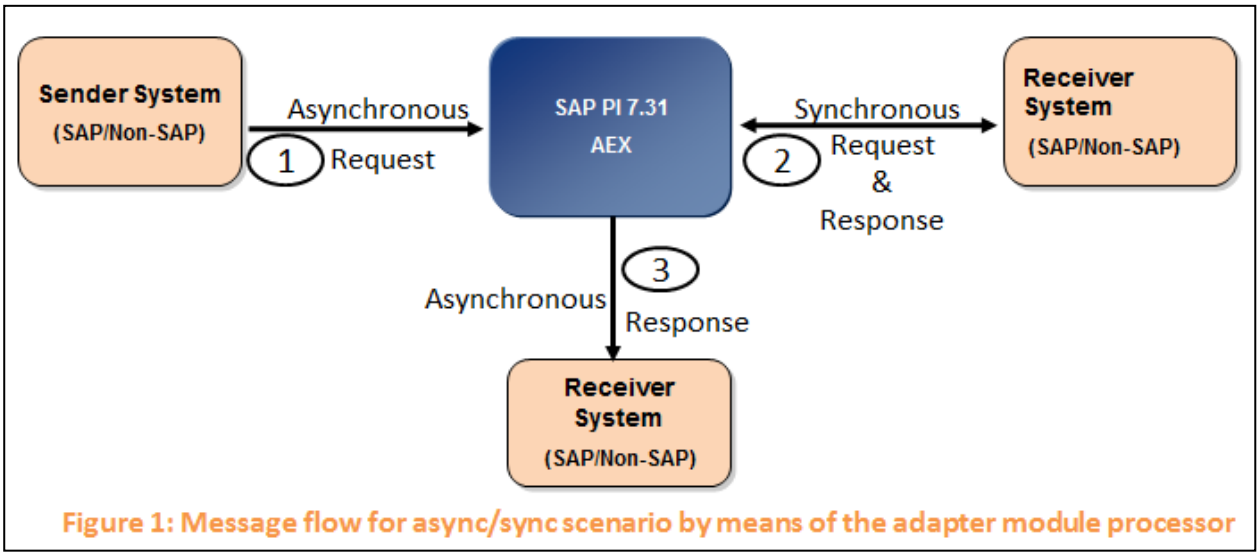


Figure 1: Message flow for async/sync scenario by means of the adapter module processor

3.3 Test scenario

The scenario in this document refers to send IDOC to RFC and response back from RFC to web service.

- PI receives the IDOC from ECC system. JIDOC Sender Adapter is used in PI to get the data from ECC.
- PI does the request mapping and sends the request to BAPI and gets back the response from BAPI via RFC Receiver adapter.
- The response from BAPI is sent to web service via SOAP adapter

4 PREPARATION

4.1 User Permissions

The tester should have permission to log on the PI test system and to open the SOA Monitors in the NetWeaver Administrator. Also the user should have access permissions to R/3 system to perform the testing.
Here PI 7.31 AEX, R/3 system, PI 7.31 Double Stack are used to execute this test. The system environment may differ in your test executions.

Note: The RFC program is created in dual stack server and the request and response is fetched by 7.31 AEX from dual stack system

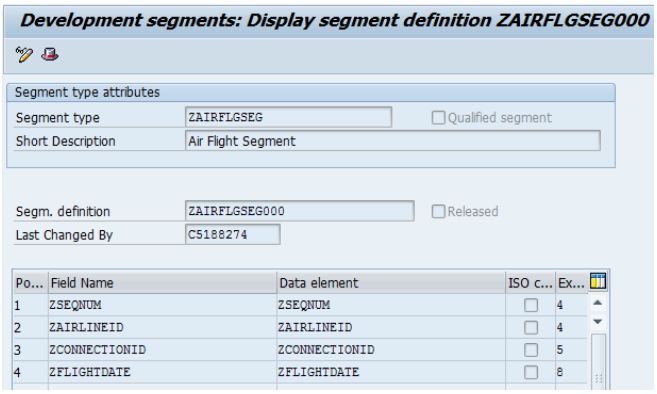
4.2 Preconditions

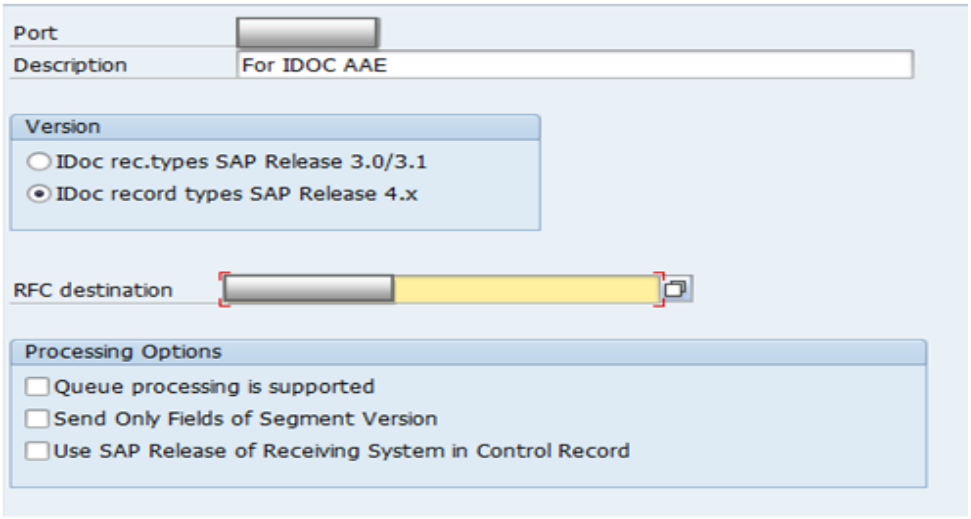
1. Configurations in NWA

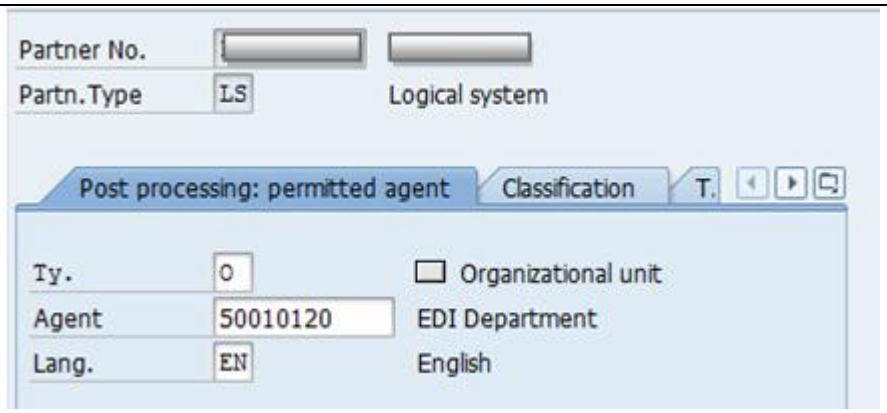
Actions	System	Details
Settings in Application Resources	PI 7.31	1. Open NWA (http://<host>:<port>/**nwa**) - Configuration **>** Infrastructure **> Application Resources** - Filter for Resource Adapter, **inboundRA** - Click on Properties tab below: • Set value for "MaxReaderThreadCount" between 5 to 10 • Set Value of ProgramID as JIDOC_[redacted]_PID_DEMO (This must be used when creating the RFC in ECC system) • Set Local as false (don't change) • Set values for default **GatewayServer** as host of PI SystemSet values for default **GatewayService** • Save Changes
Create JCO RFC Provider destination	PI 7.31	2. Open NWA (http://<host>:<port>/**nwa**) - Configuration **>** Infrastructure **> JCO RFC Provider** - Create JCO RFC Provider destination with name JIDOC_[redacted]_PID_DEMO in NWA where PIDEV refers to the PI System ID - Save Changes

2. Configurations at Sender System (SAP ECC)

   This section describes all the configurations needed in the Sender SAP System (ECC) for sending an IDoc to PI

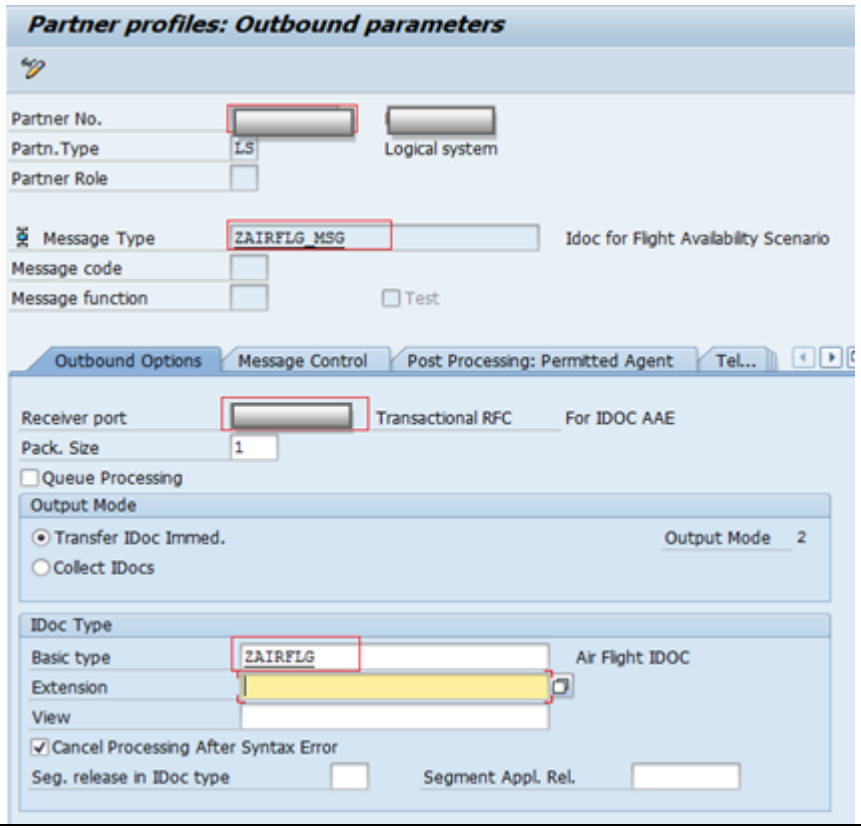
Actions	Details
IDOC Creation	The IDoc used in variant1 (JIDOC-RFC-SOAP) is a custom IDoc (ZAIRFLG) In SAP ECC system, • Goto transaction code(tcode) we31 to create segment (ZAIRFLGSEG) which will have the fields as below:  • Goto tcode we30 to create the IDOC (ZAIRFLG) Below link can be referred for creation of custom IDOC in SAP ECC system and the custom IDoc created should be imported in ESR http://wiki.scn.sap.com/wiki/display/ABAP/Steps+to+create+custom+IDOC

RFC Destination	On your SAP ECC system in order to send the IDOCs you need to create an RFC destination of type T (TCP/IP) Go to Transaction SM59 create a new RFC destination of Type T • Select the radio button Registered Server Program • In the program ID enter the program ID from inboundRA of NWA • Enter the gateway host and gateway service of your PI server After you configure that you should be able to check the configuration using the test connection button on the RFC destination.
Port	Goto Transaction WE21 • Click on the Transactional RFC from Ports (left tree panel) • Click on Create Button • Give Port Name (SAP<SID>) here or select generate port name and click on continue. • Give the description of the port and select RFC destination of type T created above. Now click on save. 
Logical System	• Click on create Button • Give name for your Logical System (<SID>CLNT<ClntNo.>) • Now click on save
Partner Profile	Create Partner Profile with outbound parameter (WE20) to be send to the PI receiver system 1. Go to Transaction WE20 2. Select Partner Type LS 3. Click on Create Button 4. Give Logical System Name which we have just created above as Partner No, Partner Type should be LS, Agent (some valid data), Language (EN). Now click on Save



5. Now create an Outbound Parameter.

- Select the required Message Type.
- Select the Receiver Port (which we have created in step 2 in this example).
- Select Transfer IDoc Immediately option in Output mode for Immediate Testing. Select Basic Type Save.
- Enter all credentials in the post processing permitted agent tab.

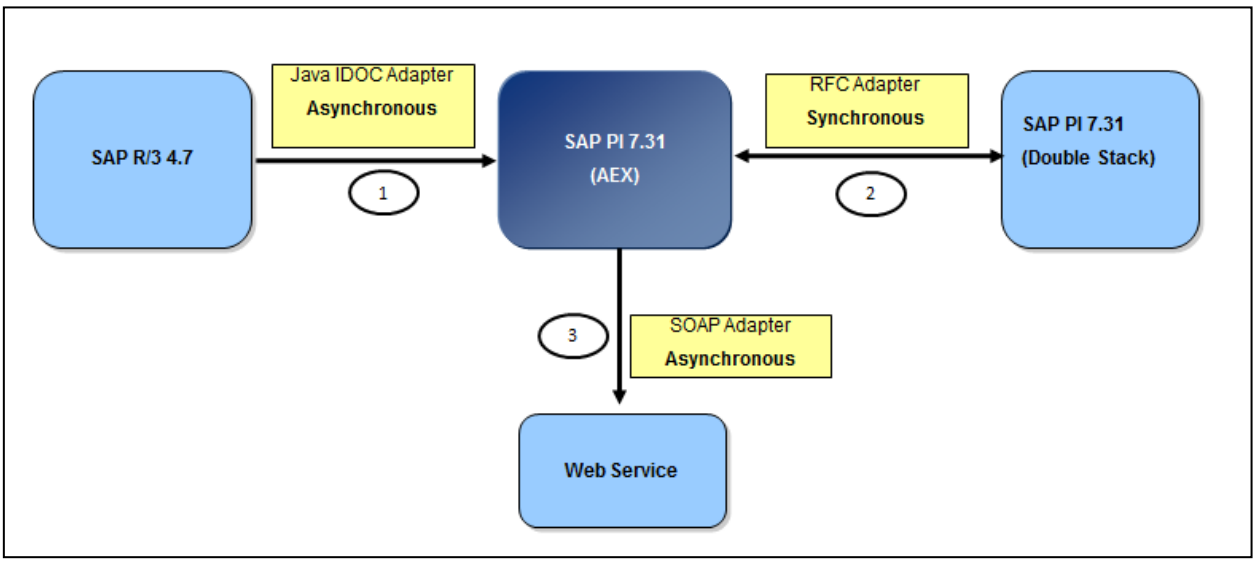


3. The tester should fulfill the ESR Objects and the Configuration Objects as shown in the subsequent sections of this document.

5 JIDOC-RFC-SOAP

5.1 Overview

This test case describes how to connect an asynchronous system to a synchronous system by means of an async/sync bridge.



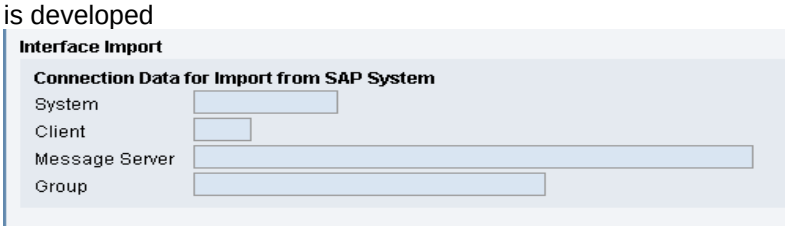
5.2 Description

SAP R/3 system sends the IDOC using Java IDOC Sender Adapter. PI does the request mapping and sends the request to BAPI and gets back the response of BAPI via RFC adapter. The response received is sent to web service via SOAP Adapter.
To implement the scenario, we need to define two ICO. One for routing the request message from the IDOC to the RFC, and one for routing back the response message to web service.

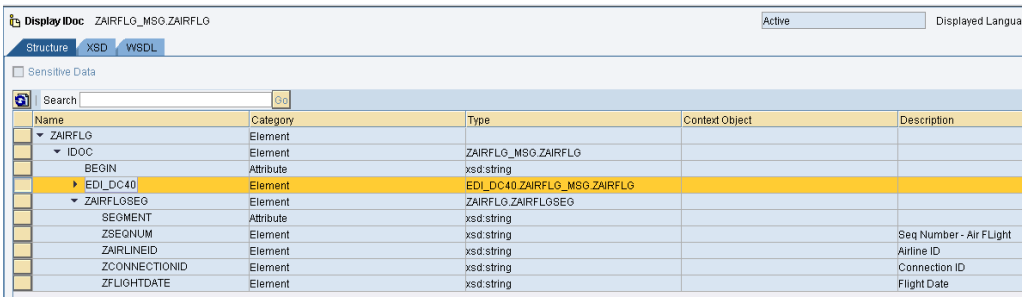
5.3 Repository Objects

In the Enterprise Service Repository – on PI test system:

- We should have the custom Idoc created as in section 4.2-2 and RFC *"SXIDEMO_AIRL_FLIGHT_CHECKAVAIL"* under Imported Objects folder. In this scenario the custom Idoc created and imported is *"ZAIRFLG_MSG.ZAIRFLG"*
  Steps to import Idoc:
  1. Double click and open the software Component version and provide the details in "*Interface import".* The Client and Message Server is of the ECC system in which IDoc is developed

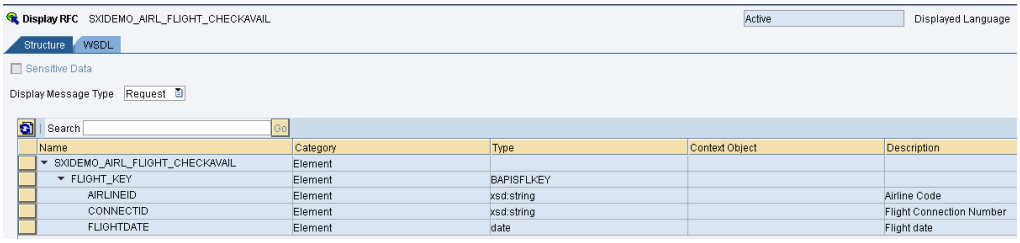


  2. Right click on *"Imported Objects-IDocs"* and click on *"Import of SAP Object"*
  3. Provide the *Application Server* and *System Number* in which the custom Idoc is been developed and also provide *UserName* and *password* and click Continue and then Finish button
  4. Once the Idoc is imported, the structure (fields) of IDOC is as below:

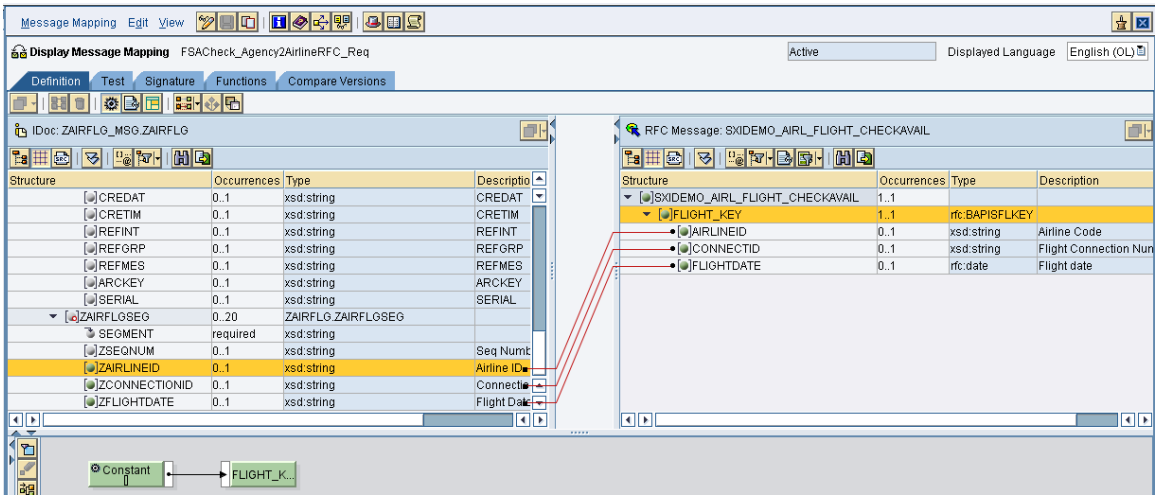


Steps to import RFC:

1. Follow the Step 1 as defined in import idoc above.
2. Right click on *"Imported Objects-RFC"* and click on *"Import of SAP Object"*
3. Provide the *Application Server* and *System Number* in which the RFC is been developed and also provide the *UserName* and *password* and click Continue and then Finish button

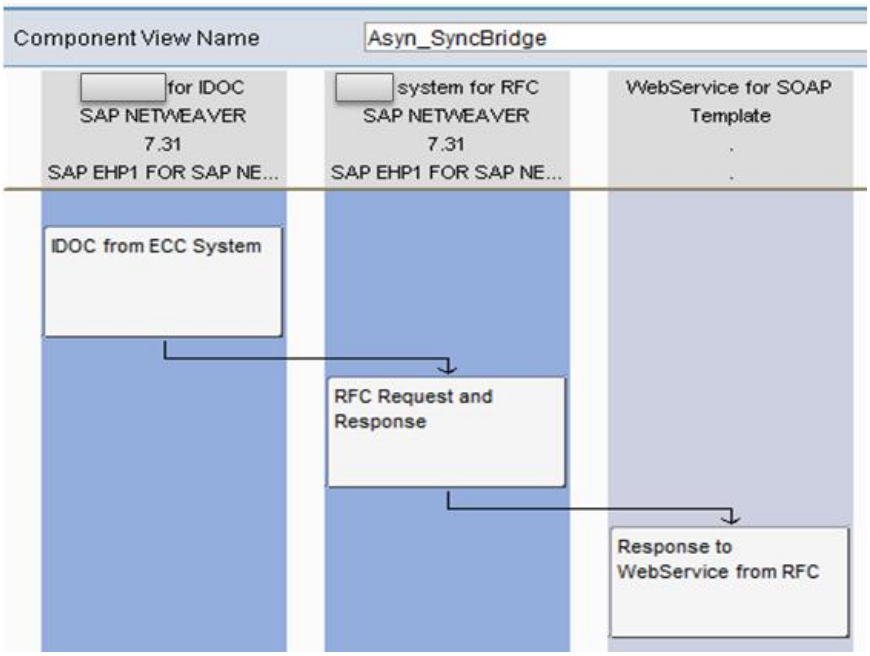


- The Message Mapping *FSACheck_Agency2AirlineRFC_Req*(as below) and Operation Mapping *FSACheck_Agency2Airline* should be created



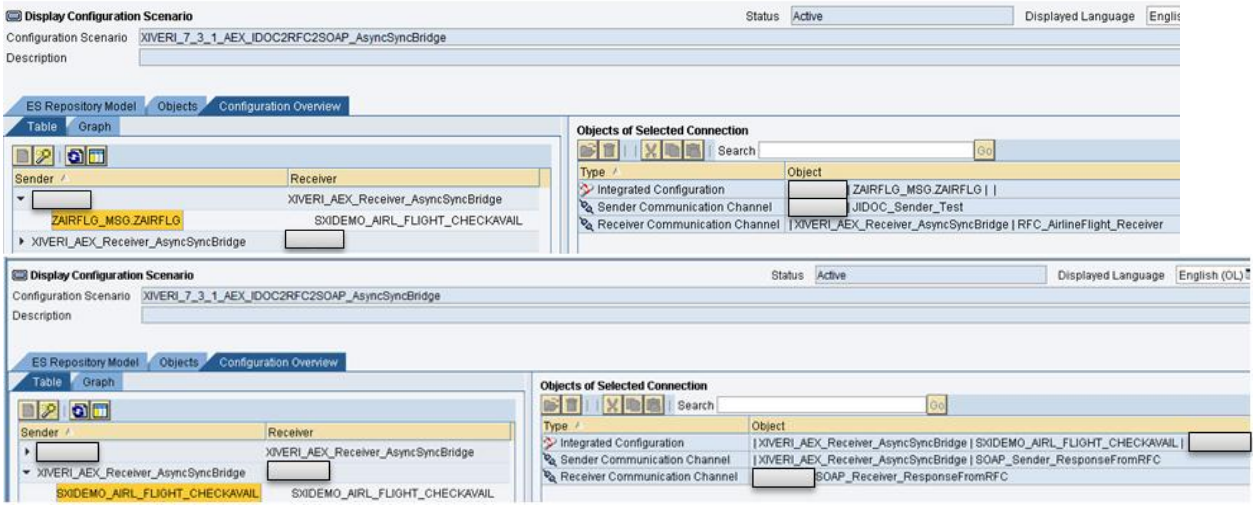
To do this – call the URL http://<server>:<host>/dir of the Process Integration system and open the link to the Enterprise Services Builder.

The following graphic shows the component view



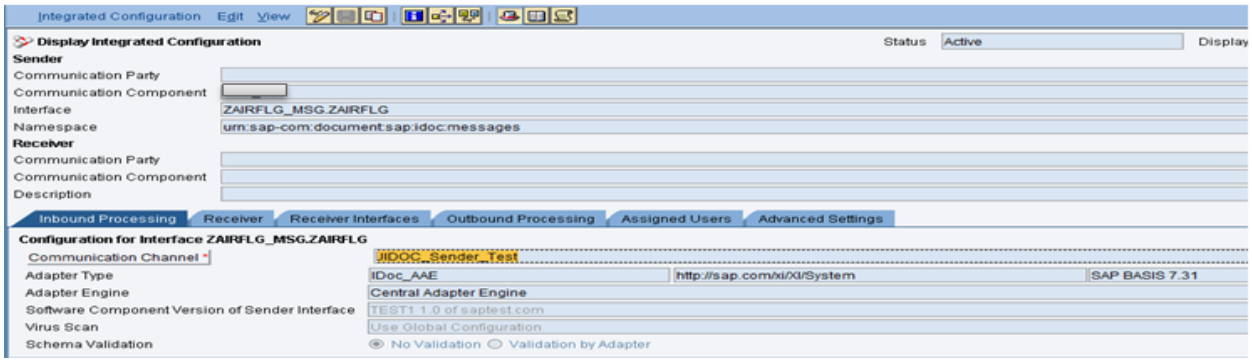
5.4 Configuration Objects

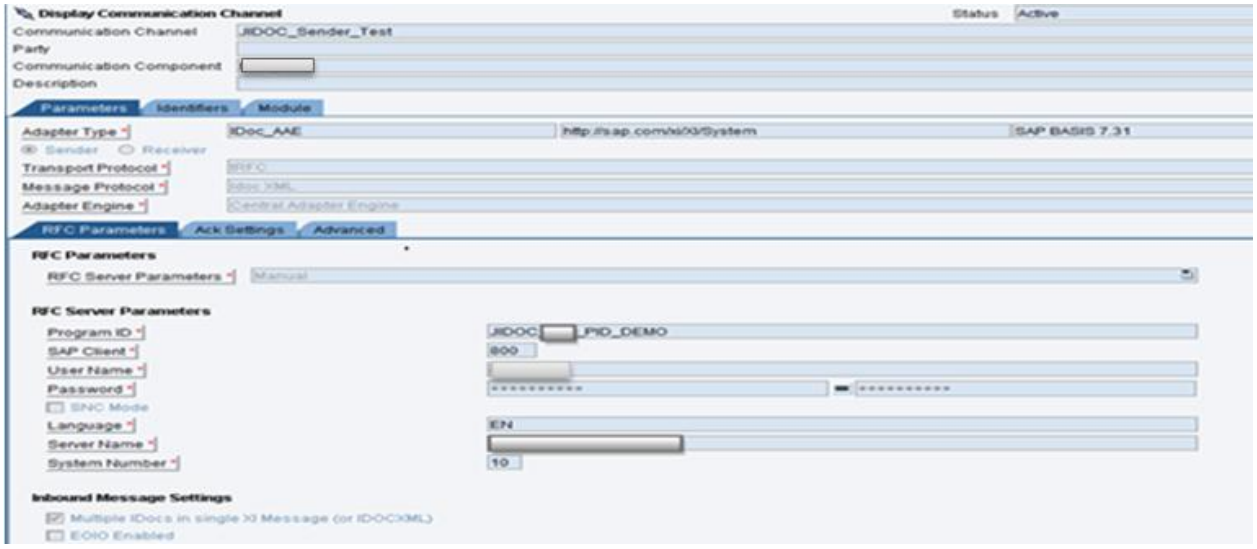
5.4.1 Configuration Overview



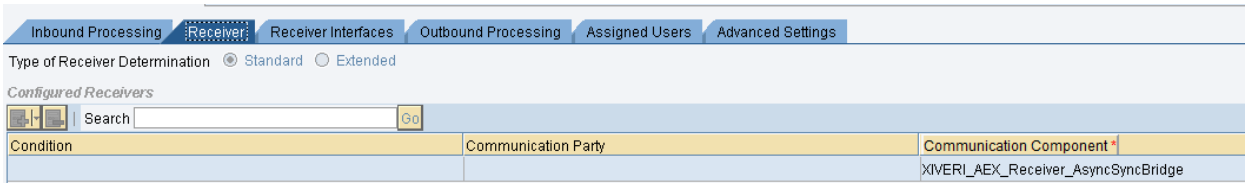
5.4.2 Integrated Configuration Objects in Integration Directory

The **first ICO** defines the routing from the JIDOC to the RFC. In the Inbound Processing, the JIDOC sender channel *(JIDOC_Sender_Test)* is specified.

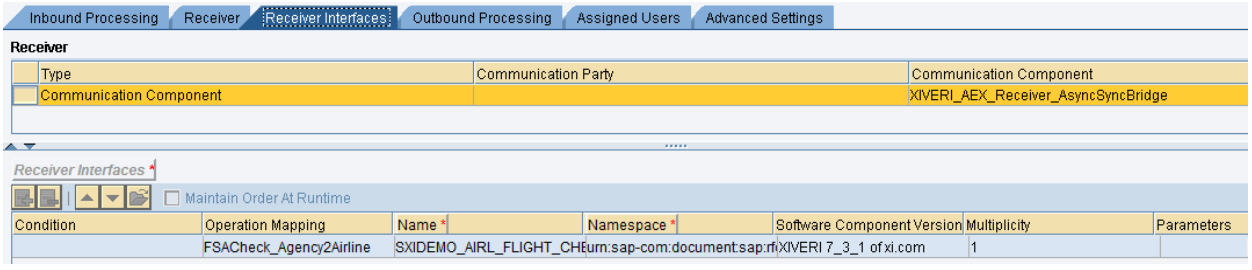




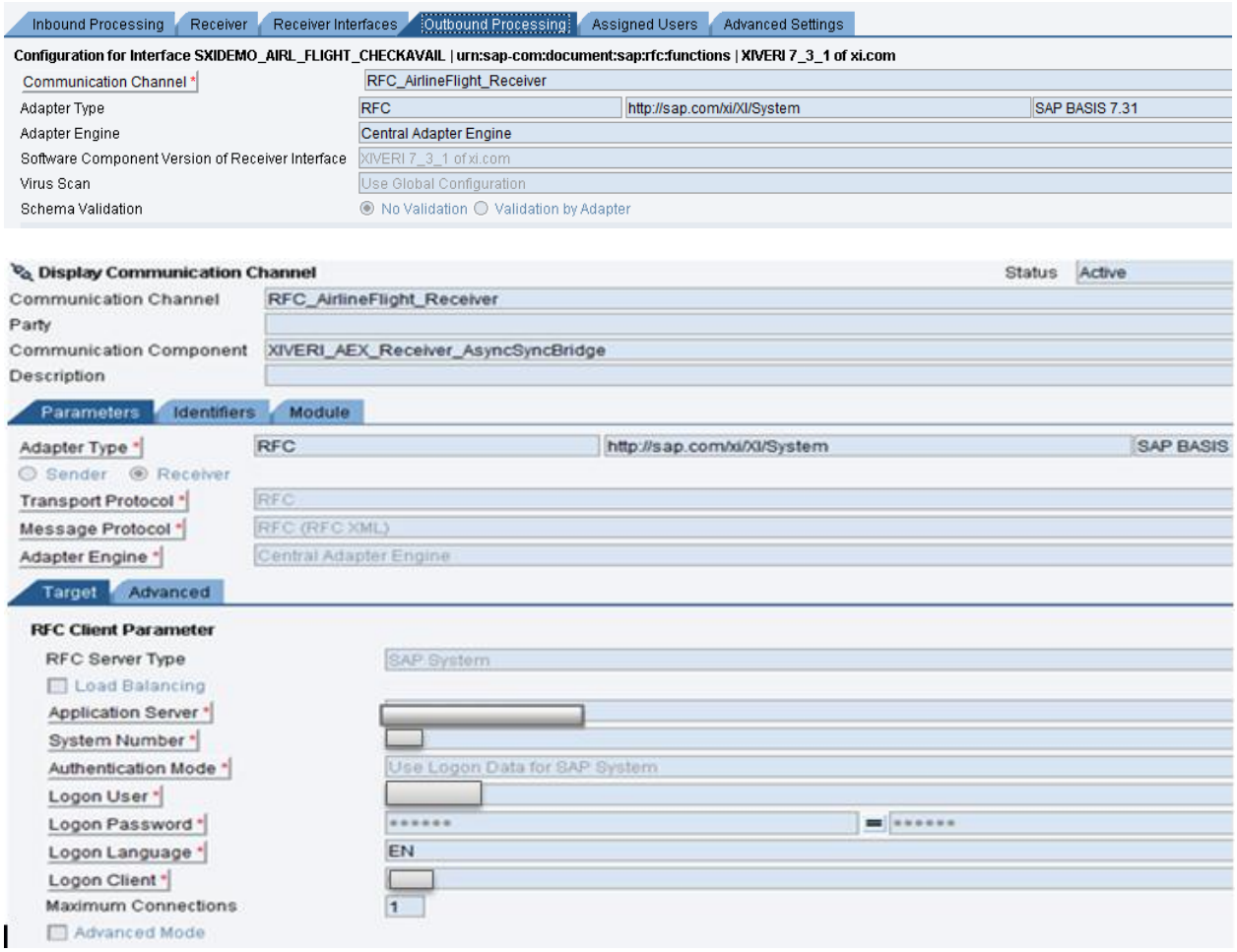
On tab *Receiver* of the ICO, the RFC Receiver Business Component is defined



On tab *Receiver Interfaces*, the Operation Mapping for request mapping is set

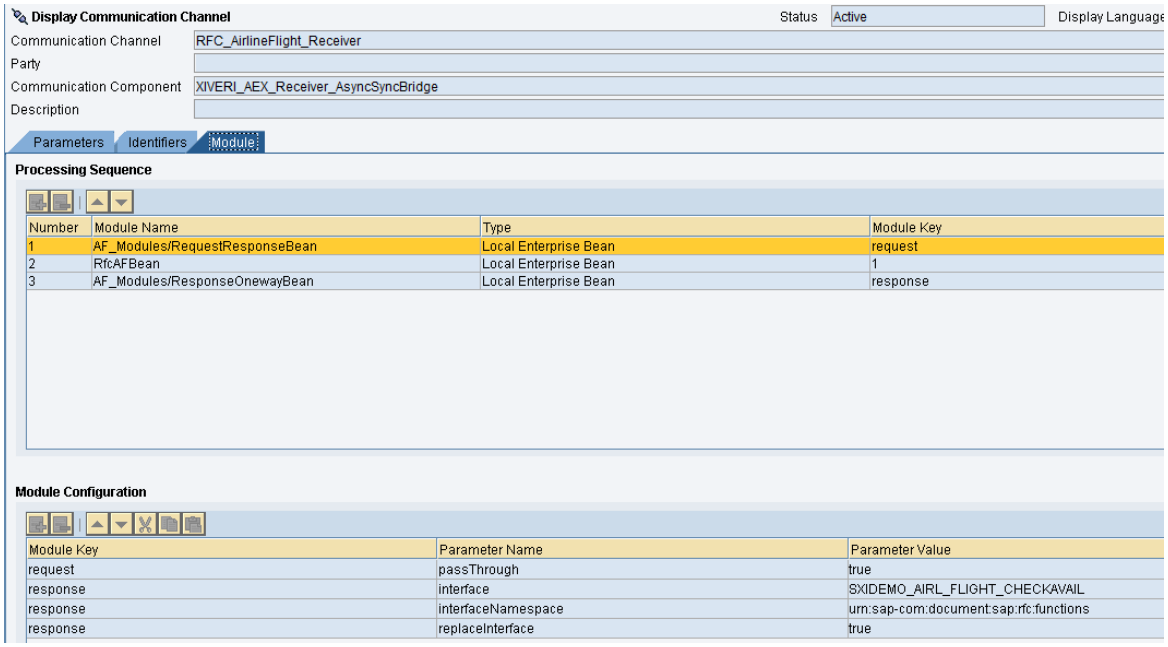


On tab *Outbound Processing*, the RFC receiver channel is specified.



In the RFC receiver channel, the modules AF_Modules/RequestResponseBean and AF_Modules/ResponseOnewayBean are defined in the right sequence in **Module** tab.

The specified parameters points to the second ICO.



RequestResponseBean

Use
RequestResponseBean module is used to convert an asynchronous request message to a synchronous request message.
If the thread that enters the module is part of a transaction, the transaction is suspended.

Entries in Module Configuration

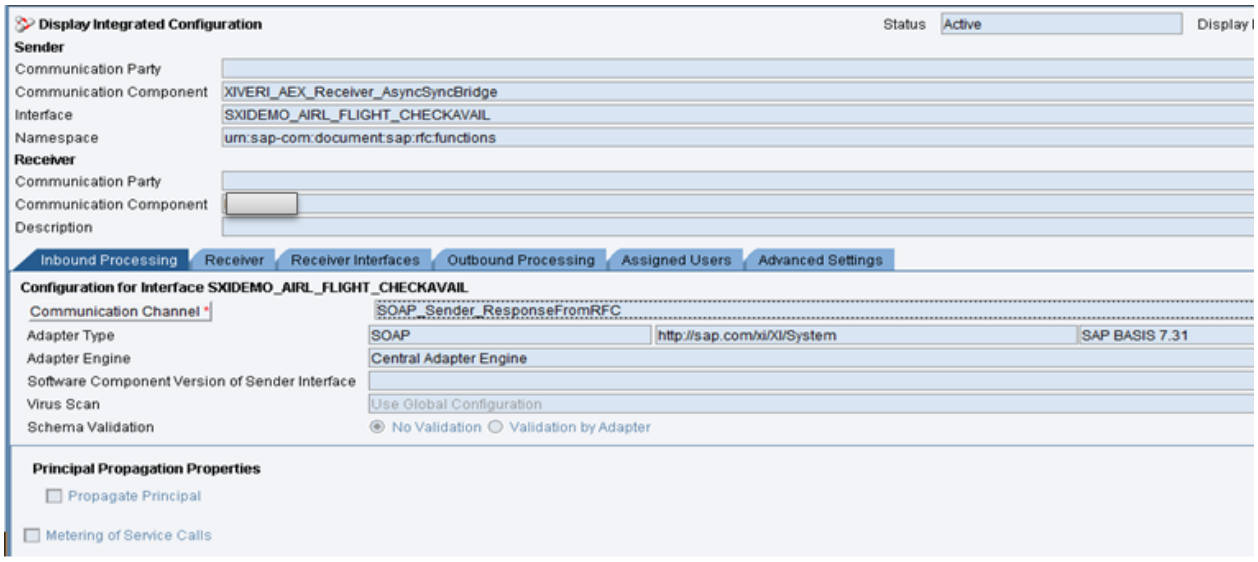
Parameter	Sender Adapter	Receiver Adapter
passThrough	If you want the module to forward the message to the next module in the module chain, enter **true**. The default value is **false**. The module then calls the messaging system.	If you want the module to forward the message to the next module in the module chain, enter **true**. The default value is **false**. The module then calls the **ResponseOnewayBean** module

ResponseOnewayBean

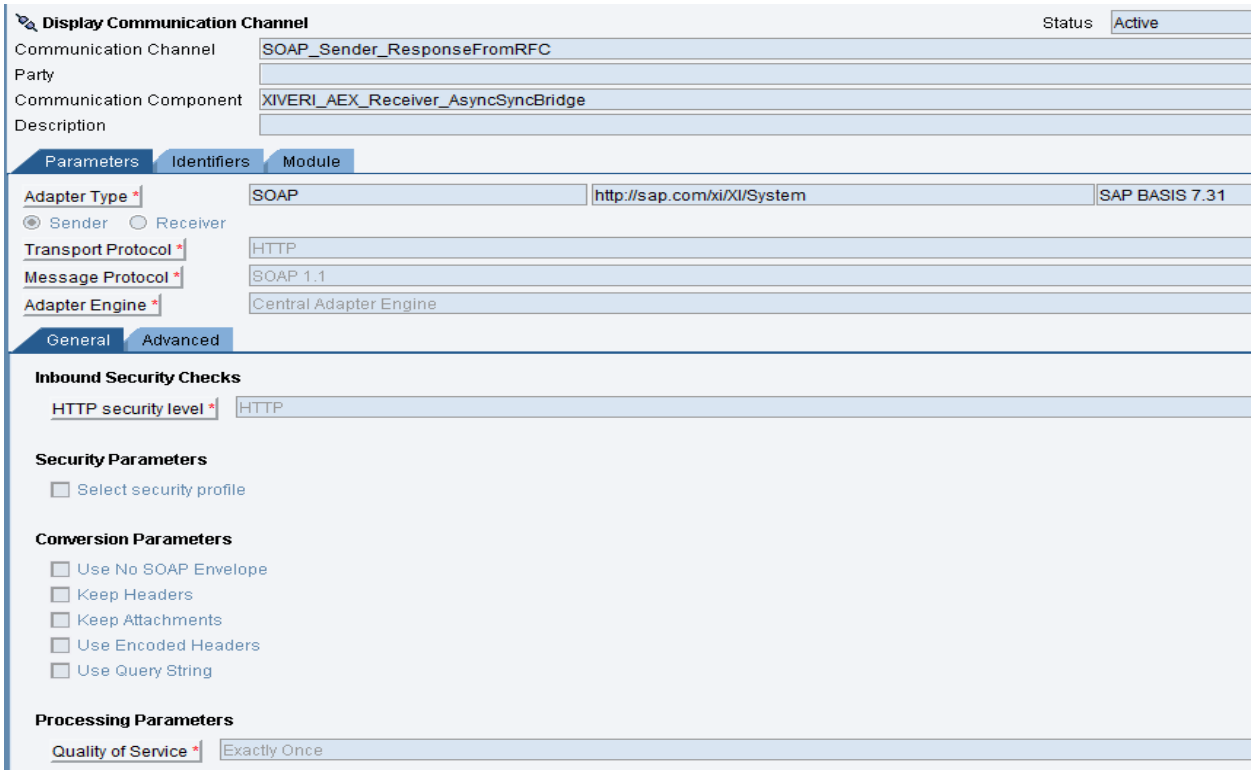
Use
A standard module used for converting an inbound message to an asynchronous message. If the thread that enters the module is part of a suspended transaction, the transaction is resumed.

Parameters (**interface, interfaceNamespace** and **replaceInterface**) mentioned in module Configuration points to the second ICO.

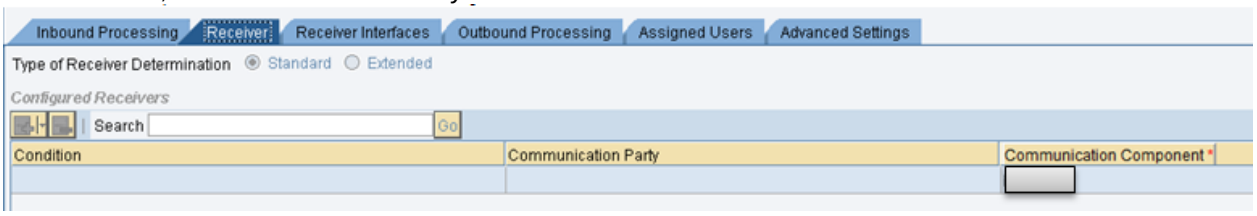
The **second ICO** defines the routing from the RFC to the Web Service. In the *Inbound Processing*, the SOAP sender channel is defined. Note, that in the header of the ICO the corresponding receiver communication component has to be specified as virtual receiver.
(The receiver of the first ICO become Sender and the Sender has to be defined as Virtual Receiver)



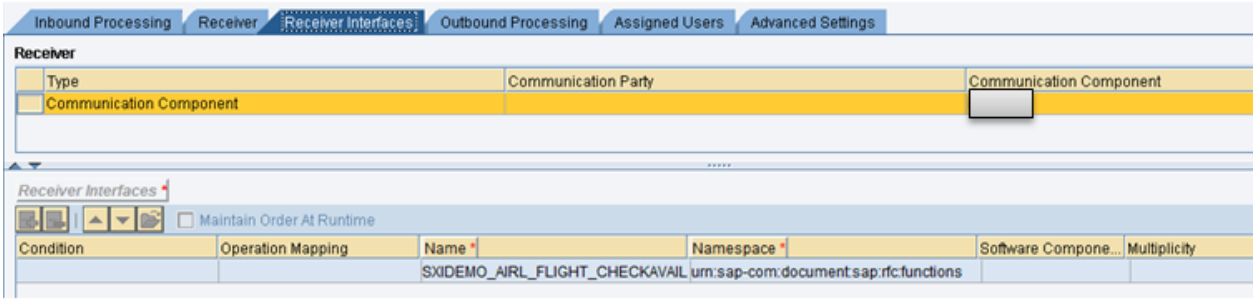
In the SOAP sender channel, the *Quality of Service* is set as *Exactly Once*.



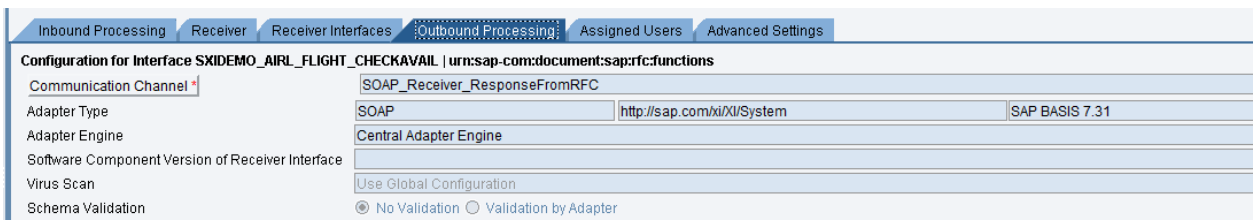
On tab *Receiver*, the receiver Business system is defined.

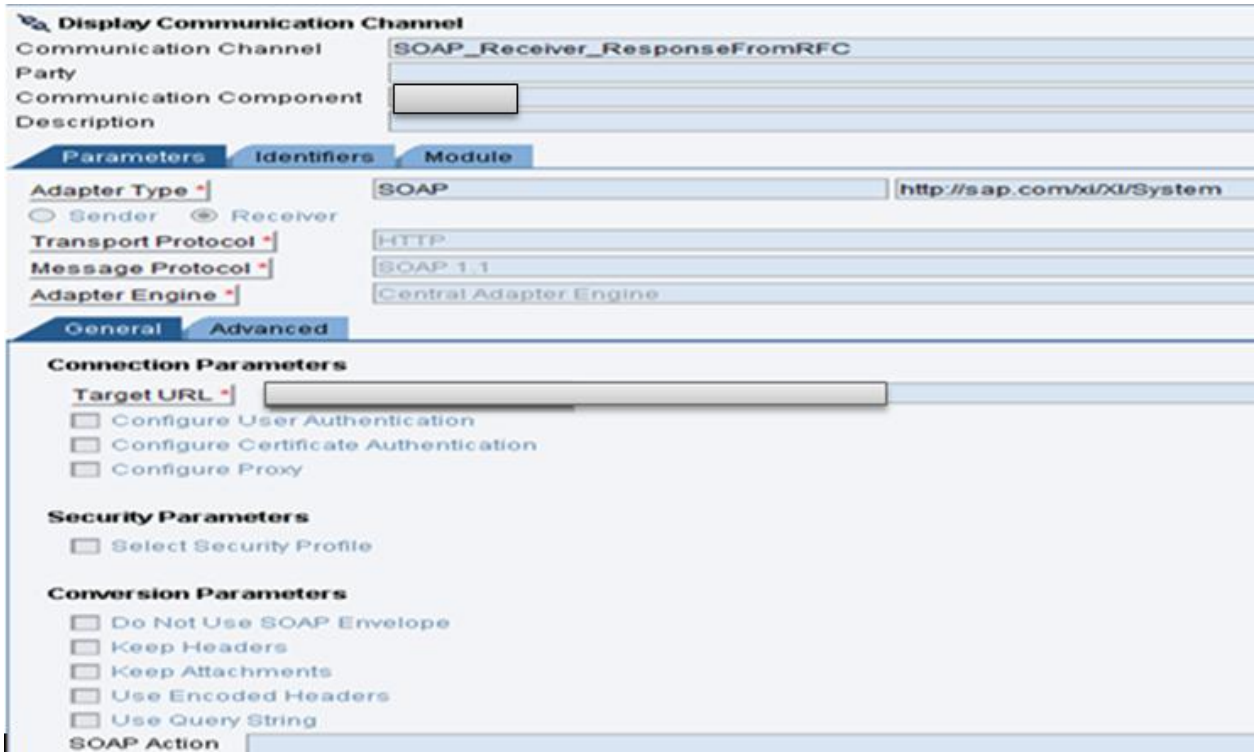


On tab *Receiver Interfaces*, the asynchronous response inbound interface is defined as receiver interfac



On tab *Outbound Processing*, the SOAP receiver Channel is set.



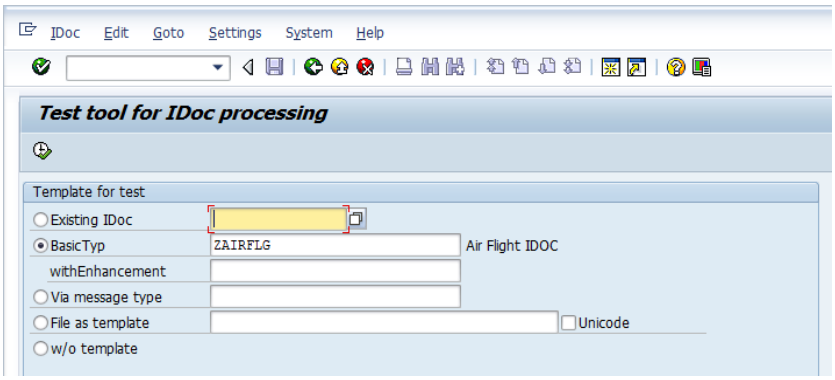


5.4.3 Test Description

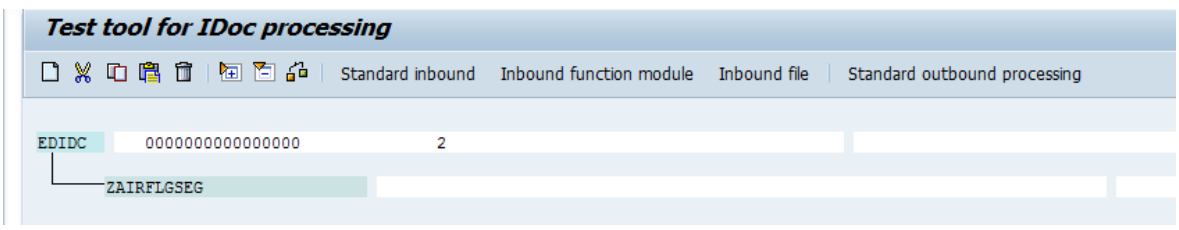
Testing the scenario JIDOC → PI → RFC → PI →SOAP please follow the steps below:

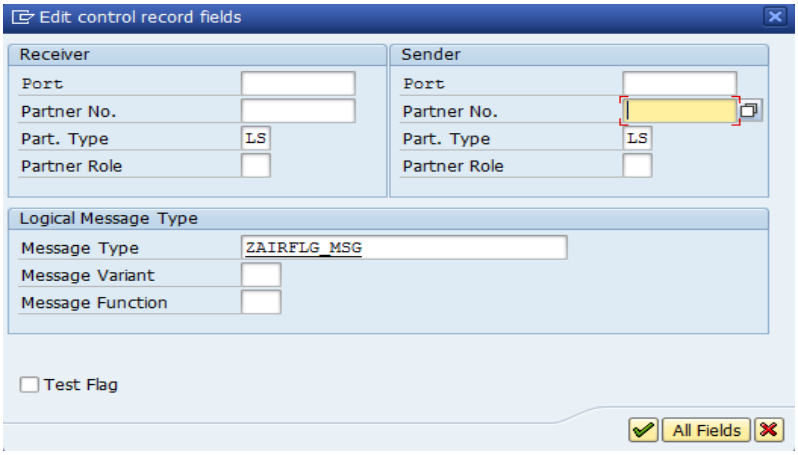
a) Check whether both (Sender & Receiver) channels being used in this scenario are started. If not then start them via Channel Monitoring of NWA/RWB.

b) Execute transaction “we19” in Sender R/3 system for triggering an IDoc. Enter IDoc Types as “ZAIRFLG”.

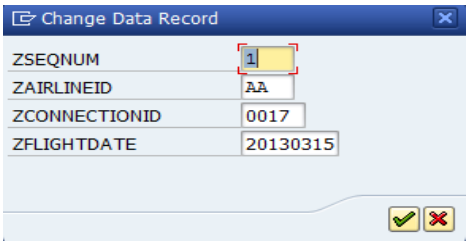


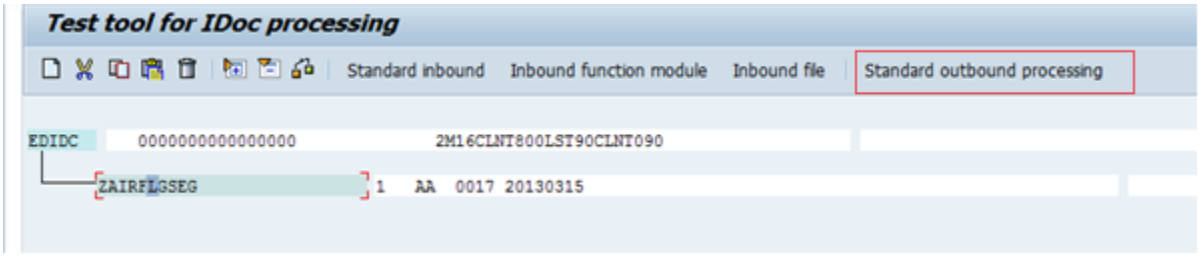
c) Double click on EDIDC(Control Record) and enter the Sender, Receiver details and Message Type ZAIRFLG_MSG





d) Double click on IDOC type and enter the record data and click on *Standard Outbound Processing*





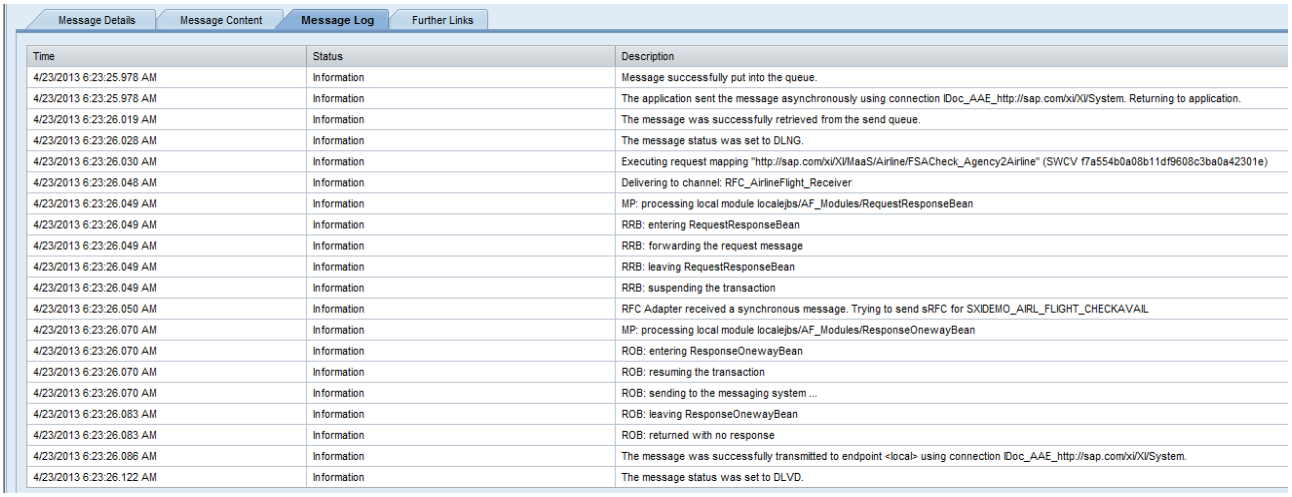
e) The IDoc is immediately sent to XI system. A message confirming the same appears at the bottom of the page. “An Idoc < IdocNo> added”.

f) The same can be confirmed by using transaction “we05” or “we02” and in the Direction box, choose “Outbound”. The list shows all the IDoc that have left the R/3 system.

On Sender Side, Check for “Idoc No.” which should be same as above sent from R/3 system. Get the Msg. ID for this flow of Idoc

This Msg. ID should be checked within Message flowing on Receiver Comm. Channel RFC_AirlineFlight_Receiver

To check on the modules process, goto Message Log of RFC Channel

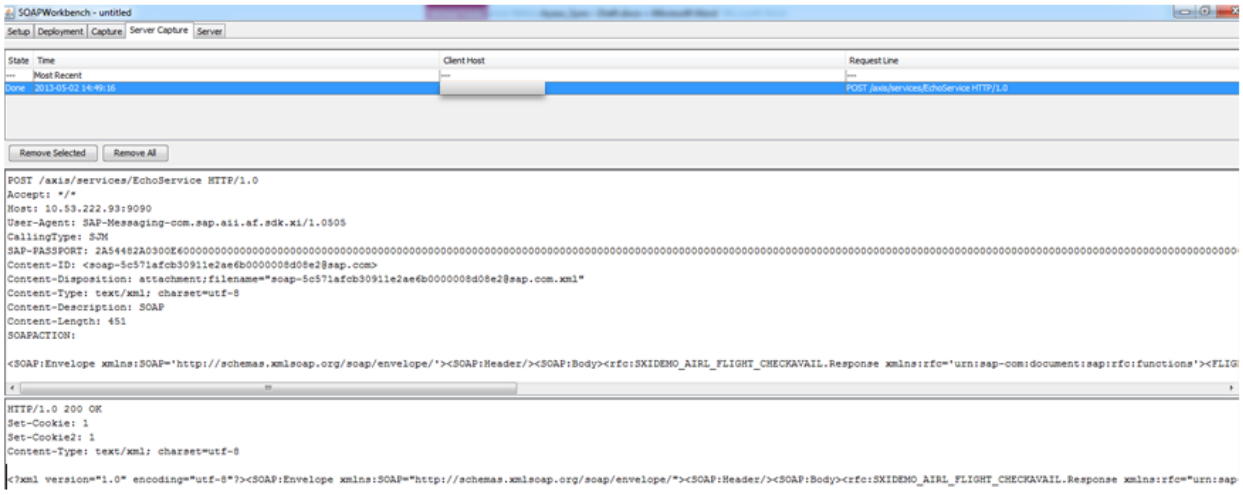


Message Details | Message Content | **Message Log** | Further Links

Time	Status	Description
4/23/2013 6:23:25.978 AM	Information	Message successfully put into the queue.
4/23/2013 6:23:25.978 AM	Information	The application sent the message asynchronously using connection IDoc_AAE_http://sap.com/xi/XI/System. Returning to application.
4/23/2013 6:23:26.019 AM	Information	The message was successfully retrieved from the send queue.
4/23/2013 6:23:26.028 AM	Information	The message status was set to DLNG.
4/23/2013 6:23:26.030 AM	Information	Executing request mapping "http://sap.com/xi/XI/MaaS/Airline/FSACheck_Agency2Airline" (SWCV f7a554b0a08b11df9608c3ba0a42301e)
4/23/2013 6:23:26.048 AM	Information	Delivering to channel: RFC_AirlineFlight_Receiver
4/23/2013 6:23:26.049 AM	Information	MP: processing local module localejbs/AF_Modules/RequestResponseBean
4/23/2013 6:23:26.049 AM	Information	RRB: entering RequestResponseBean
4/23/2013 6:23:26.049 AM	Information	RRB: forwarding the request message
4/23/2013 6:23:26.049 AM	Information	RRB: leaving RequestResponseBean
4/23/2013 6:23:26.049 AM	Information	RRB: suspending the transaction
4/23/2013 6:23:26.050 AM	Information	RFC Adapter received a synchronous message. Trying to send sRFC for SXIDEMO_AIRL_FLIGHT_CHECKAVAIL
4/23/2013 6:23:26.070 AM	Information	MP: processing local module localejbs/AF_Modules/ResponseOnewayBean
4/23/2013 6:23:26.070 AM	Information	ROB: entering ResponseOnewayBean
4/23/2013 6:23:26.070 AM	Information	ROB: resuming the transaction
4/23/2013 6:23:26.070 AM	Information	ROB: sending to the messaging system ...
4/23/2013 6:23:26.083 AM	Information	ROB: leaving ResponseOnewayBean
4/23/2013 6:23:26.083 AM	Information	ROB: returned with no response
4/23/2013 6:23:26.086 AM	Information	The message was successfully transmitted to endpoint <local> using connection IDoc_AAE_http://sap.com/xi/XI/System.
4/23/2013 6:23:26.122 AM	Information	The message status was set to DLVD.

To check the response from RFC sent to web service via SOAP Receiver Channel can be checked in the communication channel monitoring of channel SOAP_Receiver_ResponseFromRFC

The response can be checked in SOAP Workbench/external tool. Here in SOAP Workbench the response can be checked as URL mentioned in in Target URL of SOAP receiver channel is of SOAP Workbench.



6 VARIANT 2: SYNC/ASYN BRIDGE BY MEANS OF MODULE PROCESSOR

6.1 Introduction

This test case describes how to connect a synchronous system to an asynchronous system by means of an async/sync bridge without using BPM.

6.2 Sync/Async Bridge

A synchronous call is mapped to asynchronous request and response messages by means of the module processor. In the module processor of the receiver adapter, the synchronous message is converted to an asynchronous request message. The synchronous call waits until the receiver adapter sends a reply.

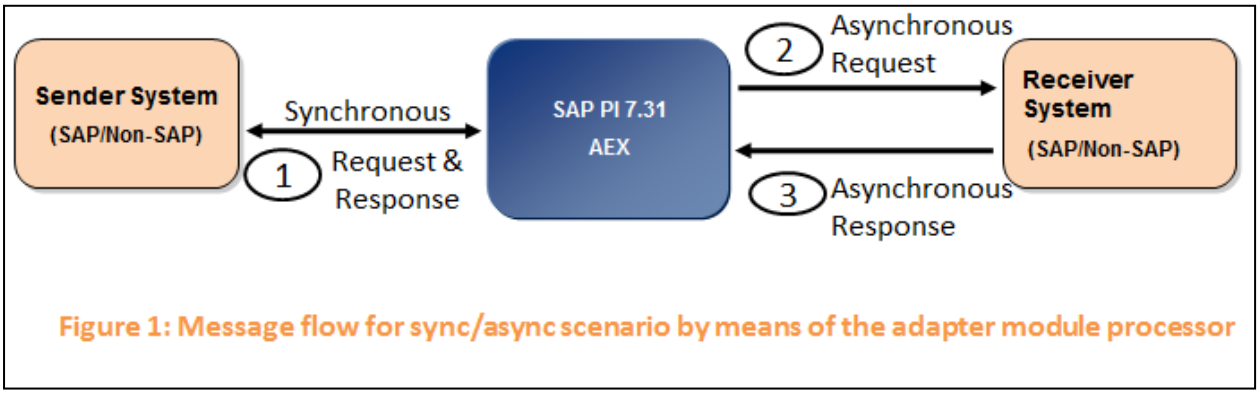


Figure 1: Message flow for sync/async scenario by means of the adapter module processor

6.3 Test scenario

The purpose of this scenario is to configure SOAP to FILE communication where SOAP is sync and FILE is Async

7 PREPARATION

7.1 User Permissions

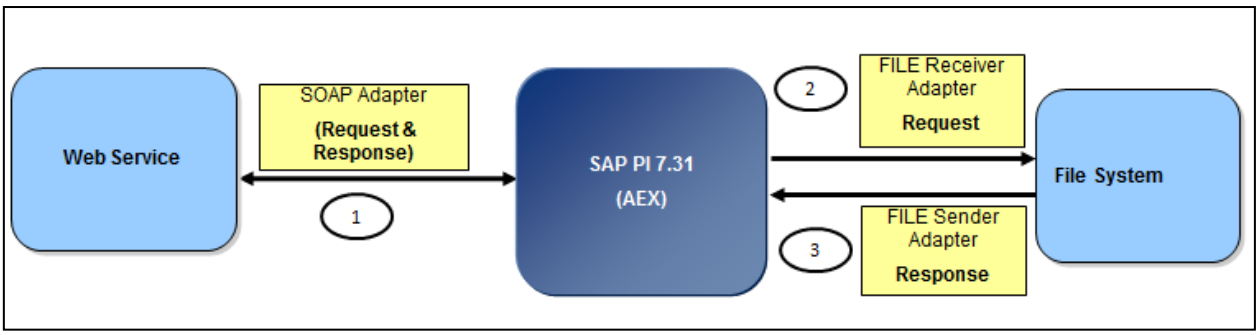
The tester should have permission to log on the PI test system and to open the SOA Monitors in the NetWeaver Administrator. Here PI 7.31 AEX is used to execute the test case. The system environment may differ in your test executions.

SOAP Workbench tool is used to respectively send and receive the messages.

8 SOAP - FILE

8.1 Overview

This test case describes how to connect an synchronous system to a aynchronous system by means of an sync/async bridge.

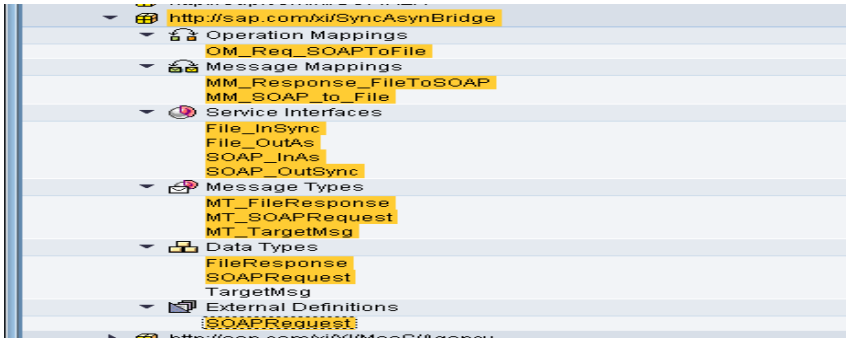


8.2 Description

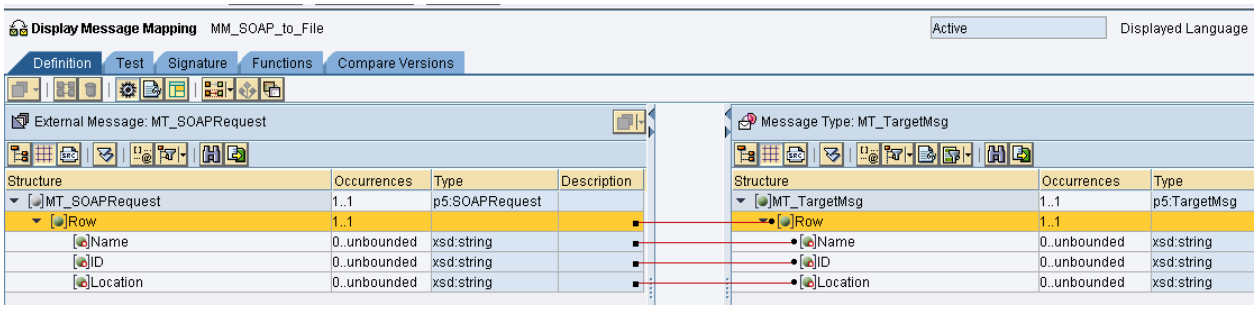
Web service sends the data to PI (SOAP Sender Adapter used) and PI does the request mapping and sends the request to FILE and the filename placed is with the message ID to correlate with the response message. Once the file is placed, the response (Status) is sent to web service via SOAP Adapter.
To implement the scenario, we need to define two ICO. One for routing the request message from the Web Service client to the File system, and one for routing back the response message.

8.3 Repository Objects

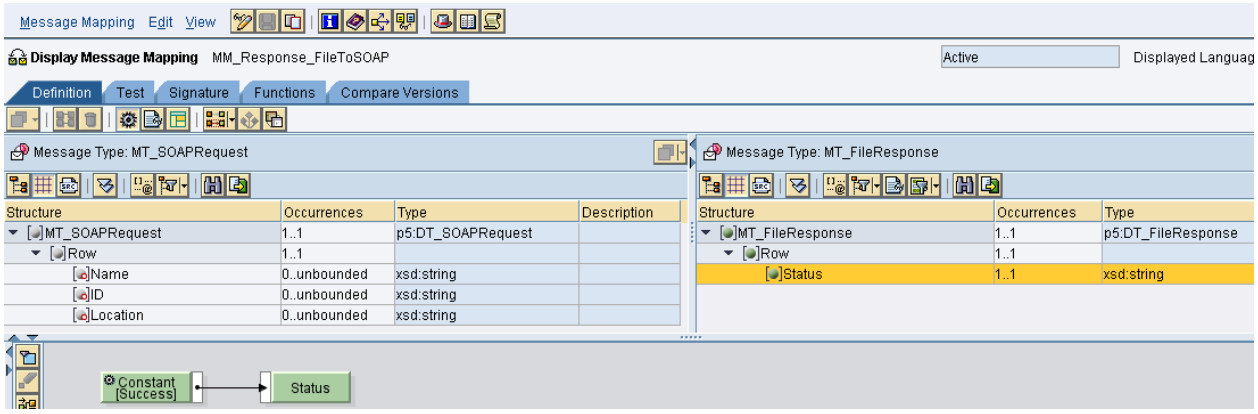
In the Enterprise Service Repository – on PI test system the below screenshot objects are created and used in this test case. The field structure can be referred in the Request and Response Mapping given below during creation.



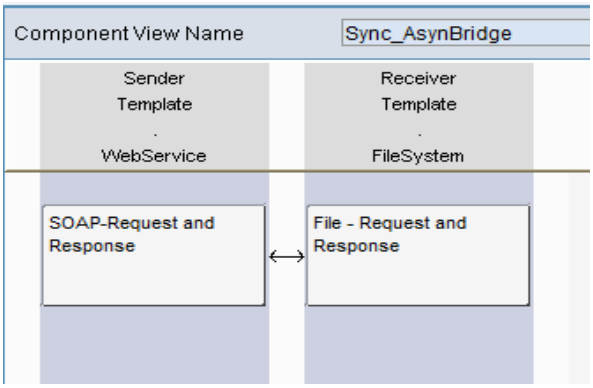
Request Mapping: MM_SOAP_to_File



Response Mapping: MM_Response_FileToSOAP

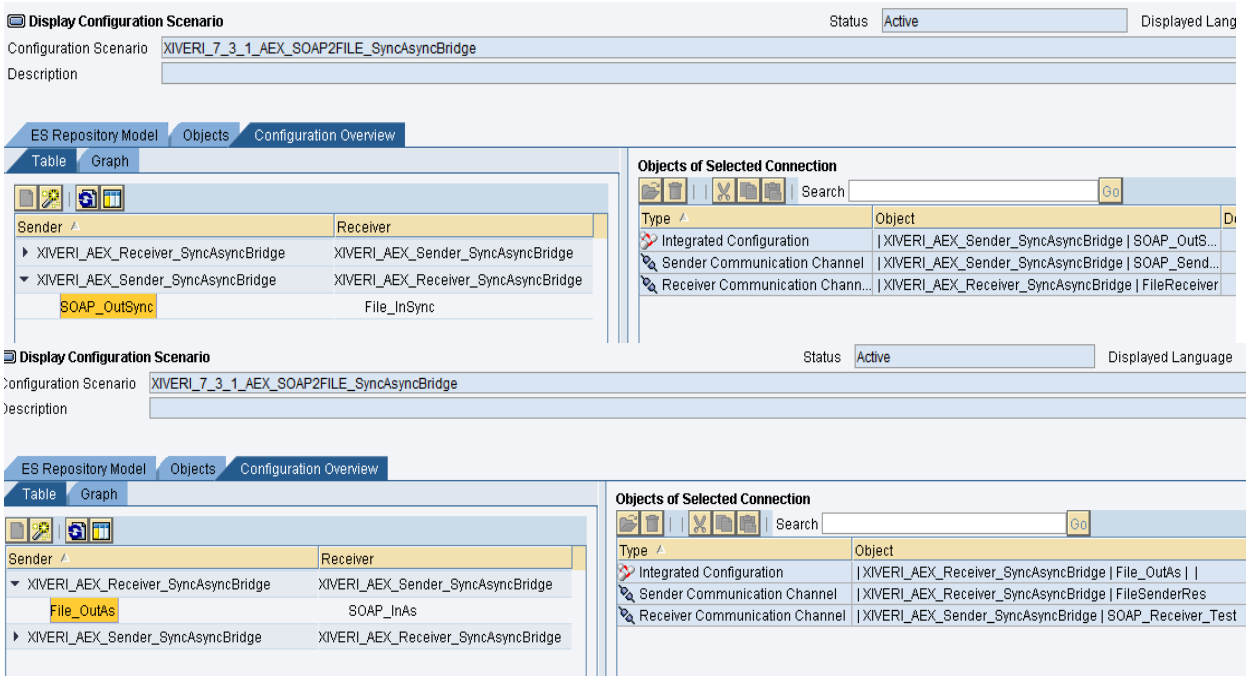


The following graphic shows the component view:



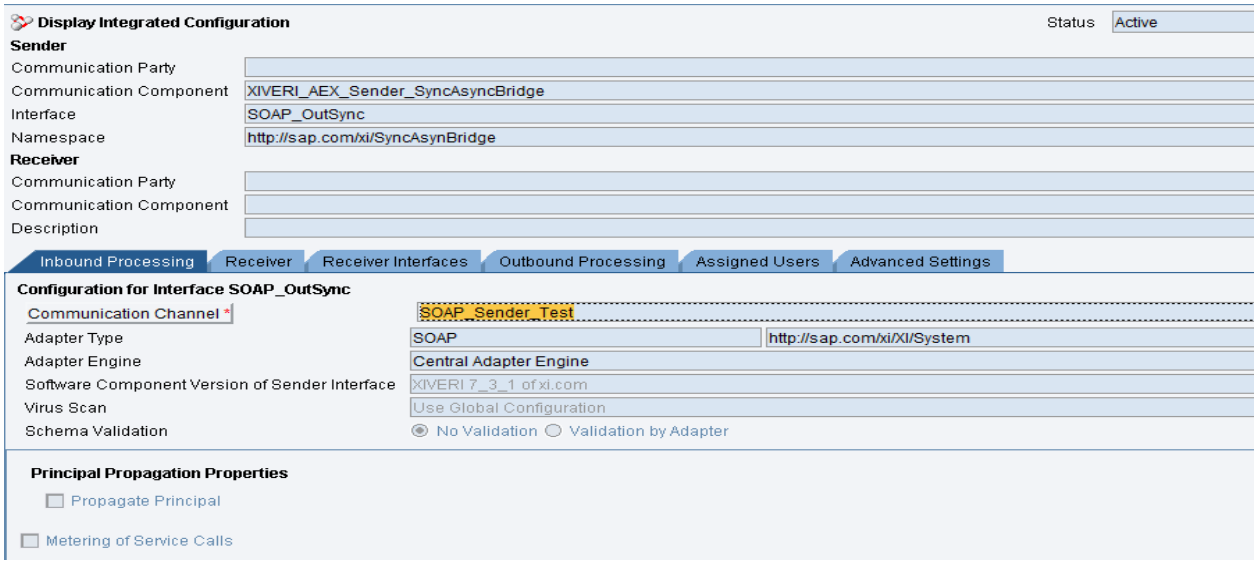
8.4 Configuration Objects

8.4.1 Configuration Overview

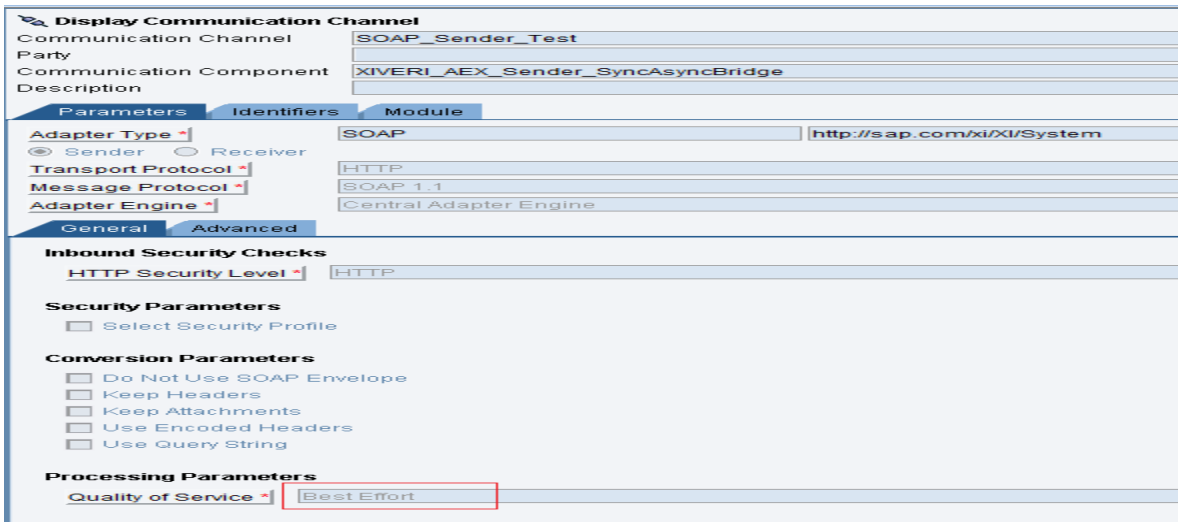


8.4.2 Integrated Configuration Objects in Integration Directory

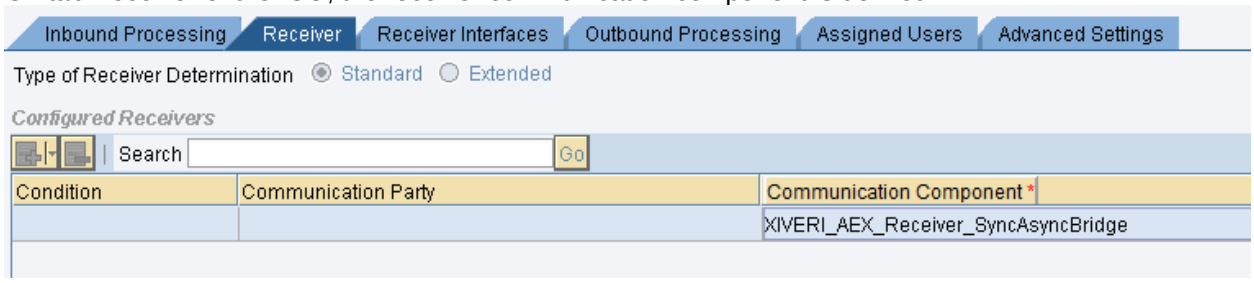
The first ICO defines the routing from the Web Service client to the File System. In the *Inbound Processing*, the SOAP sender channel is specified.



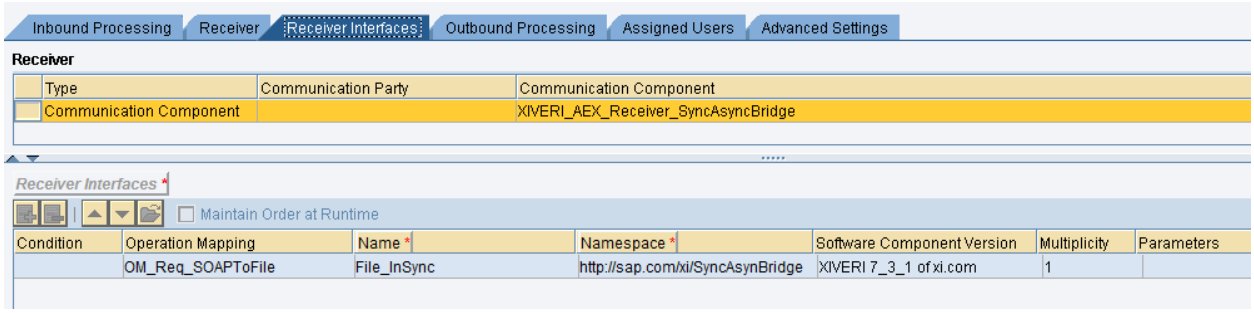
In the sender channel of type SOAP, the *Quality of Service* is set to *Best Effort*.



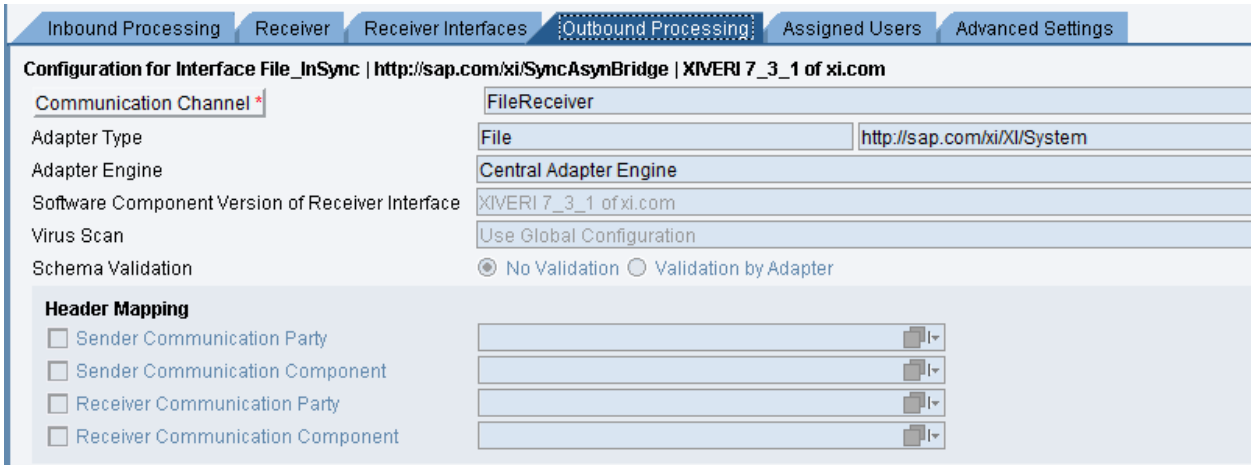
On tab *Receiver* of the ICO, the receiver communication component is defined



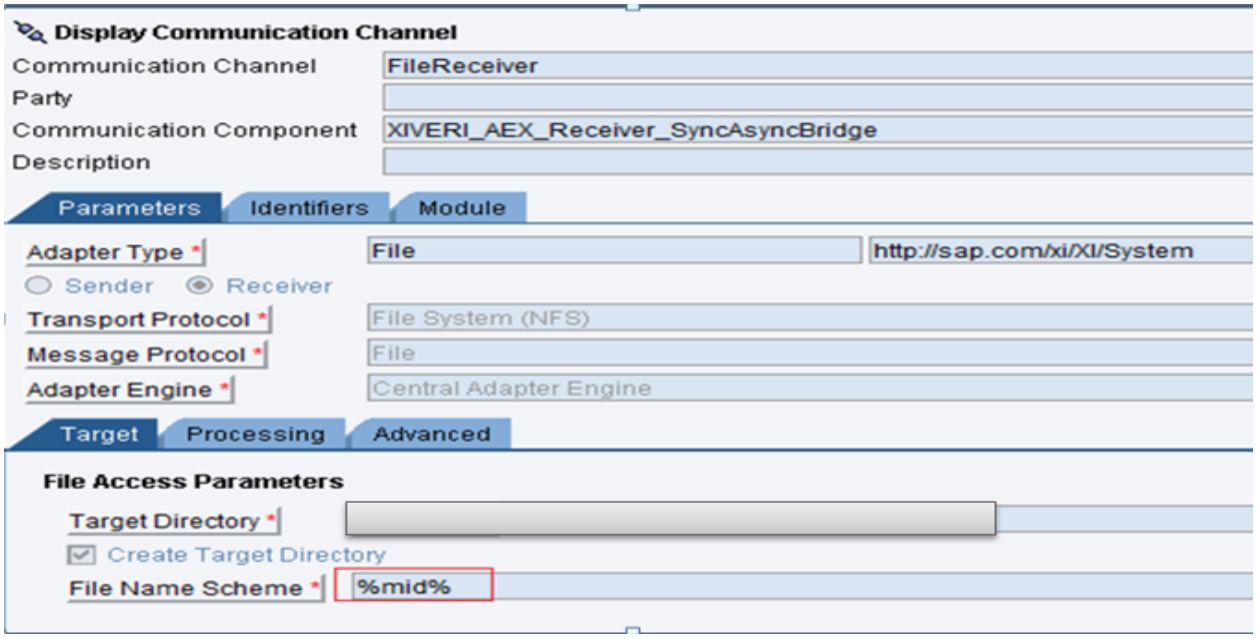
On tab *Receiver Interfaces*, the Operation Mapping for request mapping is set

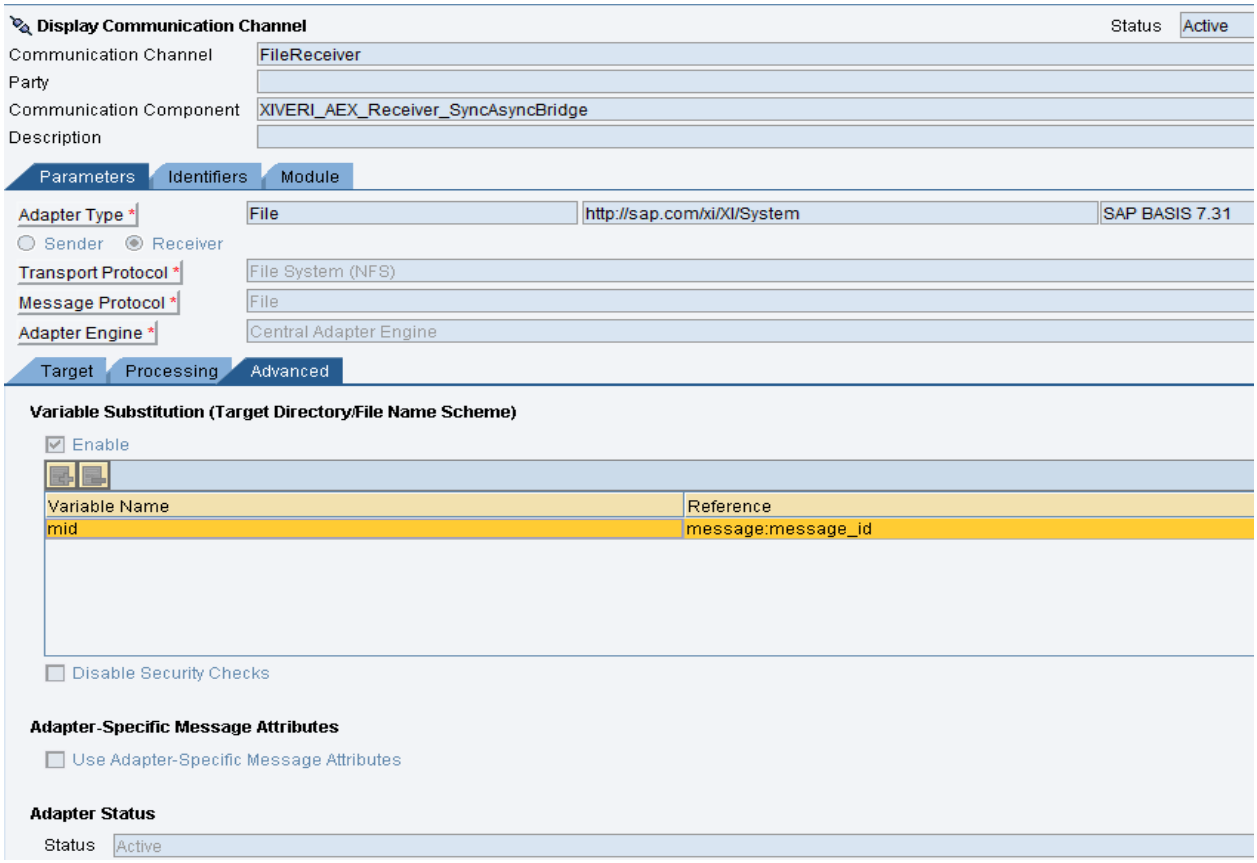


On tab *Outbound Processing*, the FILE receiver channel is specified.



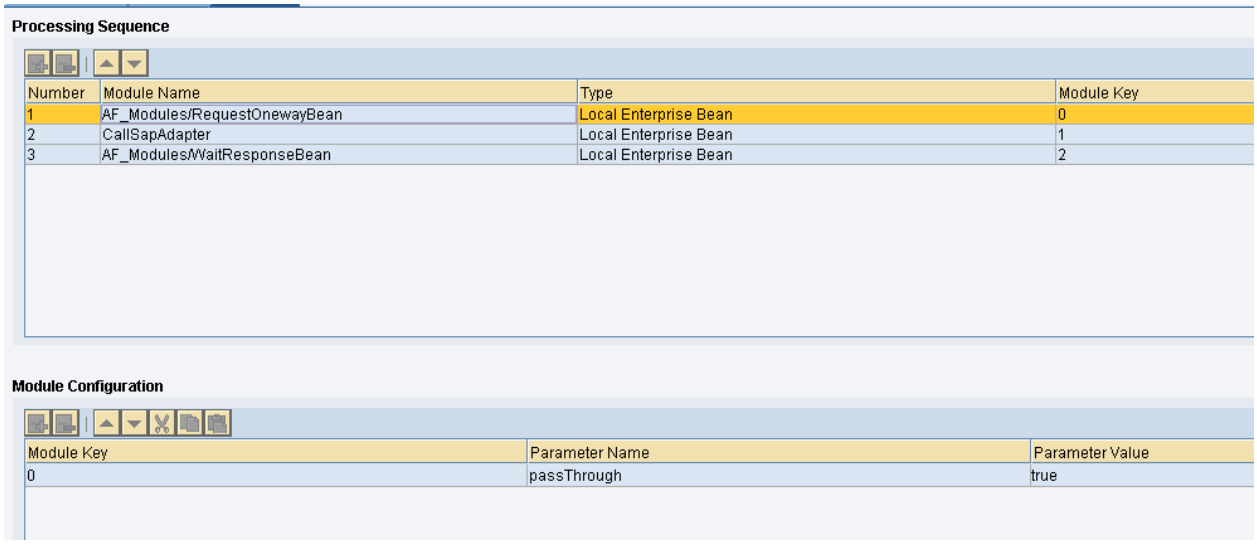
In the Parameters tab, maintain the target directory and set the file name scheme as "%mid%" (without quotes), so we can set the filename with the message id through variable substitution. Set the construction mode as create (do not append any info after the message id) and enable variable substitution, with a single entry: name="mid", reference="message:message_id" (also, without quotes).



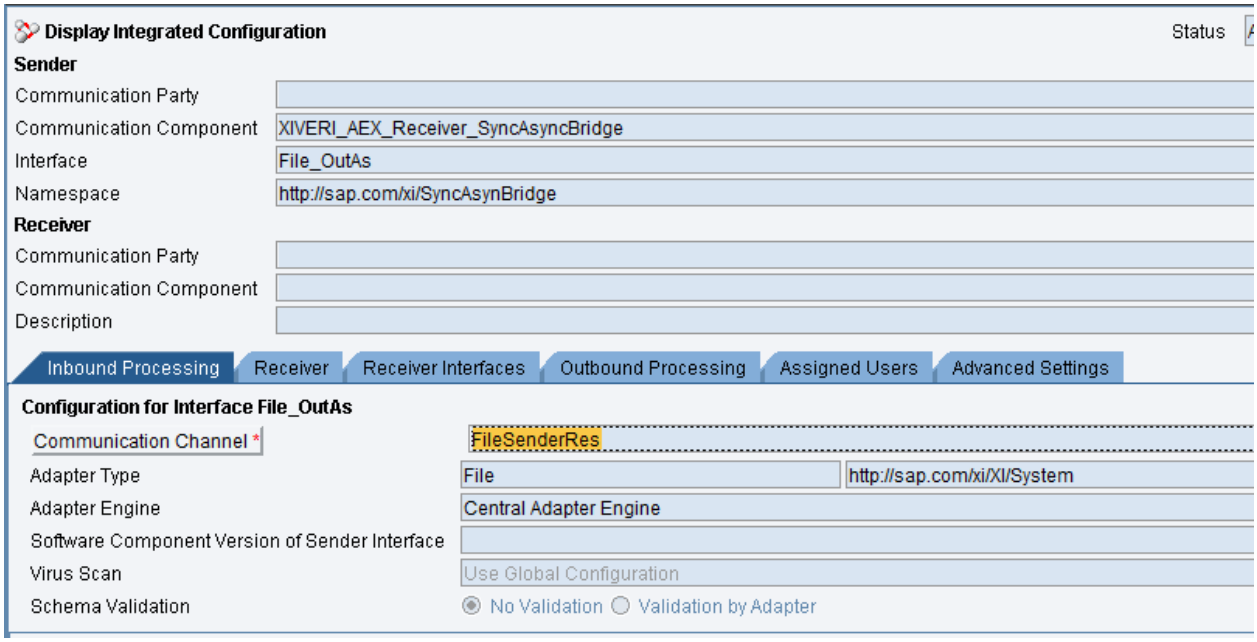


For more details on variable substitution made in File Receiver channel:

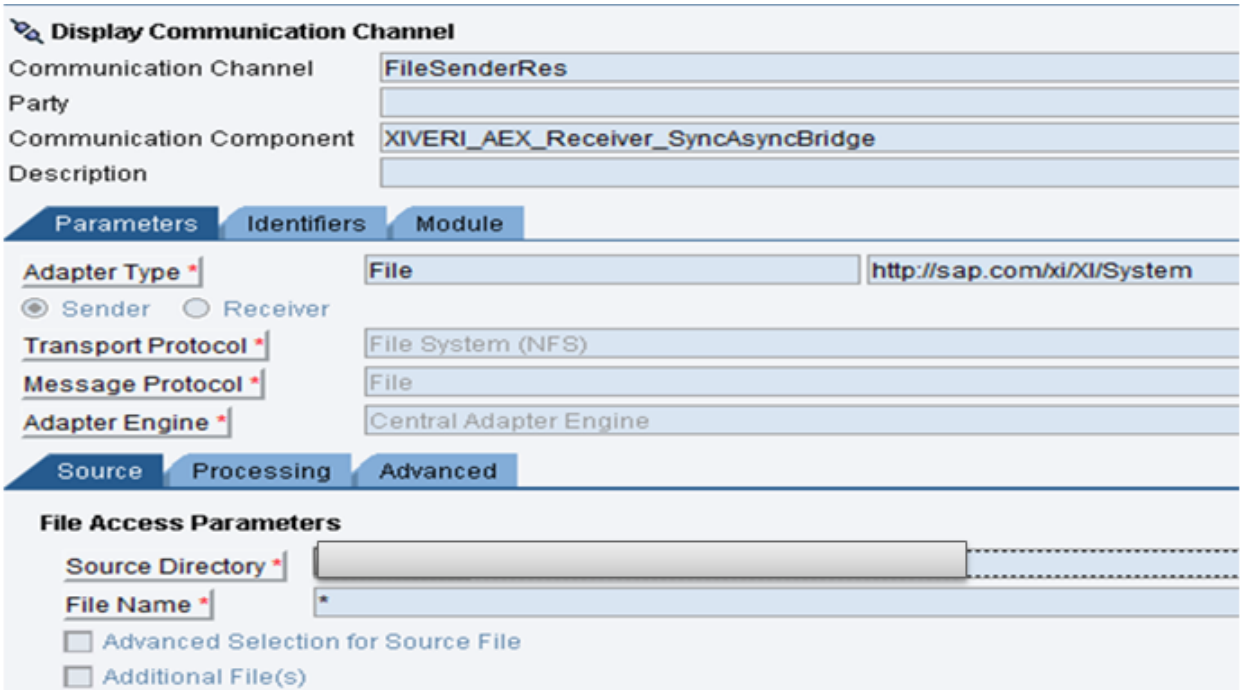
In the Module tab, insert the "AF_Modules/RequestOnewayBean" module before the CallSapAdapter module, and set its parameter "passThrough" with value "true"; also, insert the "AF_Modules/WaitResponseBean" module after the CallSapAdapter module

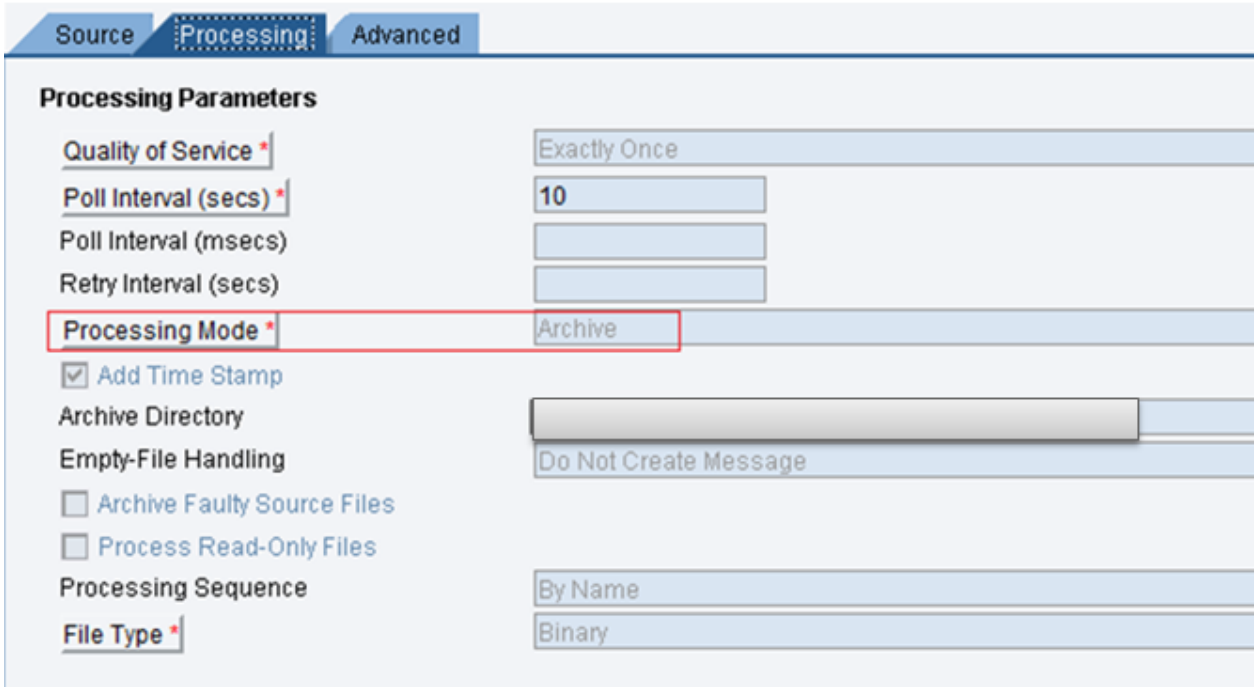


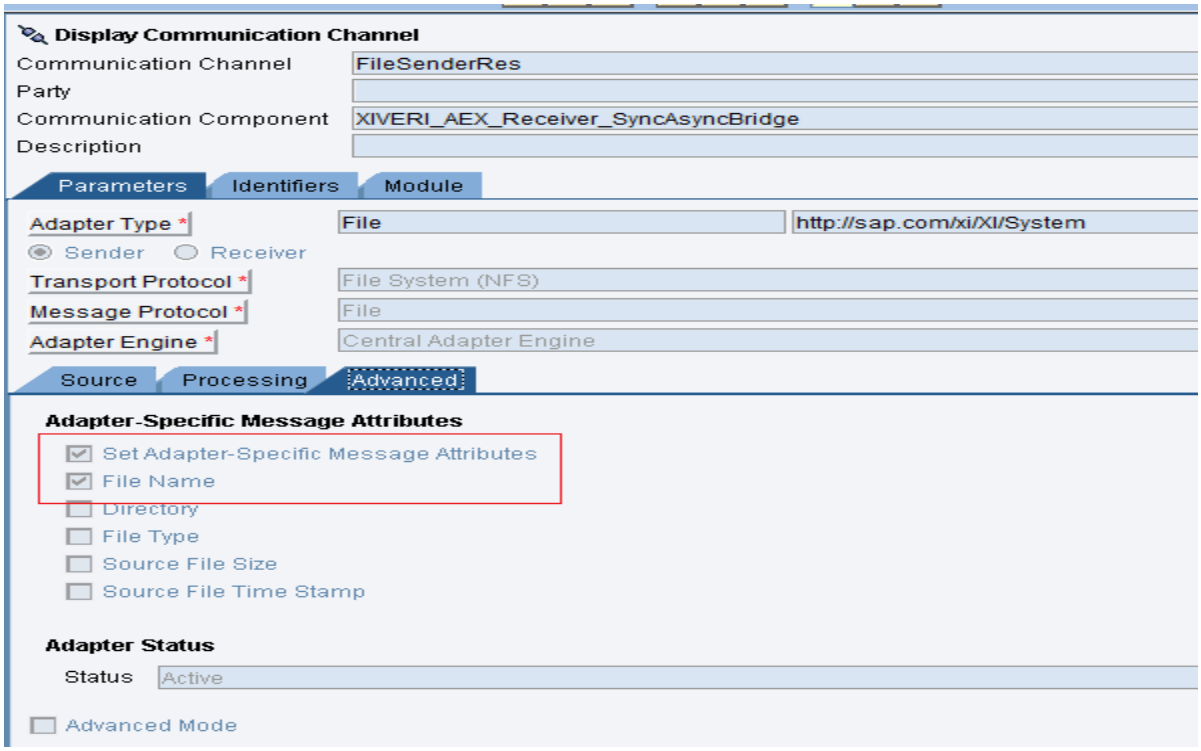
The **second ICO** defines the routing of the response message from File to the Web Service client. In the *Inbound Processing*, the File Sender channel is defined.



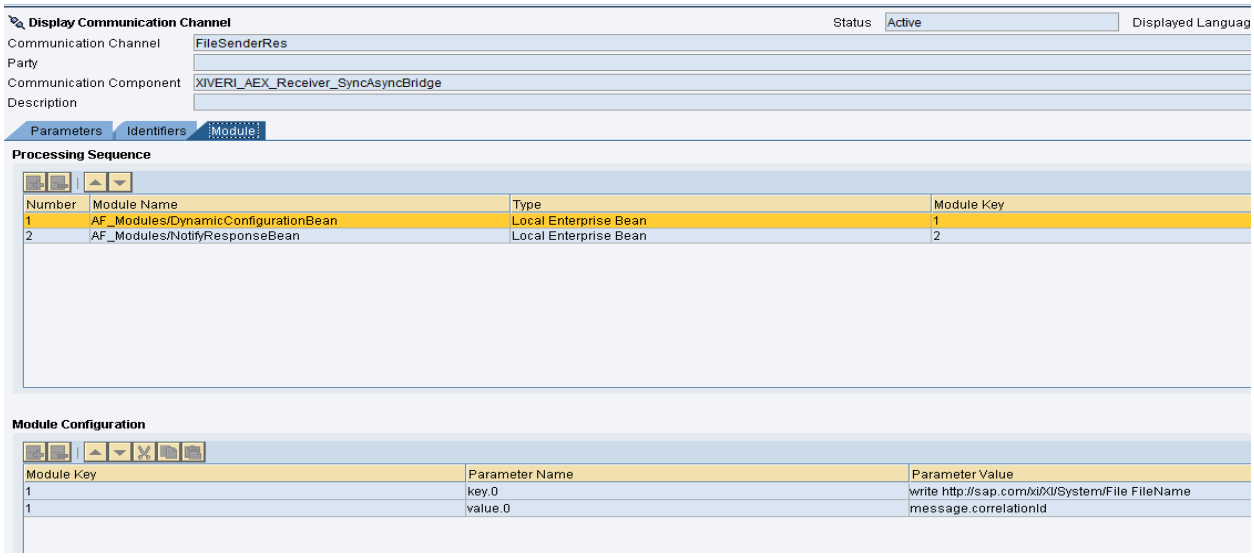
In the Parameters tab, maintain the source directory and set the file name as "*" (without extension); set the polling interval for a short time, for example, 10 seconds (a long polling time may cause a timeout in the sync SOAP); make sure you set the "Set Adapter-Specific Message Attributes" flag, with the File Name flag active
The file picked is been archived.







In the Module tab, replace the "CallSapAdapter" module with the "AF_Modules/NotifyResponseBean" (since the message doesn't need to be sent to the Integration Engine, but rather to the waiting WaitResponseBean); also, before the NotifyResponseBean, insert the "AF_Modules/DynamicConfigurationBean" module, with the following parameters and respective values: name="key.0" value="write http://sap.com/xi/XI/System/File FileName", name="value.0" value="message.correlationId".



The DynamicConfigurationBean module will get the filename ASMA-Adapter Specific Message Attributes (which was previously set by the adapter) and write its value in the message.correlationId attribute, before the message is sent to the waiting process (WaitResponseBean) by the NotifyResponseBean module.

8.4.3 Test Description

Testing the scenario SOAP - > FILE please follow the steps below:

Pushing the Message

a) Check whether both (Sender & Receiver) channels being used in this scenario are started. If not then start them via Channel Monitoring of NWA/RWB.

b) Open the external SOAP Request/Response tool being used

c) In the field URL set the below value.

https://<HOST>:<PORT>/XISOAPAdapter/MessageServlet?senderService=XIVERI_AEX_Sender_SyncAsyncBridge&interface=SOAP_OutSync&interfaceNamespace=http://sap.com/xi/SyncAsynBridge

Set the host and port of the PI server.

Note: The URL specified here is for SOAP Workbench tool when used. If any other external tool is used, the URL should be changed accordingly.

d) Select “Use Authentication”. Then enter your username and password for the PI server.

e) Under the label SOAP delete the existing text. Now copy the content from the payload file below and paste it in the editbox under label SOAP.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
<ns0:MT_SOAPRequest xmlns:ns0="http://sap.com/xi/SyncAsynBridge">
  <Row>
```

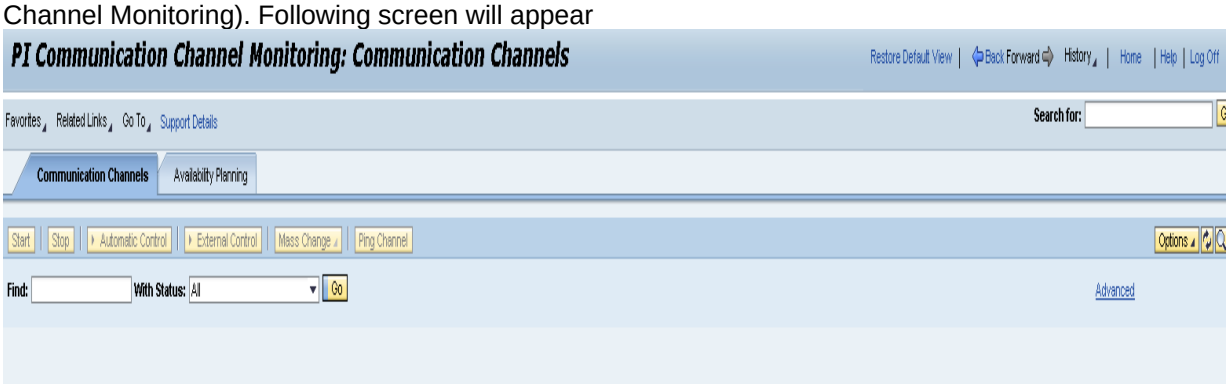
```
      <Name>ABC</Name>
      <ID>123</ID>
      <Location>Bangalore</Location>
    </Row>
</ns0:MT_SOAPRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

f) Click on the Send button. It will push a message.

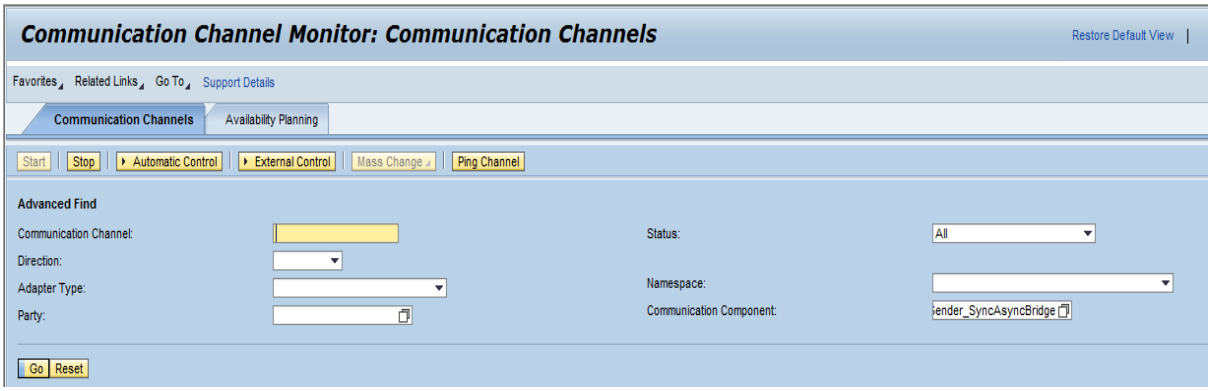
g) As QoS is Best Effort in SOAP Sender adapter the response is received as below:

Channel Monitoring

Open the **PI Communication Channel Monitoring** (NWA→SOA→ Monitoring→ PI Communication Channel Monitoring). Following screen will appear



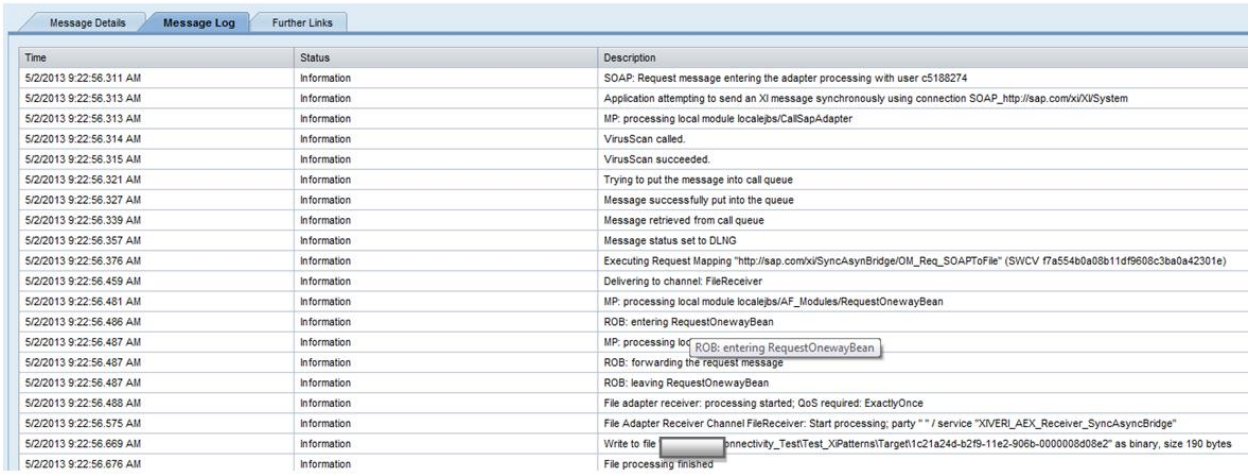
Click on the link “Advanced”. The following screen appears. Set the Sender Communication Component “XIVERI_AEX_Sender_SyncAsyncBridge” in the edit box “**Communication Component**” & hit **GO** button.



As a result we see successful message in the SOAP Sender channel

Goto File Receiver Communication Channel to check the message received.

To check on the modules process, goto Message Log of either SOAP Sender Channel or File Receiver Channel



Message Details | Message Log | Further Links

Time	Status	Description
5/2/2013 9:22:56.311 AM	Information	SOAP: Request message entering the adapter processing with user c5188274
5/2/2013 9:22:56.313 AM	Information	Application attempting to send an XI message synchronously using connection SOAP_http://sap.com/xi/XI/System
5/2/2013 9:22:56.313 AM	Information	MP: processing local module localejbs/CallSapAdapter
5/2/2013 9:22:56.314 AM	Information	VirusScan called.
5/2/2013 9:22:56.315 AM	Information	VirusScan succeeded.
5/2/2013 9:22:56.321 AM	Information	Trying to put the message into call queue
5/2/2013 9:22:56.327 AM	Information	Message successfully put into the queue
5/2/2013 9:22:56.339 AM	Information	Message retrieved from call queue
5/2/2013 9:22:56.357 AM	Information	Message status set to DLNG
5/2/2013 9:22:56.376 AM	Information	Executing Request Mapping "http://sap.com/xi/SyncAsynBridge/OM_Req_SOAPToFile" (SWCV f7a554b0a08b11df9608c3ba0a42301e)
5/2/2013 9:22:56.459 AM	Information	Delivering to channel: FileReceiver
5/2/2013 9:22:56.481 AM	Information	MP: processing local module localejbs/AF_Modules/RequestOnewayBean
5/2/2013 9:22:56.486 AM	Information	ROB: entering RequestOnewayBean
5/2/2013 9:22:56.487 AM	Information	MP: processing lo
5/2/2013 9:22:56.487 AM	Information	ROB: forwarding the request message
5/2/2013 9:22:56.487 AM	Information	ROB: leaving RequestOnewayBean
5/2/2013 9:22:56.488 AM	Information	File adapter receiver: processing started; QoS required: ExactlyOnce
5/2/2013 9:22:56.575 AM	Information	File Adapter Receiver Channel FileReceiver: Start processing; party " " / service "XIVERI_AEX_Receiver_SyncAsyncBridge"
5/2/2013 9:22:56.669 AM	Information	Write to file [redacted]onnectivity_Test\Test_XiPatterns\Target\1c21a24d-b2f9-11e2-906b-0000008d08e2" as binary, size 190 bytes
5/2/2013 9:22:56.676 AM	Information	File processing finished

ROB: entering RequestOnewayBean

5/2/2013 9:22:56.459 AM	Information	Delivering to channel: FileReceiver
5/2/2013 9:22:56.481 AM	Information	MP: processing local module localejbs/AF_Modules/RequestOnewayBean
5/2/2013 9:22:56.486 AM	Information	ROB: entering RequestOnewayBean
5/2/2013 9:22:56.487 AM	Information	MP: processing local module localejbs/CallSapAdapter
5/2/2013 9:22:56.487 AM	Information	ROB: forwarding the request message
5/2/2013 9:22:56.487 AM	Information	ROB: leaving RequestOnewayBean
5/2/2013 9:22:56.488 AM	Information	File adapter receiver: processing started; QoS required: ExactlyOnce
5/2/2013 9:22:56.575 AM	Information	File Adapter Receiver Channel FileReceiver: Start processing; party " " / service "XIVERI_AEX_Receiver_SyncAsyncBridge"
5/2/2013 9:22:56.669 AM	Information	Write to file [redacted]onnectivity_Test\Test_XiPatterns\Target\1c21a24d-b2f9-11e2-906b-0000008d08e2" as binary, size 190 bytes
5/2/2013 9:22:56.676 AM	Information	File processing finished
5/2/2013 9:22:56.679 AM	Information	MP: processing local module localejbs/AF_Modules/WaitResponseBean
5/2/2013 9:22:56.683 AM	Information	WRB: entering WaitResponseBean
5/2/2013 9:22:56.683 AM	Information	WRB: retrieving the message for 1c21a24d-b2f9-11e2-906b-0000008d08e2 ...
5/2/2013 9:23:23.134 AM	Information	WRB: retrieved the message: ApplicationResponse
5/2/2013 9:23:23.135 AM	Information	WRB: leaving WaitResponseBean
5/2/2013 9:23:23.209 AM	Information	Application sent message synchronously using connection SOAP_http://sap.com/xi/XI/System. Returning to application
5/2/2013 9:23:23.219 AM	Information	SOAP: Processing completed
5/2/2013 9:23:23.220 AM	Information	SOAP: Response message leaving the adapter
5/2/2013 9:23:23.228 AM	Information	Message was successfully transmitted to endpoint <local> using connection SOAP_http://sap.com/xi/XI/System
5/2/2013 9:23:23.228 AM	Information	Message status set to DLVD

The response is sent from File to SOAP can be checked in the communication channel monitor of channel FileSenderRes

www.sap.com