Configuring Sync/Async Bridge on SAP NetWeaver Process Orchestration

Applies to:

SAP NetWeaver Process Orchestration, release 7.31 SP4 and above.

Summary

This paper shows how to connect a synchronous system to an asynchronous system by means of a sync/async bridge. Two approaches are described: purely running within the messaging system via module processor as well as via a Business Process Management (BPM) process. The underlying scenario connects a Web Service client to a JMS broker. For correlating the asynchronous request and response messages we can either use payload data or leveraging the JMS adapter's correlation settings.

Author: Alexander Bundschuh

Company: SAP AG

Created on: 29th of January 2013

Author Bio

Alexander Bundschuh is a product manager at SAP AG focusing on SAP NetWeaver Process Integration and SAP NetWeaver Process Orchestration.

Table of Contents

Introduction

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

The scenario that the paper refers to connects a Web Service client to a JMS broker. JMS supports asynchronous communication only. A request/response model similar to synchronous communication can be implemented using a reply queue mechanism. For correlating the asynchronous request and response messages we can either use payload data or leveraging the JMS adapter's correlation settings.

Note: You need to consider that when using synchronous communication mode, reliable messaging is not guaranteed. You can achieve quality of service best effort only. Especially, for the sync/async bridge, it is not guaranteed that the asynchronous response will be sent within the Web Service response timeout period.

This paper refers to the how to guide How to Correlate JMS Messages (NW7.0) where various options to implement async/sync and sync/async scenarios were discussed. Whereas the how to guide applies to an SAP NetWeaver PI dual-stack installation option, the current paper describes the implementation on an SAP NetWeaver Process Orchestration installation option. SAP NetWeaver Process Orchestration runs on Java-only, and is a co-installation of the products Advanced Adapter Engine Extended (AEX), Business Process Management (BPM), and Business Rules Management (BRM). For more details, refer to the blog about Installation Options for Process Integration and Orchestration Use Cases on SCN. Other than in the how to guide, we focus here on the sync/async case only. A similar paper describing the async/sync pattern on an SAP NetWeaver Process Orchestration installation option has been recently released on SCN, see Configuring Async/Sync Bridge on SAP NetWeaver Process Orchestration.

We will implement two different ways to bridge the different communication modes:

- via module processor
- via a BPM process

Note: Former approach is also supported on an AEX installation option, whereas latter requires BPM, and hence only runs on a Process Orchestration.

Note: In this paper I won't describe the implementation of the scenarios in all detail. Instead I will stick to the minimum required steps to understand the concepts. I assume that you are already familiar with the new tools that come with the SAP NetWeaver Process Orchestration such as ESR in Eclipse, Integration Flows, and Process Composer. Otherwise, I would propose that you start with reading the paper Configuring Async/Sync Bridge on SAP NetWeaver Process Orchestration that I recently published handling the async/sync bridge case. There, all new concepts are described in very detail. So, it can be seen as a sort of tutorial, and is particularly addressed to PI integration developers who haven't gained much experience so far with the new development environment of SAP NetWeaver Process Orchestration.

Prerequisites

System Setup

If not otherwise stated, the scenarios can be implemented on an SAP NetWeaver Process Orchestration 7.31 SP4 system. The only exception is if you like to use integrated monitoring between PI's message monitor and BPM's process monitor, i.e., navigating from message monitor to process monitor and vice versa. This will be only supported as of release 7.31 SP6.

JMS Provider

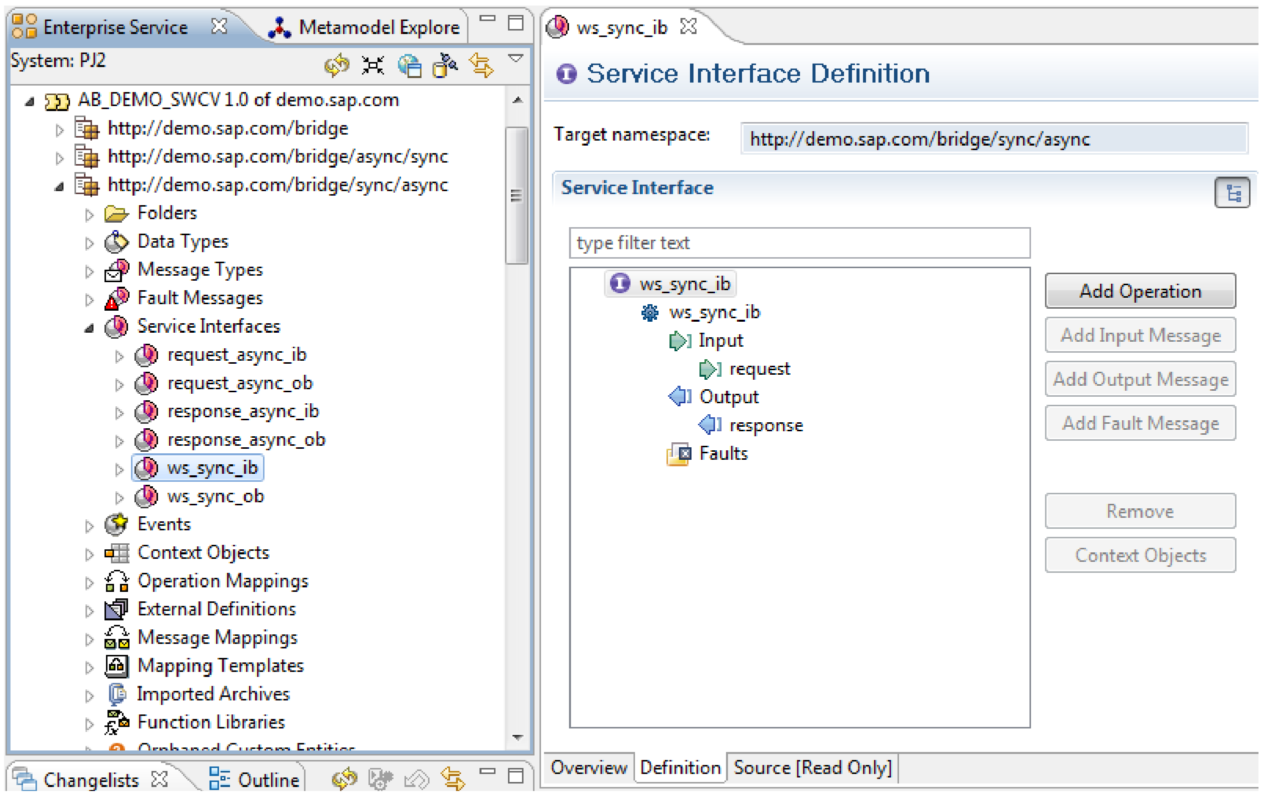
For the asynchronous communication I used SonicMQ JMS provider. You can use any other JMS provider which is supported by SAP NetWeaver PI. For more details about PI's JMS adapter, refer to Configuring the JMS Adapter in the SAP Help Portal.

ESR Objects

Both Implementation options discussed in this paper share the same ESR objects.

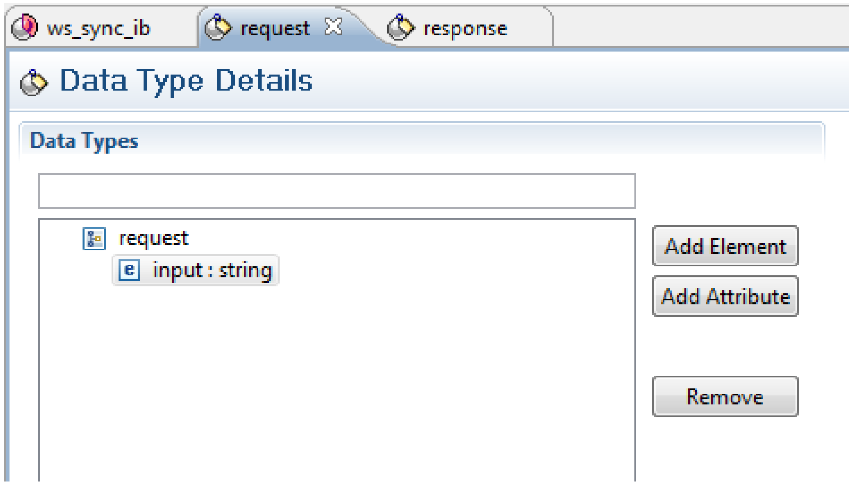
Start the *NetWeaver Developer Studio* (NWDS), and open the *Enterprise Service Repository* perspective. We need to define six service interfaces in total:

- An asynchronous inbound interface referring to data type request, here **request_async_ib**
- An asynchronous outbound interface referring to data type request, here **request_async_ob**
- An asynchronous inbound interface referring to data type response, here **response_async_ib**
- An asynchronous outbound interface referring to data type response, here **response_async_ob**
- A synchronous inbound interface referring to data type request as input, and data type response as output, here **ws_sync_ib**
- A synchronous outbound interface referring to data type request as input, and data type response as output, here **ws_sync_ob**



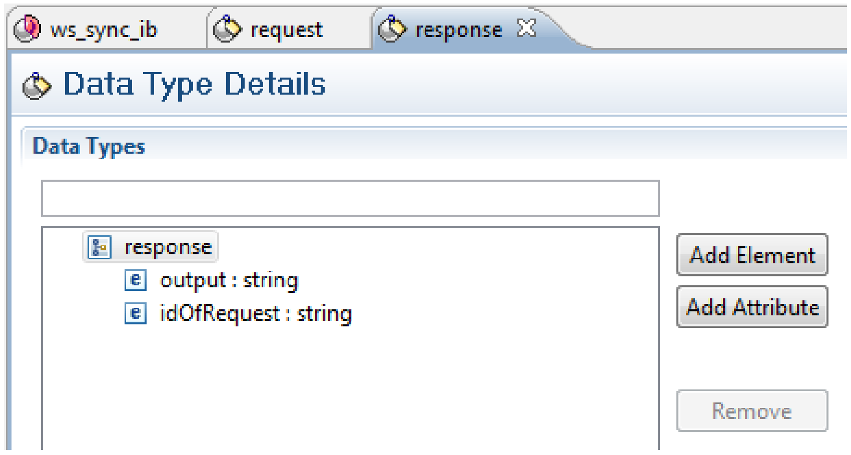
Target namespace: http://demo.sap.com/bridge/sync/async

The associated *request* data type simply contains one *input* field of type *string*.



request
input : string

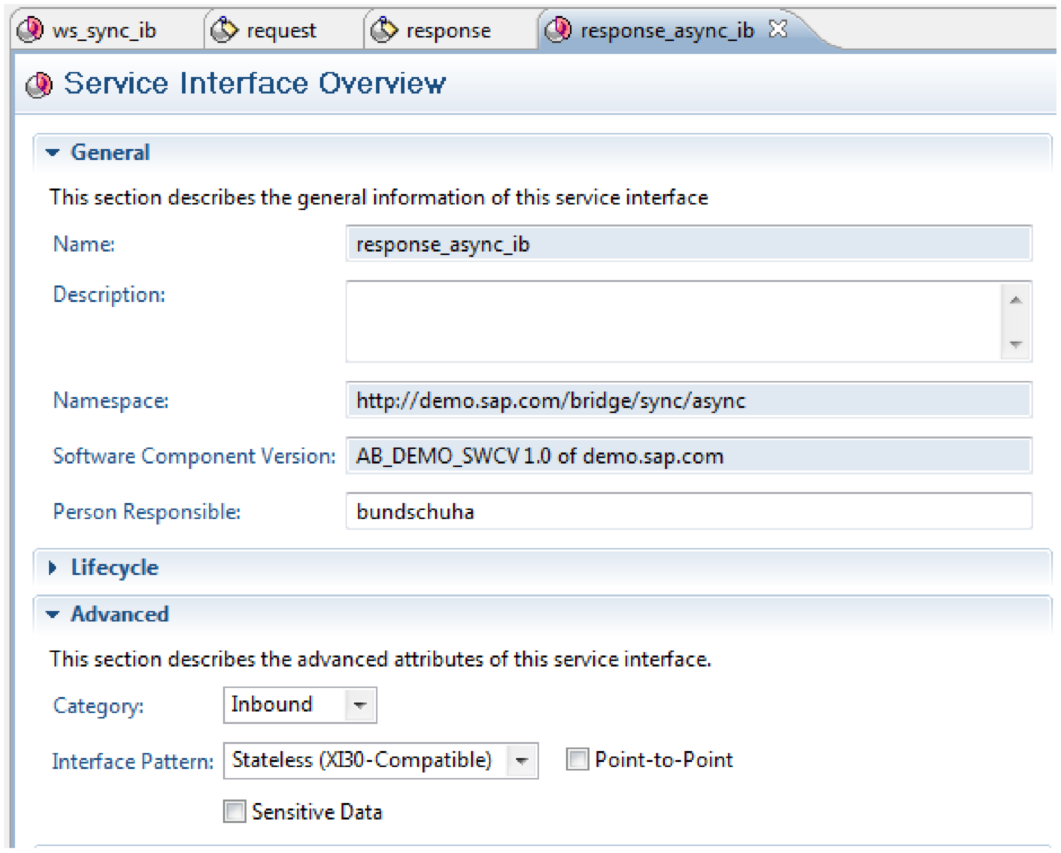
The associated *response* data type contains an *output* field, and another field *idOfRequest* where we will map the reference to the original request message to, both of type *string*.



response
output : string
idOfRequest : string

For the asynchronous inbound interfaces which are used to exchange messages from the AEX to the BPM process in the second option, we need to set the interface pattern as *Stateless XI3.0 Compatible*. Since we do not use multiple operations in those scenarios anyway, we can actually choose *Stateless XI 3.0 Compatible* pattern for all interfaces used here.

Note: Reliable communication between AEX and BPM is ensured via Java Proxy Runtime using the XI 3.0 protocol. Only if the inbound interface used within a message start event or an intermediate message event in BPM is of *Stateless XI 3.0 Compatible* pattern, an XI end point is created.



Service Interface Overview

General

This section describes the general information of this service interface

Name:	response_async_ib
Description:	
Namespace:	http://demo.sap.com/bridge/sync/async
Software Component Version:	AB_DEMO_SWCV 1.0 of demo.sap.com
Person Responsible:	bundschuha

Lifecycle

Advanced

This section describes the advanced attributes of this service interface.

Category: Inbound

Interface Pattern: Stateless (XI30-Compatible) ☐ Point-to-Point

☐ Sensitive Data

Sync/Async Bridge by means of the module processor

A synchronous call is mapped to asynchronous JMS request and response messages by means of the module processor. In the module processor of the SOAP sender adapter, the synchronous message is converted to an asynchronous request message and sent to a JMS queue. The synchronous call waits until the JMS provider sends a reply. In the module processor of the SOAP receiver adapter the asynchronous response is passed to the module processor of the SOAP sender adapter where the asynchronous response is converted to a synchronous response and sent to the waiting Web Service client.

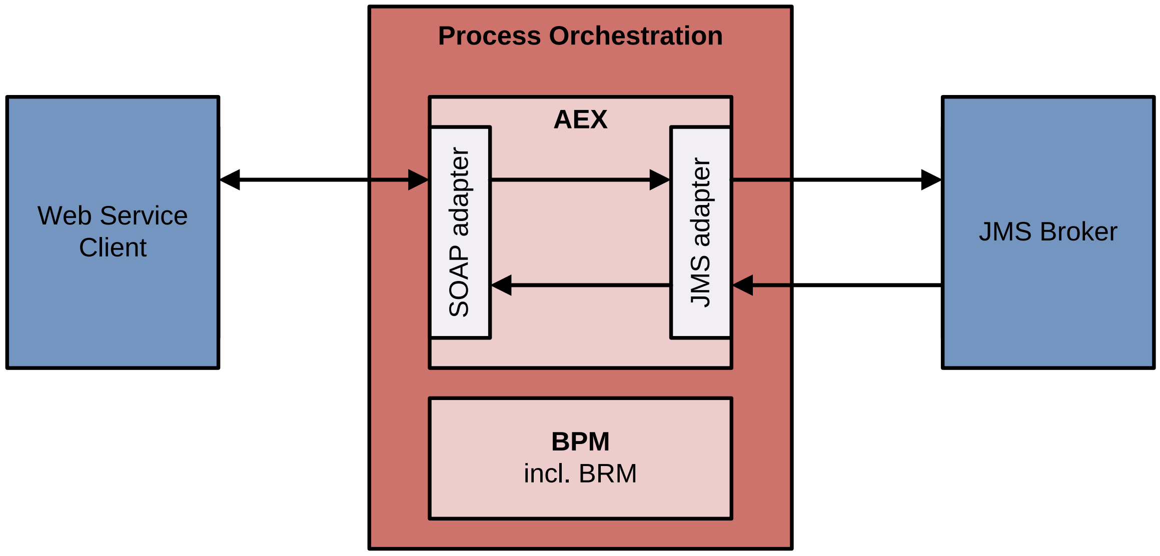


Figure 1: Message flow for sync/async scenario by means of the adapter module processor

To implement the scenario, we need to define two Integration Flows. One for routing the request message from the Web Service client to the JMS broker, and one for routing back the response message.

In order to correlate the response message to the request message, the correlation settings of the JMS adapter are applied, i.e., the *PI Conversation ID* of the response needs to keep the *PI Message ID* of the original request message to ensure that it is routed to the right session. See also [Configuring Async/Sync and Sync/Async Bridge in the JMS Adapter]() on the SAP Help Portal.

At a glance, the following settings have to be made:

- Create an Integration Flow from Web Service client to JMS broker
  - Add module *AF_Modules/RequestOnewayBean* at the beginning of the module chain of the SOAP sender channel to convert the synchronous request message to an asynchronous request message
  - Add module *AF_Modules/WaitResponseBean* at the end of the module chain of the SOAP sender channel to wait for the response message
  - Correlation settings in the JMS receiver communication channel: Set the *JMS Correlation ID* to the *PI Message ID*, and select the *Store JMS Correlation ID of request* check box to save the JMS Correlation ID of the request. The *JMS Correlation ID* is just a means to get the *PI Message ID* matched to the *JMS Message ID*. Latter is created once the request message is passed to the JMS queue, and stored on PO as a key tuple together with the *PI Message ID*.
- Create an Integration Flow from JMS broker to Web Service client
  - Correlation settings in the JMS sender communication channel: Set the *PI Conversation ID* to the *Stored JMS Correlation ID of request*. For the response message, we need to ensure that its *JMS Correlation ID* keeps the *JMS Message ID* of the request message. In the JMS sender adapter, the beforehand stored key mapping is looked up, and the *JMS Message ID* is then matched to the *PI Message ID* which is put into the *PI Conversation ID*.

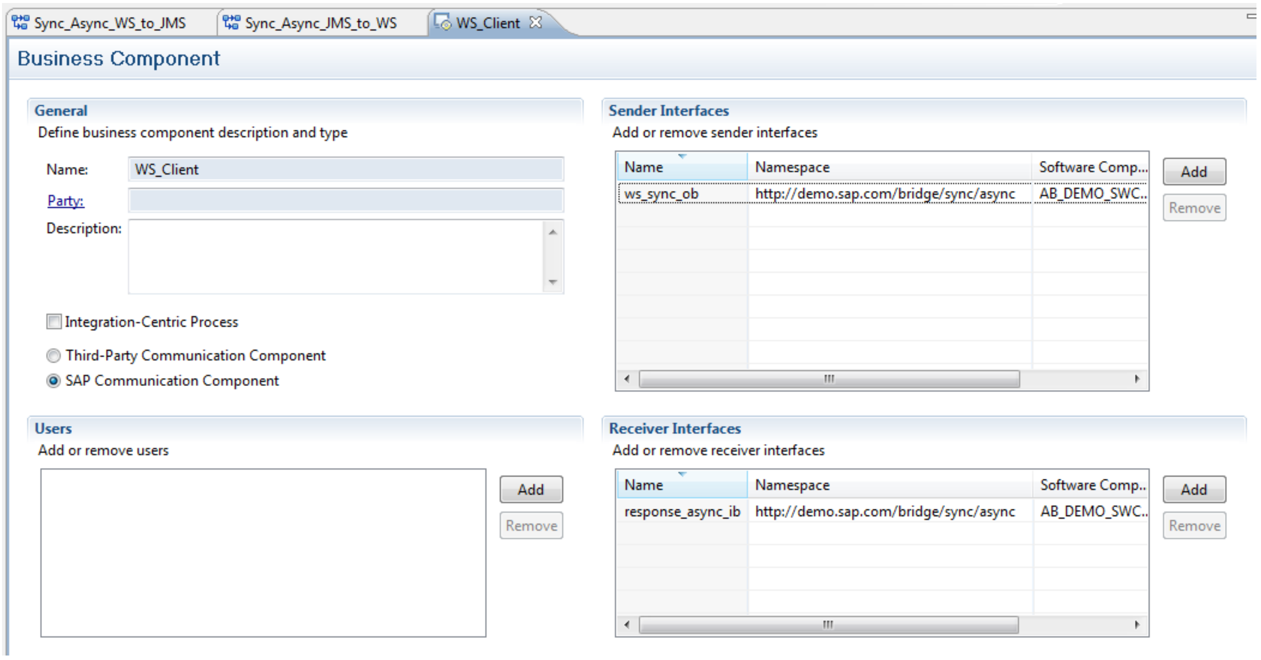
- Add module *AF_Modules/NotifyResponseBean* to the module chain of the SOAP receiver channel to pass the response message to the module processor of the SOAP sender adapter

Note: In this paper, the configuration is mainly done in the *SAP NetWeaver Developer Studio* (NWDS) using the new User Interfaces in Eclipse such as the *SAP Process Integration Designer* perspective to model the Integration Flows. Once you deploy an Integration Flow, a corresponding Integrated Configuration Object (ICO) is created in the Integration Directory. Furthermore, the approach via the module processors is also supported on a PI dual-stack system from release 7.11 on. You may like to implement the approach on the Advanced Adapter Engine of a dual-stack PI system however Integration Flows are not supported on a PI dual-stack system. For this reason, I have added at the end of this chapter screenshots of the corresponding ICOs.

Integration Flow from Web Service client to JMS Broker

First, we need to define Business Components for the Web Service client and the JMS provider.

In the NWDS, open the *SAP Process Integration Designer* perspective, and create a Business Component for the Web Service client, here **WS_Client**, and add sender interface **ws_sync_ob** and receiver interface **response_async_ib** to the same. The sender interface chosen is of mode synchronous since the Web Service client will call a synchronous Web Service. The receiver interface however is of mode asynchronous since the JMS provider will reply with an asynchronous message which is then converted into a synchronous response.



General
Define business component description and type

Name: WS_Client
Party:
Description:

Integration-Centric Process
Third-Party Communication Component
[✓ SAP Communication Component]

Users
Add or remove users

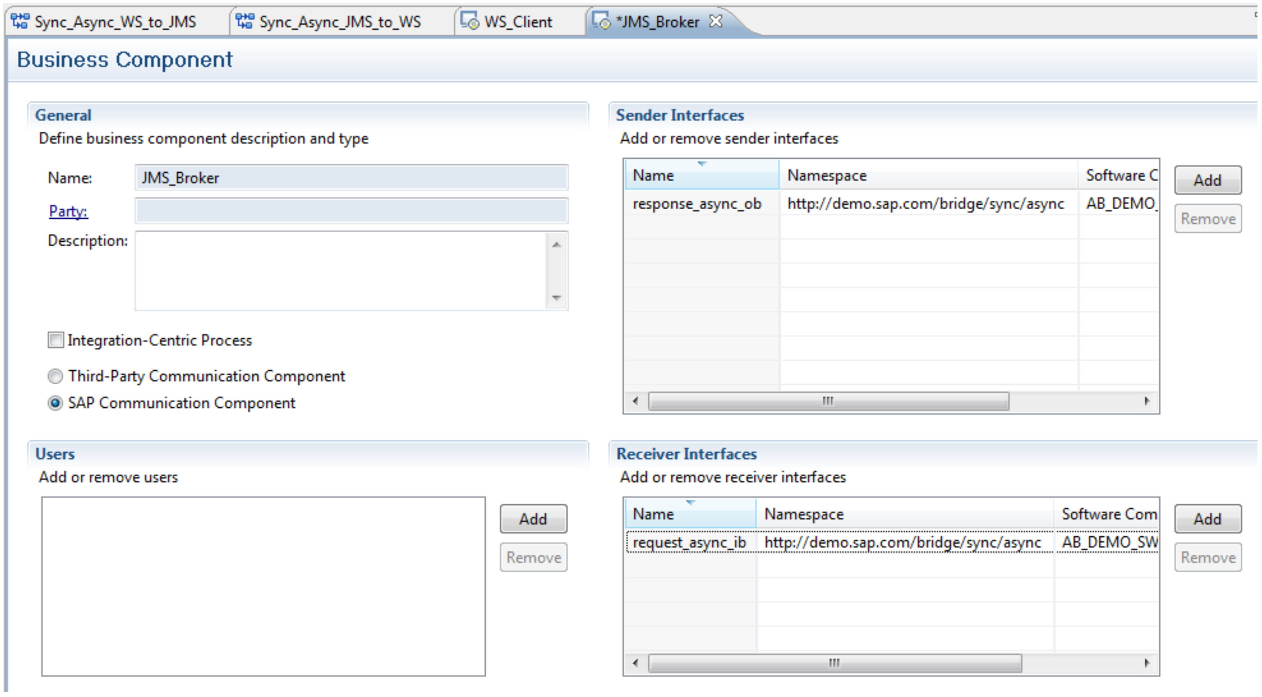
Sender Interfaces
Add or remove sender interfaces

Name	Namespace	Software Comp...
ws_sync_ob	http://demo.sap.com/bridge/sync/async	AB_DEMO_SWC...

Receiver Interfaces
Add or remove receiver interfaces

Name	Namespace	Software Comp...
response_async_ib	http://demo.sap.com/bridge/sync/async	AB_DEMO_SWC...

Create a Business Component for the JMS provider, here **JMS_Broker**, and add sender interface **response_async_ob** and receiver interface **request_async_ib** to the same.



Business Component

General
Define business component description and type

Name:	JMS_Broker
Party:	
Description:	

☐ Integration-Centric Process

○ Third-Party Communication Component

● SAP Communication Component

Users
Add or remove users

Sender Interfaces
Add or remove sender interfaces

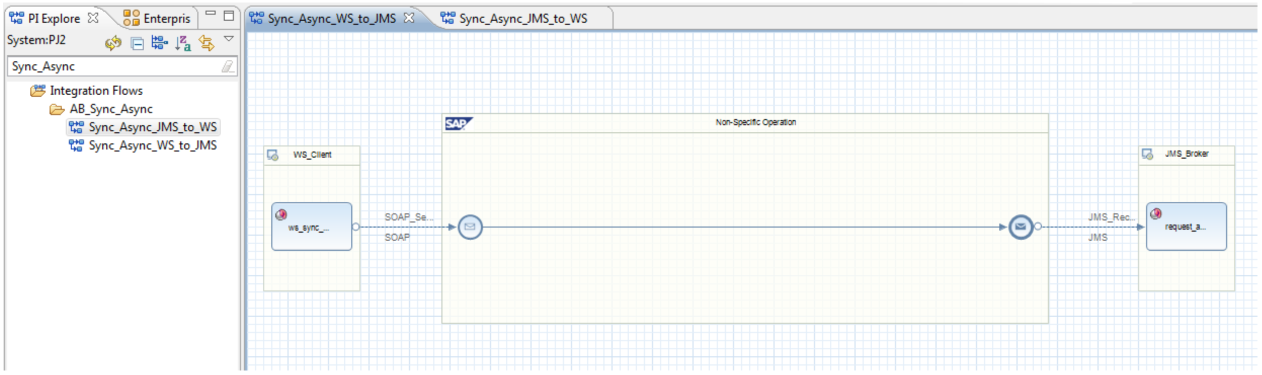
Name	Namespace	Software C
response_async_ob	http://demo.sap.com/bridge/sync/async	AB_DEMO_

Receiver Interfaces
Add or remove receiver interfaces

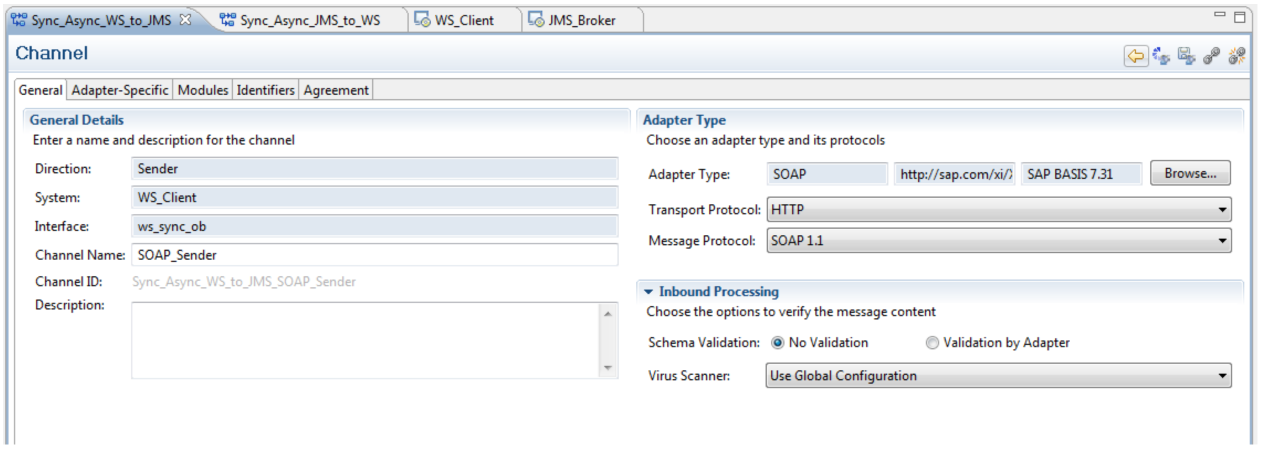
Name	Namespace	Software Com
request_async_ib	http://demo.sap.com/bridge/sync/async	AB_DEMO_SW

Create a new Integration Flow describing the routing from the Web Service client to the JMS broker.

- Assign Business Component **WS_Client** and the sender interface **ws_sync_ob** to the sender of the Integration Flow
- Assign Business Component **JMS_Broker** and the receiver interface **request_async_ib** to the receiver of the Integration Flow
- Maintain sender and receiver channel as described further below



Maintain sender channel of adapter type *SOAP*.



General Details

Enter a name and description for the channel

Direction: Sender
System: WS_Client
Interface: ws_sync_ob
Channel Name: SOAP_Sender
Channel ID: Sync_Async_WS_to_JMS_SOAP_Sender
Description:

Adapter Type

Choose an adapter type and its protocols

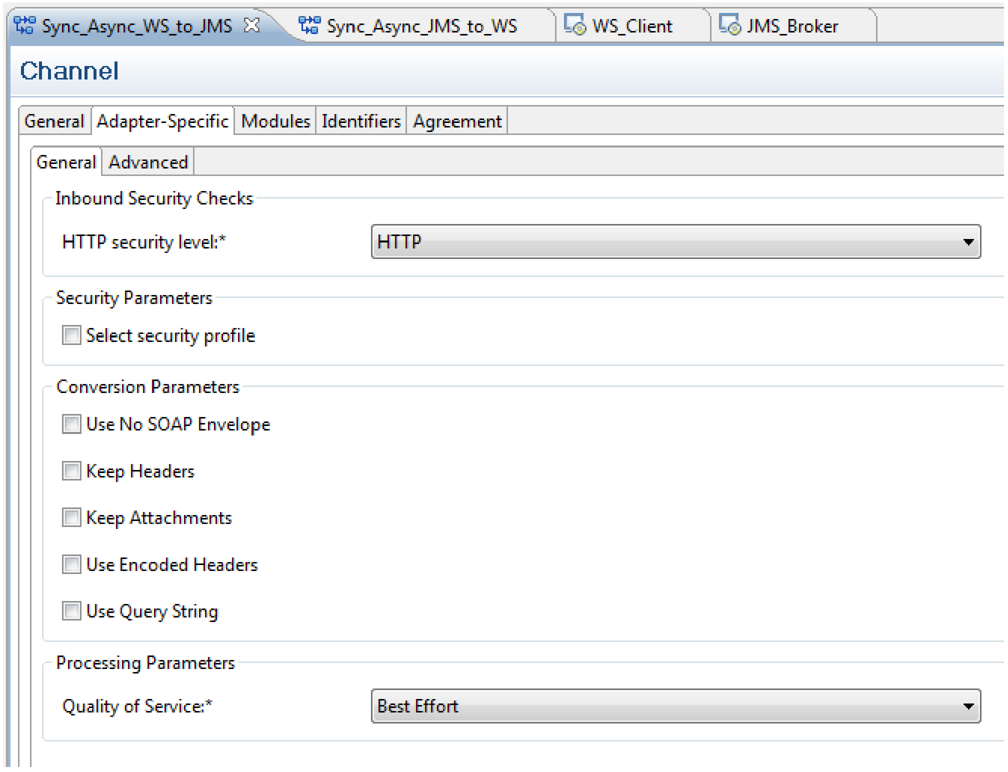
Adapter Type: SOAP http://sap.com/xi/) SAP BASIS 7.31
Transport Protocol: HTTP
Message Protocol: SOAP 1.1

Inbound Processing

Choose the options to verify the message content

Schema Validation: [✓ No Validation] Validation by Adapter
Virus Scanner: Use Global Configuration

Switch to tab *Adapter-Specific*, and set the *Quality of Service* to *Best Effort*.



Inbound Security Checks

HTTP security level:* HTTP

Security Parameters

☐ Select security profile

Conversion Parameters

☐ Use No SOAP Envelope
☐ Keep Headers
☐ Keep Attachments
☐ Use Encoded Headers
☐ Use Query String

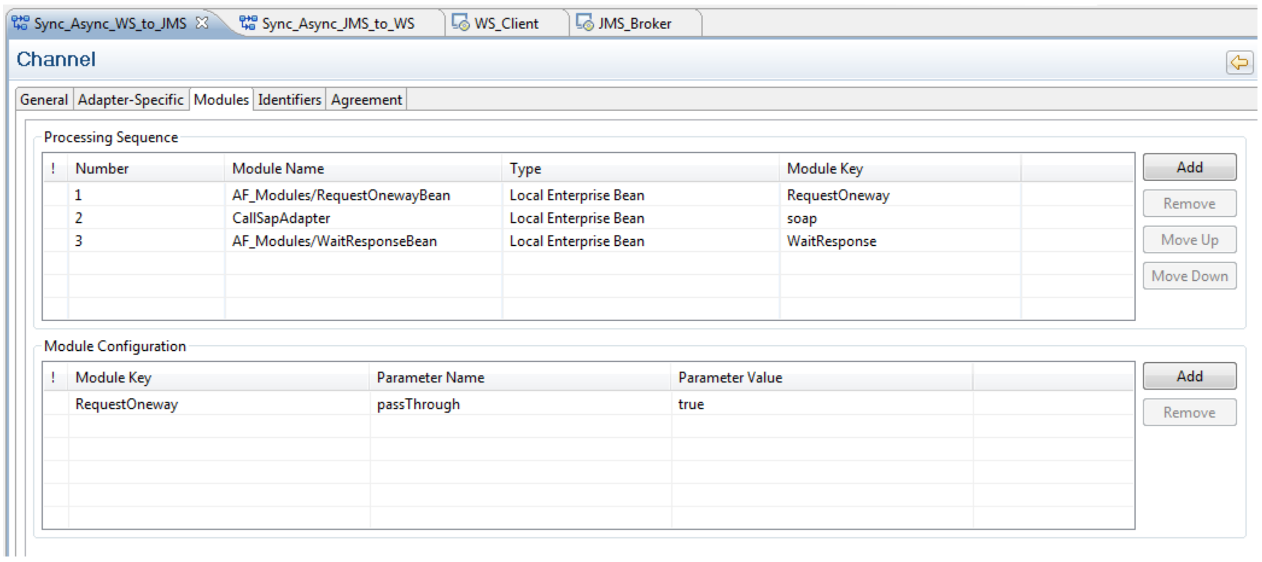
Processing Parameters

Quality of Service:* Best Effort

The conversion from synchronous to asynchronous communication will happen in the SOAP adapter, so we have to add the modules here. Switch to tab *Modules*, and add two new modules.

Add module *AF_Modules/RequestOnewayBean* at the beginning of the module chain. Maintain Parameter *passThrough* with **true**. The synchronous request message is converted to an asynchronous request message, and passed to the next module in sequence, i.e., the standard module calling the SOAP adapter.

Add module *AF_Modules/WaitResponseBean* at the end of the module chain. The module waits for a response message.



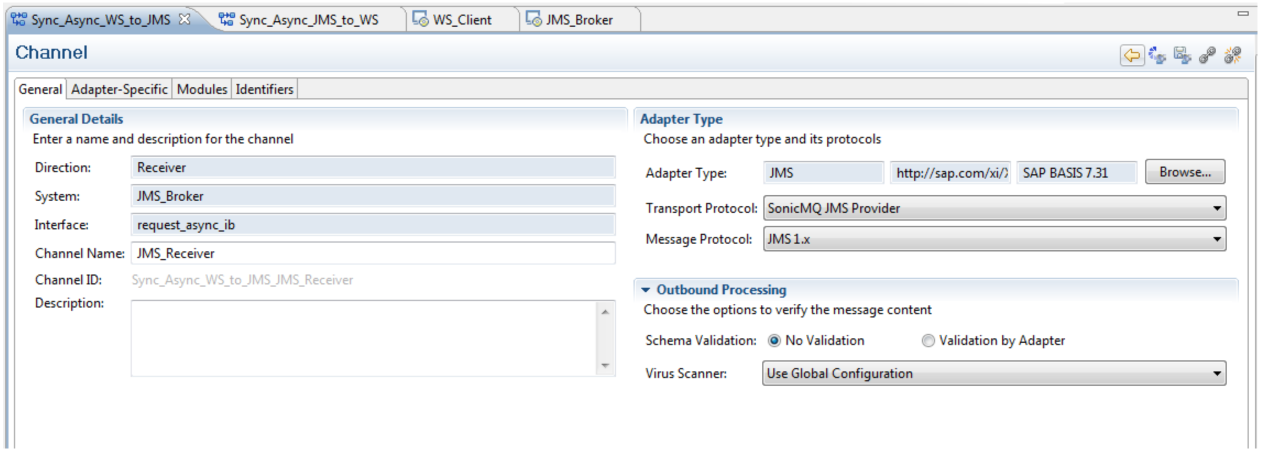
Processing Sequence

!	Number	Module Name	Type	Module Key
	1	AF_Modules/RequestOnewayBean	Local Enterprise Bean	RequestOneway
	2	CallSapAdapter	Local Enterprise Bean	soap
	3	AF_Modules/WaitResponseBean	Local Enterprise Bean	WaitResponse

Module Configuration

!	Module Key	Parameter Name	Parameter Value
	RequestOneway	passThrough	true

Maintain receiver channel of adapter type *JMS*.



General Details
Enter a name and description for the channel

Direction:	Receiver
System:	JMS_Broker
Interface:	request_async_ib
Channel Name:	JMS_Receiver
Channel ID:	Sync_Async_WS_to_JMS_JMS_Receiver
Description:	

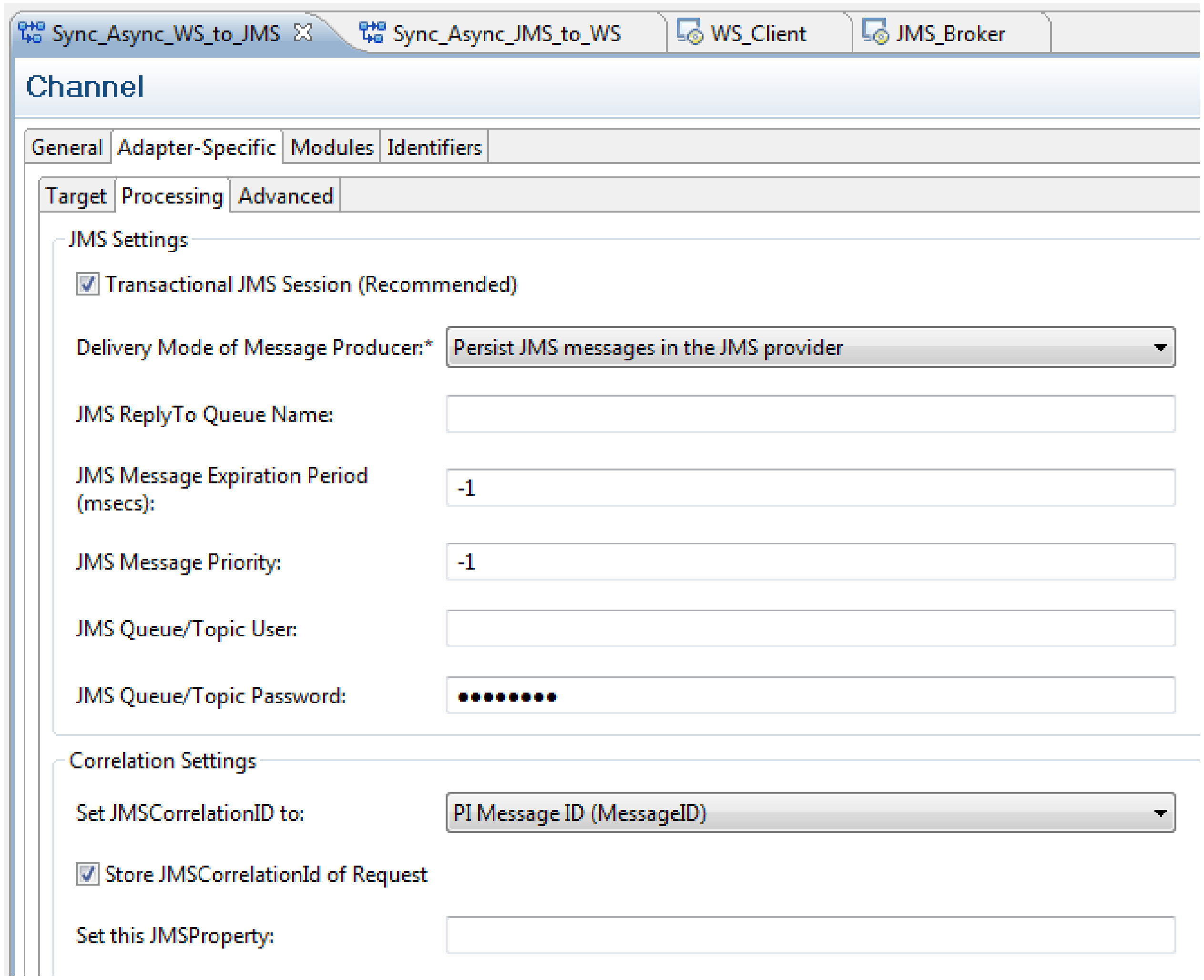
Adapter Type
Choose an adapter type and its protocols

Adapter Type:	JMS · http://sap.com/xi/› · SAP BASIS 7.31
Transport Protocol:	SonicMQ JMS Provider
Message Protocol:	JMS 1.x

Outbound Processing
Choose the options to verify the message content

Schema Validation:	[✓ No Validation] Validation by Adapter
Virus Scanner:	Use Global Configuration

Switch to tab *Adapter-Specific*, sub tab *Processing*, and set the *JMS Correlation ID* to the *PI Message ID*. Select the *Store JMS Correlation ID of request* flag in order to save the *JMS Correlation ID* of the request message.



JMS Settings

- [✓] Transactional JMS Session (Recommended)

Delivery Mode of Message Producer:*	Persist JMS messages in the JMS provider
JMS ReplyTo Queue Name:	
JMS Message Expiration Period (msecs):	-1
JMS Message Priority:	-1
JMS Queue/Topic User:	
JMS Queue/Topic Password:	●●●●●●●●●

Correlation Settings

Set JMSCorrelationID to:	PI Message ID (MessageID)

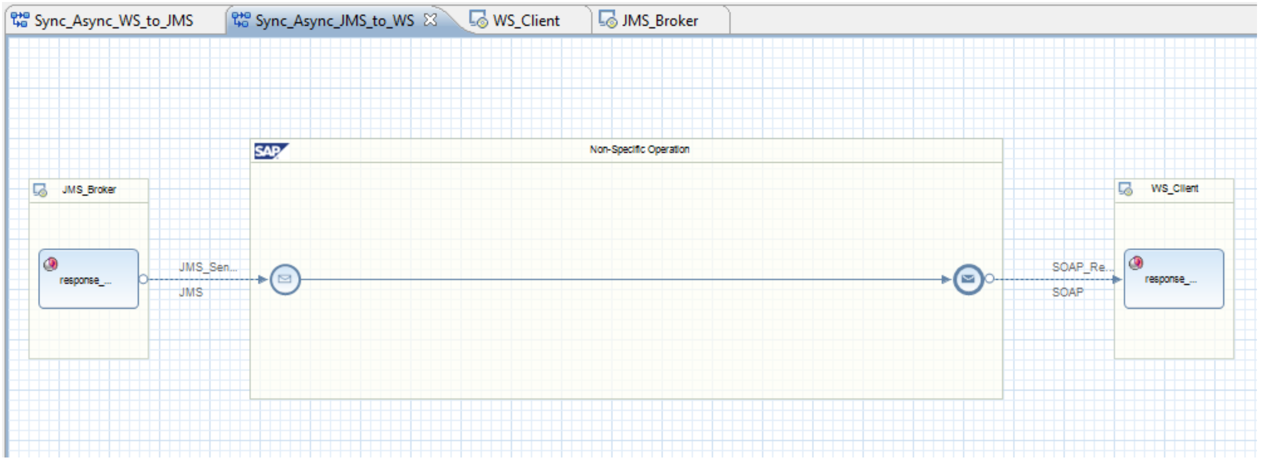
- [✓] Store JMSCorrelationId of Request

Set this JMSProperty:	

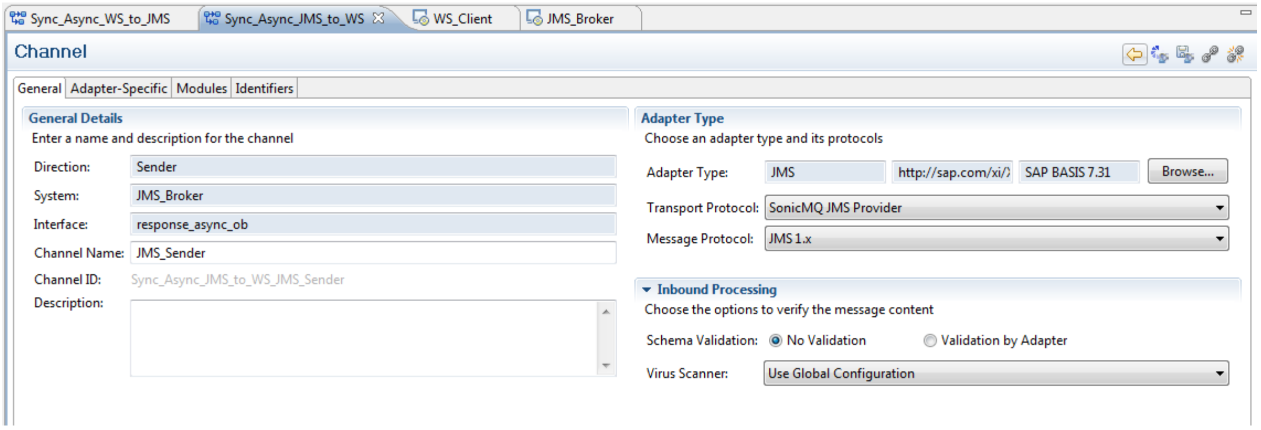
Integration Flow from JMS Broker to Web Service client

Create another Integration Flow describing the routing of the response from the JMS Broker to the Web Service client.

- Assign Business Component **JMS_Broker** and the sender interface **response_async_ob** to the sender of the Integration Flow
- Assign Business Component **WS_Client** and the receiver interface **response_async_ib** to the receiver of the Integration Flow
- Maintain sender and receiver channel as described further below



Maintain sender channel of adapter type *JMS*.



Channel

General Details

Enter a name and description for the channel

Direction:	Sender
System:	JMS_Broker
Interface:	response_async_ob
Channel Name:	JMS_Sender
Channel ID:	Sync_Async_JMS_to_WS_JMS_Sender
Description:	

Adapter Type

Choose an adapter type and its protocols

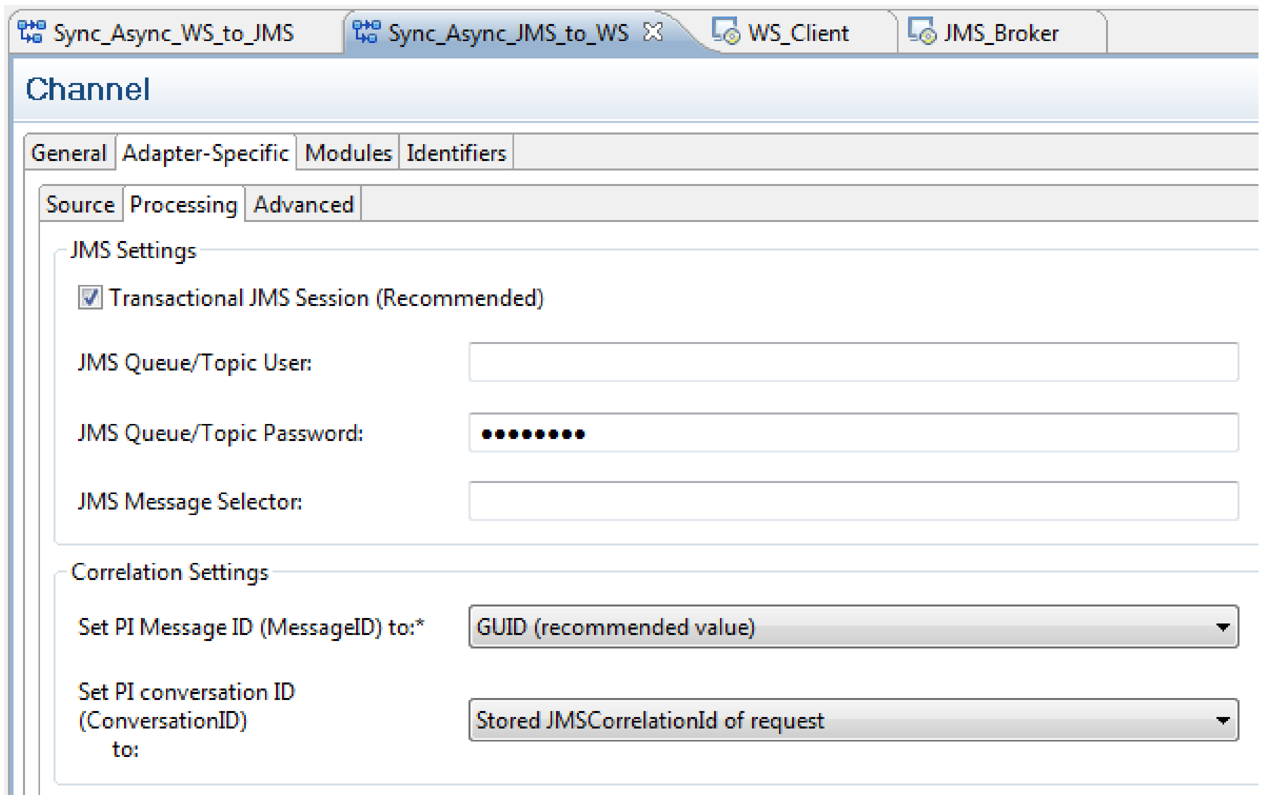
Adapter Type:	JMS http://sap.com/xi/X SAP BASIS 7.31
Transport Protocol:	SonicMQ JMS Provider
Message Protocol:	JMS 1.x

Inbound Processing

Choose the options to verify the message content

Schema Validation:	[✓ No Validation] Validation by Adapter
Virus Scanner:	Use Global Configuration

Switch to tab *Adapter-Specific*, sub tab *Processing*, and set the *PI Conversation ID* to the Stored *JMS Correlation ID of request*.



Channel

JMS Settings

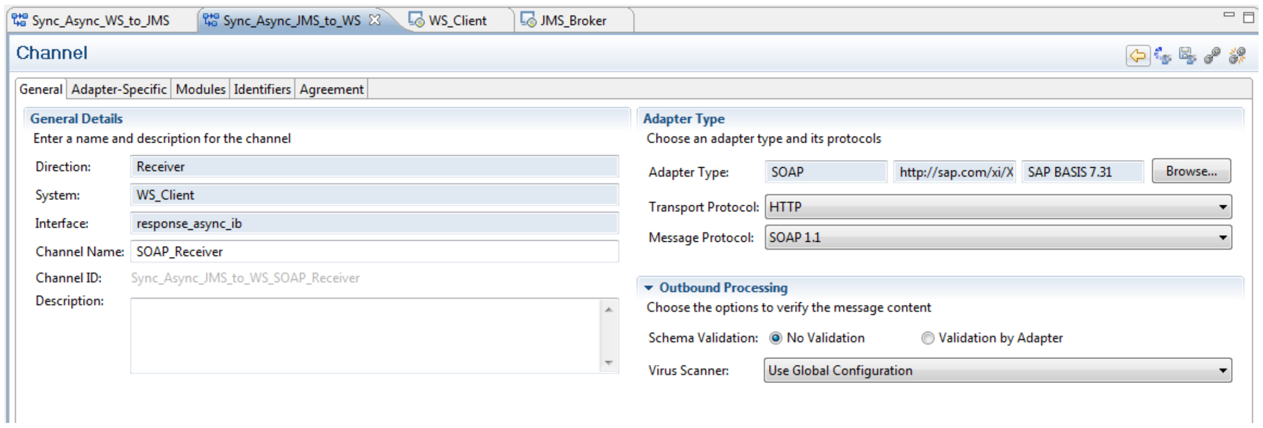
✓ Transactional JMS Session (Recommended)

JMS Queue/Topic User:	
JMS Queue/Topic Password:	●●●●●●●●●
JMS Message Selector:	

Correlation Settings

Set PI Message ID (MessageID) to:*	GUID (recommended value)
Set PI conversation ID (ConversationID) to:	Stored JMSCorrelationId of request

Maintain receiver channel of adapter type *SOAP*.



Channel

General Details

Enter a name and description for the channel

Direction:	Receiver
System:	WS_Client
Interface:	response_async_ib
Channel Name:	SOAP_Receiver
Channel ID:	Sync_Async_JMS_to_WS_SOAP_Receiver
Description:	

Adapter Type

Choose an adapter type and its protocols

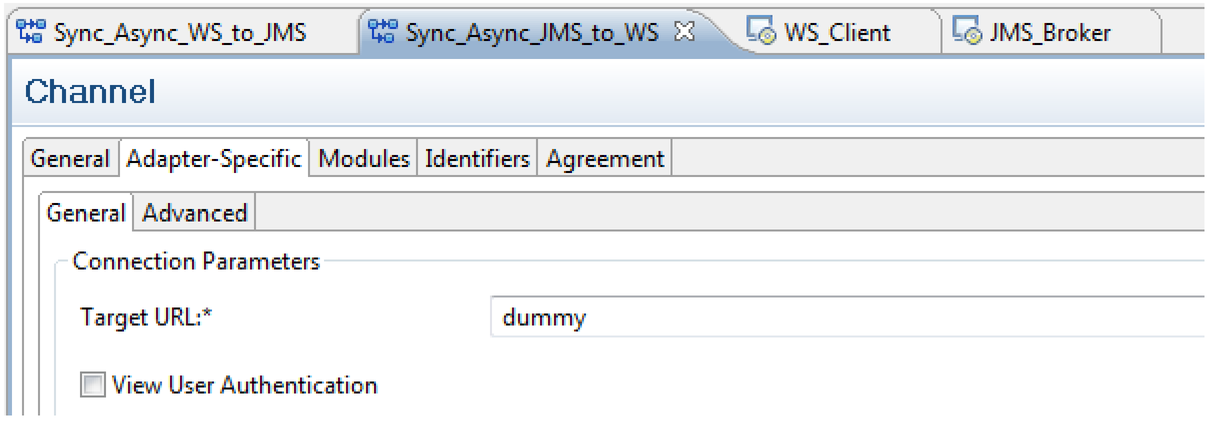
Adapter Type:	SOAP http://sap.com/xi/X SAP BASIS 7.31
Transport Protocol:	HTTP
Message Protocol:	SOAP 1.1

Outbound Processing

Choose the options to verify the message content

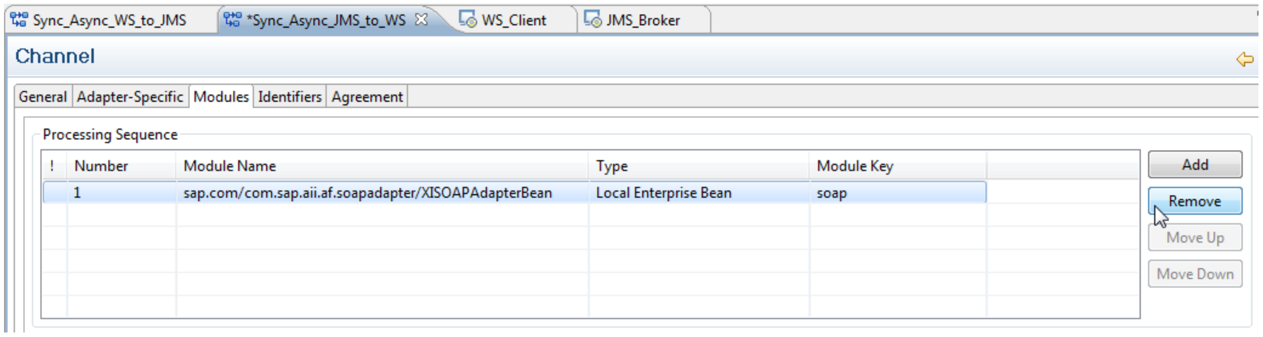
Schema Validation:	[✓ No Validation] Validation by Adapter
Virus Scanner:	Use Global Configuration

Switch to tab *Adapter-Specific*, and maintain any dummy target URL. We do not intend to call any Web Service here, instead will use this channel to pass the response message to the waiting SOAP sender adapter of the first Integration Flow defined beforehand. For this reason we also need to remove the standard module calling the SOAP adapter.



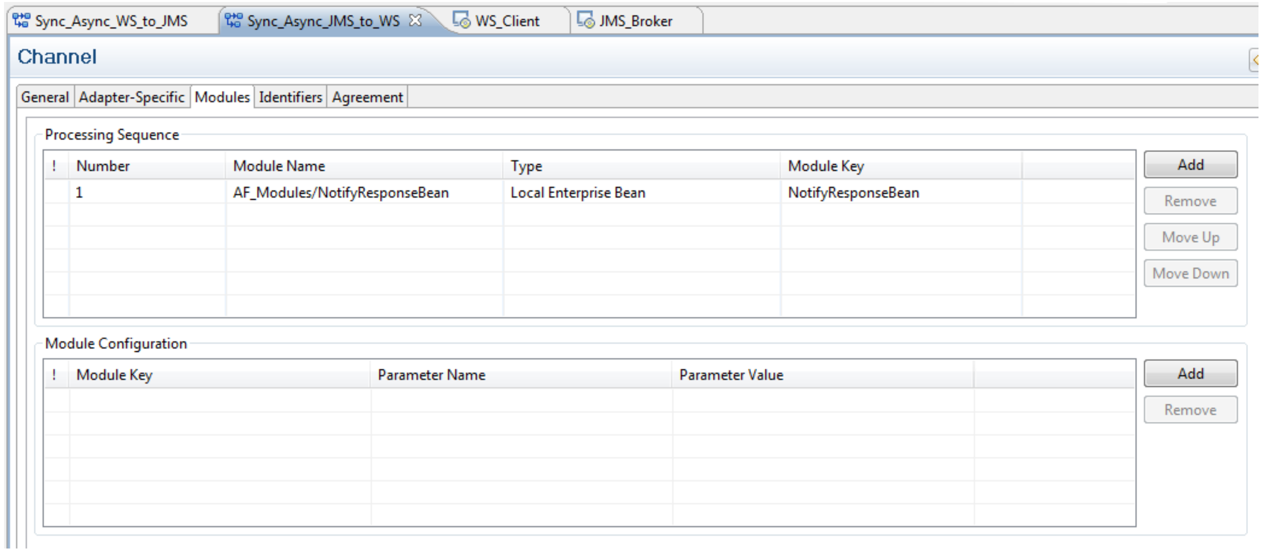
Target URL:* dummy

Switch to tab *Modules*. Pick the standard module that calls the adapter, and select *Remove*.



!	Number	Module Name	Type	Module Key
	1	sap.com/com.sap.aii.af.soapadapter/XISOAPAdapterBean	Local Enterprise Bean	soap

Add adapter module *AF_Modules/NotifyResponseBean*. The asynchronous response message is passed to the *AF_Modules/WaitResponseBean* of the SOAP sender channel of the first Integration Flow created beforehand.



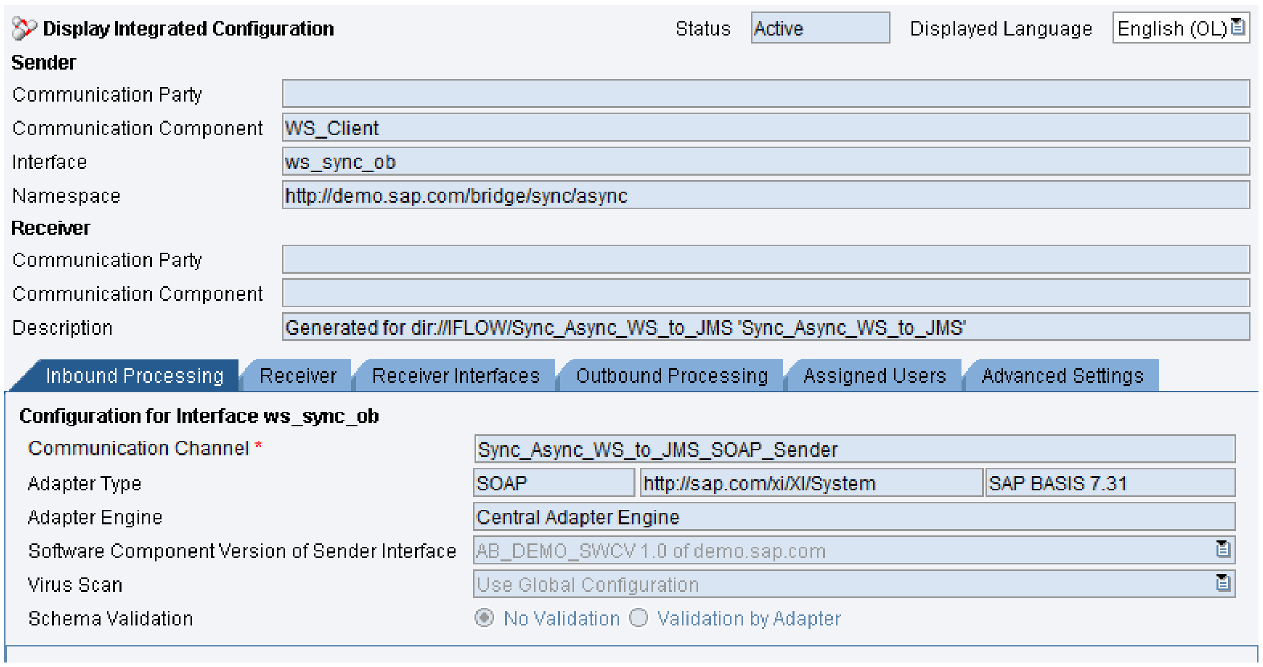
!	Number	Module Name	Type	Module Key
	1	AF_Modules/NotifyResponseBean	Local Enterprise Bean	NotifyResponseBean

!	Module Key	Parameter Name	Parameter Value

Integrated Configuration Objects in Integration Directory

Let's take a look at the objects which have been created in the Integration Directory. For each Integration Flow, an Integrated Configuration Object (ICO) and two channels have been created. From the description you can see that the ICO has been generated based on an Integration Flow.

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



Display Integrated Configuration Status: Active Displayed Language: English (OL)

Sender

Communication Party	
Communication Component	WS_Client
Interface	ws_sync_ob
Namespace	http://demo.sap.com/bridge/sync/async

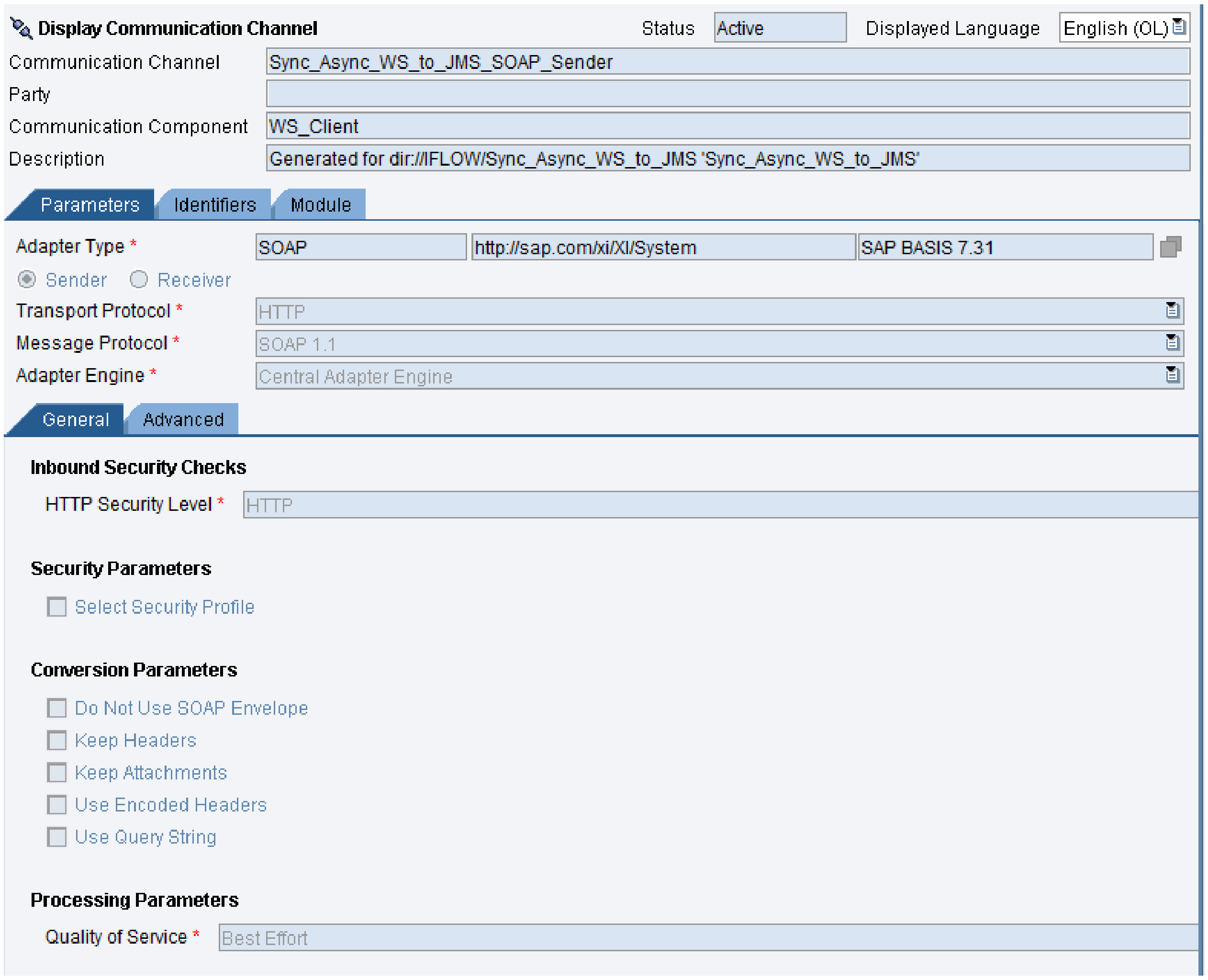
Receiver

Communication Party	
Communication Component	
Description	Generated for dir://IFLOW/Sync_Async_WS_to_JMS 'Sync_Async_WS_to_JMS'

Configuration for Interface ws_sync_ob

Communication Channel *	Sync_Async_WS_to_JMS_SOAP_Sender
Adapter Type	SOAP · http://sap.com/xi/XI/System · SAP BASIS 7.31
Adapter Engine	Central Adapter Engine
Software Component Version of Sender Interface	AB_DEMO_SWCV 1.0 of demo.sap.com
Virus Scan	Use Global Configuration
Schema Validation	[✓ No Validation] Validation by Adapter

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



Display Communication Channel

Status	Active
Displayed Language	English (OL)
Communication Channel	Sync_Async_WS_to_JMS_SOAP_Sender
Party	
Communication Component	WS_Client
Description	Generated for dir://IFLOW/Sync_Async_WS_to_JMS 'Sync_Async_WS_to_JMS'

Adapter Type *	SOAP / http://sap.com/xi/XI/System / SAP BASIS 7.31
	[✓ Sender] Receiver
Transport Protocol *	HTTP
Message Protocol *	SOAP 1.1
Adapter Engine *	Central Adapter Engine

Inbound Security Checks

HTTP Security Level * HTTP

Security Parameters

☐ Select Security Profile

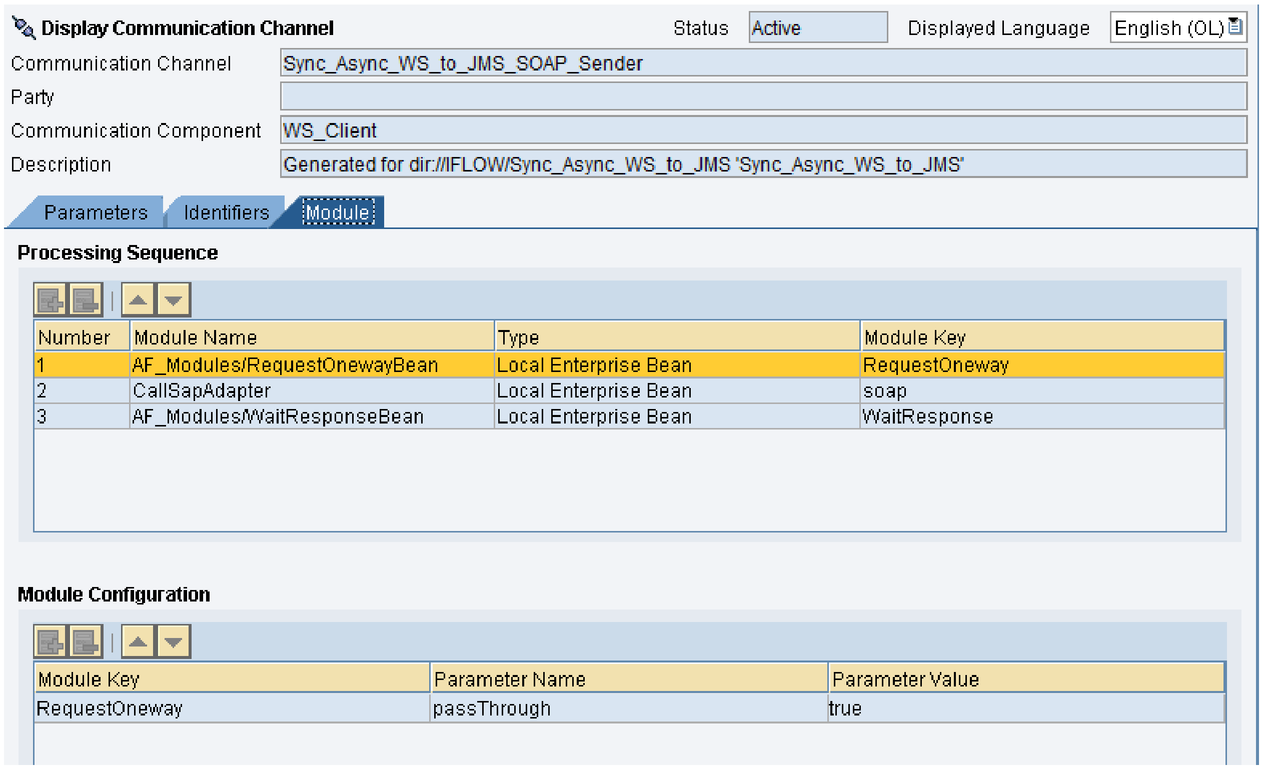
Conversion Parameters

☐ Do Not Use SOAP Envelope
☐ Keep Headers
☐ Keep Attachments
☐ Use Encoded Headers
☐ Use Query String

Processing Parameters

Quality of Service * Best Effort

Switch to tab *Module*. The modules *AF_Modules/RequestOnewayBean* and *AF_Modules/WaitResponseBean* are defined in the right sequence. The parameter *passThrough* of the first module is set to *true*.



Display Communication Channel Status: Active Displayed Language: English (OL)

Communication Channel	Sync_Async_WS_to_JMS_SOAP_Sender
Party	
Communication Component	WS_Client
Description	Generated for dir://IFLOW/Sync_Async_WS_to_JMS 'Sync_Async_WS_to_JMS'

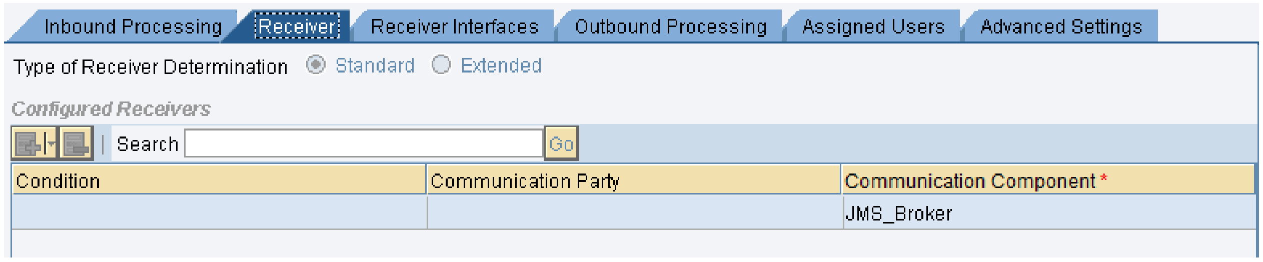
Processing Sequence

Number	Module Name	Type	Module Key
1	AF_Modules/RequestOnewayBean	Local Enterprise Bean	RequestOneway
2	CallSapAdapter	Local Enterprise Bean	soap
3	AF_Modules/WaitResponseBean	Local Enterprise Bean	WaitResponse

Module Configuration

Module Key	Parameter Name	Parameter Value
RequestOneway	passThrough	true

On tab *Receiver* of the ICO, the JMS provider is defined as receiver communication component.

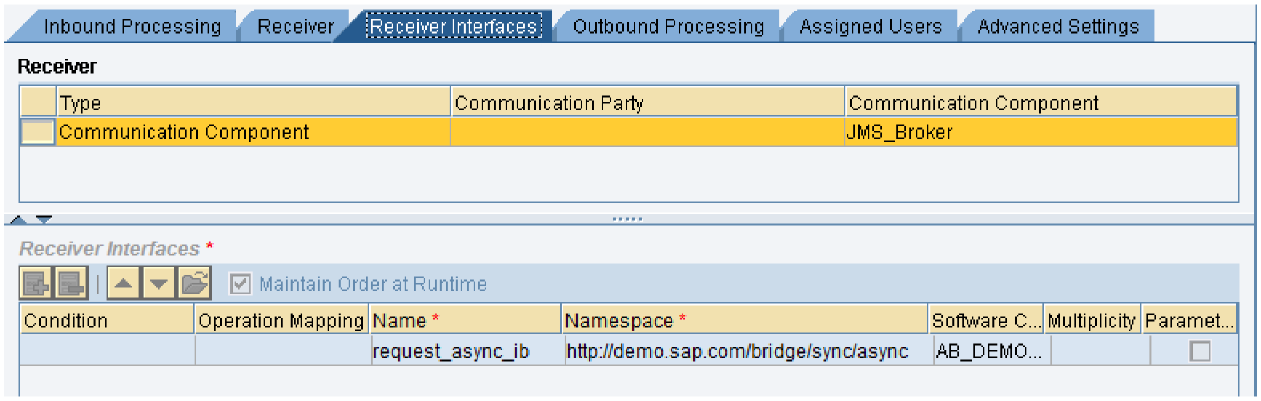


Type of Receiver Determination [✓ Standard] Extended

Configured Receivers

Condition	Communication Party	Communication Component *
		JMS_Broker

On tab *Receiver Interfaces*, the asynchronous request inbound interface is set.



Receiver

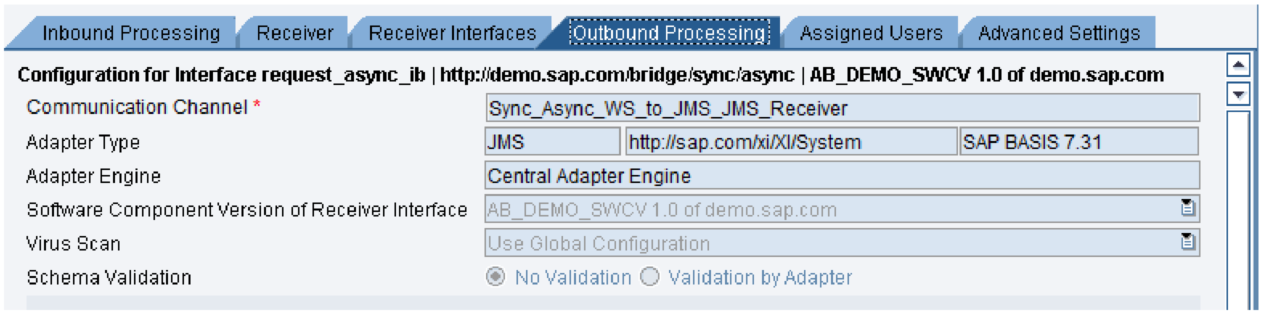
Type	Communication Party	Communication Component
Communication Component		JMS_Broker

Receiver Interfaces *

Maintain Order at Runtime [✓]

Condition	Operation Mapping	Name *	Namespace *	Software C...	Multiplicity	Paramet...
		request_async_ib	http://demo.sap.com/bridge/sync/async	AB_DEMO...		☐

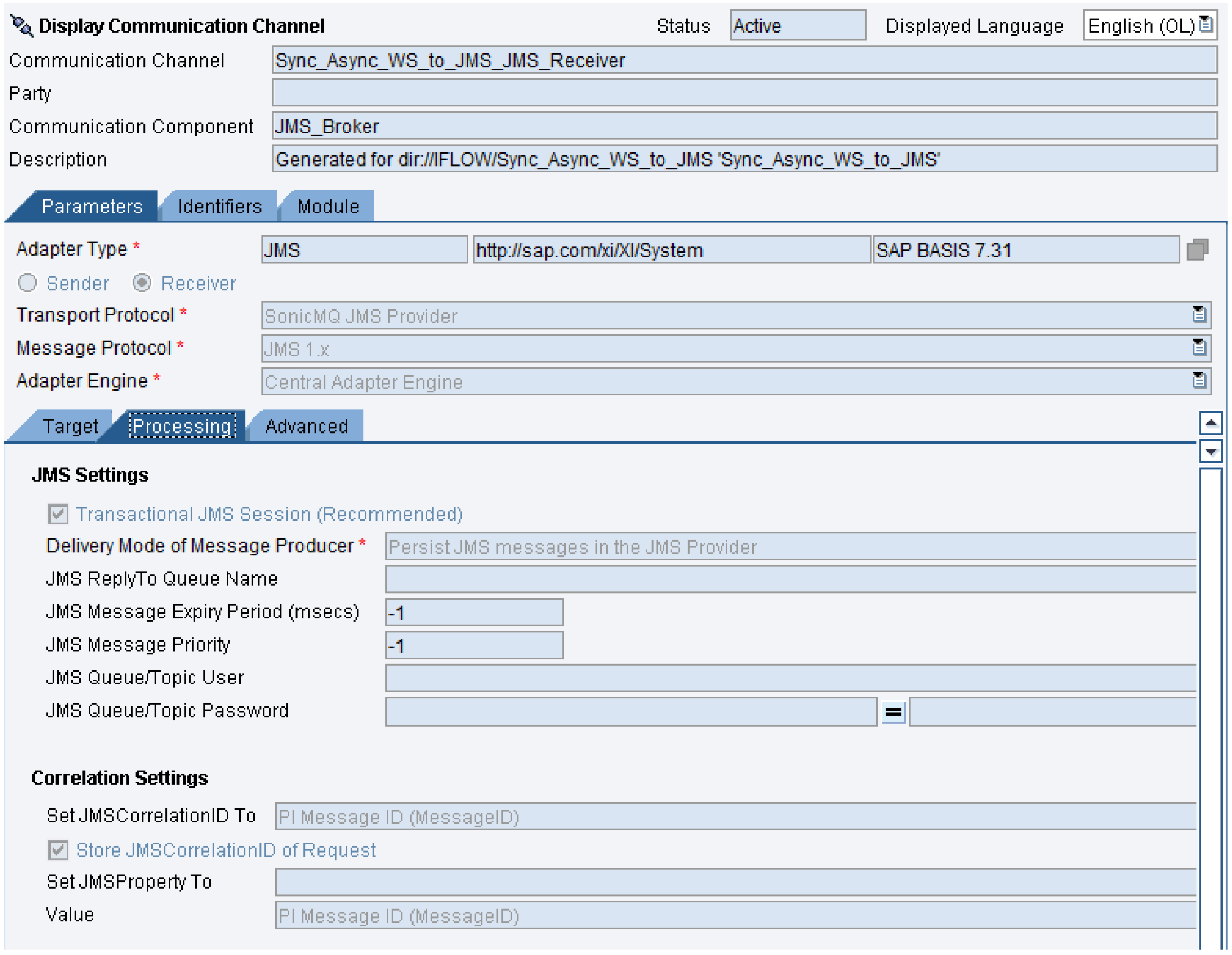
On tab *Outbound Processing*, the JMS receiver channel is specified.



Configuration for Interface request_async_ib | http://demo.sap.com/bridge/sync/async | AB_DEMO_SWCV 1.0 of demo.sap.com

Communication Channel *	Sync_Async_WS_to_JMS_JMS_Receiver
Adapter Type	JMS | http://sap.com/xi/XI/System | SAP BASIS 7.31
Adapter Engine	Central Adapter Engine
Software Component Version of Receiver Interface	AB_DEMO_SWCV 1.0 of demo.sap.com
Virus Scan	Use Global Configuration
Schema Validation	[✓ No Validation] Validation by Adapter

In the receiver channel of type JMS, the *JMS Correlation ID* is set to the *PI Message ID*. Furthermore, the *Store JMS Correlation ID of Request* flag is selected.



Status	Active
Displayed Language	English (OL)
Communication Channel	Sync_Async_WS_to_JMS_JMS_Receiver
Party	
Communication Component	JMS_Broker
Description	Generated for dir://IFLOW/Sync_Async_WS_to_JMS 'Sync_Async_WS_to_JMS'
Adapter Type *	JMS | http://sap.com/xi/XI/System | SAP BASIS 7.31
	Sender [✓ Receiver]
Transport Protocol *	SonicMQ JMS Provider
Message Protocol *	JMS 1.x
Adapter Engine *	Central Adapter Engine

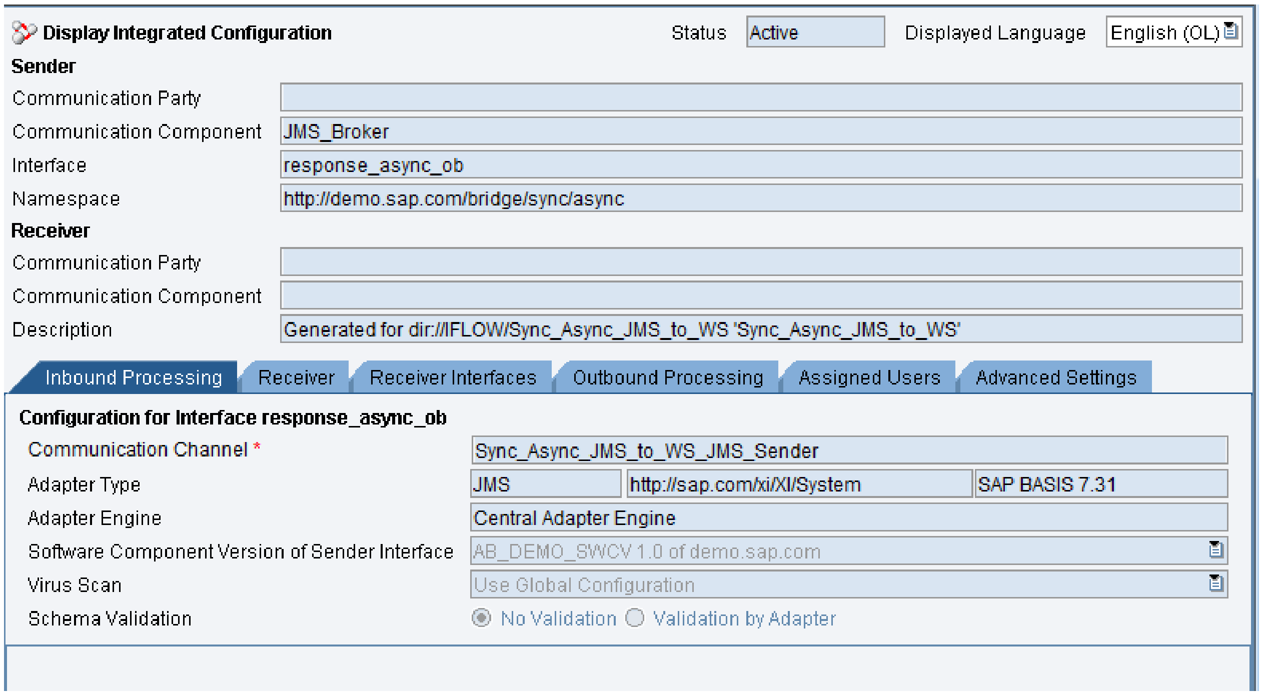
JMS Settings

✓ Transactional JMS Session (Recommended)	
Delivery Mode of Message Producer *	Persist JMS messages in the JMS Provider
JMS ReplyTo Queue Name	
JMS Message Expiry Period (msecs)	-1
JMS Message Priority	-1
JMS Queue/Topic User	
JMS Queue/Topic Password	

Correlation Settings

Set JMSCorrelationID To	PI Message ID (MessageID)
✓ Store JMSCorrelationID of Request	
Set JMSProperty To	
Value	PI Message ID (MessageID)

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



Display Integrated Configuration Status: Active Displayed Language: English (OL)

Sender

Communication Party	
Communication Component	JMS_Broker
Interface	response_async_ob
Namespace	http://demo.sap.com/bridge/sync/async

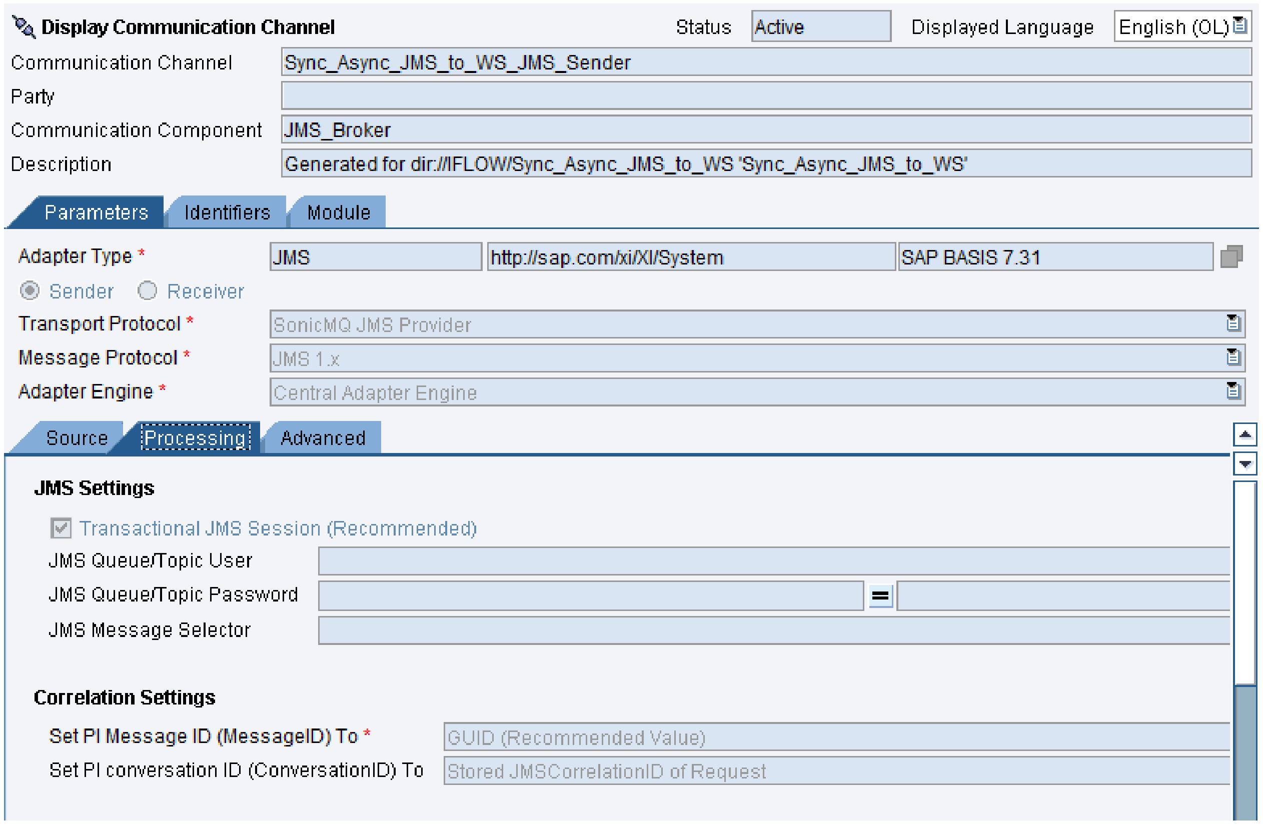
Receiver

Communication Party	
Communication Component	
Description	Generated for dir://IFLOW/Sync_Async_JMS_to_WS 'Sync_Async_JMS_to_WS'

Configuration for Interface response_async_ob

Communication Channel *	Sync_Async_JMS_to_WS_JMS_Sender
Adapter Type	JMS / http://sap.com/xi/XI/System / SAP BASIS 7.31
Adapter Engine	Central Adapter Engine
Software Component Version of Sender Interface	AB_DEMO_SWCV 1.0 of demo.sap.com
Virus Scan	Use Global Configuration
Schema Validation	[✓ No Validation] Validation by Adapter

In the JMS sender channel, the *PI Conversation ID* is set to the *Stored JMS Correlation ID of Request*.



Display Communication Channel Status: Active Displayed Language: English (OL)

Communication Channel	Sync_Async_JMS_to_WS_JMS_Sender
Party	
Communication Component	JMS_Broker
Description	Generated for dir://IFLOW/Sync_Async_JMS_to_WS 'Sync_Async_JMS_to_WS'

Adapter Type *	JMS / http://sap.com/xi/XI/System / SAP BASIS 7.31
	[✓ Sender] Receiver
Transport Protocol *	SonicMQ JMS Provider
Message Protocol *	JMS 1.x
Adapter Engine *	Central Adapter Engine

JMS Settings

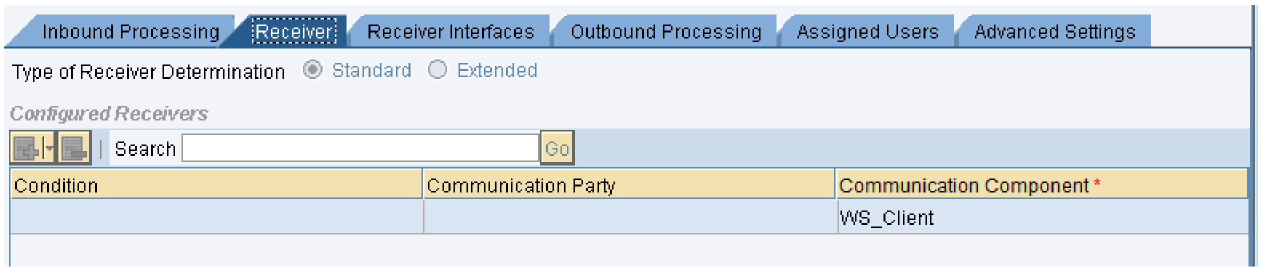
✓ Transactional JMS Session (Recommended)

JMS Queue/Topic User	
JMS Queue/Topic Password	
JMS Message Selector	

Correlation Settings

Set PI Message ID (MessageID) To *	GUID (Recommended Value)
Set PI conversation ID (ConversationID) To	Stored JMSCorrelationID of Request

On tab *Receiver* of the ICO, the Web Service client is defined as receiver.

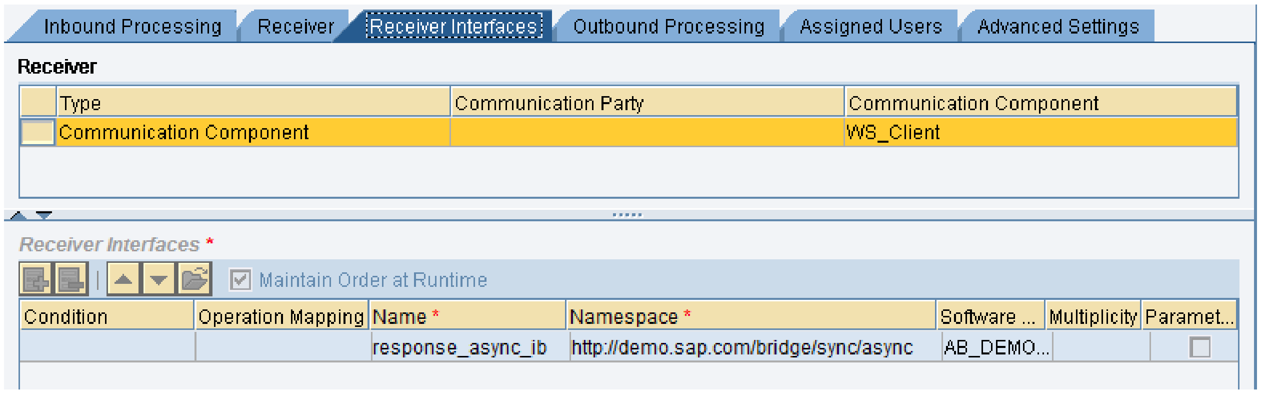


Type of Receiver Determination [✓ Standard] Extended

Configured Receivers

Condition	Communication Party	Communication Component *
		WS_Client

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



Receiver

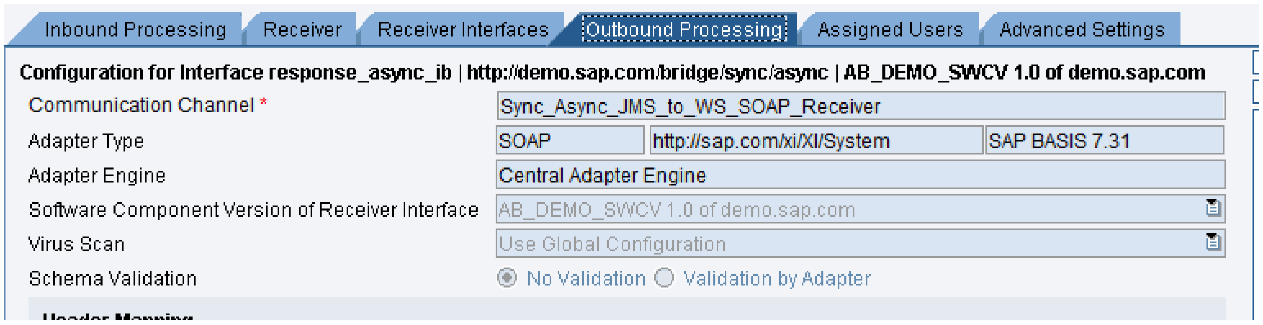
Type	Communication Party	Communication Component
Communication Component		WS_Client

Receiver Interfaces *

[✓ Maintain Order at Runtime]

Condition	Operation Mapping	Name *	Namespace *	Software ...	Multiplicity	Paramet...
		response_async_ib	http://demo.sap.com/bridge/sync/async	AB_DEMO...		☐

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

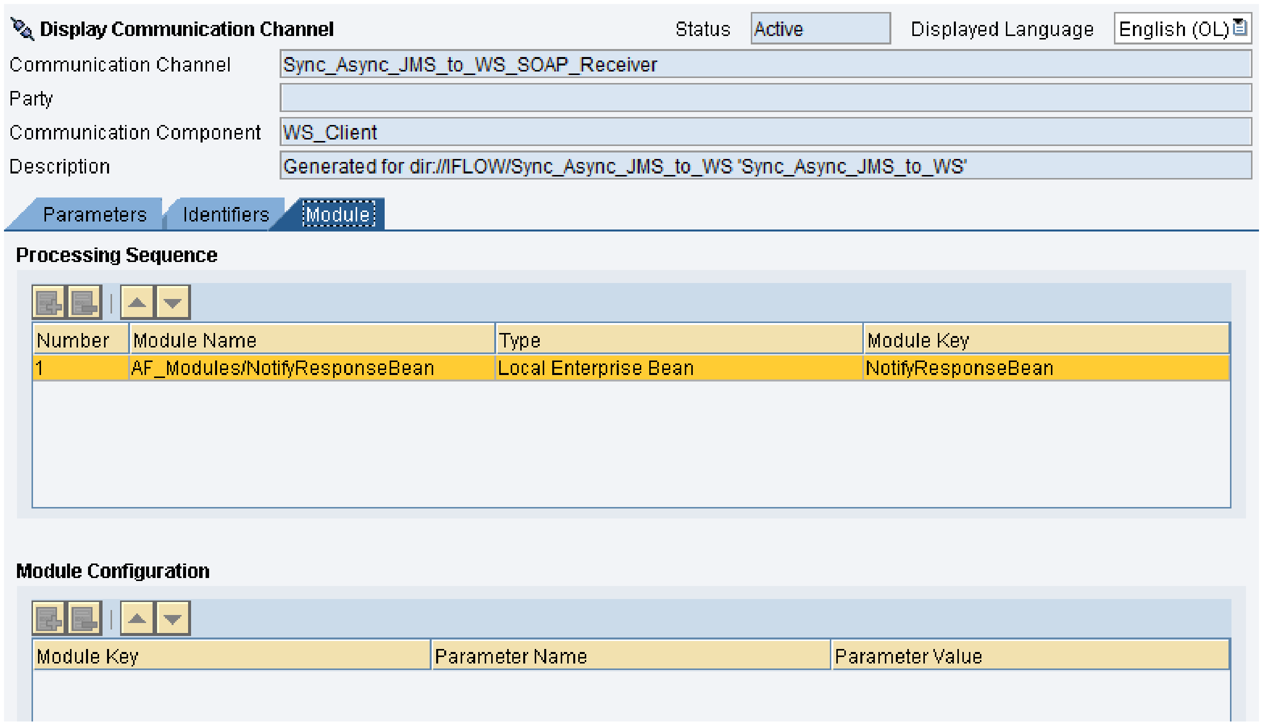


Configuration for Interface response_async_ib | http://demo.sap.com/bridge/sync/async | AB_DEMO_SWCV 1.0 of demo.sap.com

Communication Channel *	Sync_Async_JMS_to_WS_SOAP_Receiver
Adapter Type	SOAP | http://sap.com/xi/XI/System | SAP BASIS 7.31
Adapter Engine	Central Adapter Engine
Software Component Version of Receiver Interface	AB_DEMO_SWCV 1.0 of demo.sap.com
Virus Scan	Use Global Configuration
Schema Validation	[✓ No Validation] Validation by Adapter

Header Mapping

In the SOAP receiver channel, the module *AF_Modules/NotifyResponseBean* is defined. The standard adapter module has been removed.



Display Communication Channel Status Active Displayed Language English (OL)

Communication Channel	Sync_Async_JMS_to_WS_SOAP_Receiver
Party	
Communication Component	WS_Client
Description	Generated for dir://IFLOW/Sync_Async_JMS_to_WS 'Sync_Async_JMS_to_WS'

Processing Sequence

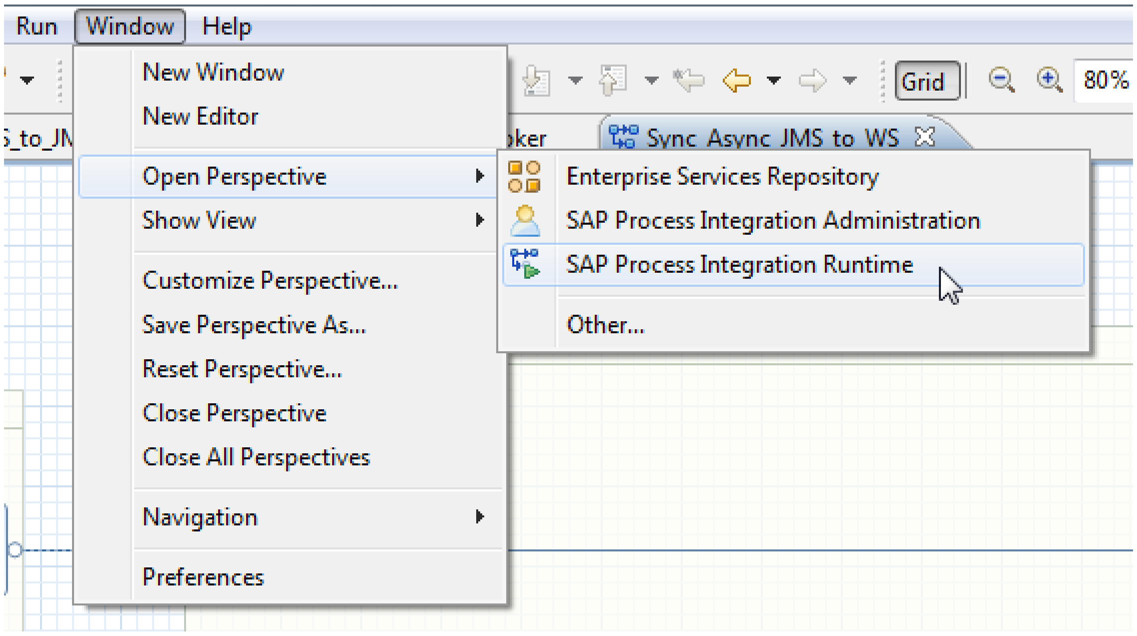
Number	Module Name	Type	Module Key
1	AF_Modules/NotifyResponseBean	Local Enterprise Bean	NotifyResponseBean

Module Configuration

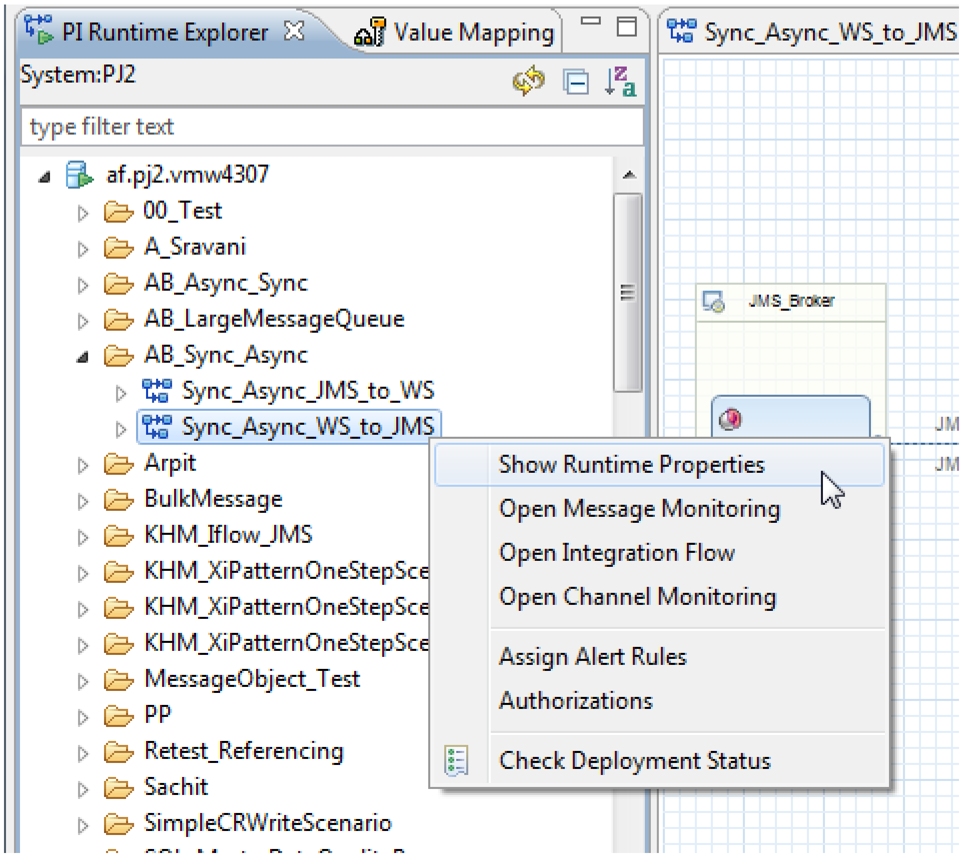
Module Key	Parameter Name	Parameter Value

Runtime

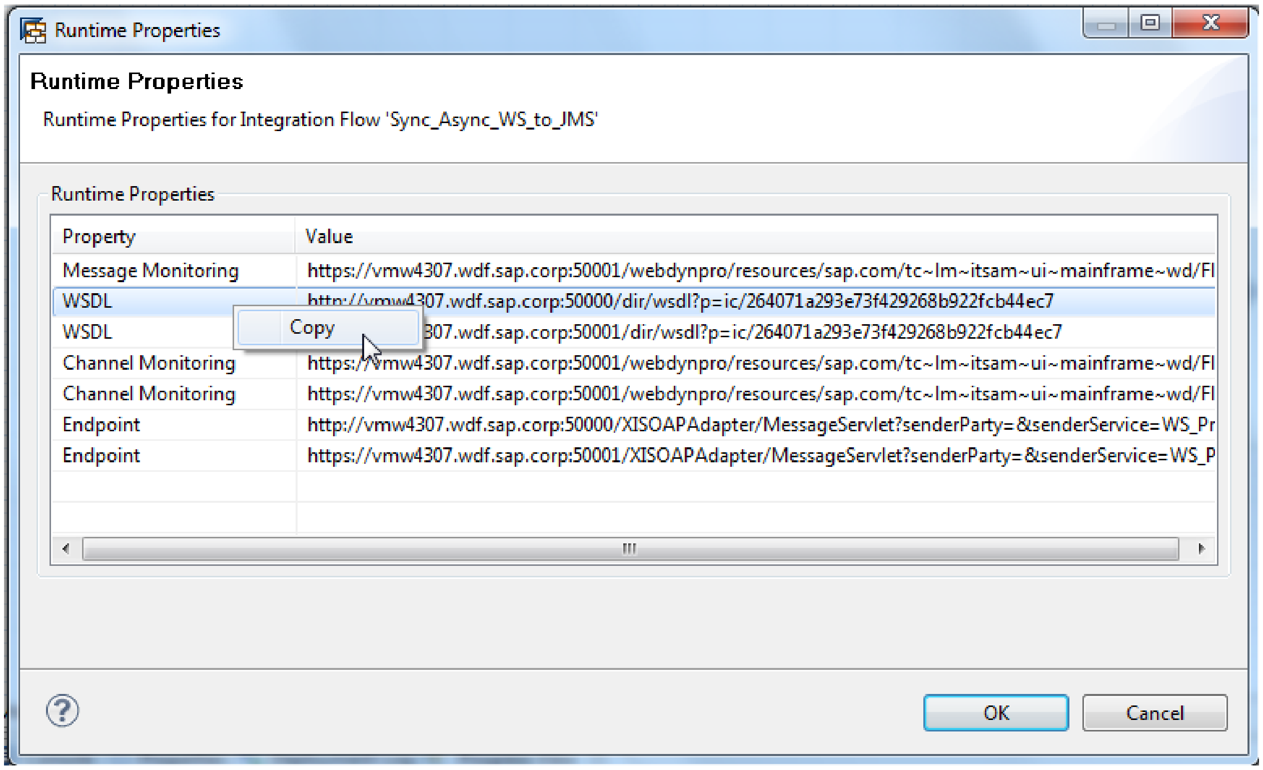
We will test the scenario in the *Web Services Navigator*. First, we need to get the WSDL URL of the first Integration Flow. In the NWDS, switch to the *SAP Process Integration Runtime* perspective by selecting *Window → Open Perspective → SAP Process Integration Runtime* from the main menu.



Pick the *Sync_Async_WS_to_JMS* Integration Flow, and select *Show Runtime Properties* from the context menu.



In the upcoming dialog, pick the *WSDL* property entry, and select *Copy* from the context menu to copy the WSDL URL into the clipboard.



Runtime Properties

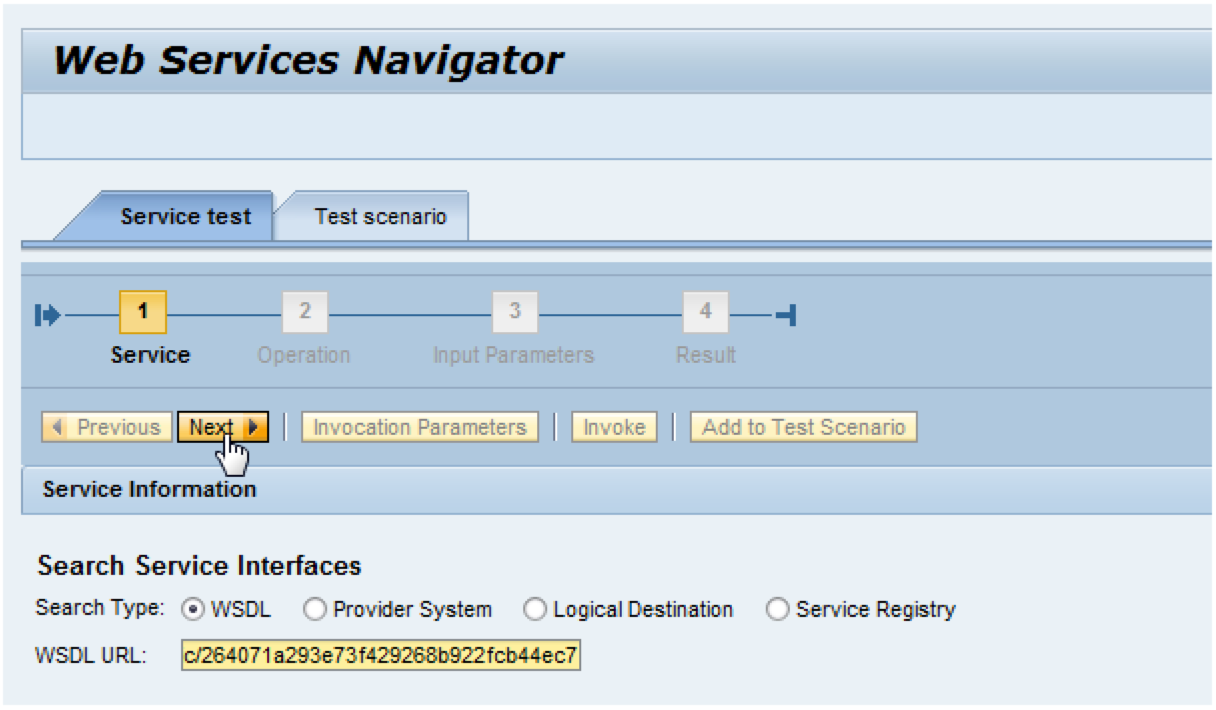
Runtime Properties

Runtime Properties for Integration Flow 'Sync_Async_WS_to_JMS'

Runtime Properties

Property	Value
Message Monitoring	https://vmw4307.wdf.sap.corp:50001/webdynpro/resources/sap.com/tc~lm~itsam~ui~mainframe~wd/Fl
WSDL	http://vmw4307.wdf.sap.corp:50000/dir/wsdl?p=ic/264071a293e73f429268b922fcb44ec7
WSDL	307.wdf.sap.corp:50001/dir/wsdl?p=ic/264071a293e73f429268b922fcb44ec7
Channel Monitoring	https://vmw4307.wdf.sap.corp:50001/webdynpro/resources/sap.com/tc~lm~itsam~ui~mainframe~wd/Fl
Channel Monitoring	https://vmw4307.wdf.sap.corp:50001/webdynpro/resources/sap.com/tc~lm~itsam~ui~mainframe~wd/Fl
Endpoint	http://vmw4307.wdf.sap.corp:50000/XISOAPAdapter/MessageServlet?senderParty=&senderService=WS_Pr
Endpoint	https://vmw4307.wdf.sap.corp:50001/XISOAPAdapter/MessageServlet?senderParty=&senderService=WS_P

Open the *Web Services Navigator* in your browser. Paste the copied WSDL location from the clipboard into the *WSDL URL* field, and click on *Next*.



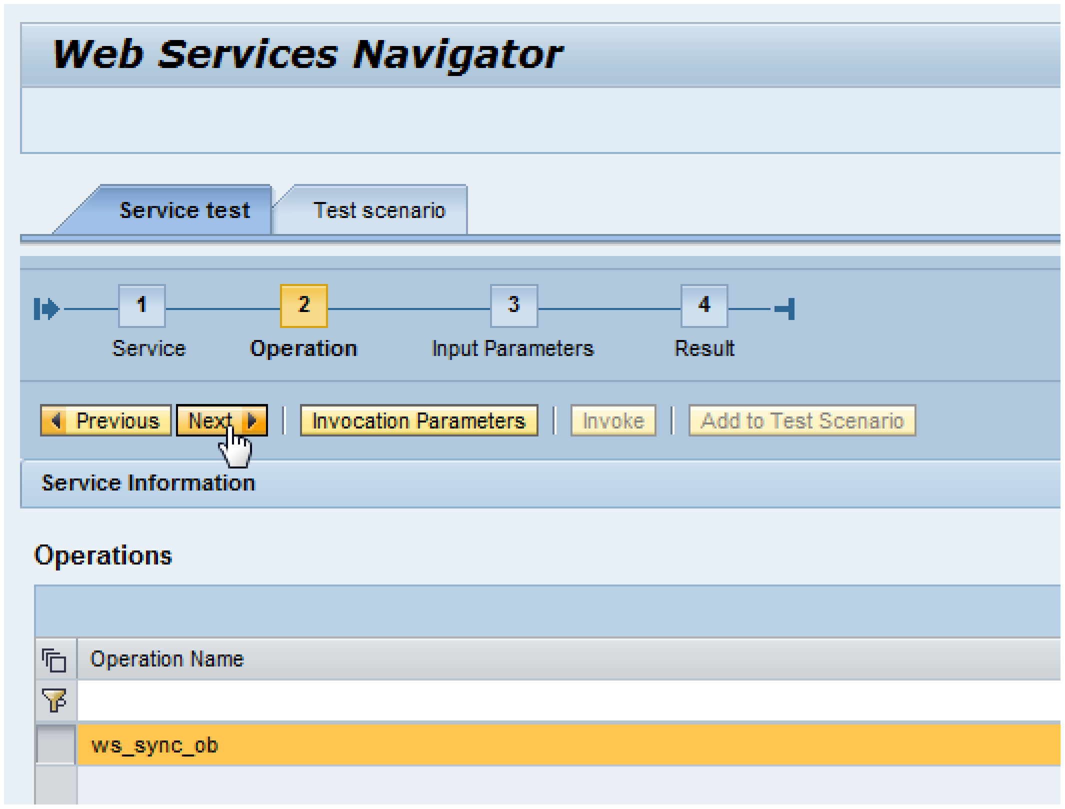
Web Services Navigator

Search Service Interfaces

Search Type: [✓ WSDL] Provider System Logical Destination Service Registry

WSDL URL: c/264071a293e73f429268b922fcb44ec7

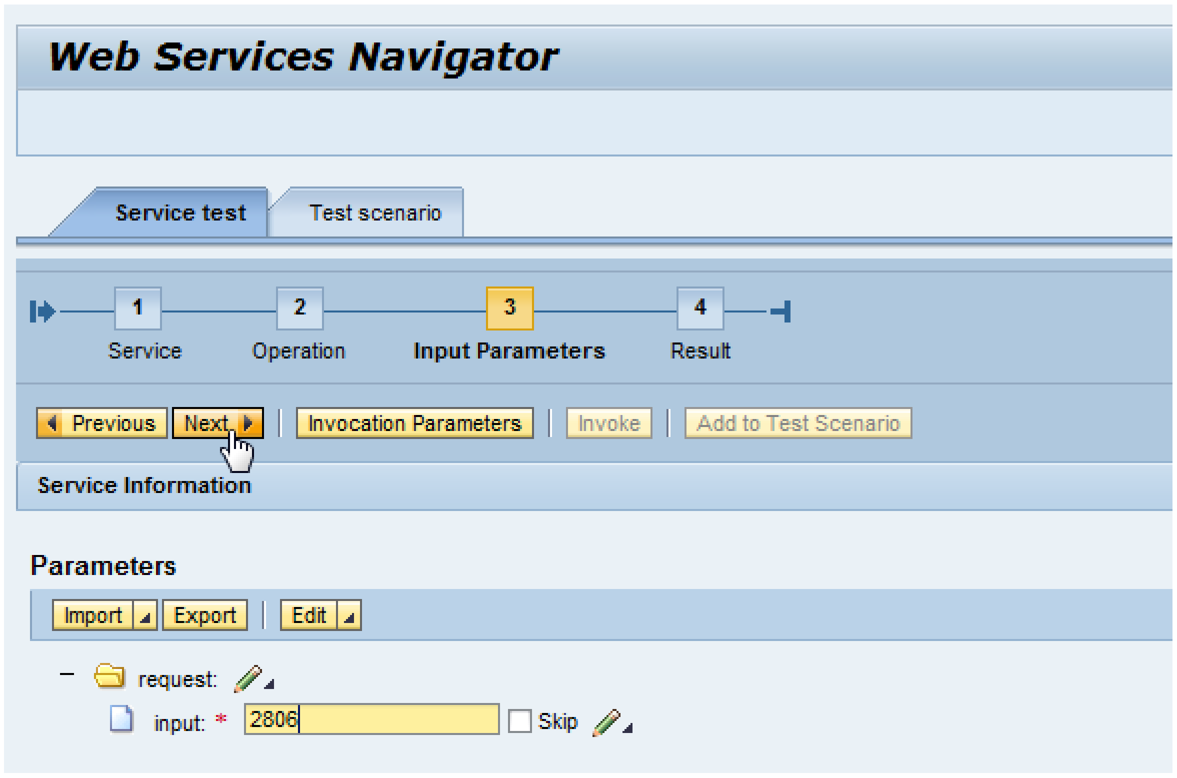
Select the operation, and click on *Next*.



Operations

Operation Name
ws_sync_ob

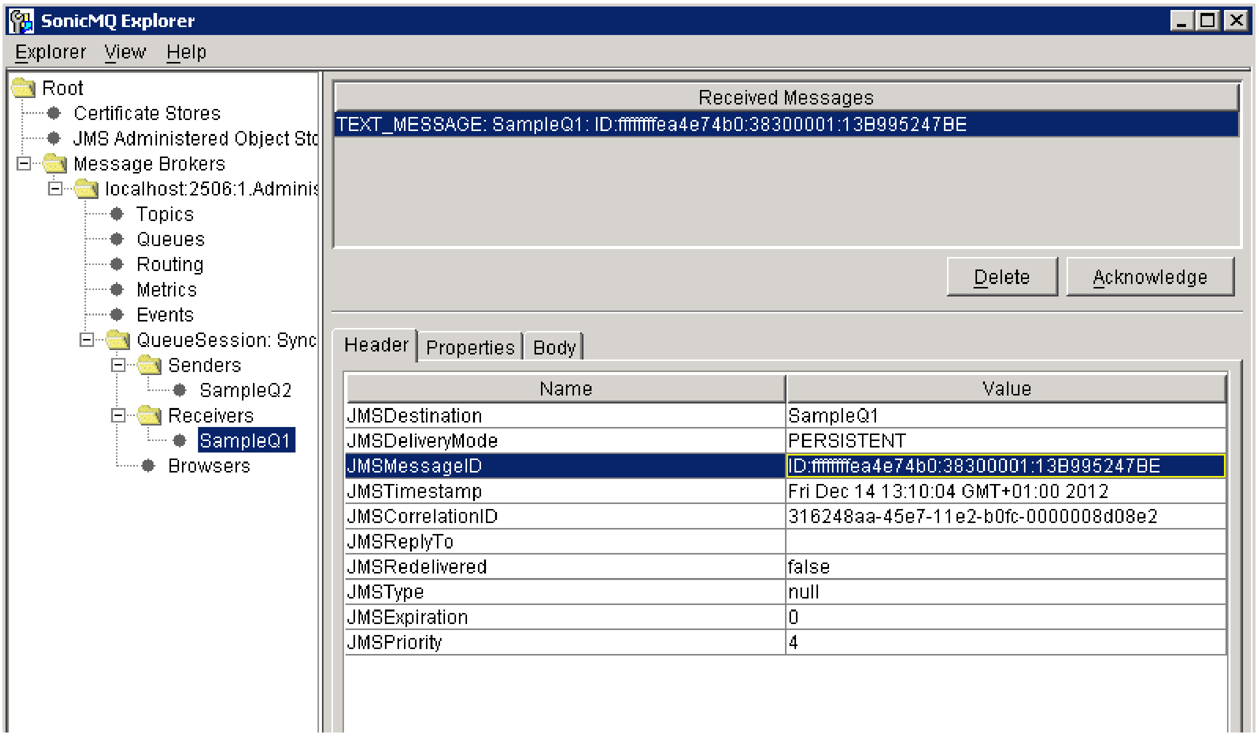
Enter any string into the *input* field, and click on *Next*.



request:

input: * 2806

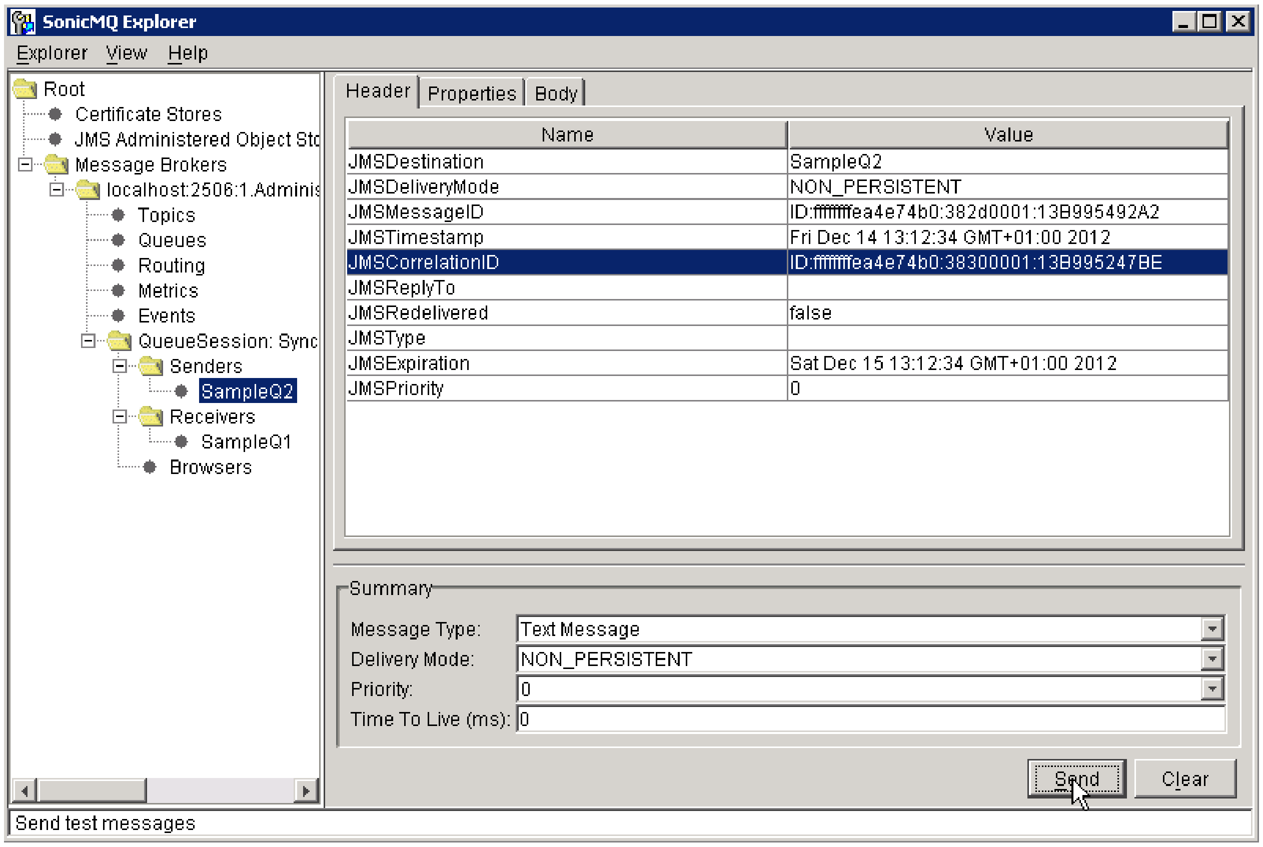
Open the *SonicMQ Explorer*. A new message was put into the *SampleQ1* receiver queue. Copy the *JMS Message ID* into the clipboard.



Received Messages
TEXT_MESSAGE: SampleQ1: ID:ffffffffea4e74b0:38300001:13B995247BE

Name	Value
JMSDestination	SampleQ1
JMSDeliveryMode	PERSISTENT
JMSMessageID	ID:ffffffffea4e74b0:38300001:13B995247BE
JMSTimestamp	Fri Dec 14 13:10:04 GMT+01:00 2012
JMSCorrelationID	316248aa-45e7-11e2-b0fc-0000008d08e2
JMSReplyTo	
JMSRedelivered	false
JMSType	null
JMSExpiration	0
JMSPriority	4

Before sending the response message, we paste the beforehand copied *JMS Message ID* of the request message into the *JMS Correlation ID* of the response's header. Once we put the message into the *SampleQ2* sender queue, it is read from the JMS adapter, and passed to the Web Service client.

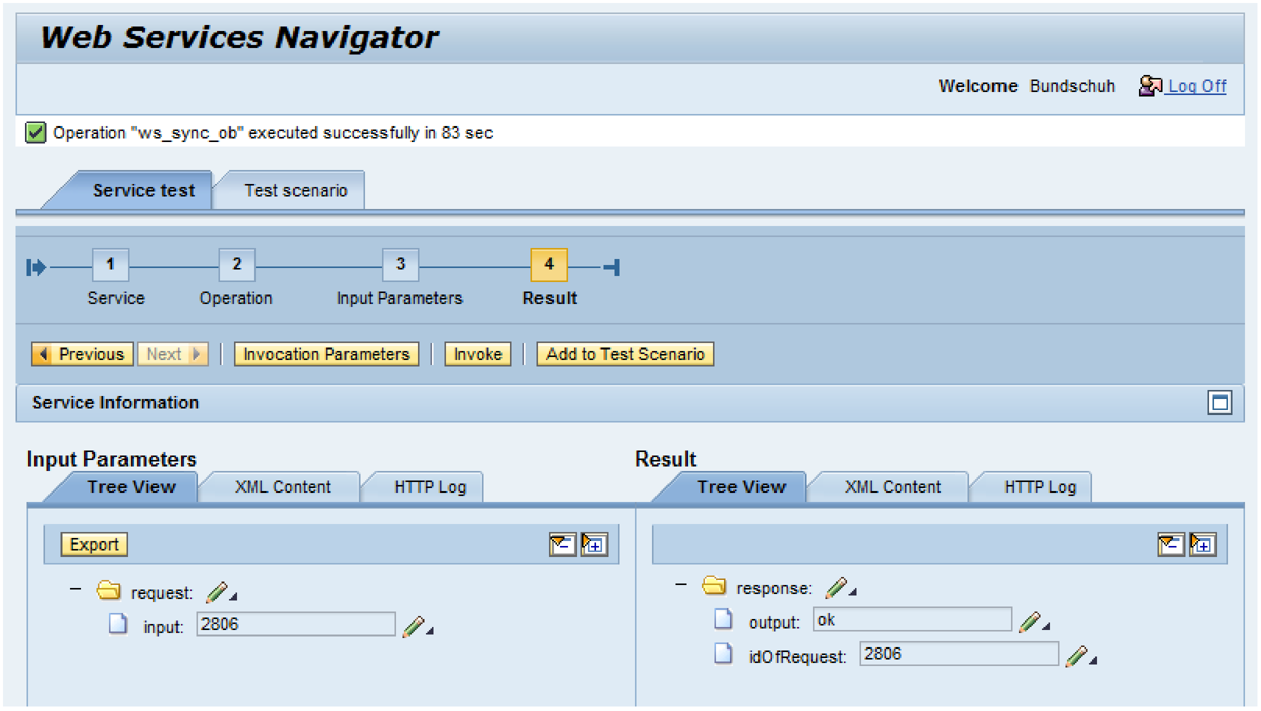


Name	Value
JMSDestination	SampleQ2
JMSDeliveryMode	NON_PERSISTENT
JMSMessageID	ID:ffffffffea4e74b0:382d0001:13B995492A2
JMSTimestamp	Fri Dec 14 13:12:34 GMT+01:00 2012
JMSCorrelationID	ID:ffffffffea4e74b0:38300001:13B995247BE
JMSReplyTo	
JMSRedelivered	false
JMSType	
JMSExpiration	Sat Dec 15 13:12:34 GMT+01:00 2012
JMSPriority	0

Summary

Message Type:	Text Message
Delivery Mode:	NON_PERSISTENT
Priority:	0
Time To Live (ms):	0

In the SOAP adapter, the asynchronous response is converted into a synchronous response and passed back to the Web Service client. You will get a confirmation that the operation executed successfully.

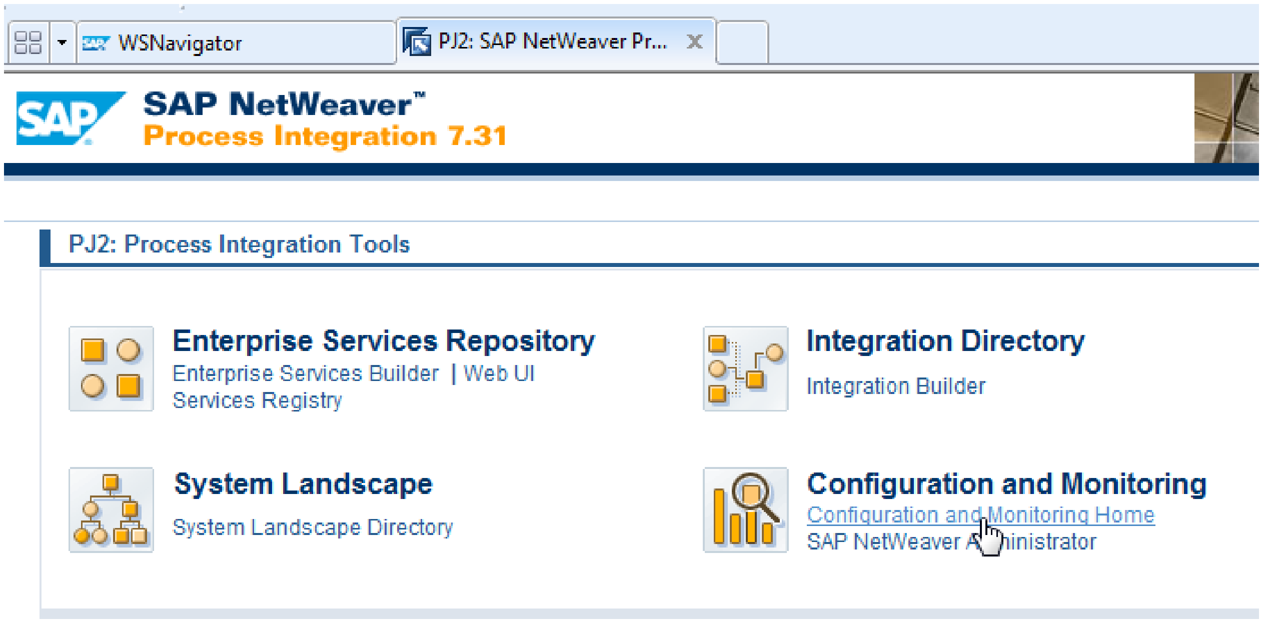


Operation "ws_sync_ob" executed successfully in 83 sec

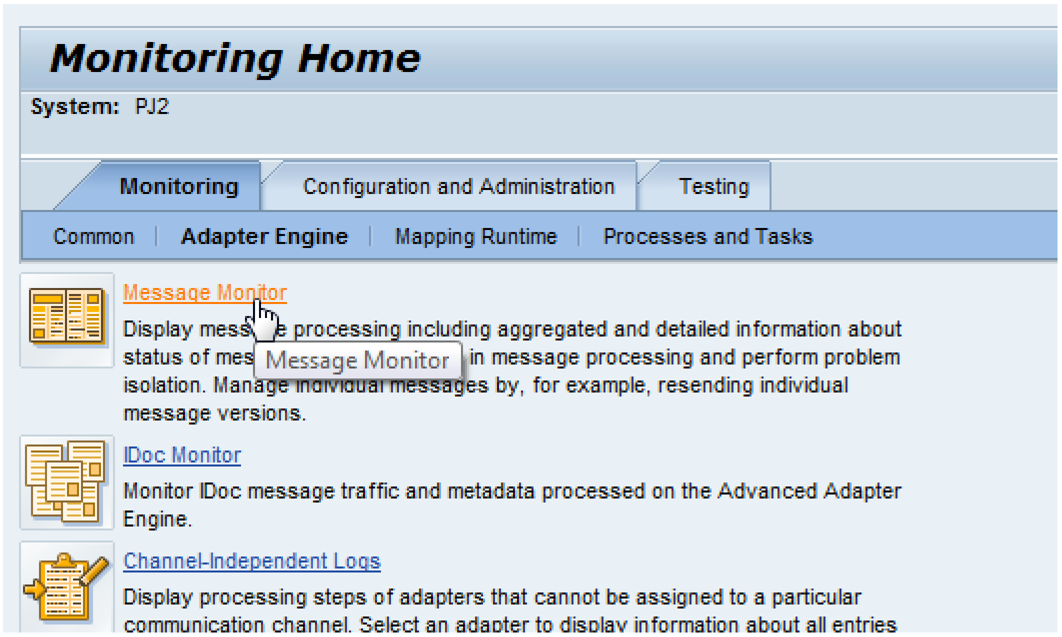
request:
input: 2806

response:
output: ok
idOfRequest: 2806

To monitor the test run, navigate to the *Configuration and Monitoring Home* page from the Process Orchestration landing page.


WSNavigator
PJ2: SAP NetWeaver Pr...
SAP NetWeaver™
Process Integration 7.31
PJ2: Process Integration Tools
Enterprise Services Repository
Enterprise Services Builder | Web UI
Services Registry
Integration Directory
Integration Builder
System Landscape
System Landscape Directory
Configuration and Monitoring
Configuration and Monitoring Home
SAP NetWeaver Administrator

Switch to tab *Monitoring*, sub tab *Adapter Engine*, and select link *Message Monitor*.

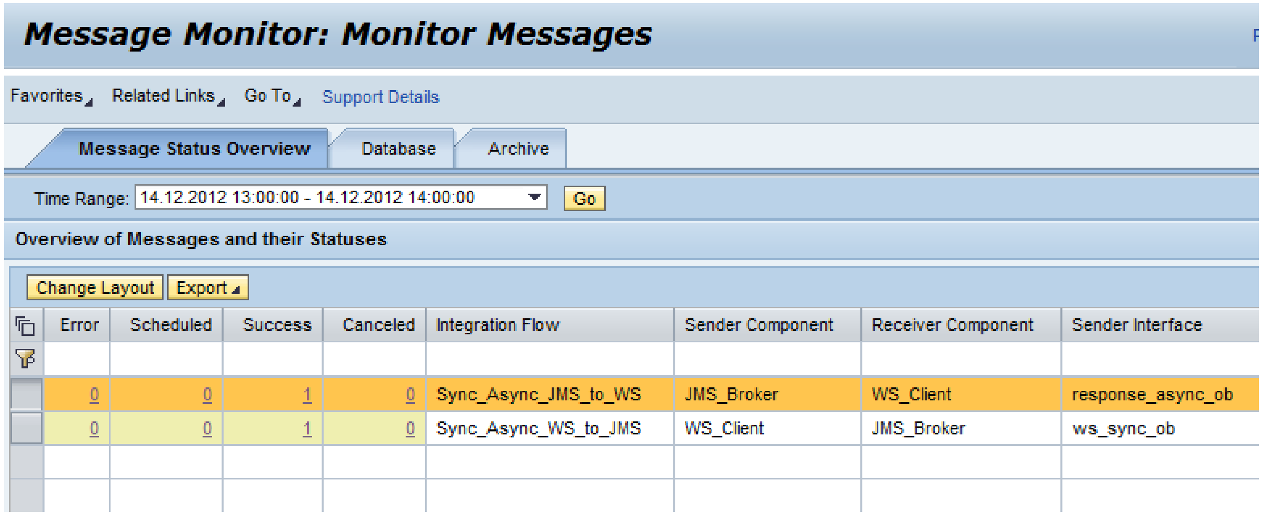


Display message processing including aggregated and detailed information about status of messages in message processing and perform problem isolation. Manage individual messages by, for example, resending individual message versions.

Monitor IDoc message traffic and metadata processed on the Advanced Adapter Engine.

Display processing steps of adapters that cannot be assigned to a particular communication channel. Select an adapter to display information about all entries

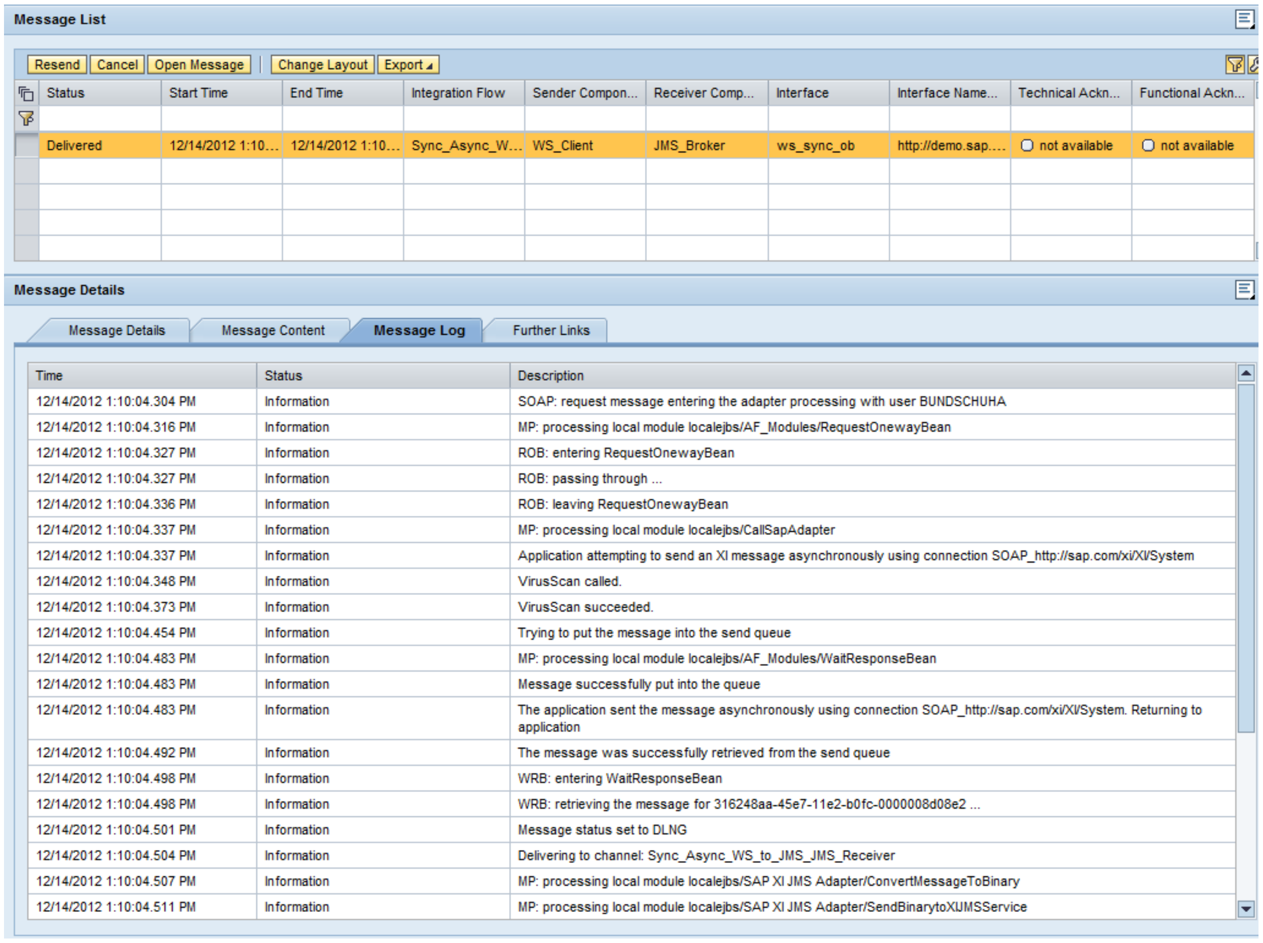
In the *Message Status Overview*, you will see two successful messages. Click on the second message to navigate to the details of the request message.



Time Range: 14.12.2012 13:00:00 - 14.12.2012 14:00:00

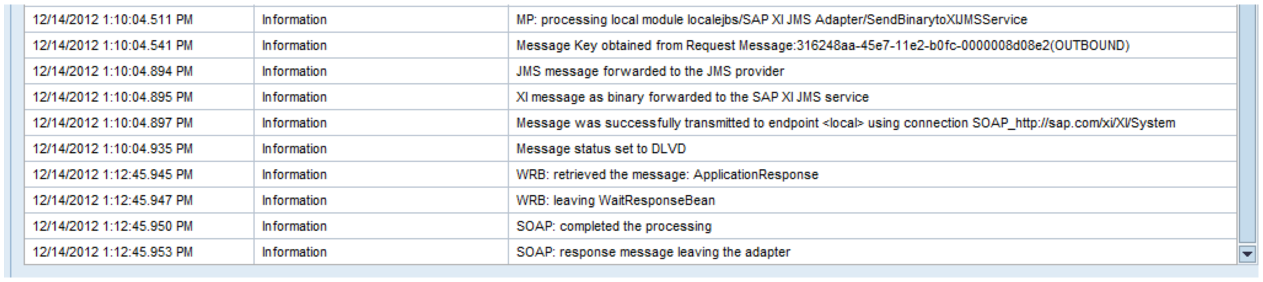
Error	Scheduled	Success	Canceled	Integration Flow	Sender Component	Receiver Component	Sender Interface
0	0	1	0	Sync_Async_JMS_to_WS	JMS_Broker	WS_Client	response_async_ob
0	0	1	0	Sync_Async_WS_to_JMS	WS_Client	JMS_Broker	ws_sync_ob

Switch to tab *Message Log*. You can see from the message log in which sequence the modules were processed.



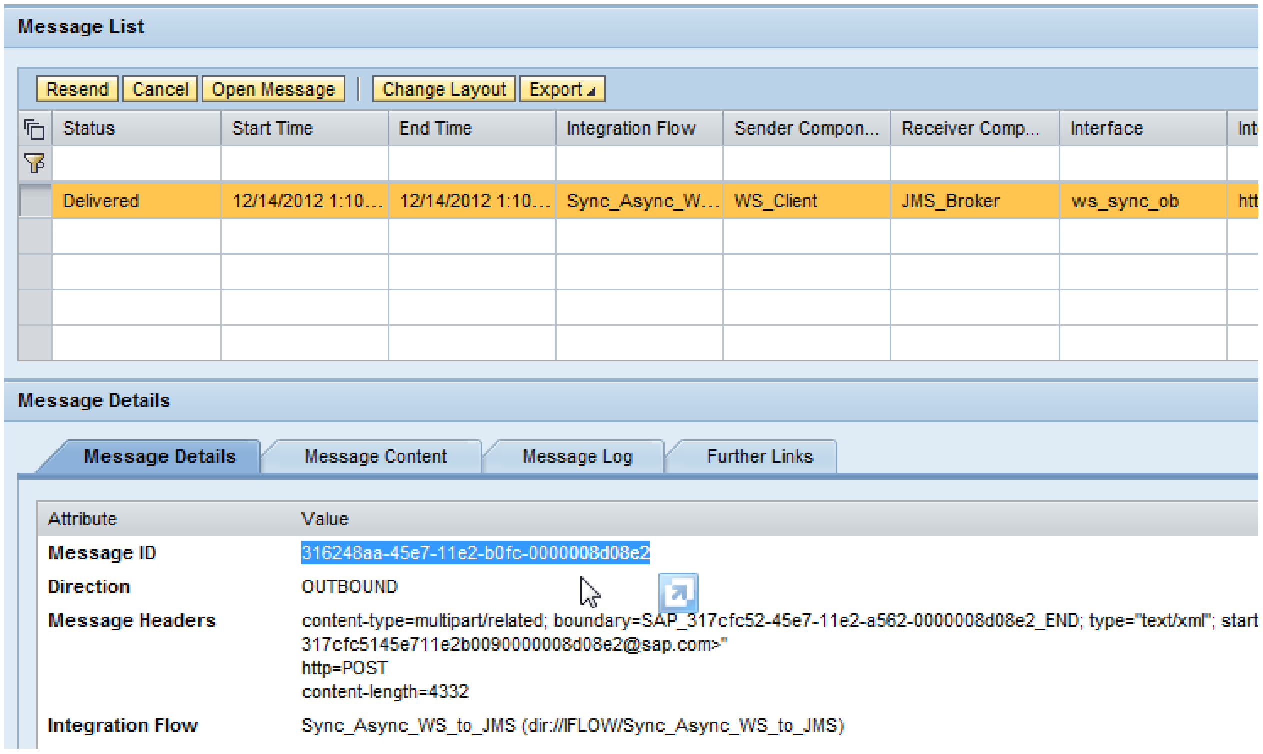
Status	Start Time	End Time	Integration Flow	Sender Compon...	Receiver Comp...	Interface	Interface Name...	Technical Ackn...	Functional Ackn...
Delivered	12/14/2012 1:10...	12/14/2012 1:10...	Sync_Async_W...	WS_Client	JMS_Broker	ws_sync_ob	http://demo.sap....	○ not available	○ not available

Time	Status	Description
12/14/2012 1:10:04.304 PM	Information	SOAP: request message entering the adapter processing with user BUNDSCHUHA
12/14/2012 1:10:04.316 PM	Information	MP: processing local module localejbs/AF_Modules/RequestOnewayBean
12/14/2012 1:10:04.327 PM	Information	ROB: entering RequestOnewayBean
12/14/2012 1:10:04.327 PM	Information	ROB: passing through ...
12/14/2012 1:10:04.336 PM	Information	ROB: leaving RequestOnewayBean
12/14/2012 1:10:04.337 PM	Information	MP: processing local module localejbs/CallSapAdapter
12/14/2012 1:10:04.337 PM	Information	Application attempting to send an XI message asynchronously using connection SOAP_http://sap.com/xi/XI/System
12/14/2012 1:10:04.348 PM	Information	VirusScan called.
12/14/2012 1:10:04.373 PM	Information	VirusScan succeeded.
12/14/2012 1:10:04.454 PM	Information	Trying to put the message into the send queue
12/14/2012 1:10:04.483 PM	Information	MP: processing local module localejbs/AF_Modules/WaitResponseBean
12/14/2012 1:10:04.483 PM	Information	Message successfully put into the queue
12/14/2012 1:10:04.483 PM	Information	The application sent the message asynchronously using connection SOAP_http://sap.com/xi/XI/System. Returning to application
12/14/2012 1:10:04.492 PM	Information	The message was successfully retrieved from the send queue
12/14/2012 1:10:04.498 PM	Information	WRB: entering WaitResponseBean
12/14/2012 1:10:04.498 PM	Information	WRB: retrieving the message for 316248aa-45e7-11e2-b0fc-0000008d08e2 ...
12/14/2012 1:10:04.501 PM	Information	Message status set to DLNG
12/14/2012 1:10:04.504 PM	Information	Delivering to channel: Sync_Async_WS_to_JMS_JMS_Receiver
12/14/2012 1:10:04.507 PM	Information	MP: processing local module localejbs/SAP XI JMS Adapter/ConvertMessageToBinary
12/14/2012 1:10:04.511 PM	Information	MP: processing local module localejbs/SAP XI JMS Adapter/SendBinarytoXIJMSService



Time	Status	Description
12/14/2012 1:10:04.511 PM	Information	MP: processing local module localejbs/SAP XI JMS Adapter/SendBinarytoXIJMSService
12/14/2012 1:10:04.541 PM	Information	Message Key obtained from Request Message:316248aa-45e7-11e2-b0fc-0000008d08e2(OUTBOUND)
12/14/2012 1:10:04.894 PM	Information	JMS message forwarded to the JMS provider
12/14/2012 1:10:04.895 PM	Information	XI message as binary forwarded to the SAP XI JMS service
12/14/2012 1:10:04.897 PM	Information	Message was successfully transmitted to endpoint <local> using connection SOAP_http://sap.com/xi/XI/System
12/14/2012 1:10:04.935 PM	Information	Message status set to DLVD
12/14/2012 1:12:45.945 PM	Information	WRB: retrieved the message: ApplicationResponse
12/14/2012 1:12:45.947 PM	Information	WRB: leaving WaitResponseBean
12/14/2012 1:12:45.950 PM	Information	SOAP: completed the processing
12/14/2012 1:12:45.953 PM	Information	SOAP: response message leaving the adapter

Switch to tab *Message Details*. Below, you see the *PI Message ID* of the request message.



Message List

Resend | Cancel | Open Message | Change Layout | Export

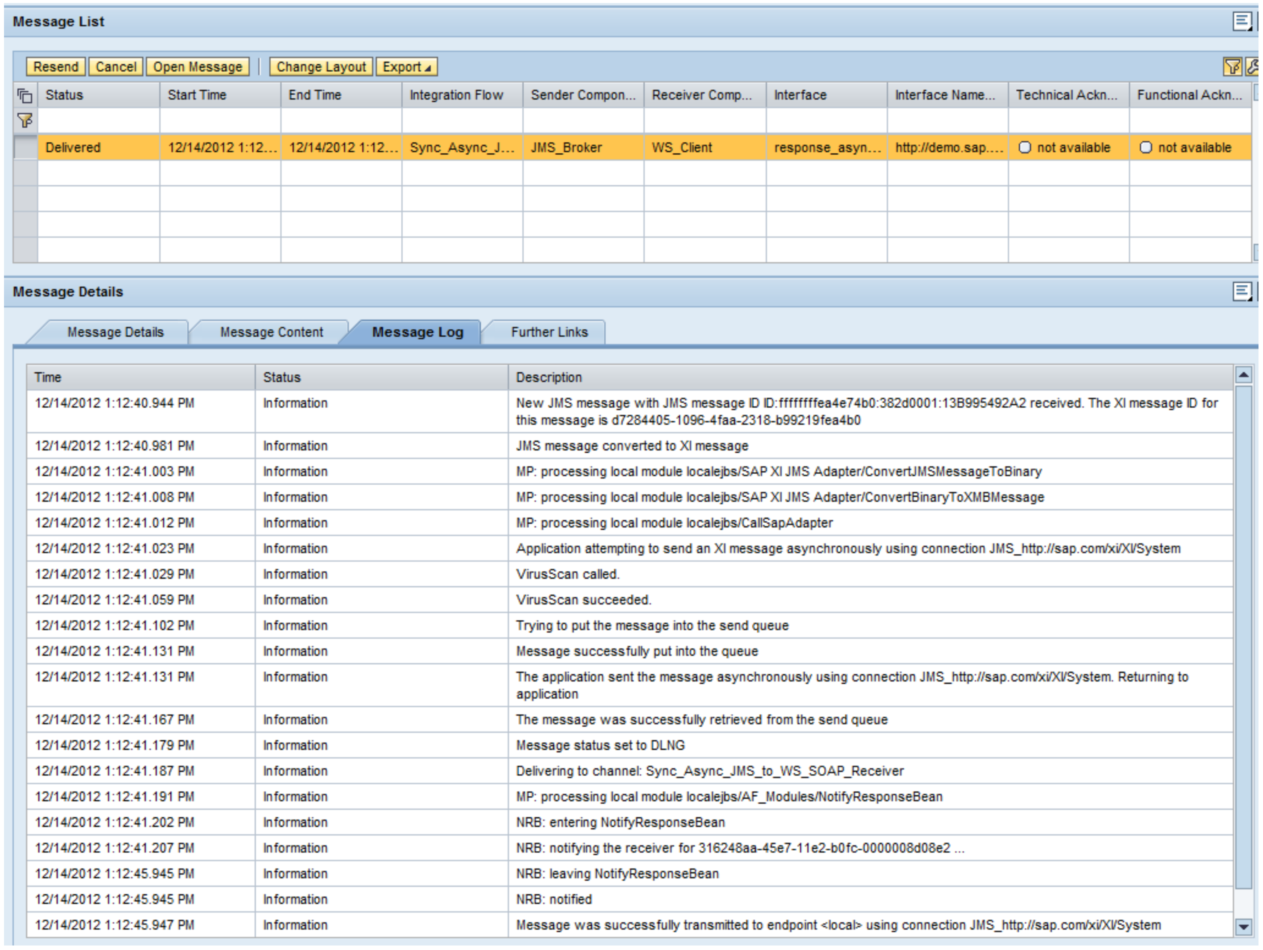
Status	Start Time	End Time	Integration Flow	Sender Compon...	Receiver Comp...	Interface	Int
Delivered	12/14/2012 1:10...	12/14/2012 1:10...	Sync_Async_W...	WS_Client	JMS_Broker	ws_sync_ob	htt

Message Details

Message Details | Message Content | Message Log | Further Links

Attribute	Value
Message ID	316248aa-45e7-11e2-b0fc-0000008d08e2
Direction	OUTBOUND
Message Headers	content-type=multipart/related; boundary=SAP_317cfc52-45e7-11e2-a562-0000008d08e2_END; type="text/xml"; start 317cfc5145e711e2b0090000008d08e2@sap.com>" http=POST content-length=4332
Integration Flow	Sync_Async_WS_to_JMS (dir://IFLOW/Sync_Async_WS_to_JMS)

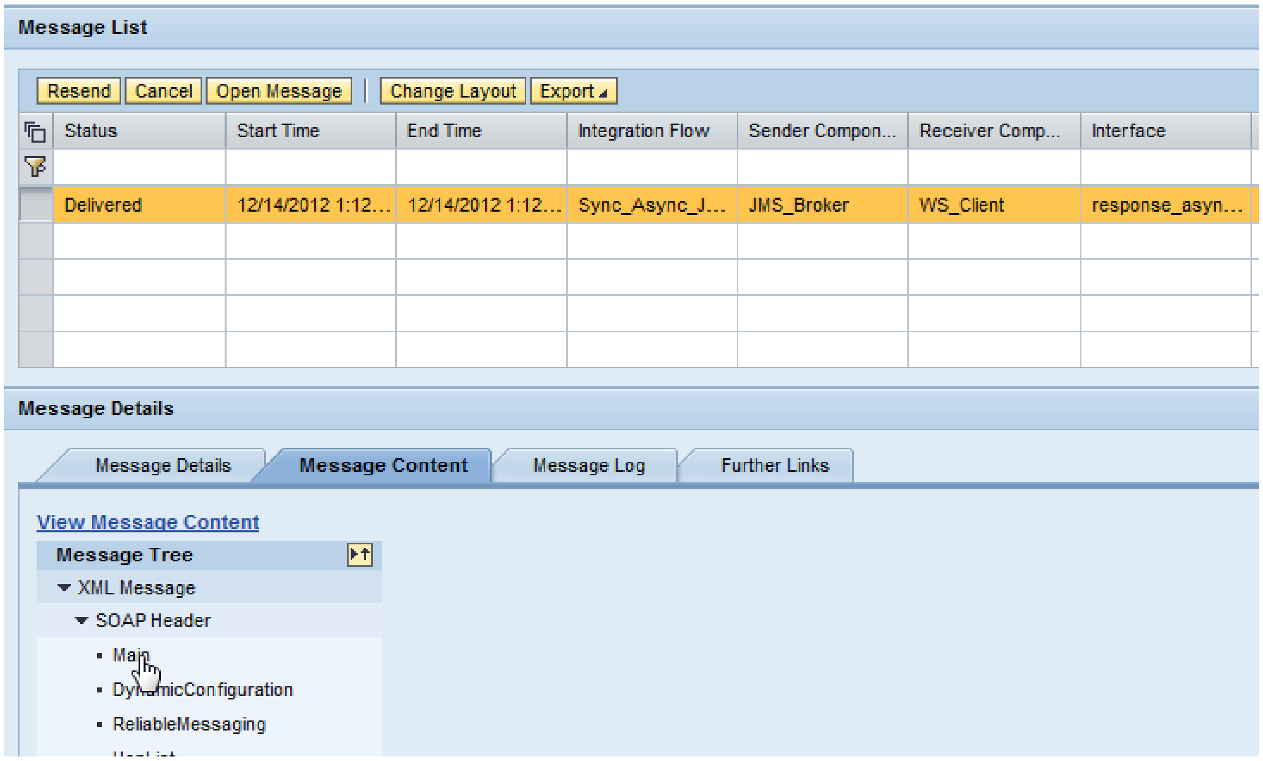
Select the response message, and switch to tab *Message Log*. Below you see the message log of the response message.



Status	Start Time	End Time	Integration Flow	Sender Compon...	Receiver Comp...	Interface	Interface Name...	Technical Ackn...	Functional Ackn...
Delivered	12/14/2012 1:12...	12/14/2012 1:12...	Sync_Async_J...	JMS_Broker	WS_Client	response_asyn...	http://demo.sap....	not available	not available

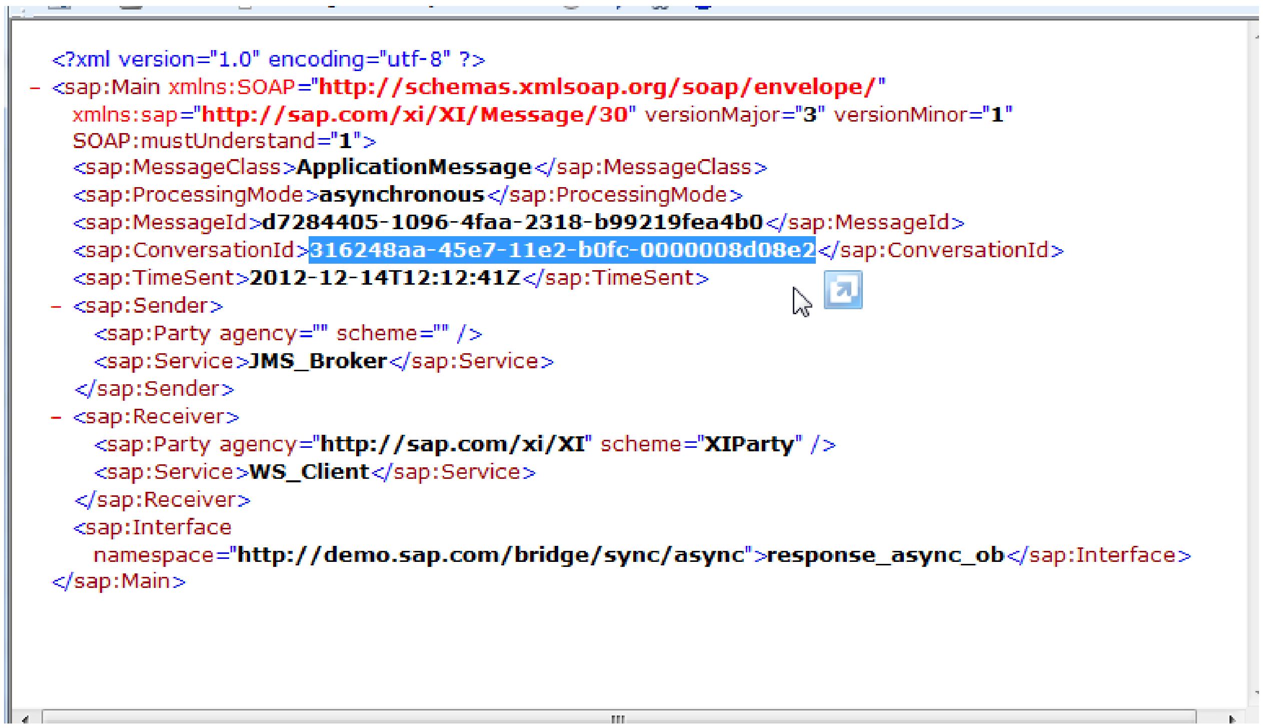
Time	Status	Description
12/14/2012 1:12:40.944 PM	Information	New JMS message with JMS message ID ID:ffffffffea4e74b0:382d0001:13B995492A2 received. The XI message ID for this message is d7284405-1096-4faa-2318-b99219fea4b0
12/14/2012 1:12:40.981 PM	Information	JMS message converted to XI message
12/14/2012 1:12:41.003 PM	Information	MP: processing local module localejbs/SAP XI JMS Adapter/ConvertJMSMessageToBinary
12/14/2012 1:12:41.008 PM	Information	MP: processing local module localejbs/SAP XI JMS Adapter/ConvertBinaryToXMBMessage
12/14/2012 1:12:41.012 PM	Information	MP: processing local module localejbs/CallSapAdapter
12/14/2012 1:12:41.023 PM	Information	Application attempting to send an XI message asynchronously using connection JMS_http://sap.com/xi/XI/System
12/14/2012 1:12:41.029 PM	Information	VirusScan called.
12/14/2012 1:12:41.059 PM	Information	VirusScan succeeded.
12/14/2012 1:12:41.102 PM	Information	Trying to put the message into the send queue
12/14/2012 1:12:41.131 PM	Information	Message successfully put into the queue
12/14/2012 1:12:41.131 PM	Information	The application sent the message asynchronously using connection JMS_http://sap.com/xi/XI/System. Returning to application
12/14/2012 1:12:41.167 PM	Information	The message was successfully retrieved from the send queue
12/14/2012 1:12:41.179 PM	Information	Message status set to DLNG
12/14/2012 1:12:41.187 PM	Information	Delivering to channel: Sync_Async_JMS_to_WS_SOAP_Receiver
12/14/2012 1:12:41.191 PM	Information	MP: processing local module localejbs/AF_Modules/NotifyResponseBean
12/14/2012 1:12:41.202 PM	Information	NRB: entering NotifyResponseBean
12/14/2012 1:12:41.207 PM	Information	NRB: notifying the receiver for 316248aa-45e7-11e2-b0fc-0000008d08e2 ...
12/14/2012 1:12:45.945 PM	Information	NRB: leaving NotifyResponseBean
12/14/2012 1:12:45.945 PM	Information	NRB: notified
12/14/2012 1:12:45.947 PM	Information	Message was successfully transmitted to endpoint <local> using connection JMS_http://sap.com/xi/XI/System

Switch to tab *Message Content*, and select the *Main* part of the SOAP header.



Status	Start Time	End Time	Integration Flow	Sender Compon...	Receiver Comp...	Interface
Delivered	12/14/2012 1:12...	12/14/2012 1:12...	Sync_Async_J...	JMS_Broker	WS_Client	response_asyn...

The *Conversation ID* holds the *PI Message ID* of the request message.



```
  <?xml version="1.0" encoding="utf-8" ?>
- <sap:Main xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:sap="http://sap.com/xi/XI/Message/30" versionMajor="3" versionMinor="1"
    SOAP:mustUnderstand="1">
    <sap:MessageClass>ApplicationMessage</sap:MessageClass>
    <sap:ProcessingMode>asynchronous</sap:ProcessingMode>
    <sap:MessageId>d7284405-1096-4faa-2318-b99219fea4b0</sap:MessageId>
    <sap:ConversationId>316248aa-45e7-11e2-b0fc-0000008d08e2</sap:ConversationId>
    <sap:TimeSent>2012-12-14T12:12:41Z</sap:TimeSent>
  - <sap:Sender>
      <sap:Party agency="" scheme="" />
      <sap:Service>JMS_Broker</sap:Service>
    </sap:Sender>
  - <sap:Receiver>
      <sap:Party agency="http://sap.com/xi/XI" scheme="XIParty" />
      <sap:Service>WS_Client</sap:Service>
    </sap:Receiver>
    <sap:Interface
      namespace="http://demo.sap.com/bridge/sync/async">response_async_ob</sap:Interface>
  </sap:Main>
```

Sync/Async Bridge by means of BPM process

A synchronous call is mapped to asynchronous JMS request and response messages by means of a sync/async bridge modeled in BPM. A synchronous request message is sent to the Business Process Engine (BPE) triggering a new process instance. Within the process, an asynchronous message is sent to the AEX where it is routed to the JMS broker. The asynchronous response from the JMS broker is then passed to the BPE where it is matched to the corresponding running process instance. To correlate the JMS response message to the right process instance, a correlation condition needs to be defined within the BPM process based on payload data.

The asynchronous messages are reliably exchanged between BPM and AEX via Java Proxy runtime based on the XI 3.0 protocol. The synchronous Web Service call can be directly passed to the BPM process. Since synchronous calls are of Quality of Service *Best Effort* anyway, the communication does not necessarily have to go via the AEX.

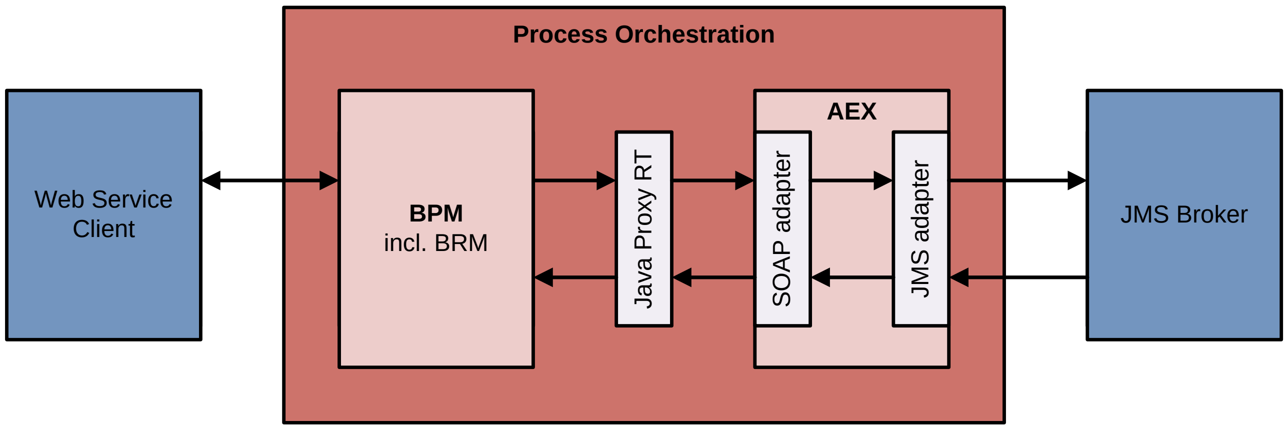


Figure 2: Message flow for sync/async scenario by means of a BPM process

To implement the scenario, beside the BPM process, we need to define two Integration Flows. One for routing the request message from the BPM process to the JMS broker, and one for routing back the response message.

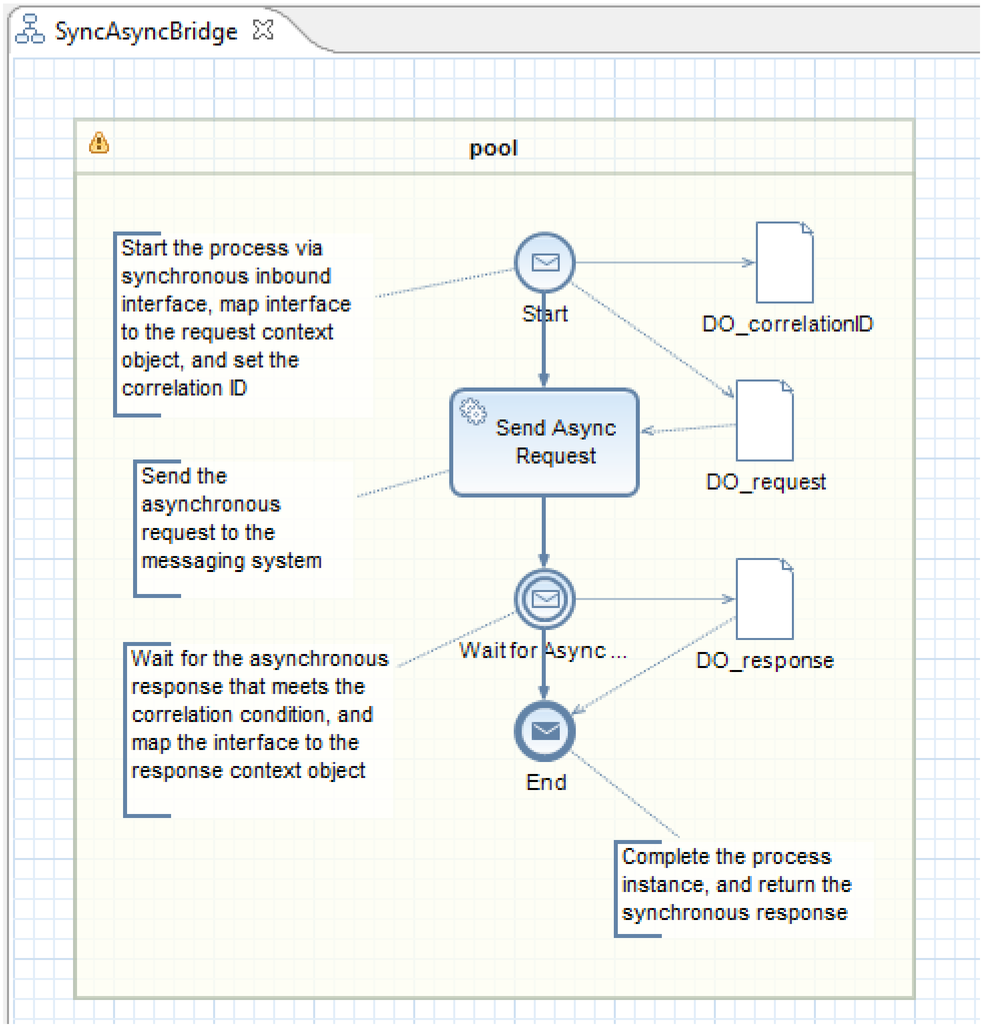
The correlation between the response message and the request message is based on payload data. In the BPM process we need to define a correlation condition for the incoming response message.

At a glance, the following settings have to be made:

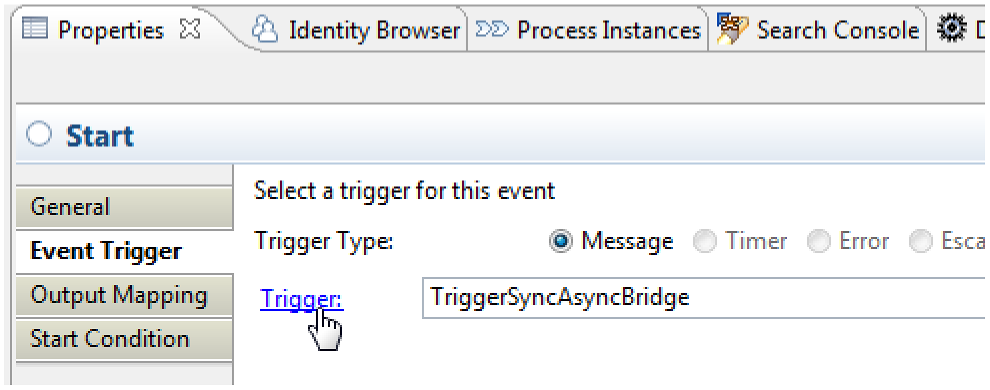
- Create a BPM process acting as sync/async broker
- Create an Integration Flow from the BPM process to the JMS broker
- Create an Integration Flow from the JMS broker to the BPM process

BPM process definition

We model the BPM process in the *Process Development* perspective of the NWDS. The complete process model is shown in figure below. The process begins with a message start event. The trigger of the message start event refers to the synchronous inbound interface which is then mapped to the *DO_request* process context object holding the payload data. Furthermore, the correlation ID is set. The next step is an automated activity (service task) calling the asynchronous request interface to pass the request message to the AEX where it is routed to the JMS broker. The asynchronous response which meets the correlation condition is delivered to an intermediate message event, and mapped to the *DO_response* process context object. Finally, the process is completed via a message end event closing the synchronous call.


SyncAsyncBridge
pool
Start the process via synchronous inbound interface, map interface to the request context object, and set the correlation ID
Start
DO_correlationID
Send Async Request
DO_request
Send the asynchronous request to the messaging system
Wait for Async ...
DO_response
Wait for the asynchronous response that meets the correlation condition, and map the interface to the response context object
End
Complete the process instance, and return the synchronous response

In the following, we will go through the relevant properties for each of the steps. First, pick the message start event *Start*, and switch to tab *Event Trigger*. For the message start event, a new trigger *TriggerSyncAsyncBridge* has been created. Click on the *Trigger* link to get more details.

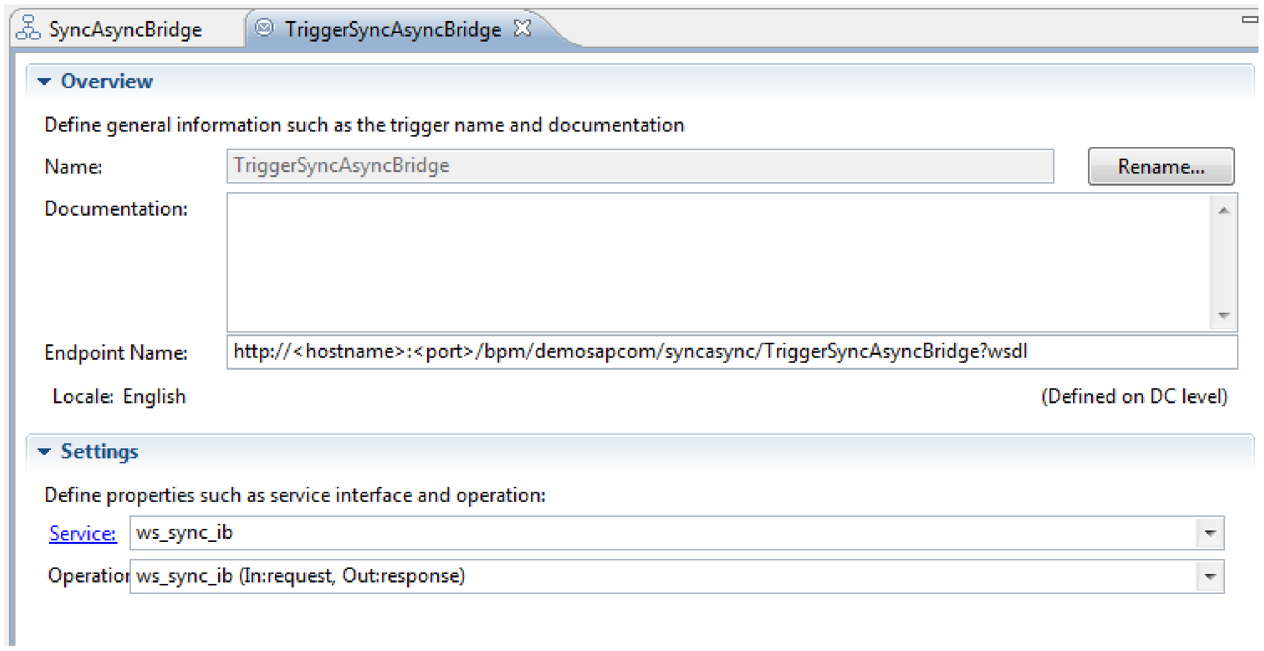


Select a trigger for this event

Trigger Type: [✓ Message] Timer Error Esca

Trigger: TriggerSyncAsyncBridge

In the *Settings* section of the trigger, the synchronous inbound interface has been selected as service interface.



Define general information such as the trigger name and documentation

Name: TriggerSyncAsyncBridge

Documentation:

Endpoint Name: http://<hostname>:<port>/bpm/demosapcom/syncasync/TriggerSyncAsyncBridge?wsdl

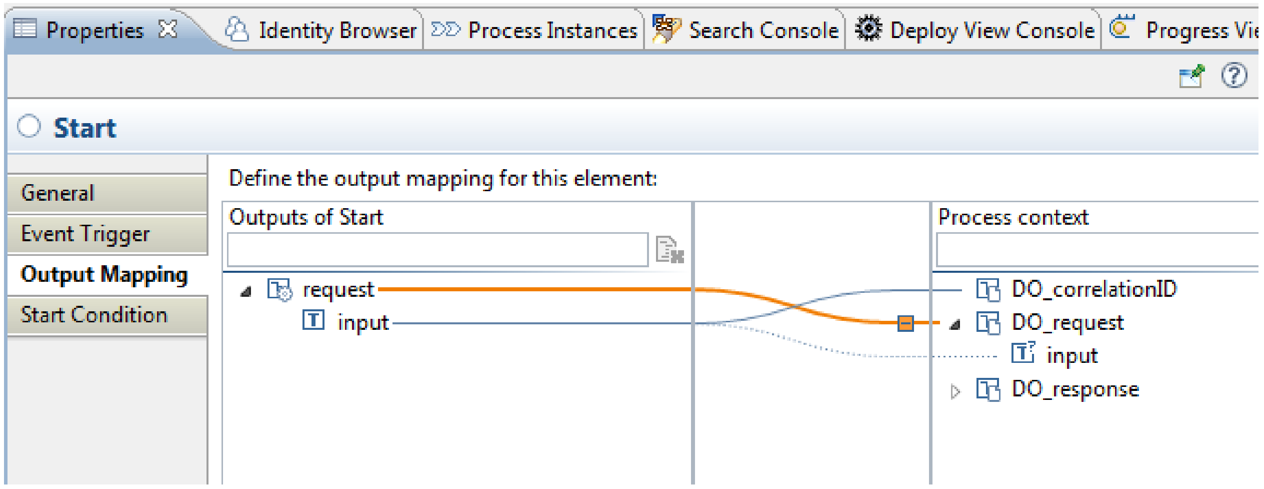
Locale: English (Defined on DC level)

Define properties such as service interface and operation:

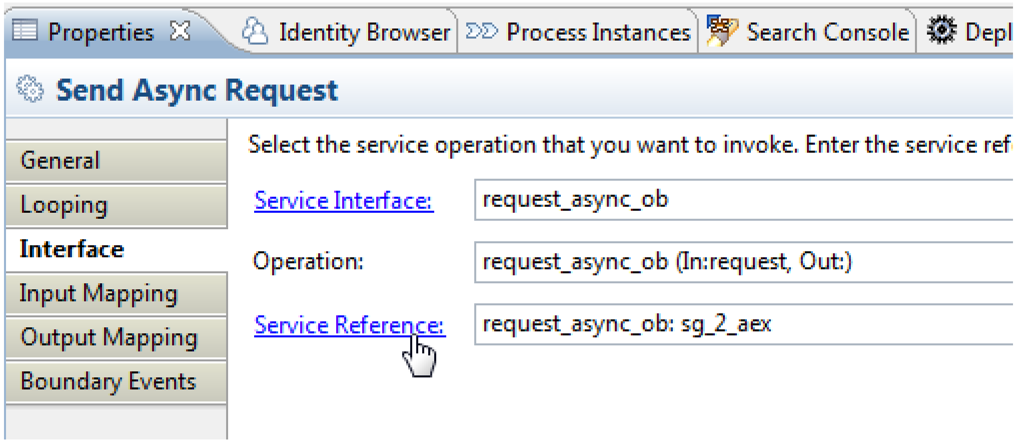
Service: ws_sync_ib

Operatio ws_sync_ib (In:request, Out:response)

Go back to the *Start* event, and switch to tab *Output Mapping*. Map the *request* message type of the synchronous interface to the *DO_request* context object. Furthermore, map the *input* field to the *DO_correlationID*. We assume that the *input* field contains a unique value, and hence can be used to correlate the asynchronous response message to the right process instance, see below.



Pick the automated activity *Send Async Request*, and switch to tab *Interface*. As interface, the asynchronous request outbound interface has been chosen. Reliable connectivity between BPM and AEX is done via the Java Proxy runtime using the XI 3.0 protocol. So, we have to maintain the Service Reference type for the asynchronous request outbound interface. Select the *Service Reference* link to navigate to the Service Reference.

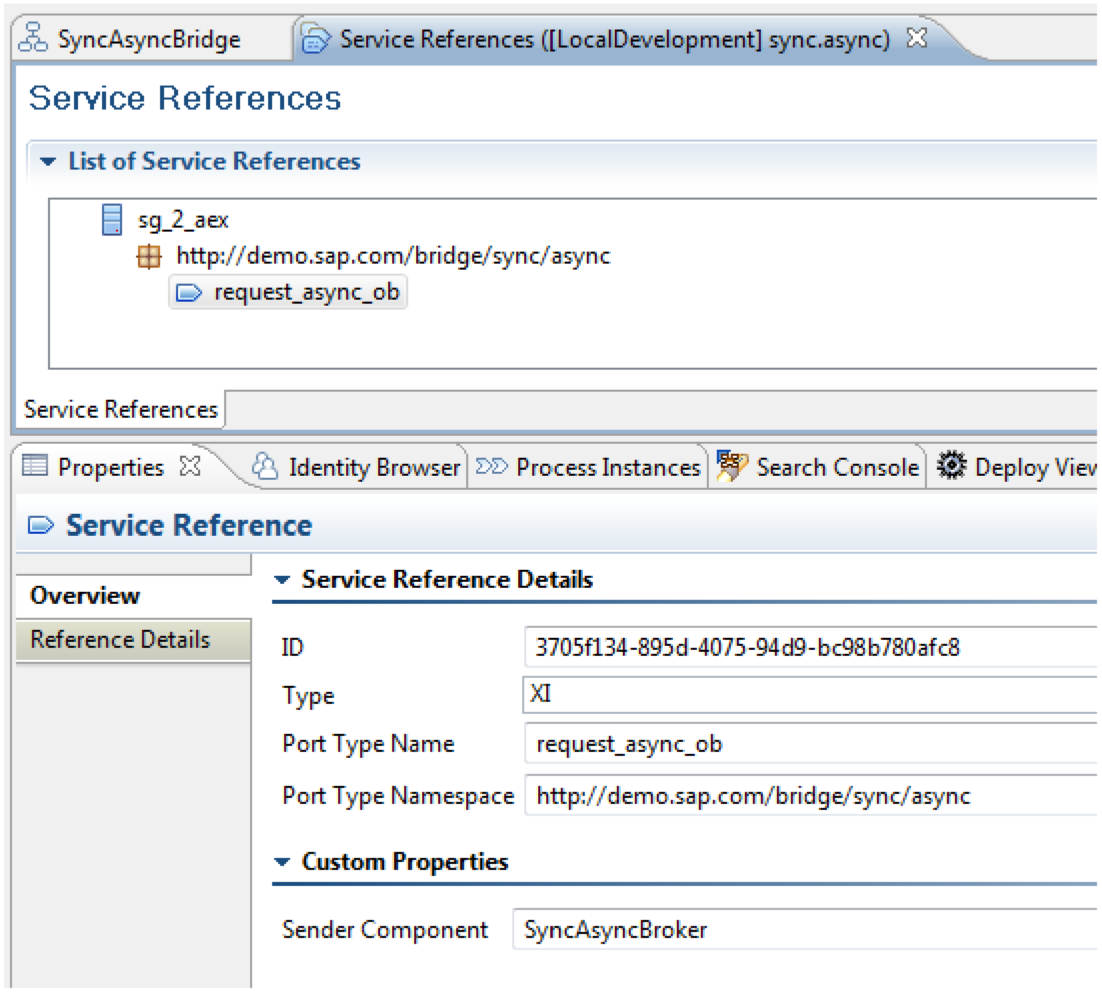


Select the service operation that you want to invoke. Enter the service ref

Service Interface:	request_async_ob
Operation:	request_async_ob (In:request, Out:)
Service Reference:	request_async_ob: sg_2_aex

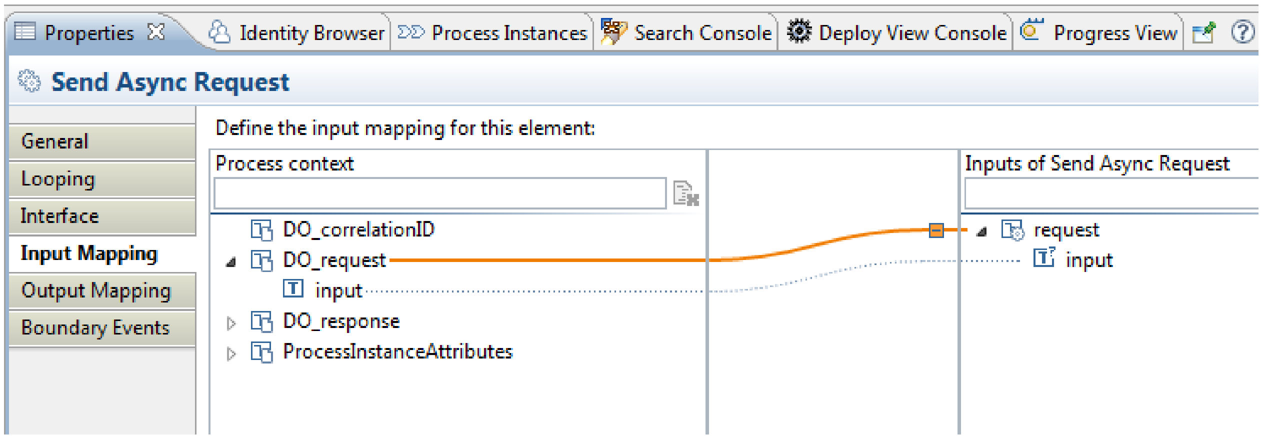
This opens a new window with the list of Service References. In the *Properties* pane of the service reference, choose the Service Reference type *XI*, and enter a *Sender Component* name representing the BPM process within the configuration, here **SyncAsyncBroker**.

Note: The Sender Component name chosen here must be identical to the Sender Component name in the Integration Flow configuration in order to link the BPM process to the Integration Flow, see also below.

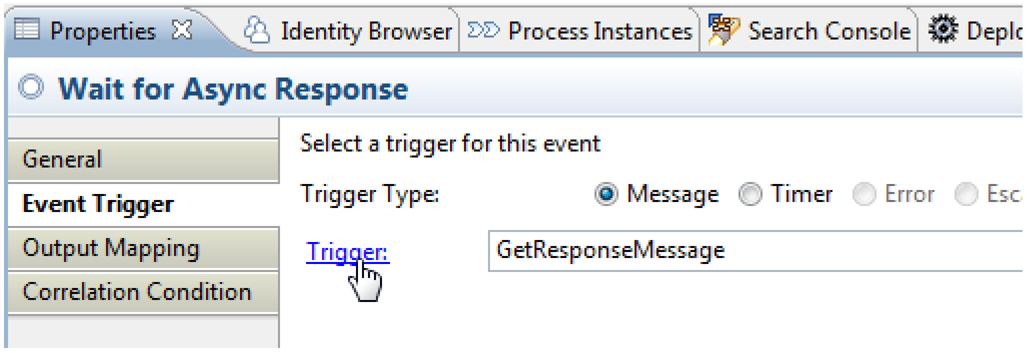


ID	3705f134-895d-4075-94d9-bc98b780afc8
Type	XI
Port Type Name	request_async_ob
Port Type Namespace	http://demo.sap.com/bridge/sync/async
Sender Component	SyncAsyncBroker

Go back to the automated activity, and switch to tab *Input Mapping*. Map the process context *DO_request* to the *request* interface.



Pick the intermediate message event *Wait for Async Response*, and switch to tab *Event Trigger*. For the intermediate message event, a new trigger *GetResponseMessage* has been created. Click on the *Trigger* link to get more details



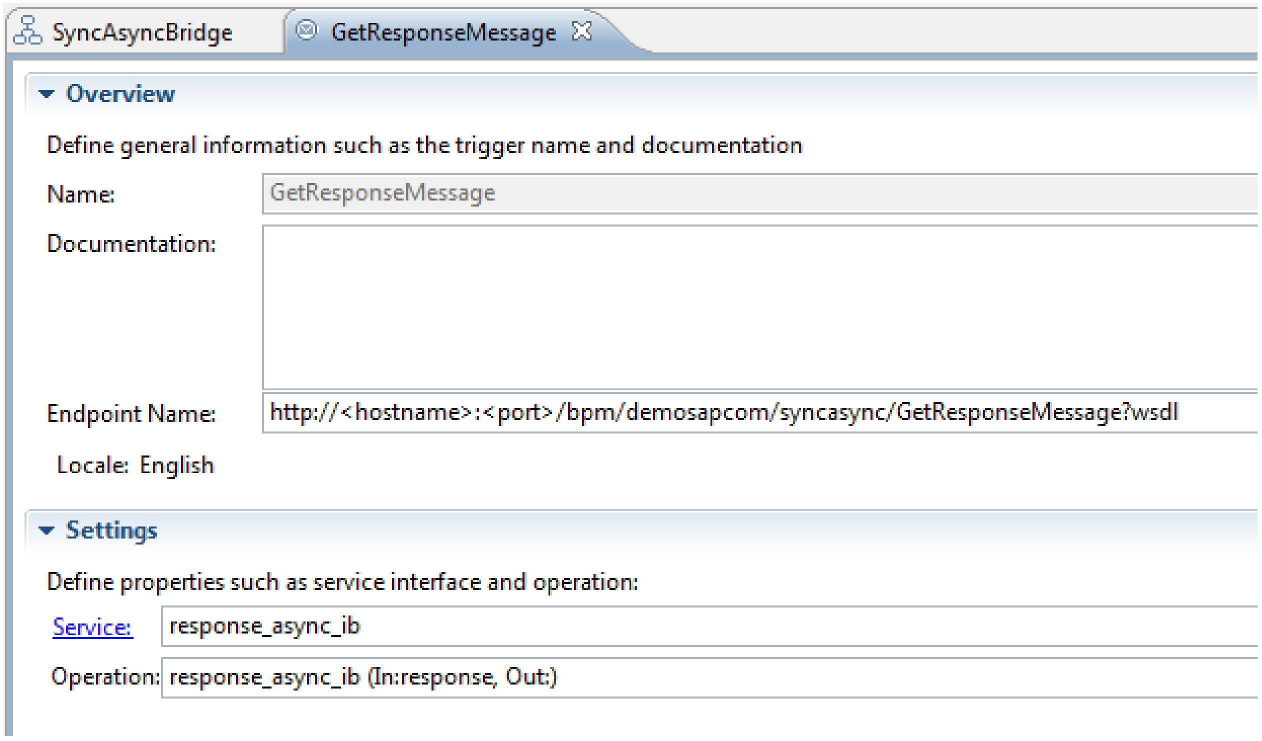
Wait for Async Response

Select a trigger for this event

Trigger Type: [✓ Message] Timer Error Esc

Trigger: GetResponseMessage

In the *Settings* section of the trigger, the asynchronous response inbound interface has been selected as service interface.



Overview

Define general information such as the trigger name and documentation

Name: GetResponseMessage

Documentation:

Endpoint Name: http://<hostname>:<port>/bpm/demosapcom/syncasync/GetResponseMessage?wsdl

Locale: English

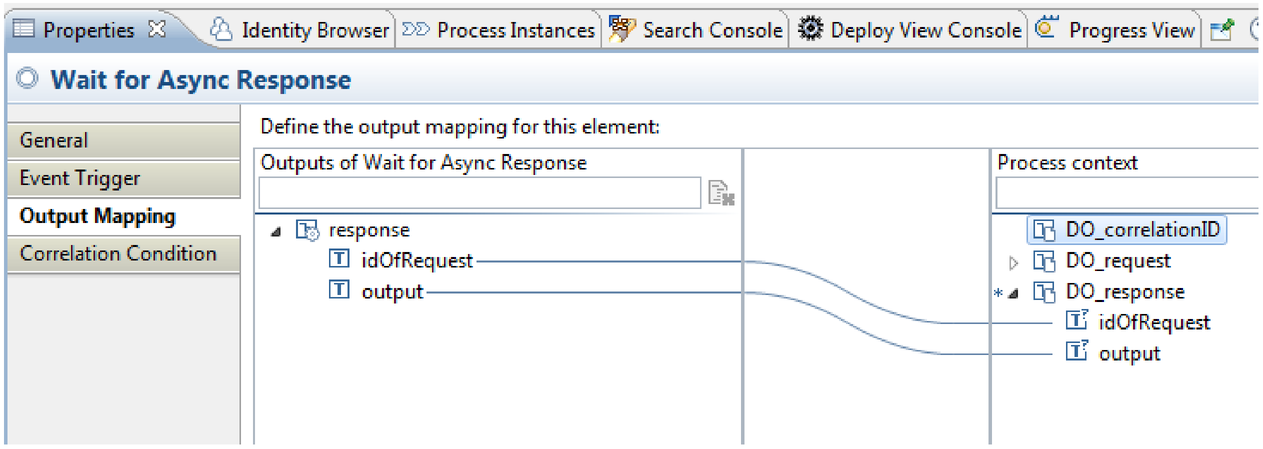
Settings

Define properties such as service interface and operation:

Service: response_async_ib

Operation: response_async_ib (In:response, Out:)

Go back to the intermediate message event *Wait for Async Response*, and switch to tab *Output Mapping*. Map the response interface to the *DO_response* process context.



Wait for Async Response

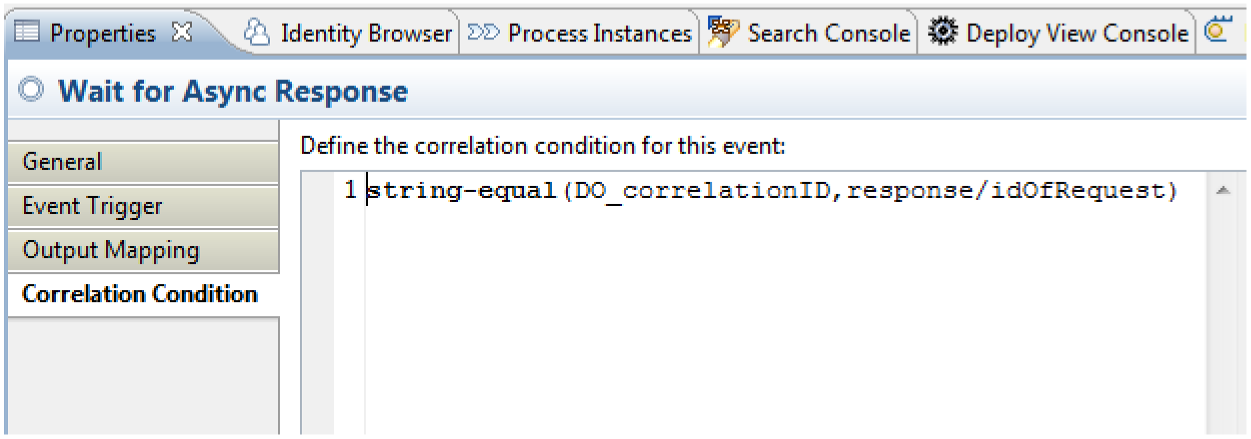
Define the output mapping for this element:

Outputs of Wait for Async Response
- response
  - idOfRequest
  - output

Process context
- DO_correlationID
- DO_request
- DO_response
  - idOfRequest
  - output

Switch to tab *Correlation Condition*, and maintain the correlation condition as follows:

String-equal(DO_correlationID,response/idOfRequest)

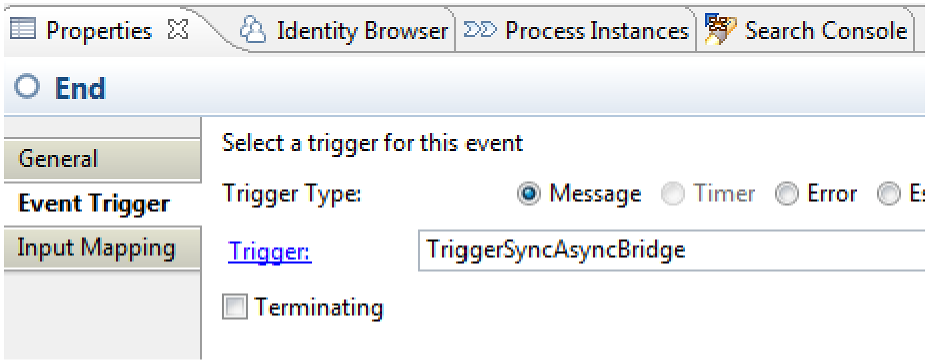


Wait for Async Response

Define the correlation condition for this event:

```
string-equal(DO_correlationID,response/idOfRequest)
```

Pick the message end event *End*, and switch to tab *Event Trigger*. Assign the same trigger *TriggerSyncAsyncBridge* that has been chosen for the message start event.



End

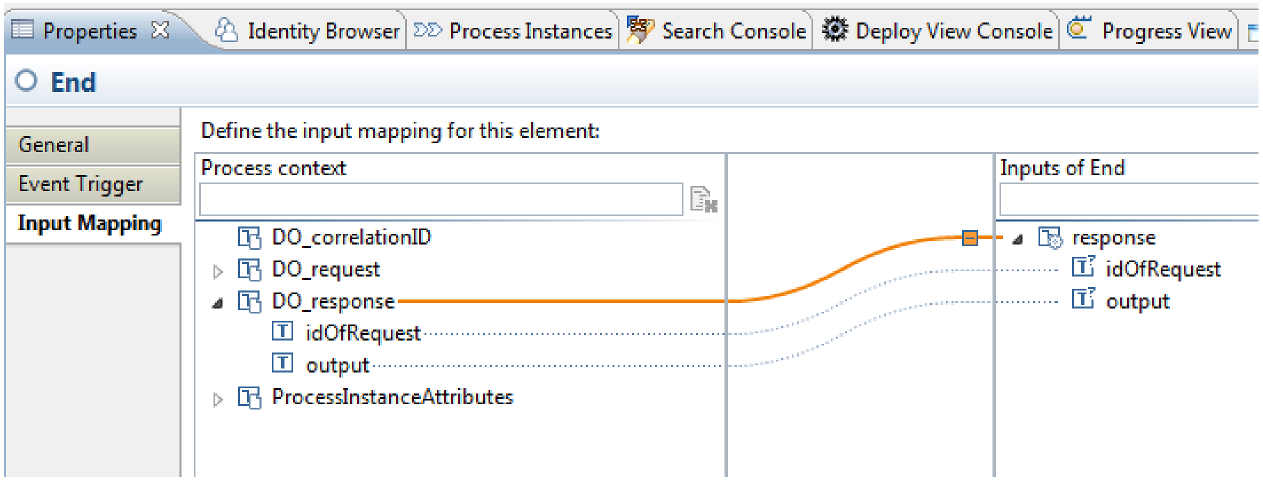
Select a trigger for this event

Trigger Type: [✓ Message] Timer Error E

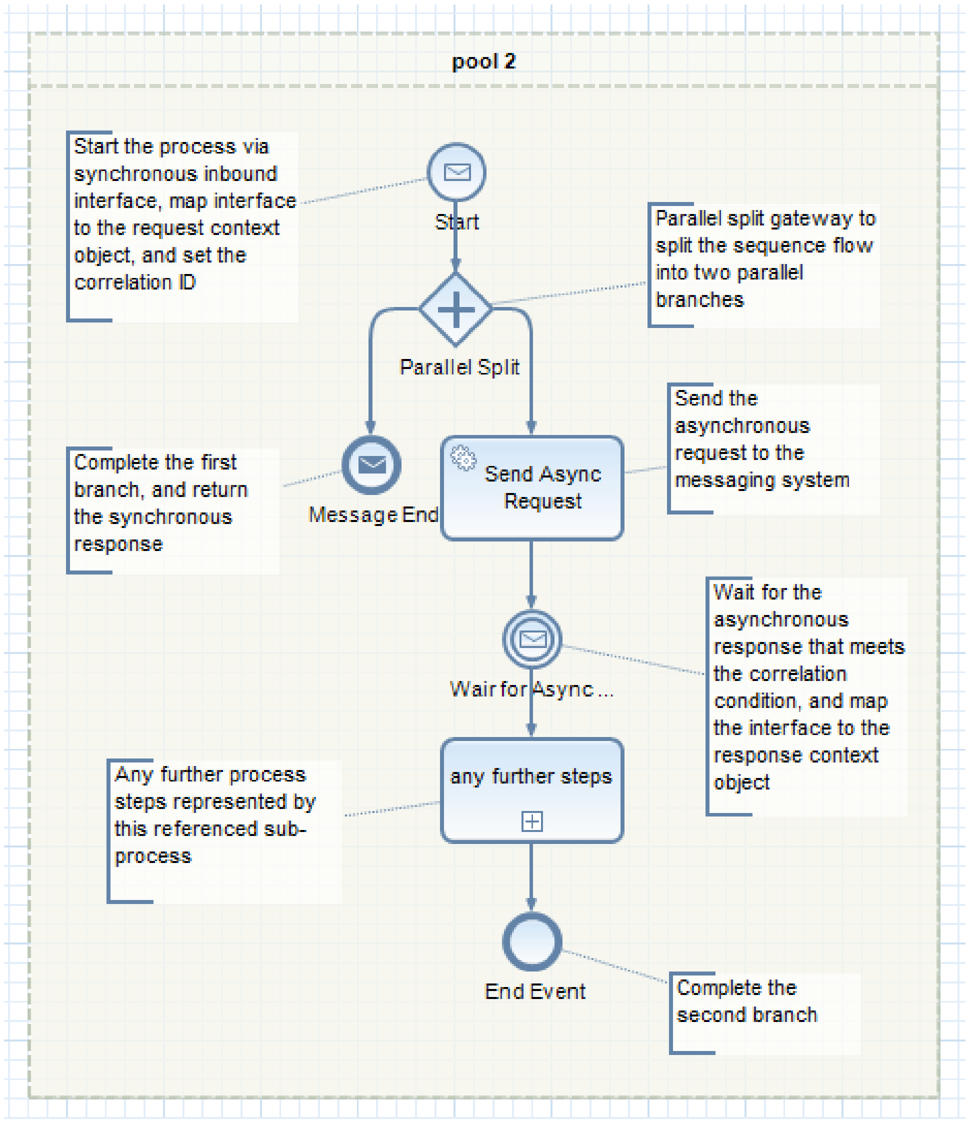
Trigger: TriggerSyncAsyncBridge

☐ Terminating

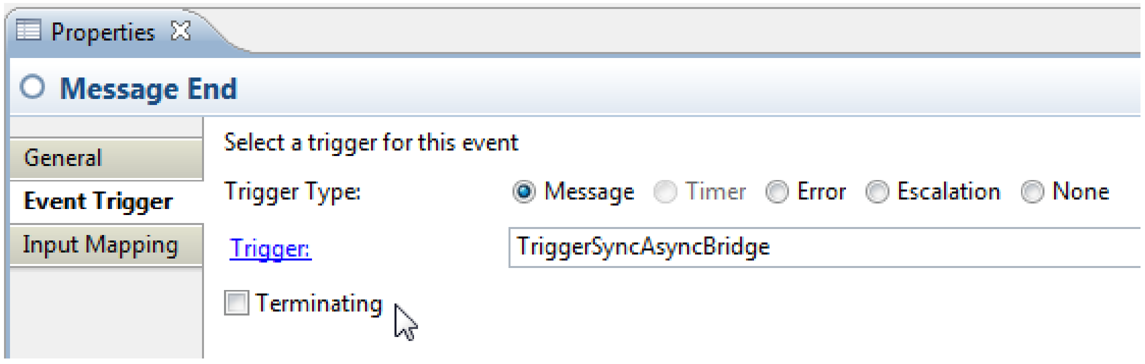
Switch to tab *Input Mapping*, and map the process context *DO_response* to the *response* message type of the synchronous interface.



As mentioned above, it is not guaranteed that the response message is sent within the Web Service timeout period, especially if there are many asynchronous steps between the start event and the end event. To avoid that the client will get a response timeout, you may implement the alternative process as seen blow. It uses a parallel split gateway so that after the start event a parallel token is created which immediately sends the response message. However, this would make sense only if the synchronous response did not depend on the asynchronous response, i.e., the client would only need a confirmation that the synchronous message was successful.


pool 2
Start the process via synchronous inbound interface, map interface to the request context object, and set the correlation ID
Start
Parallel split gateway to split the sequence flow into two parallel branches
Parallel Split
Send the asynchronous request to the messaging system
Complete the first branch, and return the synchronous response
Message End
Send Async Request
Wait for the asynchronous response that meets the correlation condition, and map the interface to the response context object
Wair for Async ...
Any further process steps represented by this referenced sub-process
any further steps
End Event
Complete the second branch

For the message end event, the *Terminating* flag must not be selected. Otherwise, it would end the complete process including the parallel flow.



Select a trigger for this event

Trigger Type: [✓ Message] Timer Error Escalation None

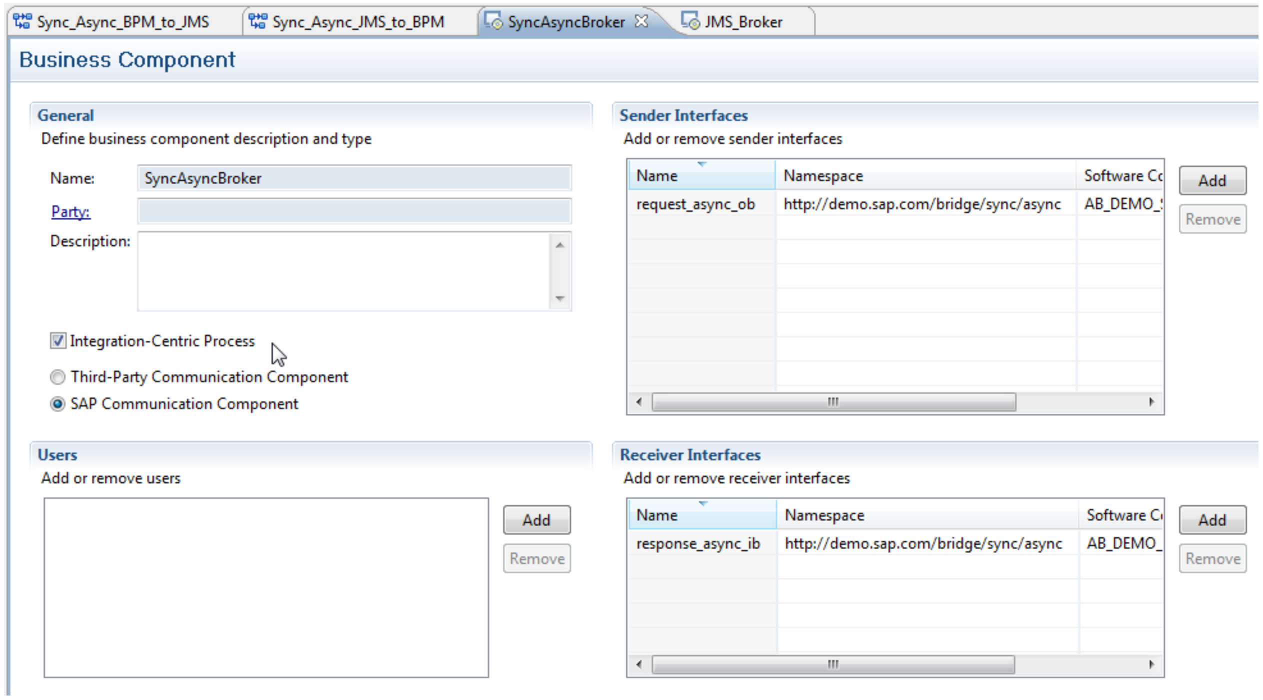
Trigger: TriggerSyncAsyncBridge

☐ Terminating

Integration Flow from BPM Process to JMS Broker

Switch to the *SAP Process Integration Designer* perspective to configure the message flow from and to the BPM process.

First, we need to create a new business component representing the BPM process. As business component name choose the name previously set in the service reference configuration, i.e., **SyncAsyncBroker**, and assign the asynchronous interfaces to the same.



General

Define business component description and type

Name: SyncAsyncBroker

Party:

Description:

☑ Integration-Centric Process

Third-Party Communication Component [✓ SAP Communication Component]

Sender Interfaces

Add or remove sender interfaces

Name	Namespace	Software C
request_async_ob	http://demo.sap.com/bridge/sync/async	AB_DEMO_

Users

Add or remove users

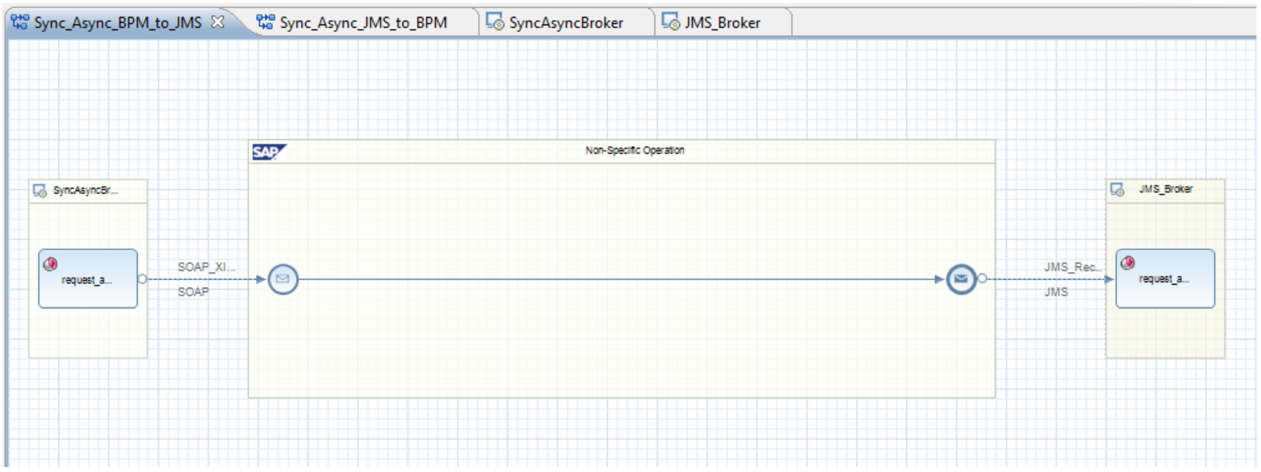
Receiver Interfaces

Add or remove receiver interfaces

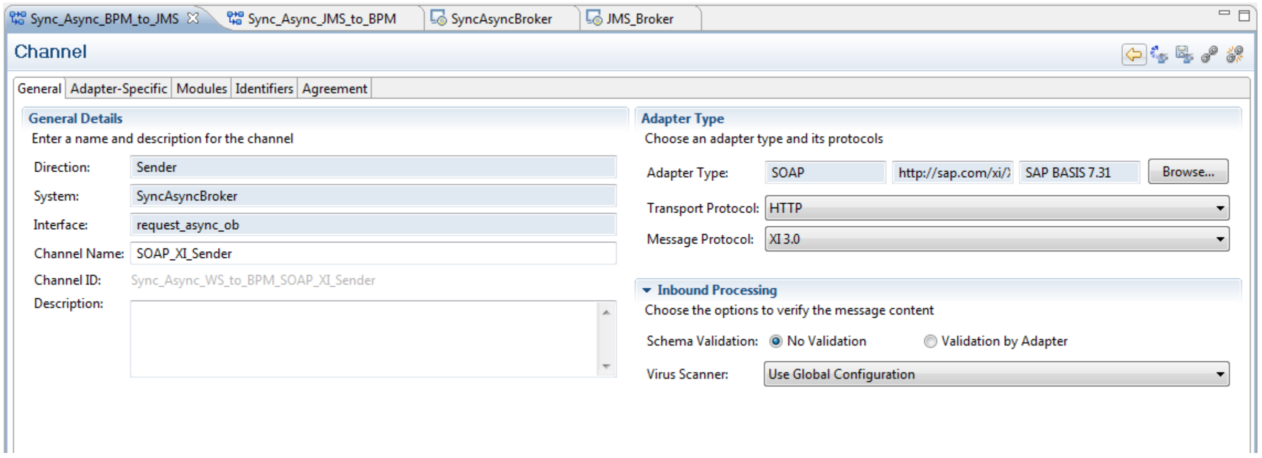
Name	Namespace	Software C
response_async_ib	http://demo.sap.com/bridge/sync/async	AB_DEMO_

Create a new Integration Flow describing the routing from the BPM process to the JMS broker.

- Assign beforehand created Business Component **SyncAsyncBroker** and the sender interface **request_async_ob** to the sender of the Integration Flow
- Assign Business Component **JMS_Broker** and the receiver interface **request_async_ib** to the receiver of the Integration Flow
- Maintain sender and receiver channel as described further below



Maintain sender channel of adapter type *SOAP*, and message protocol *XI 3.0*.



Channel

General Details
Enter a name and description for the channel

Direction:	Sender
System:	SyncAsyncBroker
Interface:	request_async_ob
Channel Name:	SOAP_XI_Sender
Channel ID:	Sync_Async_WS_to_BPM_SOAP_XI_Sender
Description:	

Adapter Type
Choose an adapter type and its protocols

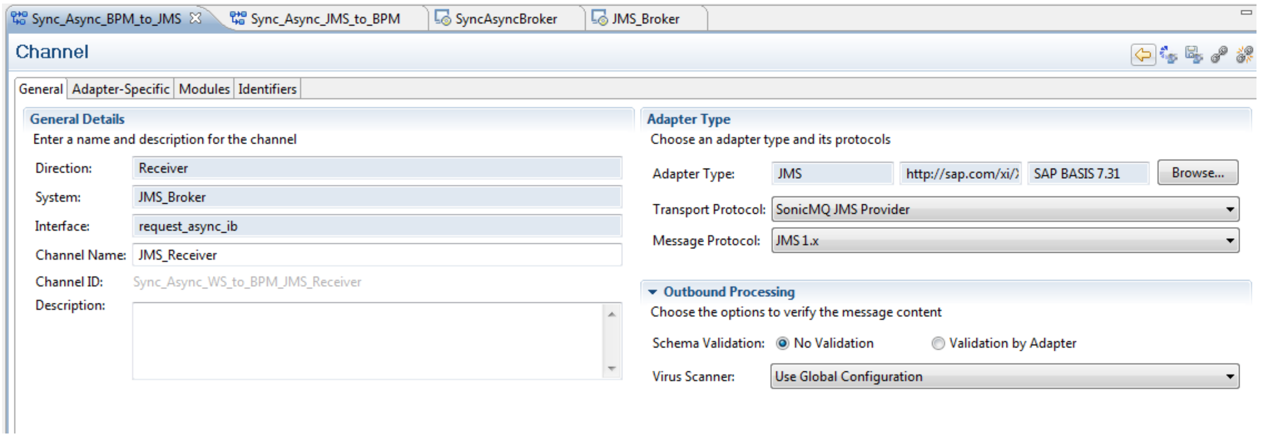
Adapter Type:	SOAP · http://sap.com/xi/X · SAP BASIS 7.31
Transport Protocol:	HTTP
Message Protocol:	XI 3.0

Inbound Processing
Choose the options to verify the message content

Schema Validation: [✓ No Validation] Validation by Adapter

Virus Scanner: Use Global Configuration

Maintain receiver channel of adapter type *JMS*.



Channel

General Details
Enter a name and description for the channel

Direction:	Receiver
System:	JMS_Broker
Interface:	request_async_ib
Channel Name:	JMS_Receiver
Channel ID:	Sync_Async_WS_to_BPM_JMS_Receiver
Description:	

Adapter Type
Choose an adapter type and its protocols

Adapter Type:	JMS · http://sap.com/xi/X · SAP BASIS 7.31
Transport Protocol:	SonicMQ JMS Provider
Message Protocol:	JMS 1.x

Outbound Processing
Choose the options to verify the message content

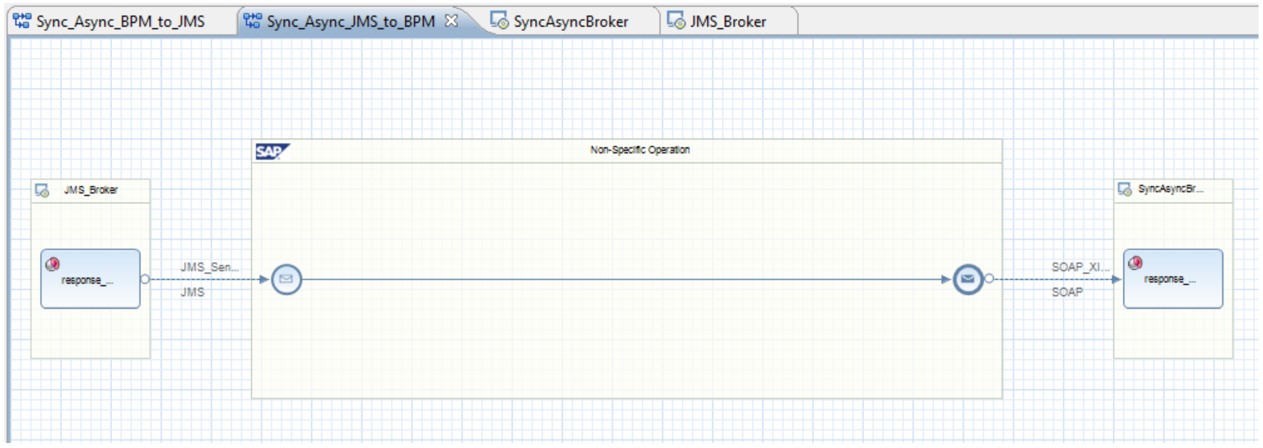
Schema Validation: [✓ No Validation] Validation by Adapter

Virus Scanner: Use Global Configuration

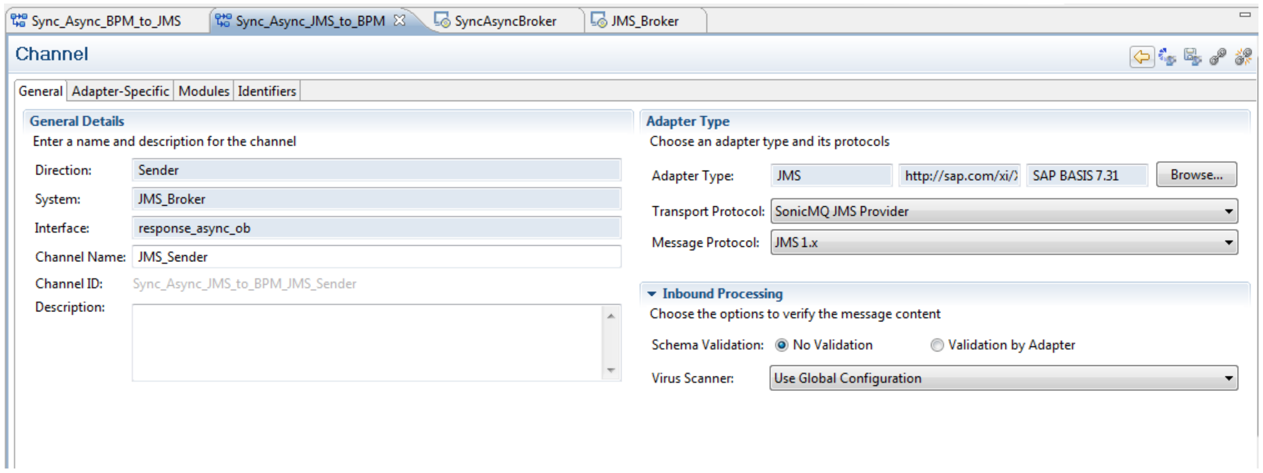
Integration Flow from JMS Broker to BPM Process

Create a new Integration Flow describing the routing from the JMS broker to the BPM process.

- Assign Business Component **JMS_Broker** and the sender interface **response_async_ob** to the sender of the Integration Flow
- Assign Business Component **SyncAsyncBridge** and the receiver interface **response_async_ib** to the receiver of the Integration Flow
- Maintain sender and receiver channel as described further below



Maintain sender channel of adapter type *JMS*.



General Details
Enter a name and description for the channel

Direction:	Sender
System:	JMS_Broker
Interface:	response_async_ob
Channel Name:	JMS_Sender
Channel ID:	Sync_Async_JMS_to_BPM_JMS_Sender
Description:	

Adapter Type
Choose an adapter type and its protocols

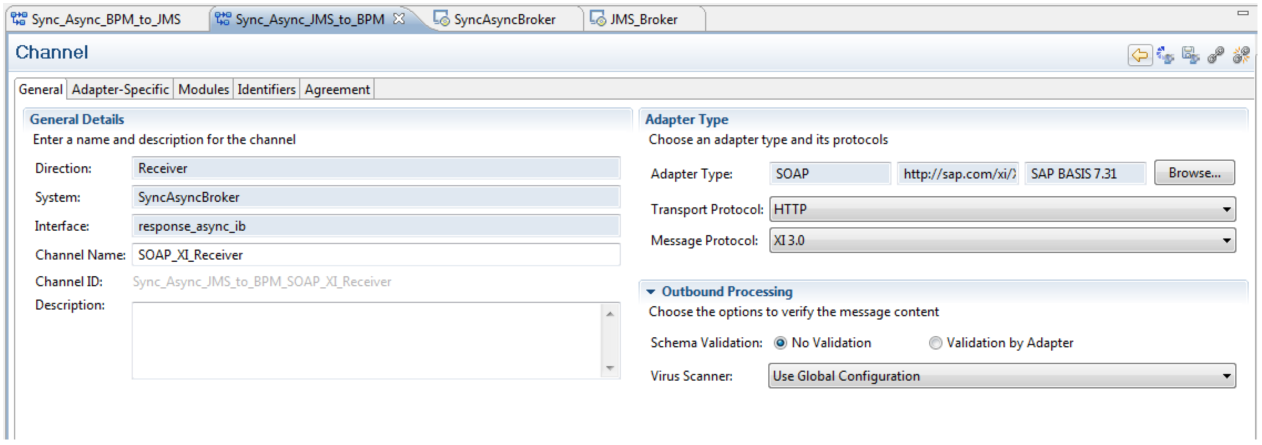
Adapter Type:	JMS http://sap.com/xi/) SAP BASIS 7.31
Transport Protocol:	SonicMQ JMS Provider
Message Protocol:	JMS 1.x

Inbound Processing
Choose the options to verify the message content

Schema Validation: [✓ No Validation] Validation by Adapter

Virus Scanner: Use Global Configuration

Maintain receiver channel of adapter type *SOAP*, and message protocol *XI 3.0*.



General Details
Enter a name and description for the channel

Direction:	Receiver
System:	SyncAsyncBroker
Interface:	response_async_ib
Channel Name:	SOAP_XI_Receiver
Channel ID:	Sync_Async_JMS_to_BPM_SOAP_XI_Receiver
Description:	

Adapter Type
Choose an adapter type and its protocols

Adapter Type:	SOAP http://sap.com/xi/) SAP BASIS 7.31
Transport Protocol:	HTTP
Message Protocol:	XI 3.0

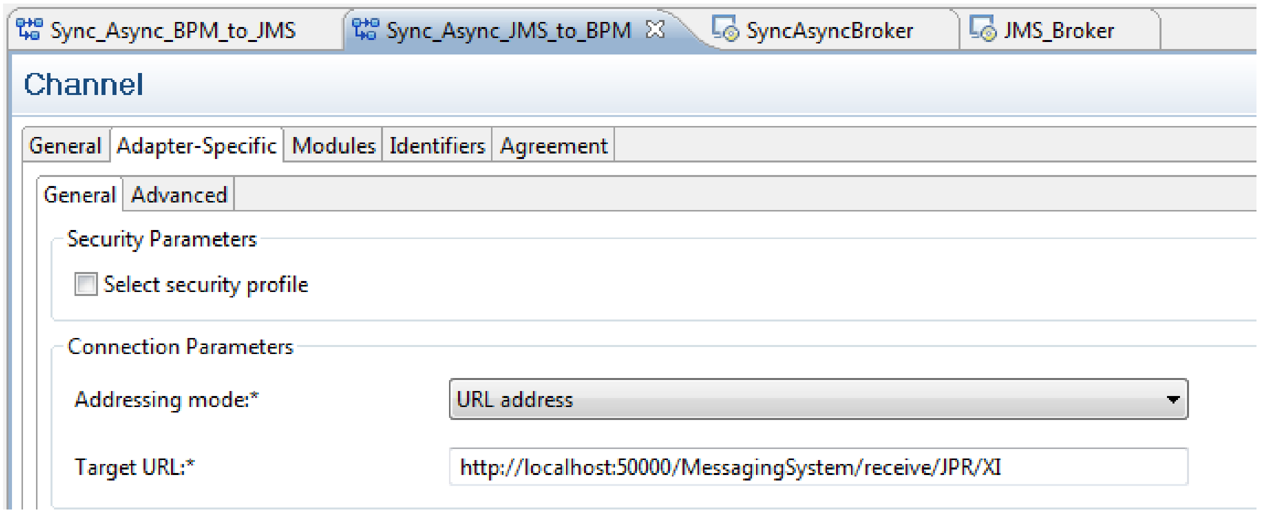
Outbound Processing
Choose the options to verify the message content

Schema Validation: [✓ No Validation] Validation by Adapter

Virus Scanner: Use Global Configuration

Switch to tab *Adapter-Specific*. The target URL needs to point to the Java Proxy runtime running on the very same system. Maintain *Target URL* as follows:

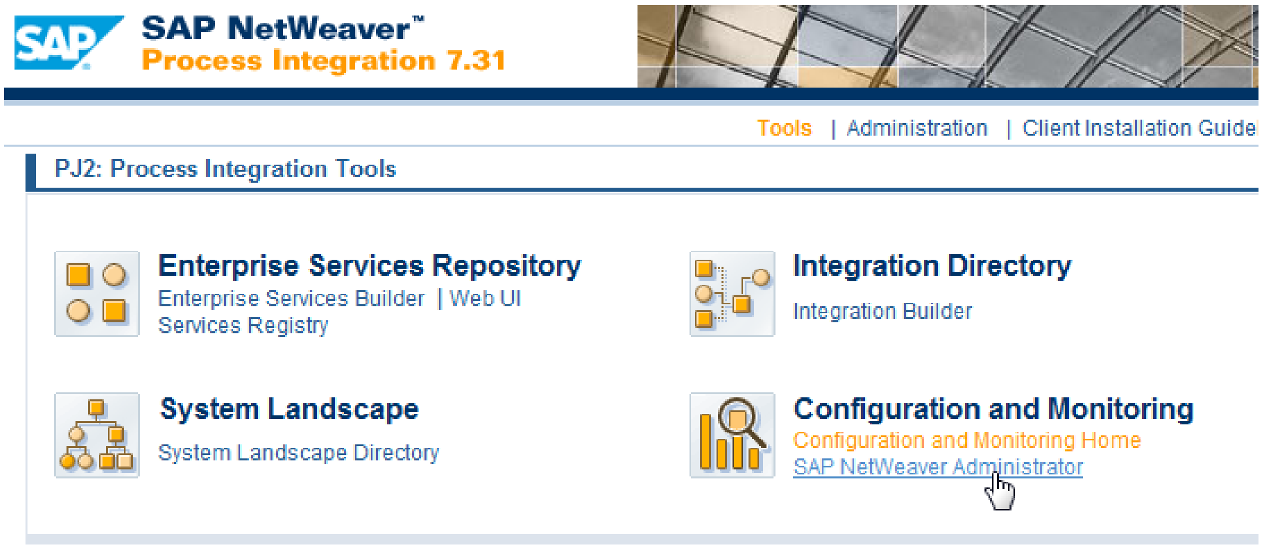
http://<host>:<port>/MessagingSystem/receive/JPR/XI



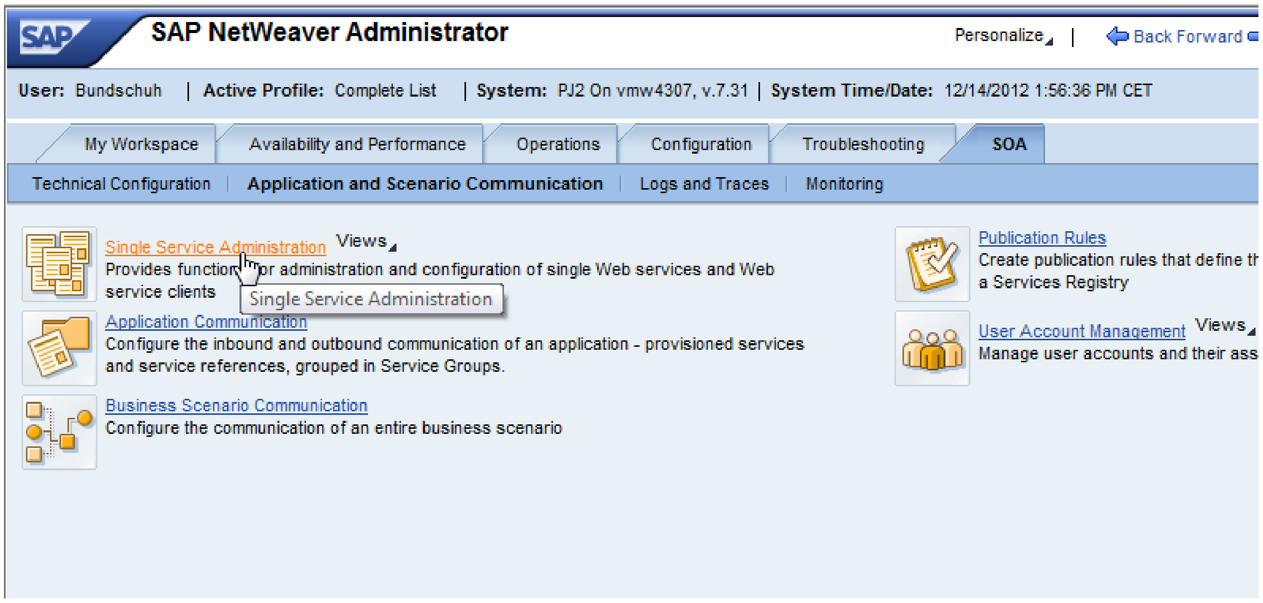
Addressing mode:*	URL address
Target URL:*	http://localhost:50000/MessagingSystem/receive/JPR/XI

Runtime

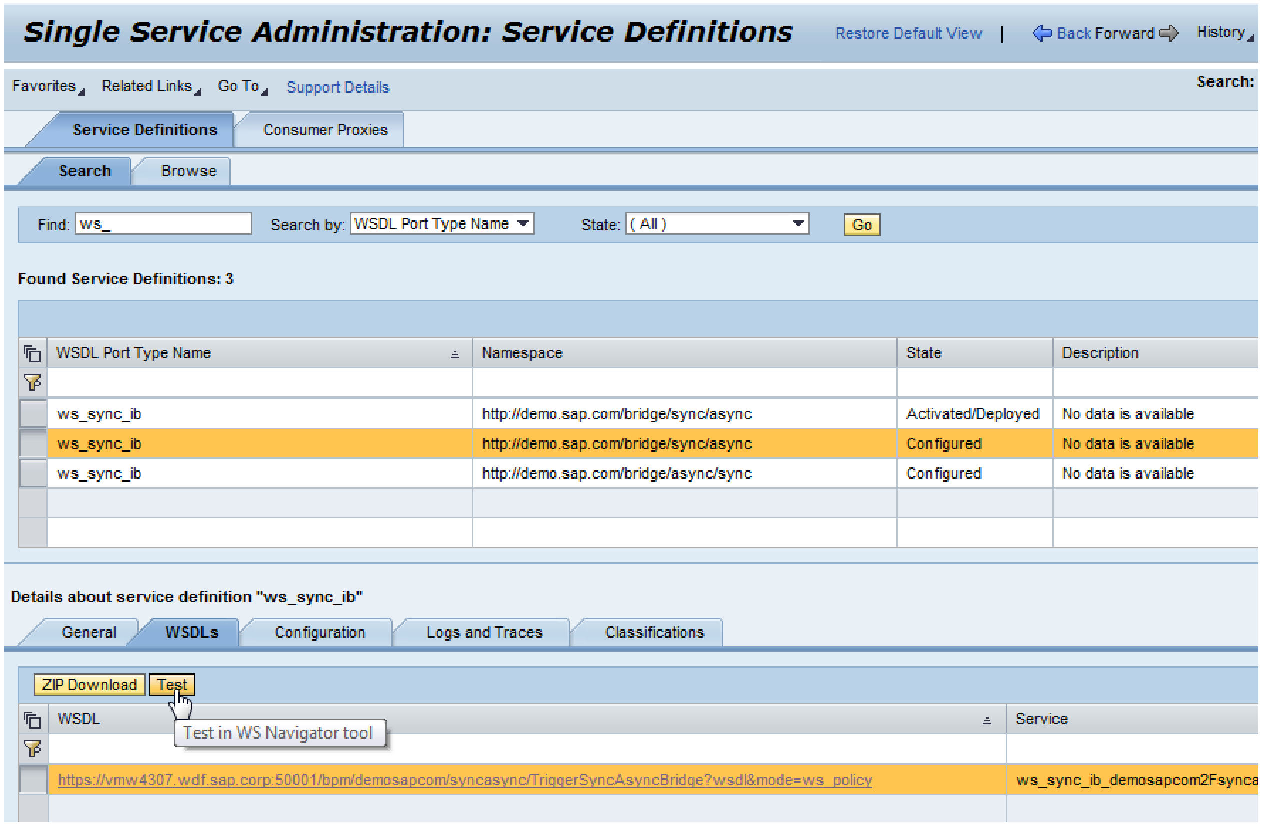
We run the scenario by calling the synchronous end point of the BPM process. Launch the *SAP NetWeaver Administrator*.


SAP NetWeaver™
Process Integration 7.31
Tools | Administration | Client Installation Guide
PJ2: Process Integration Tools
Enterprise Services Repository
Enterprise Services Builder | Web UI
Services Registry
Integration Directory
Integration Builder
System Landscape
System Landscape Directory
Configuration and Monitoring
Configuration and Monitoring Home
SAP NetWeaver Administrator

Switch to tab *SOA*, and sub-tab *Application and Scenario Communication*. Click on link *Single Service Administration.*


SAP NetWeaver Administrator
Personalize | Back Forward
User: Bundschuh | Active Profile: Complete List | System: PJ2 On vmw4307, v.7.31 | System Time/Date: 12/14/2012 1:56:36 PM CET
My Workspace
Availability and Performance
Operations
Configuration
Troubleshooting
SOA
Technical Configuration
Application and Scenario Communication
Logs and Traces
Monitoring
Single Service Administration
Views
Provides functions for administration and configuration of single Web services and Web service clients
Single Service Administration
Application Communication
Configure the inbound and outbound communication of an application - provisioned services and service references, grouped in Service Groups.
Business Scenario Communication
Configure the communication of an entire business scenario
Publication Rules
Create publication rules that define th
a Services Registry
User Account Management
Views
Manage user accounts and their ass

Search for the right service definition. The *WSDL Port Type Name* corresponds to the name of the synchronous service interface of the message start event of the BPM process. Select the *WS* end point, and switch to tab *WSDLs*. Here, click on button *Test*.



Find: ws_ Search by: WSDL Port Type Name State: (All)

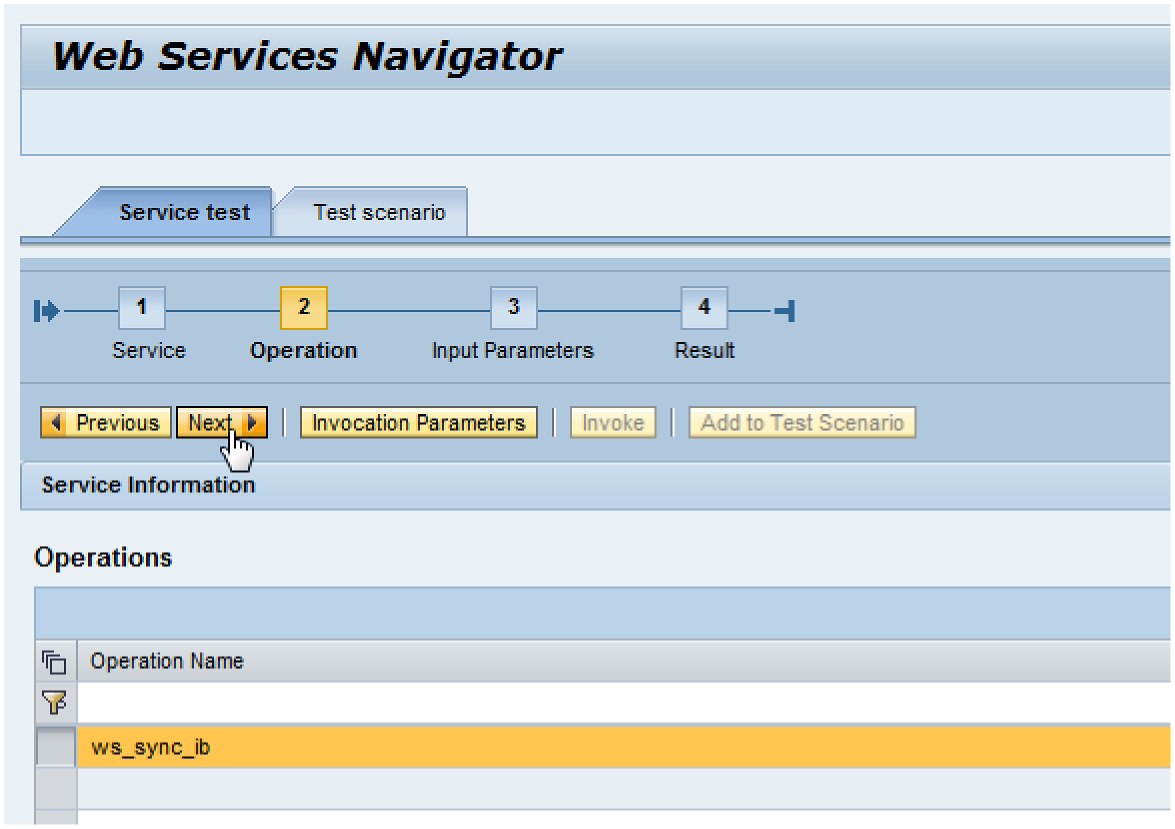
Found Service Definitions: 3

WSDL Port Type Name	Namespace	State	Description
ws_sync_ib	http://demo.sap.com/bridge/sync/async	Activated/Deployed	No data is available
ws_sync_ib	http://demo.sap.com/bridge/sync/async	Configured	No data is available
ws_sync_ib	http://demo.sap.com/bridge/async/sync	Configured	No data is available

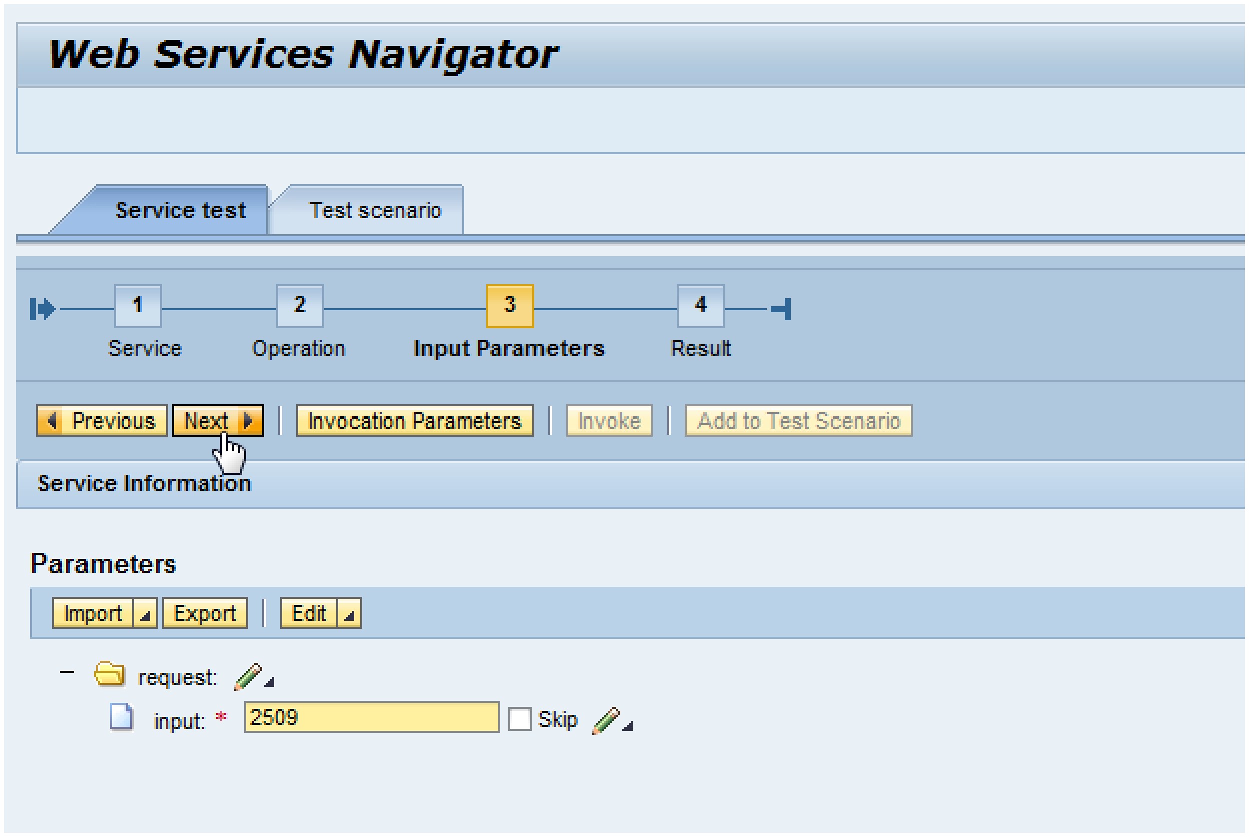
Details about service definition "ws_sync_ib"

WSDL	Service
https://vmw4307.wdf.sap.corp:50001/bpm/demosapcom/syncasync/TriggerSyncAsyncBridge?wsdl&mode=ws_policy	ws_sync_ib_demosapcom2Fsynca

This brings up the *Web Services Navigator*. Click on *Next*.

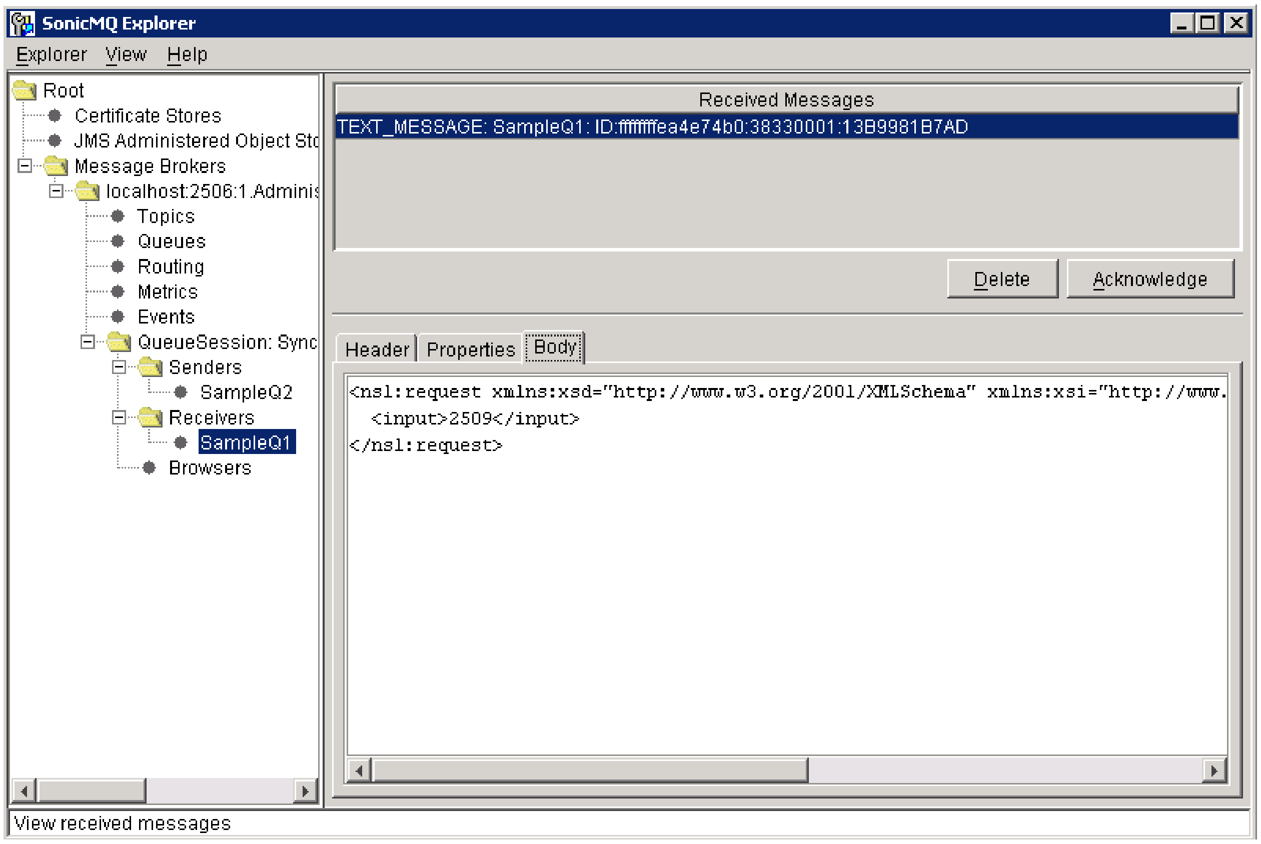


Enter any string into the *input* field, and click on *Next*.



input: * 2509

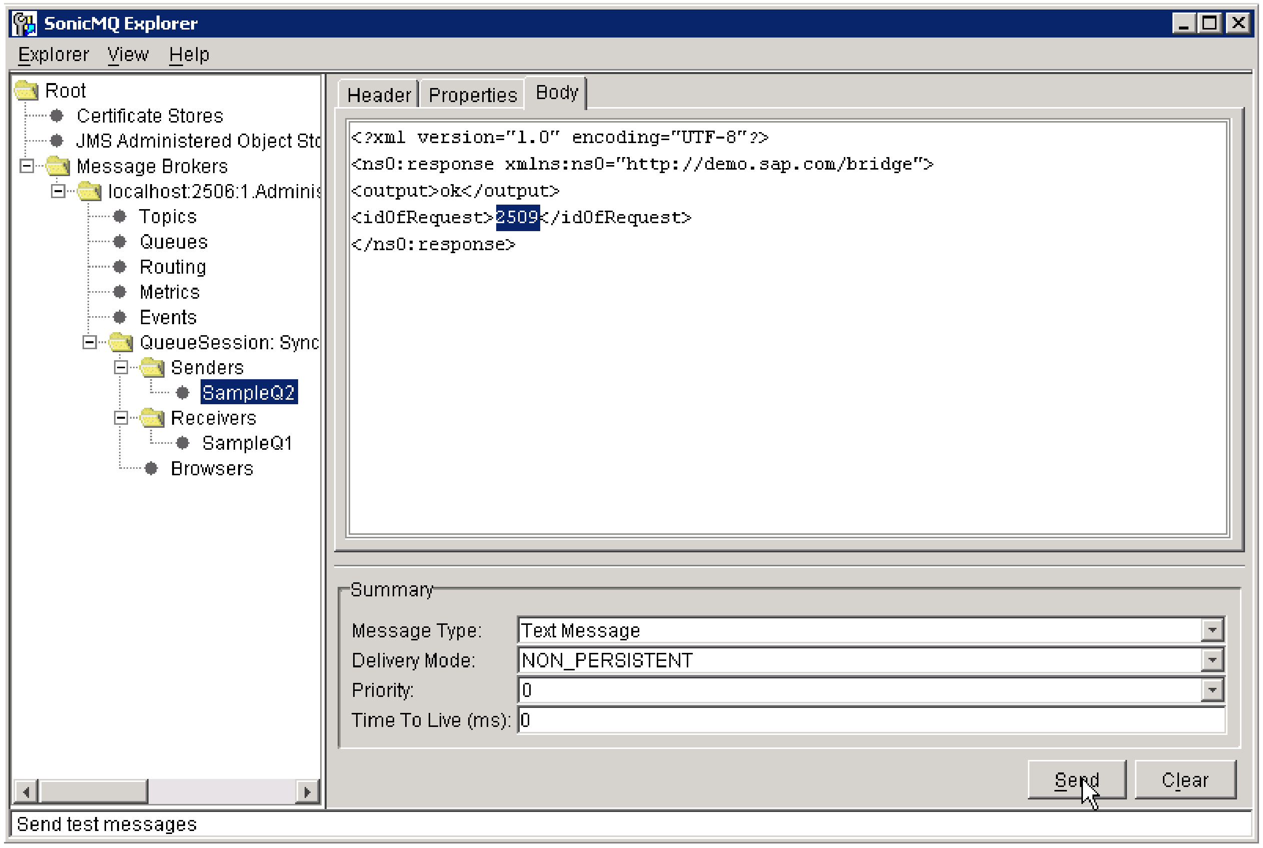
Open the *SonicMQ Explorer*. A new message was put into the *SampleQ1* receiver queue. Copy the *input* field into the clipboard.



TEXT_MESSAGE: SampleQ1: ID:ffffffffea4e74b0:38330001:13B9981B7AD

```
<ns1:request xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.
  <input>2509</input>
</ns1:request>
```

Before sending the response message, we paste the beforehand copied ID into the *idOfRequest* field. Once we put the message into the *SampleQ2* sender queue, it is read from the JMS adapter, and passed to the BPM process.

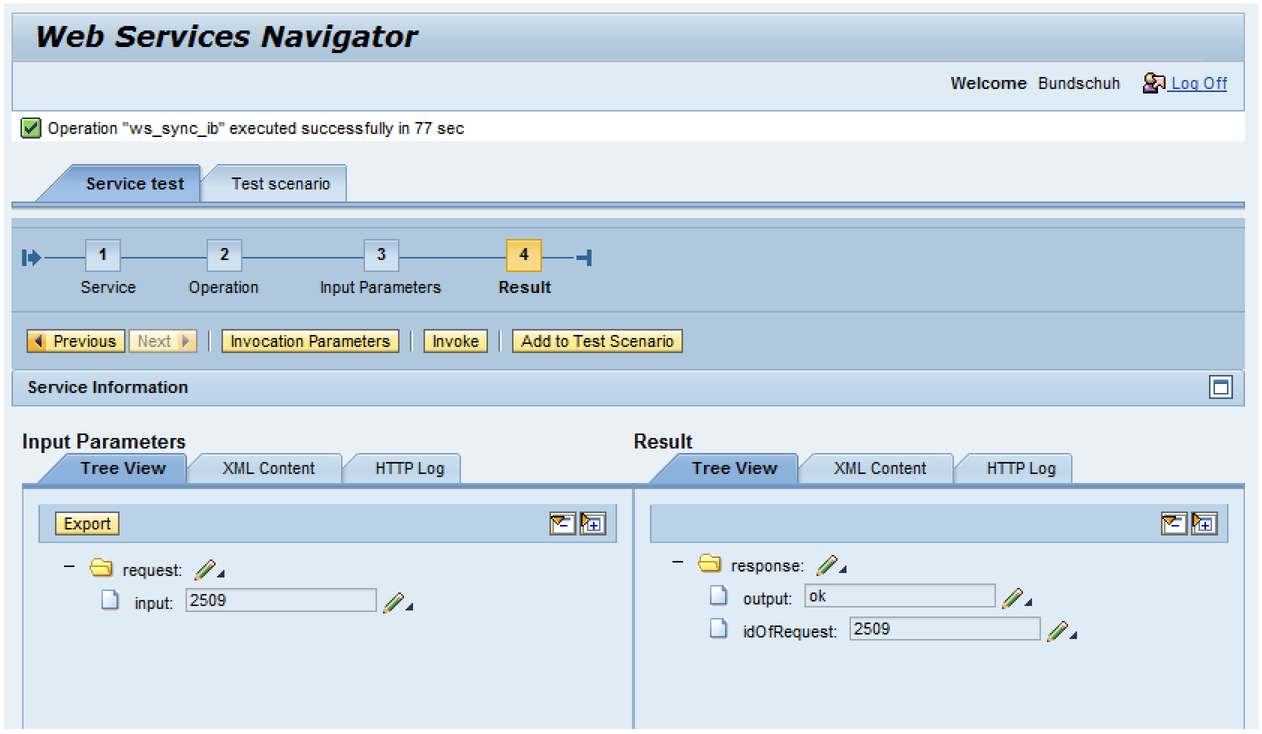


```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:response xmlns:ns0="http://demo.sap.com/bridge">
<output>ok</output>
<idOfRequest>2509</idOfRequest>
</ns0:response>
```

Summary

Message Type:	Text Message
Delivery Mode:	NON_PERSISTENT
Priority:	0
Time To Live (ms):	0

The response message is correlated to the respective process instance, whereas the process returns the synchronous response to the web service client, i.e., the Web Services Navigator. You will get a confirmation that the operation executed successfully.


Web Services Navigator

Welcome Bundschuh Log Off

Operation "ws_sync_ib" executed successfully in 77 sec

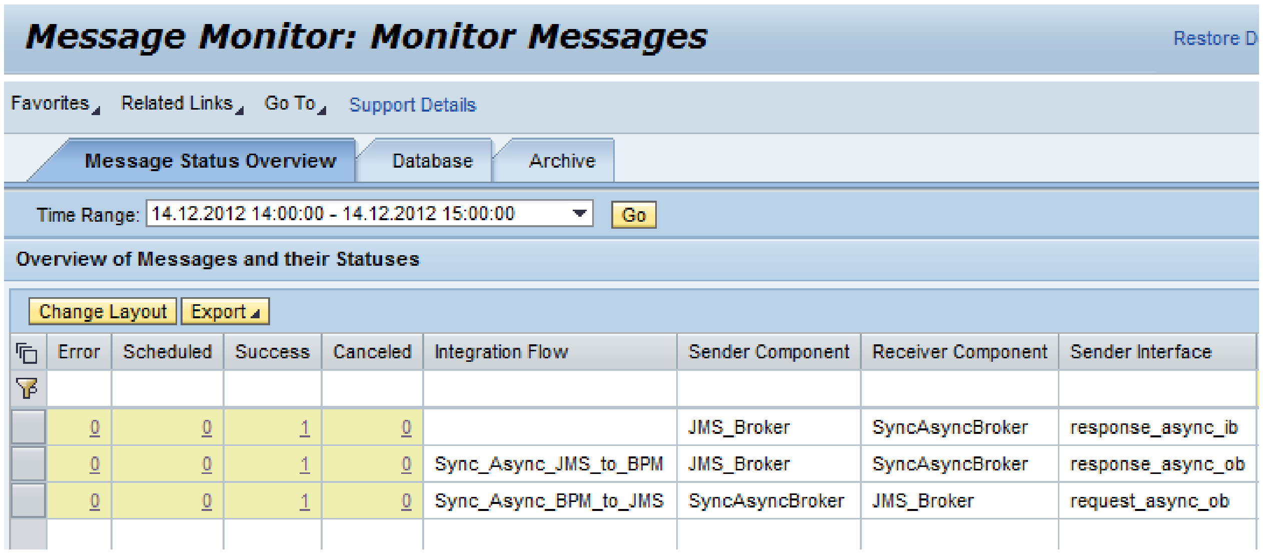
Input Parameters

request:
input: 2509

Result

response:
output: ok
idOfRequest: 2509

In the *Message Status Overview*, you will see three successful messages. From AEX to BPM, the message is persisted twice, both in the proxy system and in the messaging system. From BPM to AEX, the message is persisted only once benefitting from a proxy shortcut, hence leading to a runtime improvement. Click on the third message to navigate to the details of the request message.

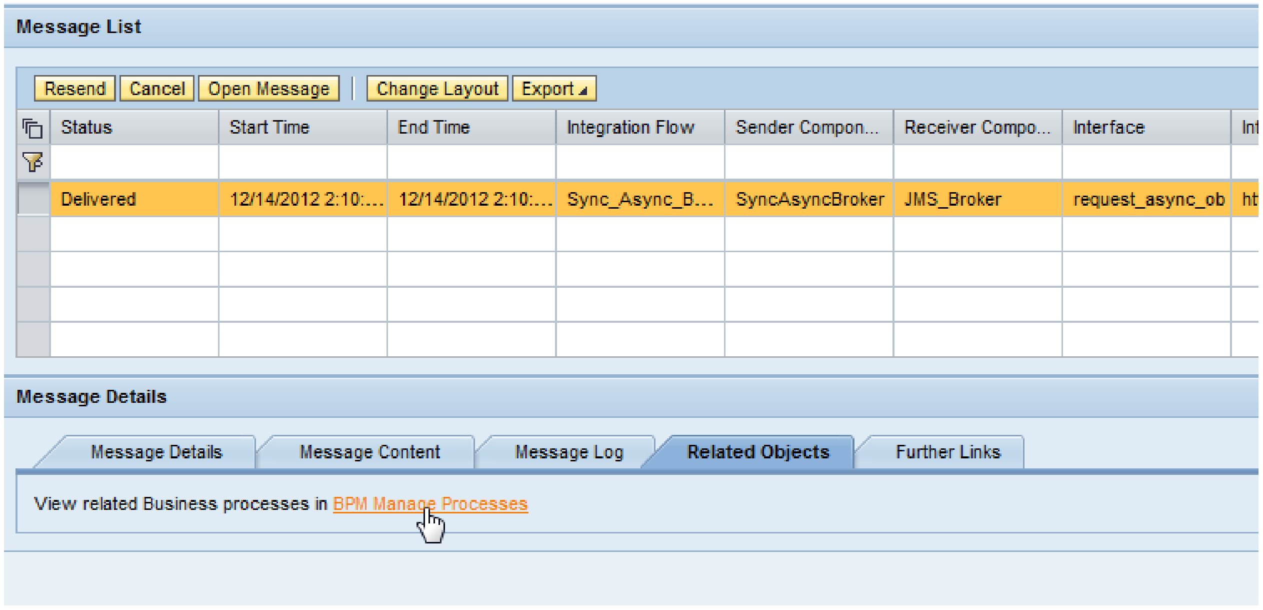

Message Monitor: Monitor Messages

Time Range: 14.12.2012 14:00:00 - 14.12.2012 15:00:00

Overview of Messages and their Statuses

Error	Scheduled	Success	Canceled	Integration Flow	Sender Component	Receiver Component	Sender Interface
0	0	1	0		JMS_Broker	SyncAsyncBroker	response_async_ib
0	0	1	0	Sync_Async_JMS_to_BPM	JMS_Broker	SyncAsyncBroker	response_async_ob
0	0	1	0	Sync_Async_BPM_to_JMS	SyncAsyncBroker	JMS_Broker	request_async_ob

Switch to tab *Related Objects*, and select link *BPM Manage Processes* to navigate to the process monitor.

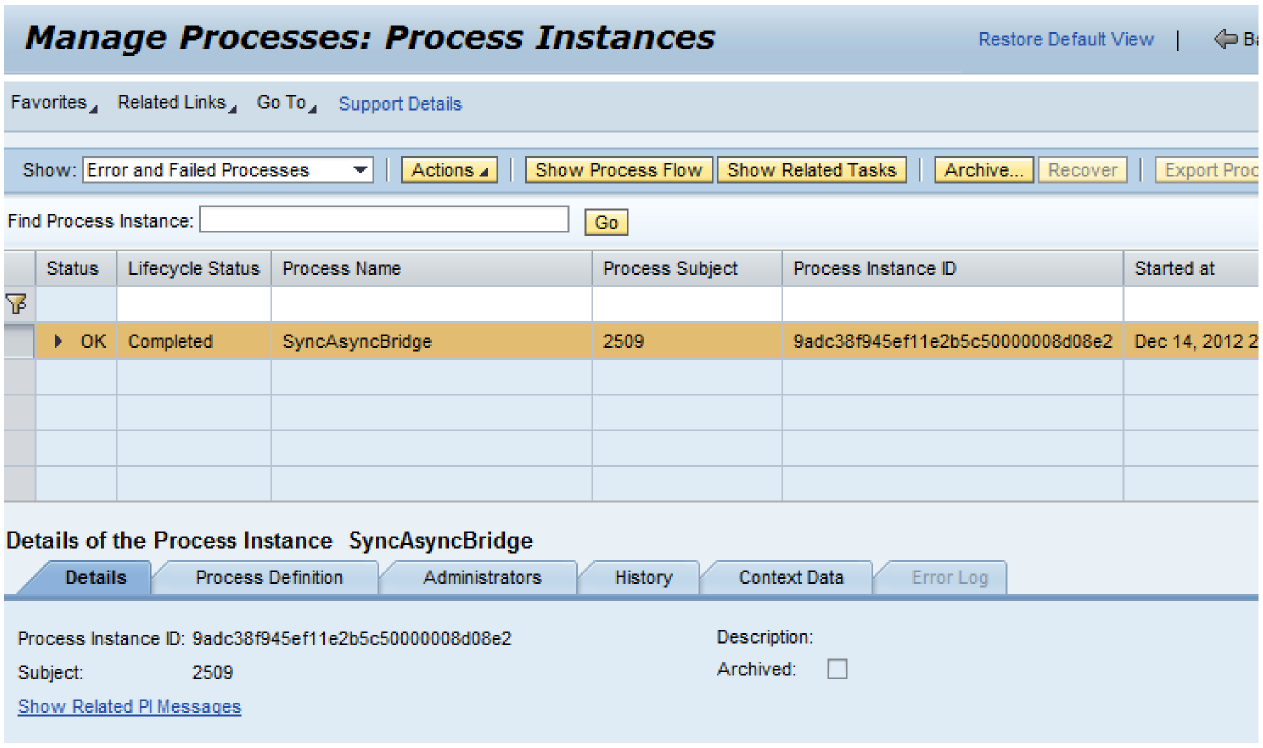


Status	Start Time	End Time	Integration Flow	Sender Compon...	Receiver Compo...	Interface	Int
Delivered	12/14/2012 2:10:...	12/14/2012 2:10:...	Sync_Async_B...	SyncAsyncBroker	JMS_Broker	request_async_ob	ht

View related Business processes in BPM Manage Processes

The navigation is context sensitive, i.e., the result is restricted to the process instance that the message is related to.

Note: The context sensitive navigation between the PI message monitor and the BPM process instance monitor is supported from release 7.31 SP6 on only.



Show: Error and Failed Processes

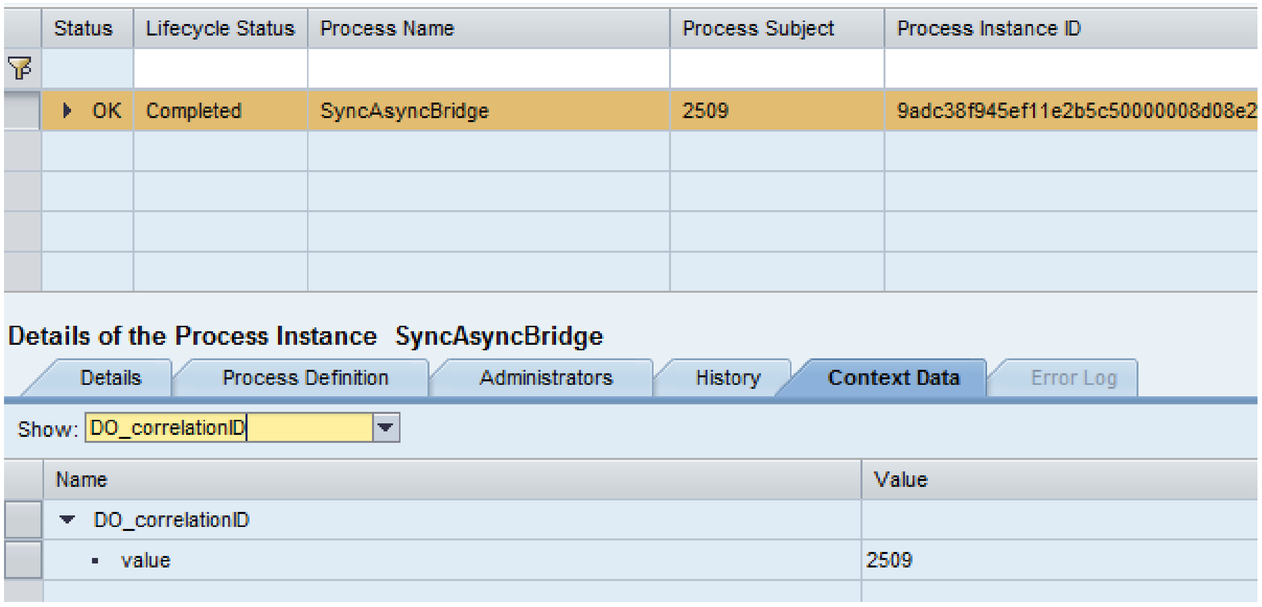
Find Process Instance:

Status	Lifecycle Status	Process Name	Process Subject	Process Instance ID	Started at
OK	Completed	SyncAsyncBridge	2509	9adc38f945ef11e2b5c50000008d08e2	Dec 14, 2012 2

Process Instance ID: 9adc38f945ef11e2b5c50000008d08e2
Subject: 2509
Show Related PI Messages

Description:
Archived: ☐

Switch to tab *Context Data*, and select *DO_correlation* in the *Show* drop down list to verify the correlation ID.



Status	Lifecycle Status	Process Name	Process Subject	Process Instance ID
OK	Completed	SyncAsyncBridge	2509	9adc38f945ef11e2b5c50000008d08e2

Details of the Process Instance SyncAsyncBridge

Show: DO_correlationID

Name	Value
DO_correlationID	
value	2509

Related Content

Blog on SCN: Installation Options for Process Integration and Orchestration Use Cases

SAP Help Portal: Configuring Async/Sync and Sync/Async Bridge in the JMS Adapter

SAP Help Portal: Configuring the JMS Adapter

How to guide on SCN: How to Correlate JMS Messages (NW7.0)

Guide on SCN: Configuring Async/Sync Bridge on SAP NetWeaver Process Orchestration

Copyright