Last Updated: April 12, 2011

ORACLE FUSION MIDDLEWARE

Oracle B2B Technical Note

Technical Note #029
Oracle B2B RosettaNet Attachments

Contents

Background

This tech note explains how to enable the attachment support for RosettaNet protocol in Oracle B2B 11g. It will explain how to create the attachment XML, the configuration required in Oracle B2B, how to pass the payload and attachment from the SOA Composite to Oracle B2B using JMS, and how to monitor and verify the attachments using the EM console and B2B reports.

Creating Attachment XML for RosettaNet

This process is same as in Oracle B2B 10g (see Appendix: Attachment Support for RosettaNet in Oracle B2B 10g for more information). The attachment XML must adhere to the following DTD:

```
<!ENTITY % common-attributes "id CDATA #IMPLIED" >
<!ELEMENT B2BAttachment (Attachment*)>
<!ELEMENT Attachment (#PCDATA) >
<!ATTLIST Attachment Type CDATA #REQUIRED
ID CDATA #REQUIRED
Encoding CDATA #REQUIRED
Description CDATA #IMPLIED
FileName CDATA #IMPLIED>
```

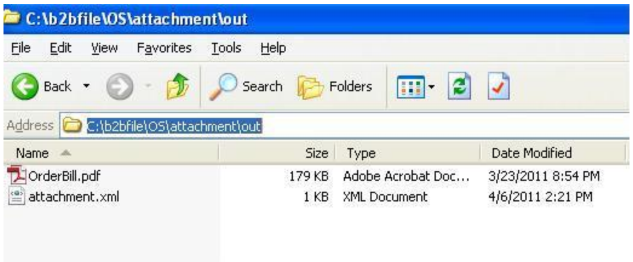
I created a XML based on above DTD which looks like -

```
<?xml version="1.0" encoding="UTF-8"?>
<B2BAttachment>
<Attachment ID="123456" Type="text/plain" Encoding="none" Description="Description">Fisrt Text
Attachment</Attachment>
<Attachment ID="123457" Type="text/plain" Encoding="none" Description="Description">Second Test
Attachment</Attachment>
<Attachment ID="123457" Type="application/pdf" Encoding="base64" Description="Description"
FileName="OrderBill.pdf">file:///C:/b2bfile/OS/attachment/out/OrderBill.pdf</Attachment>
</B2BAttachment>
```

There are three attachments included in this attachment XML:

i) A text attachment with content "First Text Attachment"
ii) Another text attachment with content "Second Test Attachment"
iii) A binary attachment (PDF) which is stored in file system in file "C:/b2bfile/OS/attachment/out/OrderBill.pdf"

Make sure to add "*Type*" and "*Encoding*" attributes correctly for all the attachments. Here "*Type*" attribute refers to MIME type and "*Encoding*" attribute refers to *Content-Transfer-Encoding*. I saved this attachment XML to directory "*C:/b2bfile/OS/attachment/out*". From this directory, composite will read the attachment XML and populate the attachment header.

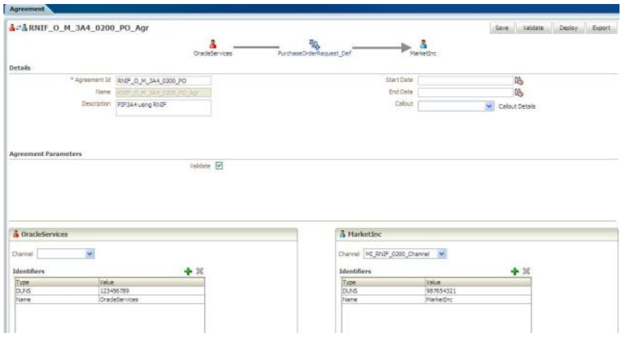


Address: C:\b2bfile\OS\attachment\out

Name	Size	Type	Date Modified
OrderBill.pdf	179 KB	Adobe Acrobat Doc...	3/23/2011 8:54 PM
attachment.xml	1 KB	XML Document	4/6/2011 2:21 PM

Configuring Oracle B2B

The first step is to create a RosettaNet agreement between the trading partners and deploy the agreement. This note assumes that you are already familiar with how to create a RosettaNet agreement as shown below:



Details

* Agreement Id: RNIF_O_M_3A4_0200_PO
Name: RNIF_O_M_3A4_0200_PO_Agr
Description: PIP3A4 using RNIF
Start Date:
End Date:
Callout:

Agreement Parameters

Validate ☑

OracleServices

Channel:

Identifiers

Type	Value
DUNS	123456789
Name	OracleServices

MarketInc

Channel: MI_RNIF_0200_Channel

Identifiers

Type	Value
DUNS	987654321
Name	MarketInc

As you can see the agreement "RNIF_O_M_3A4_0200_PO_Agr" has two participants – OracleServices (Host TP) and MarketInc (Remote TP). OracleServices is sending a PIP 3A4 PurchaseOrderRequest to MarketInc using this agreement.

Now we need to configure Oracle B2B for attachments.

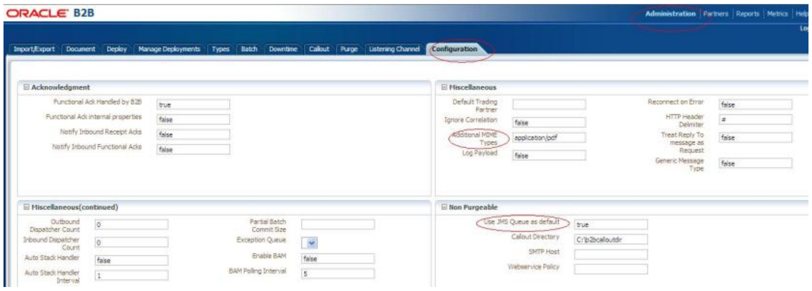
By default Oracle B2B supports the following MIME types:

text/plain : image/jpeg: text/xml : application/xml : application/octet-stream : application/EDIFACT : application/EDI-X12 : application/jpg : application/gzip : application/x-gzip : application/pkcs7-signature

If you need support for any additional MIME type then you have to add it in Oracle B2B configuration using the B2B UI.

For example, to support "application/pdf" to send PDF as attachment, we need to add it in Oracle B2B configuration. This can be done as follows:

Login to Oracle B2B console, navigate to Administration --> Configuration and in "Miscellaneous" section add "application/pdf" in "Additional MIME Types"



Acknowledgment

Functional Ack Handled by B2B: true
Functional Ack internal properties: false
Notify Inbound Receipt Acks: false
Notify Inbound Functional Acks: false

Miscellaneous

Default Trading Partner:
Ignore Correlation: false
Additional MIME Types: application/pdf
Log Payload: false
Reconnect on Error: false
HTTP Header Delimiter: #
Treat Reply To message as Request: false
Generic Message Type: false

Miscellaneous(continued)

Outbound Dispatcher Count: 0
Inbound Dispatcher Count: 0
Auto Stack Handler: false
Auto Stack Handler Interval: 1
Partial Batch Commit Size:
Exception Queue:
Enable BAM: false
BAM Polling Interval: 5

Non Purgeable

Use JMS Queue as default: true
Callout Directory: C:\b2bcalloutdir
SMTP Host:
Webservice Policy:

On the same page, set the "Use JMS Queue as default" to "true" so that B2B can start polling the default JMS queue (B2B_OUT_QUEUE) instead of the default AQ (IP_OUT_QUEUE) for messages.

If Oracle B2B is receiving the RN document with attachment then you need to add a property "b2b.AttachmentInboundDirectory" as described in B2B User Guide This property will hold the absolute directory name where incoming attachments will be stored. Make sure to restart the B2B server(s) after adding this property.

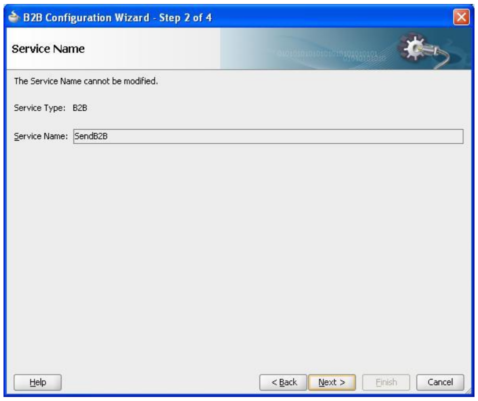
Creating SOA composite for passing the attachment and payload to Oracle B2B

Create a SOA composite to pass the message to Oracle B2B.

In JDeveloper create a SOA application and create a SOA project with an empty composite.

Drag the File Adapter into "Exposed Services" section of the composite and configure it for picking up 3A4 PurchaseOrderRequest from the local file system. Drag the B2B Adapter into the "External Reference" section of the composite and configure it as follows:

1. On Welcome screen click on "Next"
2. On Service Name screen provide the service name as "SendB2B" and click "Next"



B2B Configuration Wizard - Step 2 of 4

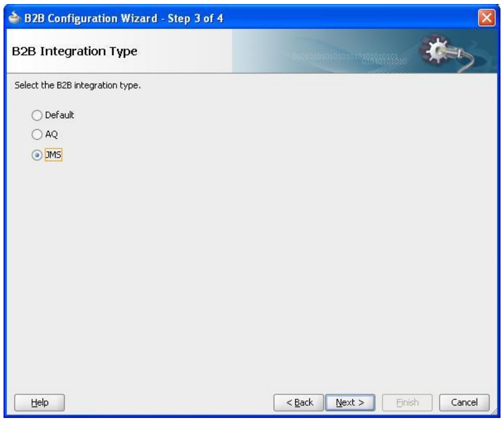
Service Name

The Service Name cannot be modified.

Service Type: B2B

Service Name: SendB2B

3. On "B2B Integration Type" screen, select "JMS"

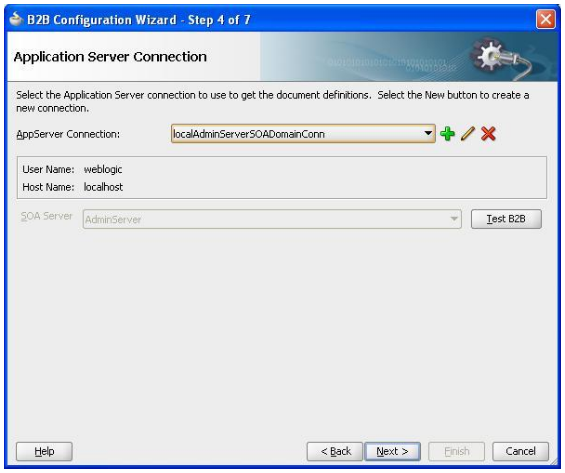


B2B Integration Type

Select the B2B integration type.

○ Default
○ AQ
● JMS

4. On "Application Server Connection" screen select the Application Server connection you have created or create a new one. This connection should be created to AdminServer of the B2B domain.



Application Server Connection

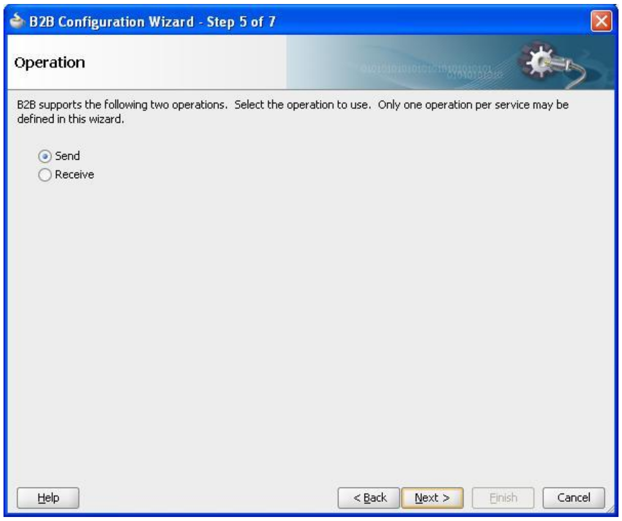
Select the Application Server connection to use to get the document definitions. Select the New button to create a new connection.

AppServer Connection: localAdminServerSOADomainConn

User Name: weblogic
Host Name: localhost

SOA Server: AdminServer

5. On the "Operation" screen, select Send



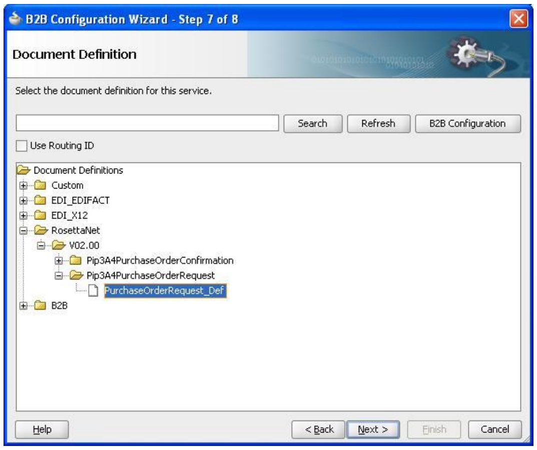
Operation

B2B supports the following two operations. Select the operation to use. Only one operation per service may be defined in this wizard.

● Send
○ Receive

6. On "Document Definition Handling" screen, just click on "Next"

7. On "Document Definition" screen, select the 3A4 PurchaseOrderRequest's Document Definition



B2B Configuration Wizard - Step 7 of 8

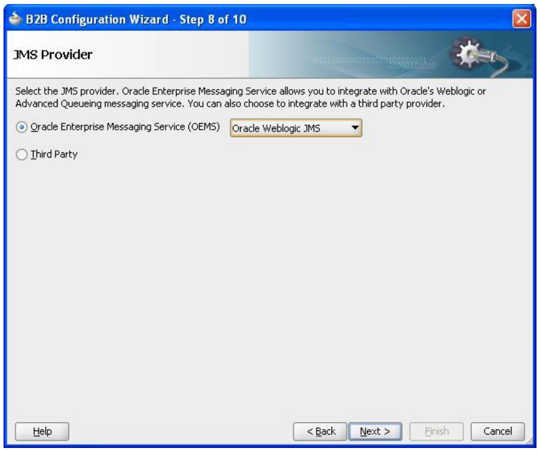
Document Definition

Select the document definition for this service.

□ Use Routing ID

- Document Definitions
  - Custom
  - EDI_EDIFACT
  - EDI_X12
  - RosettaNet
    - V02.00
      - Pip3A4PurchaseOrderConfirmation
      - Pip3A4PurchaseOrderRequest
        - PurchaseOrderRequest_Def
  - B2B

8. On "JMS Provider" screen select the "Oracle Weblogic JMS" from the drop-down against Oracle Enterprise Messaging Service (OEMS) option



B2B Configuration Wizard - Step 8 of 10

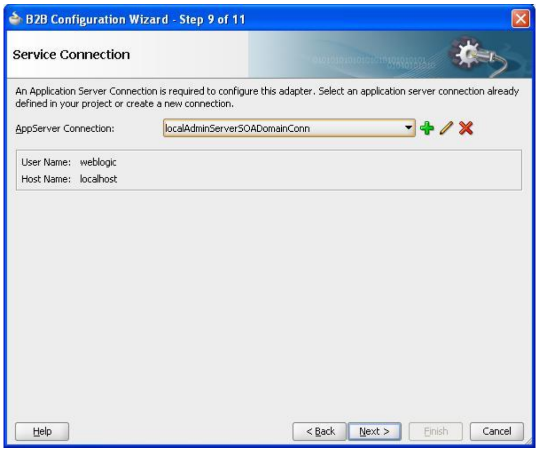
JMS Provider

Select the JMS provider. Oracle Enterprise Messaging Service allows you to integrate with Oracle's Weblogic or Advanced Queueing messaging service. You can also choose to integrate with a third party provider.

● Oracle Enterprise Messaging Service (OEMS) Oracle Weblogic JMS

○ Third Party

9. On "Service Connection" screen again select the Application Server Connection



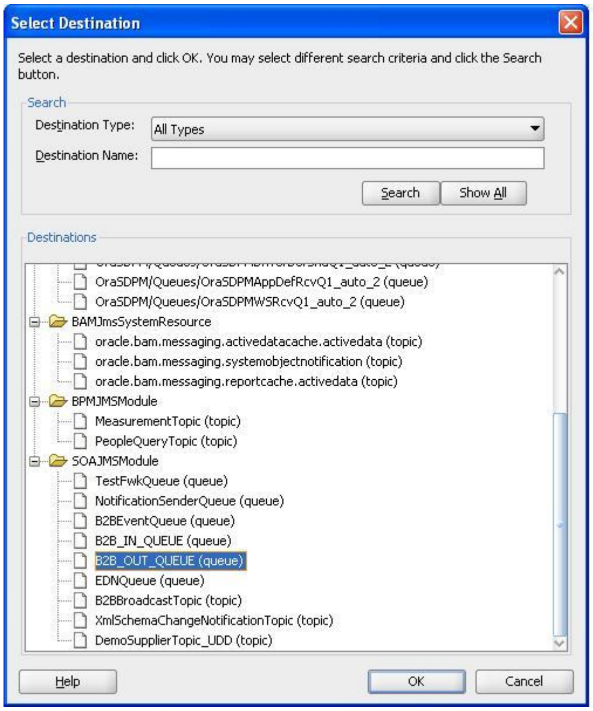
Service Connection

An Application Server Connection is required to configure this adapter. Select an application server connection already defined in your project or create a new connection.

AppServer Connection: localAdminServerSOADomainConn

User Name: weblogic
Host Name: localhost

10. On "Produce Operation Parameters" screen, click on the browse button against "Destination Name" and select "B2B_OUT_QUEUE (queue)" under "SOAJMSModule" and click ok. Accept the default for other settings and click on "Next"



Select a destination and click OK. You may select different search criteria and click the Search button.

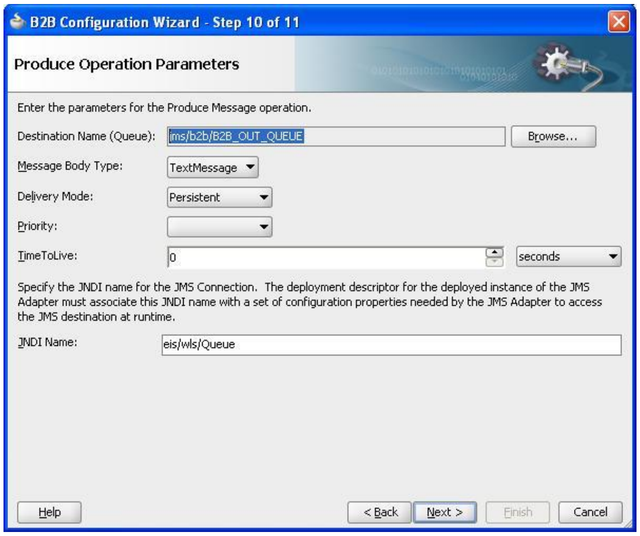
Search

Destination Type: All Types

Destination Name:

Destinations

- OraSDPM/Queues/OraSDPMAppDefRcvQ1_auto_2 (queue)
- OraSDPM/Queues/OraSDPMWSRcvQ1_auto_2 (queue)
- BAMJmsSystemResource
  - oracle.bam.messaging.activedatacache.activedata (topic)
  - oracle.bam.messaging.systemobjectnotification (topic)
  - oracle.bam.messaging.reportcache.activedata (topic)
- BPMJMSModule
  - MeasurementTopic (topic)
  - PeopleQueryTopic (topic)
- SOAJMSModule
  - TestFwkQueue (queue)
  - NotificationSenderQueue (queue)
  - B2BEventQueue (queue)
  - B2B_IN_QUEUE (queue)
  - B2B_OUT_QUEUE (queue)
  - EDNQueue (queue)
  - B2BBroadcastTopic (topic)
  - XmlSchemaChangeNotificationTopic (topic)
  - DemoSupplierTopic_UDD (topic)



B2B Configuration Wizard - Step 10 of 11

Produce Operation Parameters

Enter the parameters for the Produce Message operation.

Destination Name (Queue): jms/b2b/B2B_OUT_QUEUE

Message Body Type: TextMessage

Delivery Mode: Persistent

Priority:

TimeToLive: 0 seconds

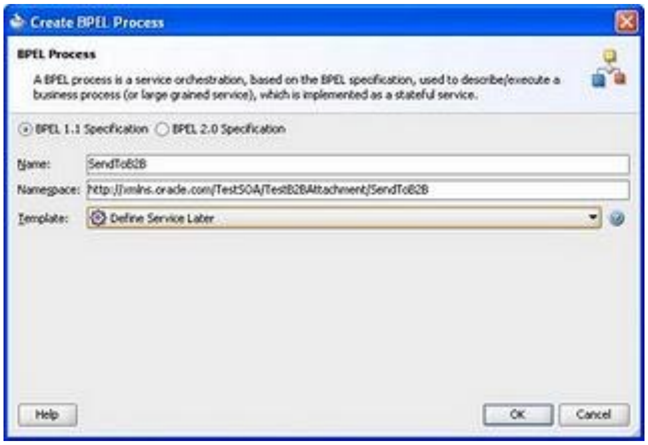
Specify the JNDI name for the JMS Connection. The deployment descriptor for the deployed instance of the JMS Adapter must associate this JNDI name with a set of configuration properties needed by the JMS Adapter to access the JMS destination at runtime.

JNDI Name: eis/wls/Queue

11. Click on "Finish" to generate the adapter configuration files

Now drag a BPEL process to the Components section of the composite and configure it as below:

Name – SendToB2B
Namespace – Accept the default
Template – Define Service Later



Create BPEL Process

BPEL Process

A BPEL process is a service orchestration, based on the BPEL specification, used to describe/execute a business process (or large grained service), which is implemented as a stateful service.

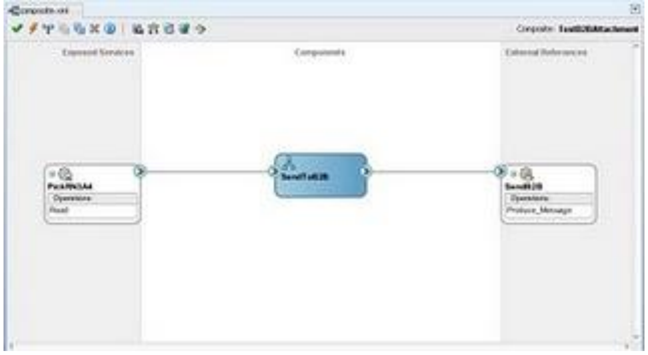
● BPEL 1.1 Specification ○ BPEL 2.0 Specification

Name: SendToB2B

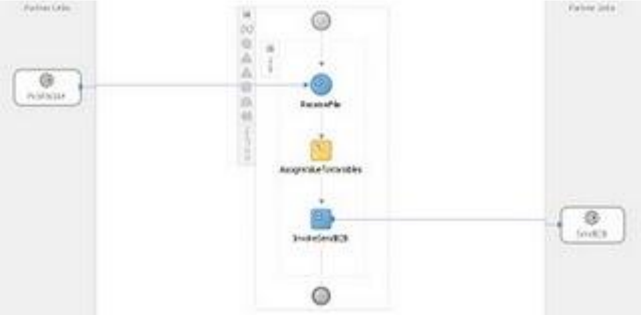
Namespace: http://xmlns.oracle.com/TestSOA/TestB2BAttachment/SendToB2B

Template: Define Service Later

Connect this BPEL process with the File Adapter and B2B by dragging wires appropriately as shown below:

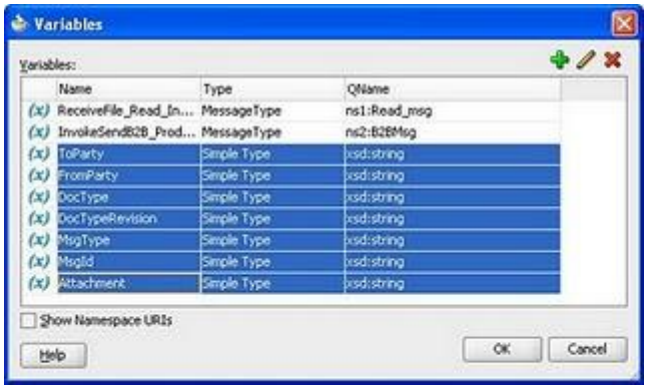


Double-click on BPEL Process SendToB2B to configure it. Add a receive activity in main scope to receive the 3A4 PurchaseOrderRequest payload from File Adapter and add an invoke activity to invoke the B2B adapter to send message to B2B. In between receive and invoke activity add an Assign activity to assign values to B2B required headers and attachment. The BPEL process should look as shown below:



In the main scope add the variables below of xsd:string type:

i) ToParty
ii) FromParty
iii) DocType
iv) DocTypeRevision
v) MsgType
vi) MsgId
vii) Attachment



Variables

Variables:

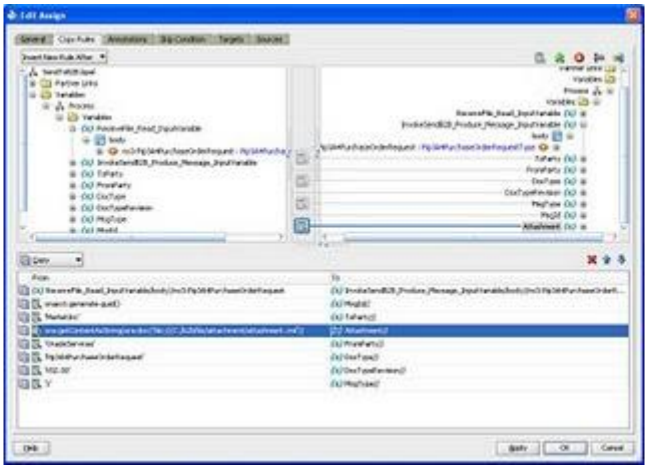
	Name	Type	QName
(x)	ReceiveFile_Read_In...	MessageType	ns1:Read_msg
(x)	InvokeSendB2B_Prod...	MessageType	ns2:B2BMsg
(x)	ToParty	Simple Type	xsd:string
(x)	FromParty	Simple Type	xsd:string
(x)	DocType	Simple Type	xsd:string
(x)	DocTypeRevision	Simple Type	xsd:string
(x)	MsgType	Simple Type	xsd:string
(x)	MsgId	Simple Type	xsd:string
(x)	Attachment	Simple Type	xsd:string

□ Show Namespace URIs

Edit the Assign activity of BPEL process and perform the following copy operations to assign values to above variables:

1. Copy the input variable (*ReceiveFile_Read_InputVariable*) body content of BPEL process to output variable (*InvokeSendB2B_Produce_Message_InputVariable*) body content. This will assign the 3A4 PurchaseOrderRequest payload read by file adapter to B2B adapter payload.
2. Copy the expression *oraext:generate-guid()* to variable *MsgId*
3. Copy the expression *'MarketInc'* to variable *ToParty*
4. Copy the expression *'OracleServices'* to variable *FromParty*
5. Copy the expression *'Pip3A4PurchaseOrderRequest'* to variable *DocType*
6. Copy the expression *'V02.00'* to variable *DocTypeRevision*
7. Copy the expression *'1'* to variable *MsgType*
8. Copy the expression*ora:getContentAsString(ora:doc('file:///C:/b2bfile/OS/attachment/out/attachment.xml'))* to variable *Attachment*

In the 8th copy operation, we are reading the content of the attachment XML using*ora:doc()* function and then converting the XML to string using the *ora:getContentAsString()* function



Edit Assign

From	To
(x) ReceiveFile_Read_InputVariable/body/ns3:Pip3A4PurchaseOrderRequest	(x) InvokeSendB2B_Produce_Message_InputVariable/body/ns3:Pip3A4PurchaseOrderR...
oraext:generate-guid()	(x) MsgId()
'MarketInc'	(x) ToParty()
ora:getContentAsString(ora:doc('file:///C:/b2bfile/attachment/attachment.xml'))	(x) Attachment()
'OracleServices'	(x) FromParty()
'Pip3A4PurchaseOrderRequest'	(x) DocType()
'V02.00'	(x) DocTypeRevision()
'1'	(x) MsgType()

Populate the JMS headers with the value of variables we assigned above. Go to the source of BPEL process and modify the invoke node as below:

```
<invoke name="InvokeSendB2B"
inputVariable="InvokeSendB2B_Produce_Message_InputVariable"
partnerLink="SendB2B" portType="ns2:Produce_Message_ptt"
operation="Produce_Message" bpelx:invokeAsDetail="no">
<bpelx:inputProperty name="jca.jms.JMSProperty.FROM_PARTY" variable="FromParty"/>
<bpelx:inputProperty name="jca.jms.JMSProperty.TO_PARTY" variable="ToParty"/>
<bpelx:inputProperty name="jca.jms.JMSProperty.DOCTYPE_NAME" variable="DocType"/>
<bpelx:inputProperty name="jca.jms.JMSProperty.DOCTYPE_REVISION" variable="DocTypeRevision"/>
<bpelx:inputProperty name="jca.jms.JMSProperty.MSG_TYPE" variable="MsgType"/>
<bpelx:inputProperty name="jca.jms.JMSProperty.MSG_ID" variable="MsgId"/>
<bpelx:inputProperty name="jca.jms.JMSProperty.ATTACHMENT" variable="Attachment"/>
</invoke>
```

Here the invoke node is populating the JMS properties required for processing in Oracle B2B with the values of the variables we declared.

Save your project and deploy it to the SOA server.

Monitoring the message with attachments

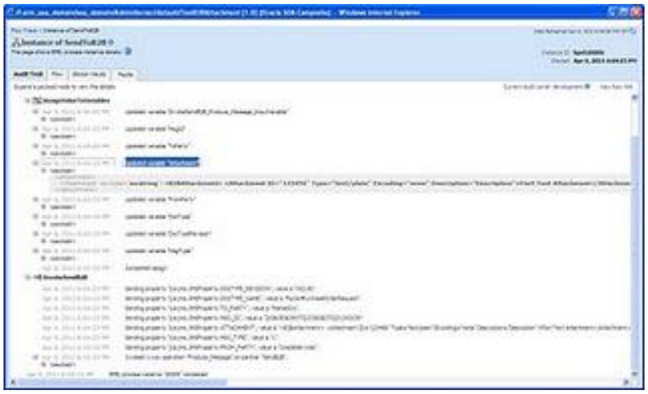
In order to test the configuration changes described above, drop a 3A4 PurchaseOrderRequest into the directory where the File Adapter is polling. Wait until the file gets deleted from that directory.

1. Login to the EM console and navigate to the SOA composite.

2. Under "Recent Instances" section, click on the Instance ID to open up the Flow Trace.

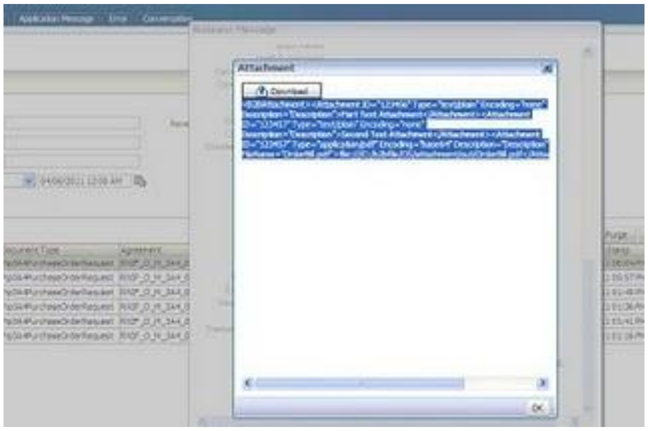
3. Click on "SendToB2B" component on Flow Trace to view the Audit Trail.

4. See the Assign Action Audit Trail and expand the payload under "Updated variable "Attachment" (You will only able to see payload if Audit level is set to Development). It should show you the exact content as the attachment XML. You can also see under the InvokeSendB2B audit trail that headers are being populated with exact values. Make sure that content of header "jca.jms.JMSProperty.ATTACHMENT" is same as our attachment XML.

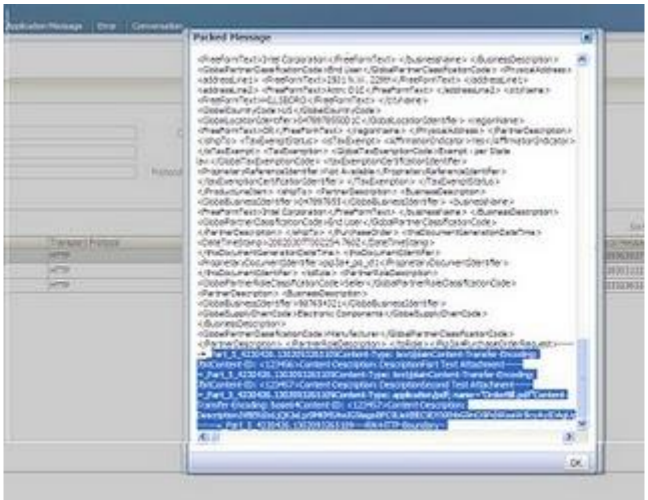


Login to B2B console and go to Reports section. In Business Message Reports, expand the details of 3A4

PurchaseOrderRequest which has been sent to TP. Scroll down and click on the link against "Attachment" in Business Message Report details. It must show you exactly same content as the attachment XML



To check whether the attachments have been included in the message sent to the TP, go to the Wire Message Reports and expand the details of 3A4 PurchaseOrderRequest which has been sent to the TP. In details, click on "Packed Message" and scroll down. You must be able to see the attachments at the bottom.



Conclusion

In summary, we are done with sending Text and Binary attachments with RosettaNet message using Oracle B2B and composite with JMS interface. We learned how to enable the attachment support for RosettaNet protocol in Oracle B2B 11g by explaining:

- How to create the attachment XML,
- How to configure the Oracle B2B to send attachments,
- How to pass the payload and attachment from the SOA Composite to Oracle B2B using JMS, and
- How to monitor and verify the attachments using the EM console and B2B reports

Appendix

Attachment Support for RosettaNet in Oracle B2B 10g

These following steps help in sending Text as an attachment in RosettaNet in Oracle B2B 10g.

1. Create an xml file, which follows the following dtd:

```
<!ENTITY % common-attributes "id CDATA #IMPLIED" >
<!ELEMENT B2BAttachment (Attachment*)>
<!ELEMENT Attachment (#PCDATA) >
<!ATTLIST Attachment Type CDATA #REQUIRED
ID CDATA #REQUIRED
Encoding CDATA #REQUIRED
Description CDATA #IMPLIED
FileName CDATA #IMPLIED>
```

2. Save the XML in UNIX FILE FORMAT. Here is the sample xml, which contains 2 attachments:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2005 sp1 U (http://www.xmlspy.com)-->
<B2BAttachment>
<Attachment ID="213123" Type="text/plain" Encoding="none" Description="Description">Sample Text message
</Attachment>

<Attachment ID="215123" Type="text/plain" Encoding="none" Description="Description">Sample Text message
another
</Attachment>
<Attachment ID="215123" Type="application/pdf" Encoding="base64" Description="Description"
FileName="Form.pdf">file:///C:/Form.pdf
</Attachment>
</B2BAttachment>
```

3. Edit the enq_3a4_req.properties to provide the file name for the "attachment" attribute

4. Provide the directory name in tip.properties for the key "oracle.tip.adapter.b2b.AttachmentInboundDirectory" where the file will be stored in the inbound scenario.
5. Restart the server and enqueue the Message.
6. By Default oracle.tip.adapter.b2b.MimeType accepts only below types,

```
@ application/xml
@ application/octet-stream
@ application/EDIFACT
@ application/EDI-X12
@ application/jpg
@ image/jpeg
@ application/gzip
@ application/x-gzip
@ application/pkcs7-signature
```

For more options you must modify tip.properties in $ORACLE_HOME/ip/config directory to include your

MimeTypes. For example, to include pdf attachments, the tip.properties would look like:

oracle.tip.adapter.b2b.MimeType=application/xml : application/octet-stream : application/EDIFACT : application/EDI-X12 : application/jpg : image/jpeg : application/gzip : application/x-gzip : application/pkcs7-signature : application/pdf

Add a new content type such as application/pdf, restart opmn and try sending the message with attachments as explained above.

Discuss this technical note in the SOA Suite Discussion Forum or Oracle B2B Discussion Forum.