

Oracle SOA Suite 11g

Oracle SOA Suite 11g HL7 Inbound Example

Back End-constructed Functional ACK

Addendum

michael@czapski.id.au

June 2010

Table of Contents

Introduction.....	1
Pre-requisites.....	1
HL7 v2 Receiver Solution	2
Inbound HL7 v2 Solution	3
HL7 ADT A01 and ACK Message Structures.....	3
Modify B2B Document Definition for ACK.....	4
Re-configure B2B Layer.....	5
Undeploy original HL7 Receiver.....	6
Build New HL7 Receiver Acker Composite	7
Exercise HL7 Inbound solution	18
Summary.....	23
References.....	23

Introduction

This article is a follow on to the “[Oracle SOA Suite 11g HL7 Inbound Example – Functional ACK Addendum](#)” article and the “[Oracle SOA Suite 11g HL7 Inbound – Customized HL7 Message Structure and Data Validation](#)” article. In these articles the B2B infrastructure was configured to return the “Functional ACK” when it validated each message. The ACK was a positive or a negative ACK depending on whether the message passed validation. The ACK was generated by the B2B Layer before the message was passed on to the SOA Layer.

In this article I expand on the previous posts by configuring the B2B Layer to pass the message to the SOA Layer and pass the Functional ACK, generated by the SOA Layer on to the requester. To process a message and produce the ACK we will build and deploy a new SOA Composite.

Pre-requisites

It is assumed that a Windows XP SP3 platform with the Oracle SOA Suite 11g, installed and configured as discussed in “Installing Oracle SOA Suite for HL7 Exploration”, published at http://blogs.czapski.id.au/wp-content/uploads/2010/06/01_Installing_Oracle_SOA_Suite_for_HL7_exploration_v1.1.pdf, is available and will be used for the work discussed in this article.

It is assumed that the HL7 solutions, discussed in blog articles [Oracle SOA Suite 11g HL7 Inbound Example](#), [Oracle SOA Suite 11g HL7 Inbound Example – Functional ACK Addendum](#) and [Oracle SOA Suite 11g HL7 Inbound – Customized HL7 Message Structure and Data Validation](#) are built and deployed.

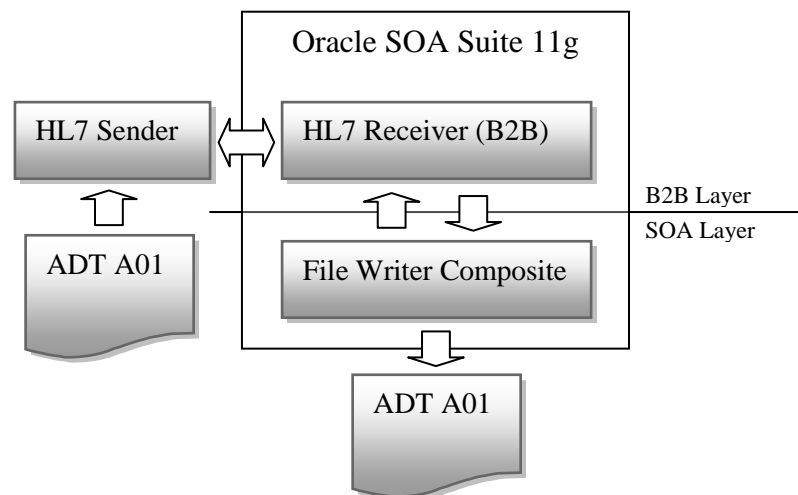
It is assumed that data in the archive, HL7_messages_sources.zip, has been unpacked to C:\hl7\adt\. This archive is available from http://blogs.czapski.id.au/wp-content/uploads/2010/06/HL7_messages_sources.zip.

It is assumed that the free HL7 Browser tool, “HL7 Browser 1.0”, available from its author’s page at <http://mac.softpedia.com/developer/Michael-Litherland-5914.html>, is available.

HL7 v2 Receiver Solution

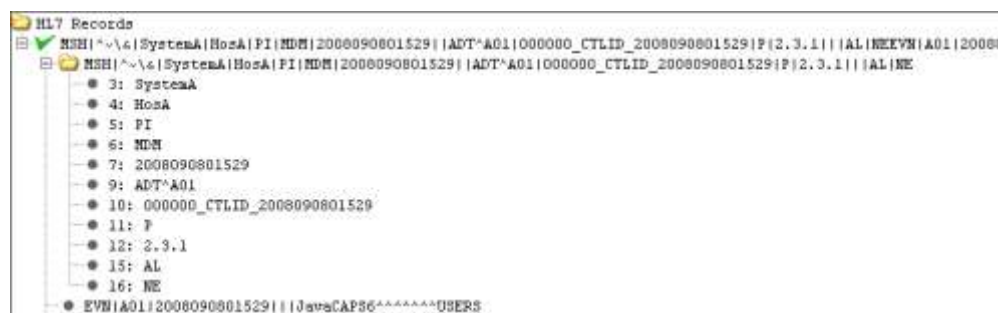
The solution we will be modifying is a HL7 Receiver, which will receive v2 delimited ADT A01 messages and will write them to files in the file system. This solution was discussed and built in the blog article [Oracle SOA Suite 11g HL7 Inbound Example – Functional ACK Addendum](#).

The solution consists of a B2B Listener Channel, to which HL7 v2 ADT A01 messages will be sent (B2b Layer), and a SOA Composite which will receive these messages and will write them to a file in the file system, and generate a HL7 ACK and return it to the B2B Layer for forwarding to the requester (SOA Layer).



Messages in the sample message set use the following identifiers:

Sending Application	SystemA
Sending Facility	HosA
Receiving Application	PI
Receiving Facility	MDM



Inbound HL7 v2 Solution

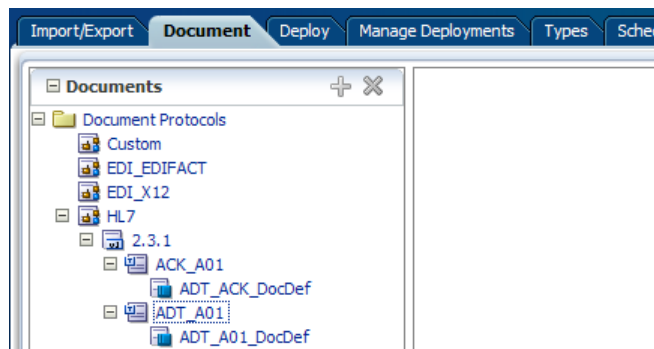
To develop the HL7 v2 inbound solution went through a number of steps. The steps were:

1. Modify HL7 ACK message structure – *ADT A01 already done in as we worked through the previous articles*
2. Modify B2B configuration to allow Functional ACK being passed between the B2B Layer and the SOA Layer
3. Undeploy the original HL7 Receiver SOA Composite
4. Create and deploy a new HL7 Receiver SOA Composite

This solution is assumed to be available for extension in this article.

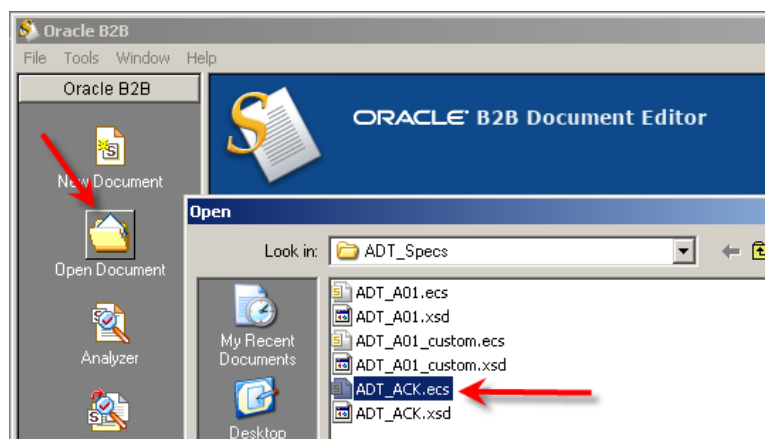
HL7 ADT A01 and ACK Message Structures

HL7 ADT A01 is already available and suitable for processing the inbound ADT A01 message so it is not necessary to go through the process with the B2B Document Editor for the inbound side.

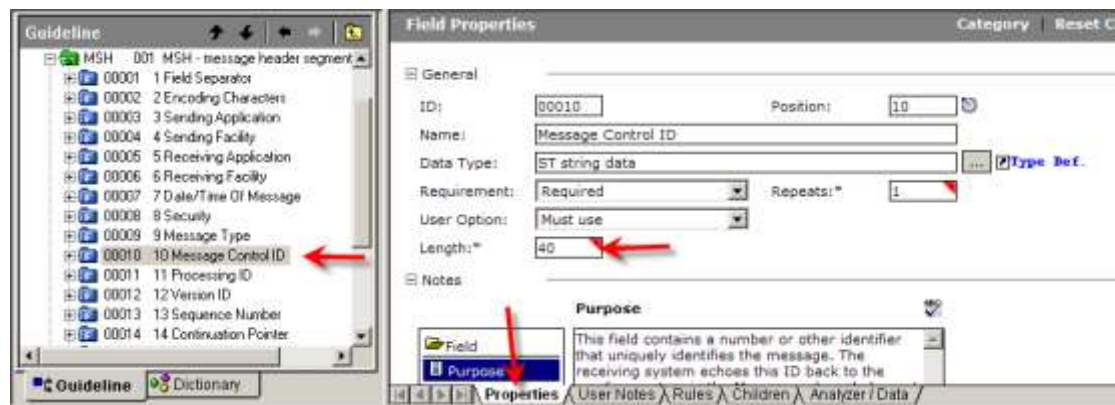


The ACK message specification must be modified to allow 32-bit GUID as MSH-10 Message Control ID and to allow the original MSH-10 to be carried as MSA-2, which is also too short to accommodate the id used in the ADT A01 message. Recall that we modified the ADT A01 structure in the previous article in this series - [Oracle SOA Suite 11g HL7 Inbound – Customized HL7 Message Structure and Data Validation](#)

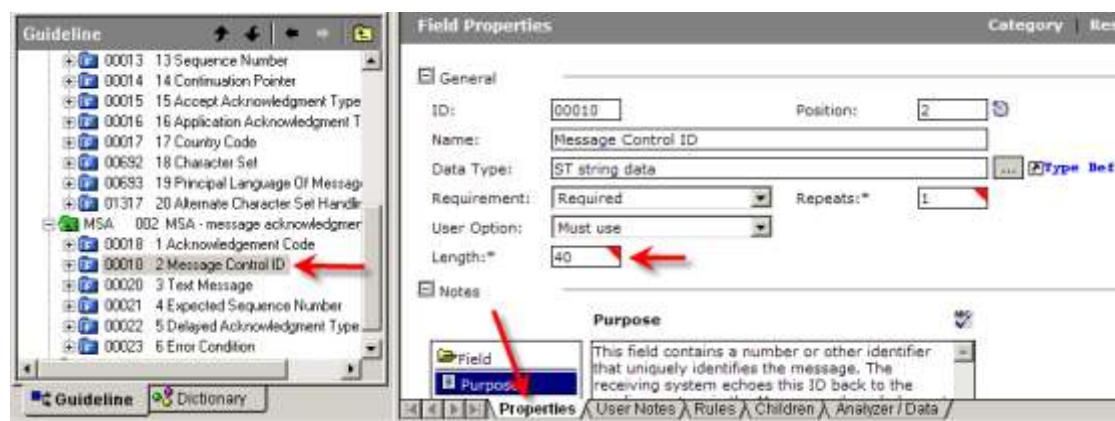
Open B2B Document Editor, choose Open Document, locate the ADT_ACK.ecs guideline and open it.



Select MSH-10 node in the left hand side panel and change the Max Length value to 40 in the right hand side Properties panel.



Repeat the process for MSA-2 Message Control ID.

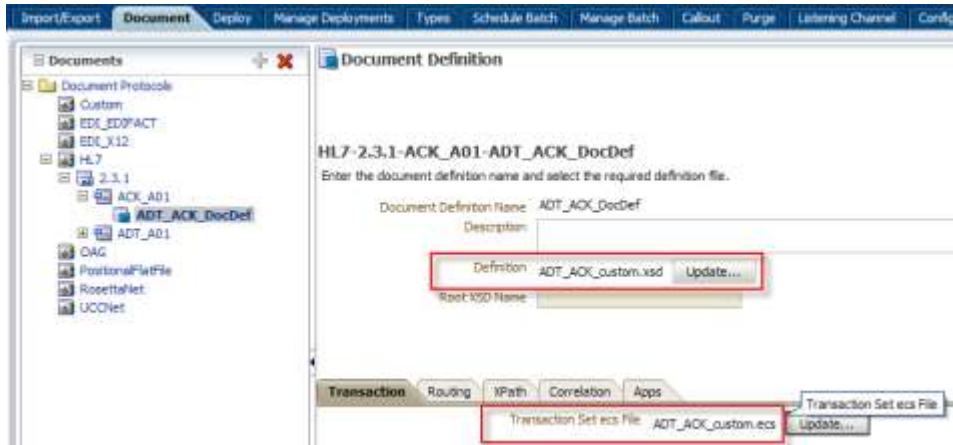


Save the modified guideline as ADT_ACK_custom.ecs. Export the XSD as ADT_ACK_custom.xsd.

Modify B2B Document Definition for ACK

Start the B2B Trading Partner Manager console by pointing the web browser at <http://localhost:7001/b2b>. Log in as [weblogic/welcome1](#).

Click the Administration link, choose Document Tab, expand HL7 → 2.3.1 → ACK_A01. Click ADT_ACK_DocDef. Update Definition and Transaction Set files. Click Save.



Click Partners link, select the Remote HL7Sender partner and Save and Deploy both ACK Trading Partnership Agreements. These will be HL7ReceiverOutboundTPA_ACK_Agr and HL7ReceiverTPA_ACK_Agr.



Re-configure B2B Layer

Click the Administration link, select the Configuration Tab and change the value of the “Functional Ack Handled by B2B” to false. Also ensure that the value of the “Notify Inbound Functional Acks” property is set to false. Click Save.



Note, as discussed in Oracle B2B User’s Guide [2], pp. B-2 and B-3, this change places specific limitations on functionality:

The following limitations apply when generating the FA from the back-end application:

- The FA is correlated with the original message based on the ReferToMsgID value set in the enqueue properties. The FA is correlated based on control numbers also.
- If the FA indicates that there was an error in the received message, the status of the correlated message is not updated to indicate an error. The correlated message is updated to MSG_COMPLETE.

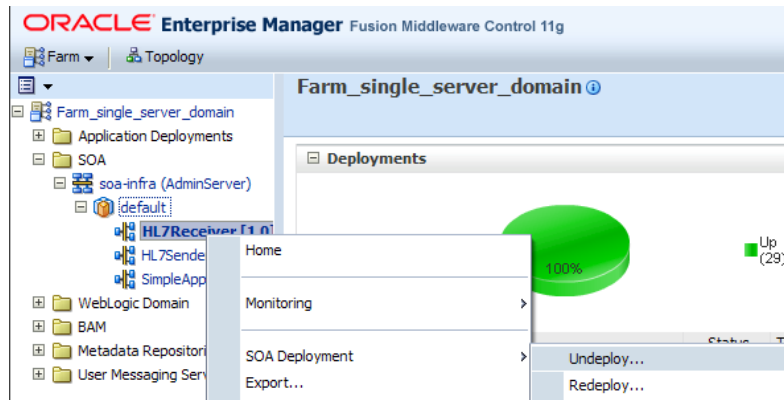
These limitations are not present when the FA is generated by B2B (that is, when `b2b.FAHandledByB2B` is true).

This change takes effect immediately upon Save and affects the behavior of all deployed trading partnership agreements.

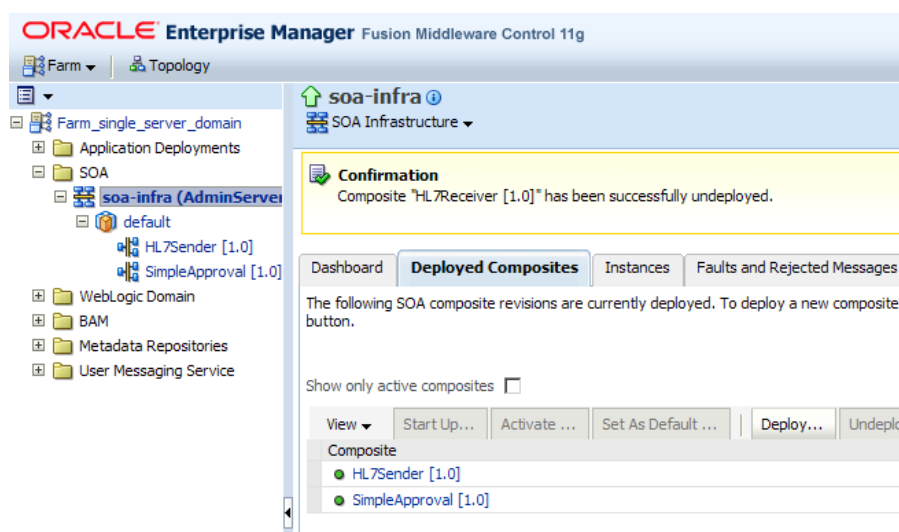
If this is not a desirable behavior for some partnership agreements a separate WebLogic Domain will have to be created for a separate B2B infrastructure where the default behavior (Functional Ack Handled By B2B == true) is in force.

Undeploy original HL7 Receiver

Log into the Enterprise Manager, <http://localhost:7001/em>, expand SOA, soa-infra through to default. Right-click HL7Receiver, choose SOA Deployment → Undeploy.



Confirm undeployment and verify that the composite is no longer deployed.

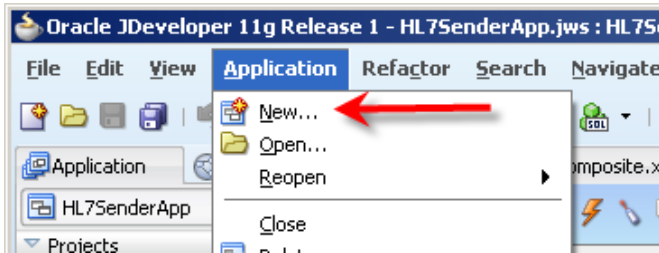


Close the Enterprise Manager window.

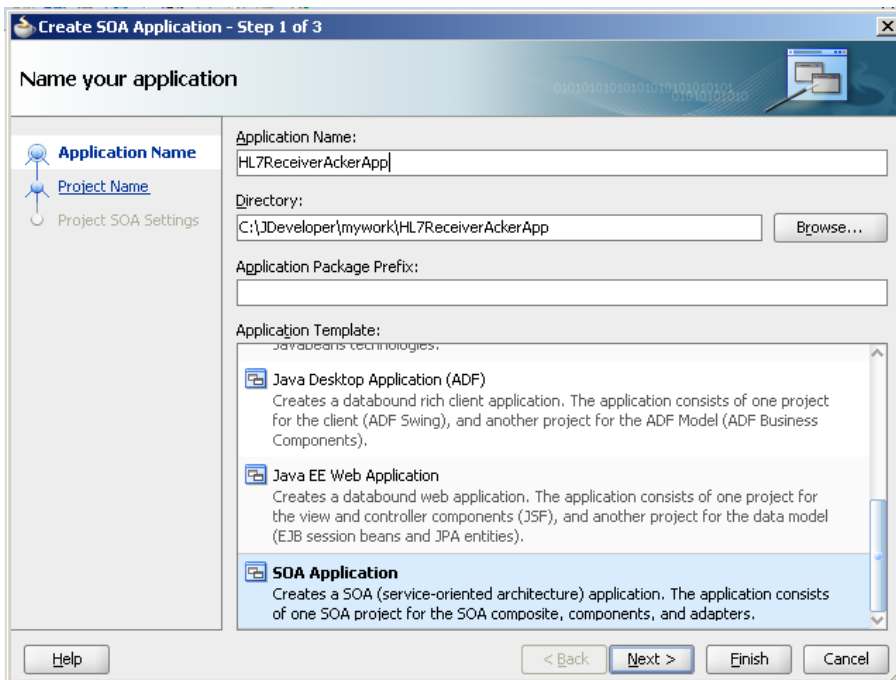
Build New HL7 Receiver Acker Composite

Oracle SOA Suite offers a number of components which can host application logic. In this example we will use the Mediator, as we have done before. We will use it to implement a “splitter” pattern – one copy of the message will be written to a file in the file system, as we did in the original HL7 Receiver solution, and one copy will be used to construct a HL7 ACK and send it back to the B2B Layer.

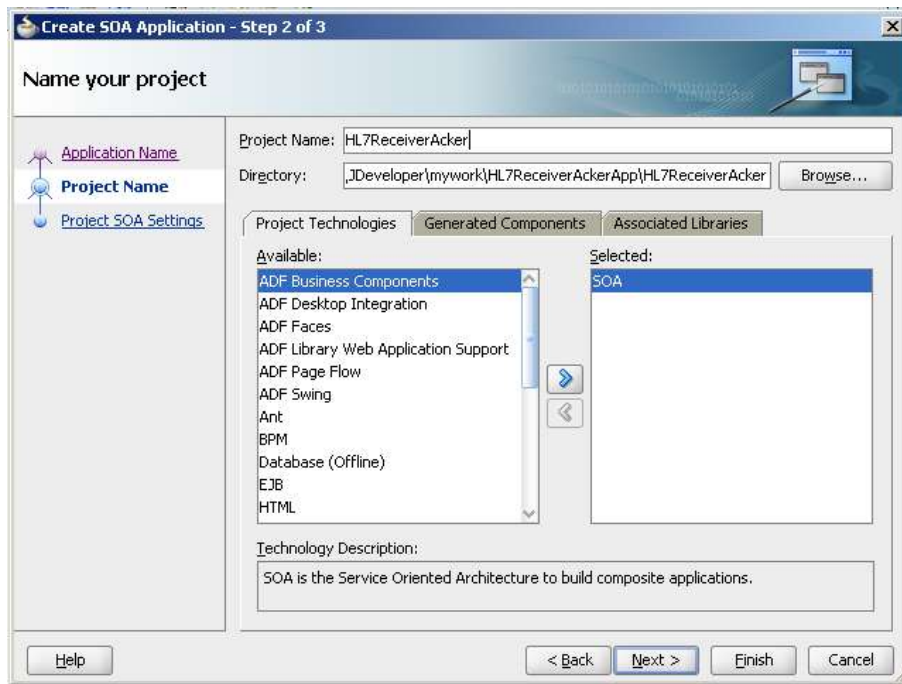
Let’s start JDveloped IDE and create a new SOA Application.



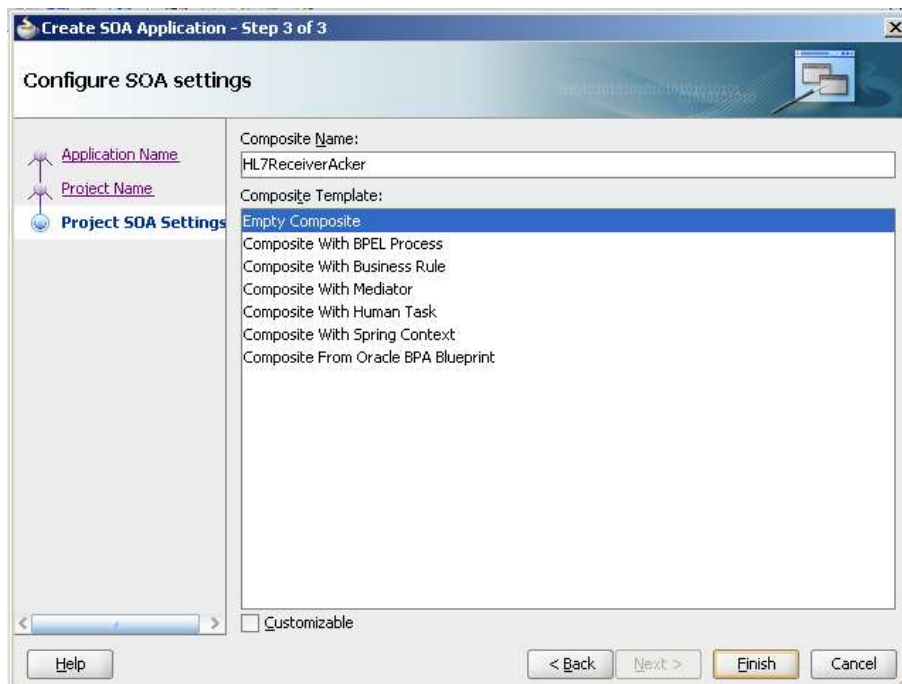
Let’s call this application HL7ReceiverAckerApp.



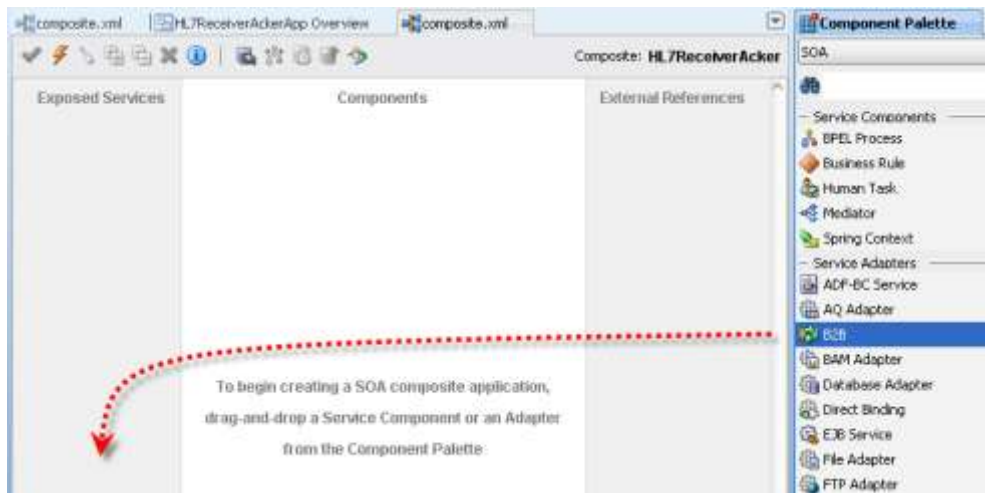
Let’s name the project HL7ReceiverAcker.



Accept Empty Composite and click Finish.



Drag B2B Adapter to the “Exposed Services” swim line.



Name the service ADT_A01_In and click Next.

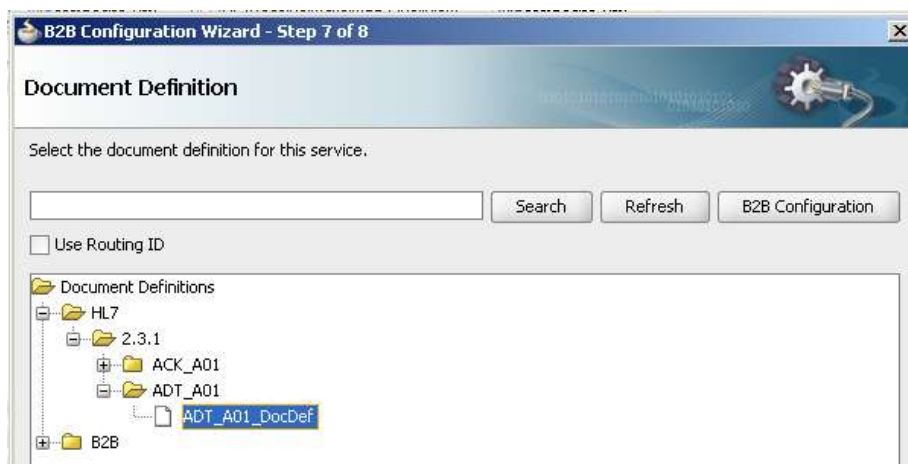
Accept default for “B2B integration type” and click Next.

Choose WebLogic_Local for AppServer Connection and Test B2B (this requires that the app server be up and running). Click Next.

Choose “Receive” as Operation and click Next.

Accept default of Basic for Document Definition Handling and click Next.

Choose ADT_A01_DocDef HL7 Document Definition and click Next, than Finish.



The inbound B2B Adapter is configured.

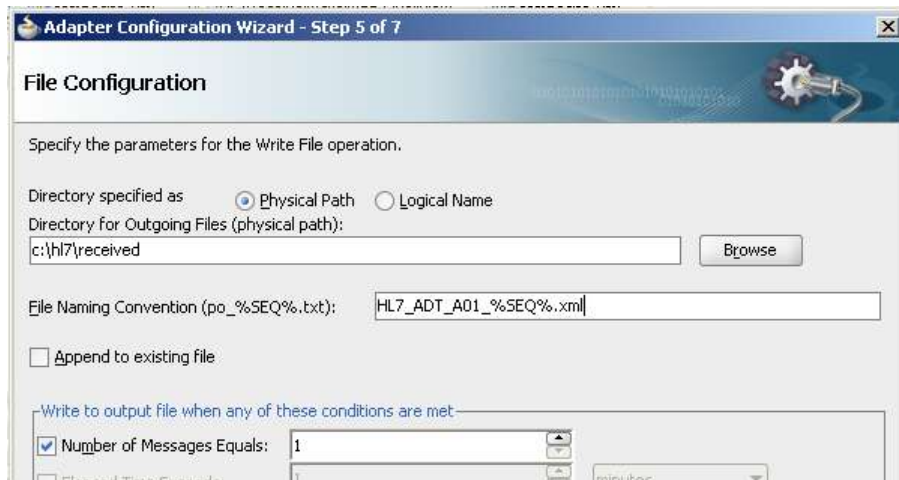
Drag the File Adapter onto the External References swim line.

Name the service WriteADTA01 and click Next.

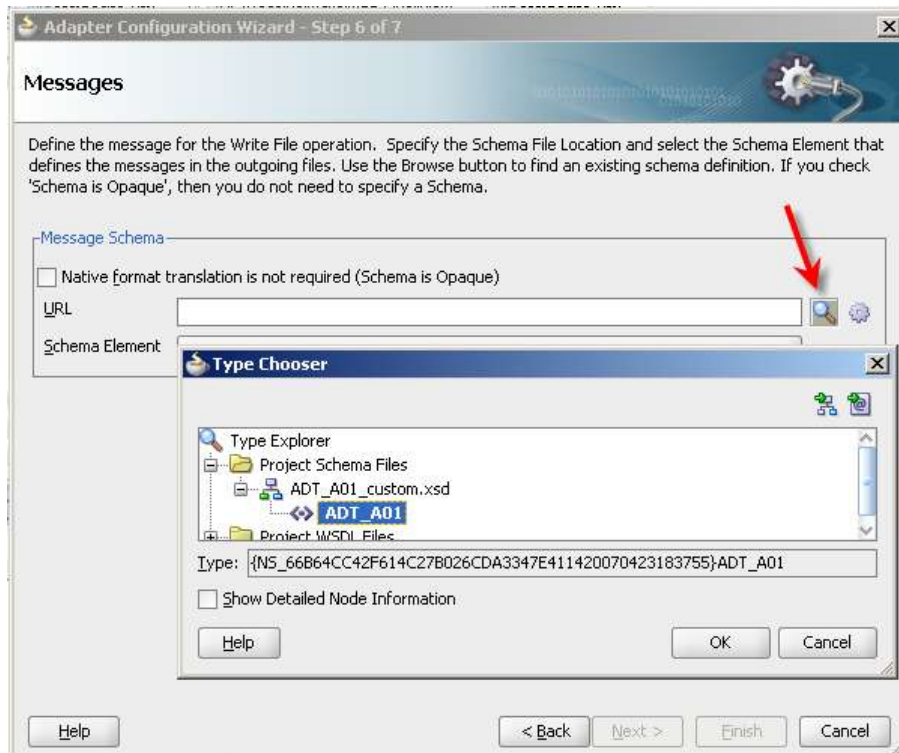
Accept the default for “Adapter Interface” and click Next.

Select Write File as Operation Type and click Next.

Enter c:\hl7\received as physical path, HL7_ADT_A01_%SEQ%.xml as File Naming Convention and click Next.



Locate the ADT_A01 XML Schema file, choose the ADT_A01 Element, click OK, Next and Finish.



Drag the B2B Adapter to the External References swim line.

Give the service the name ADT_ACK_Out and click Next.

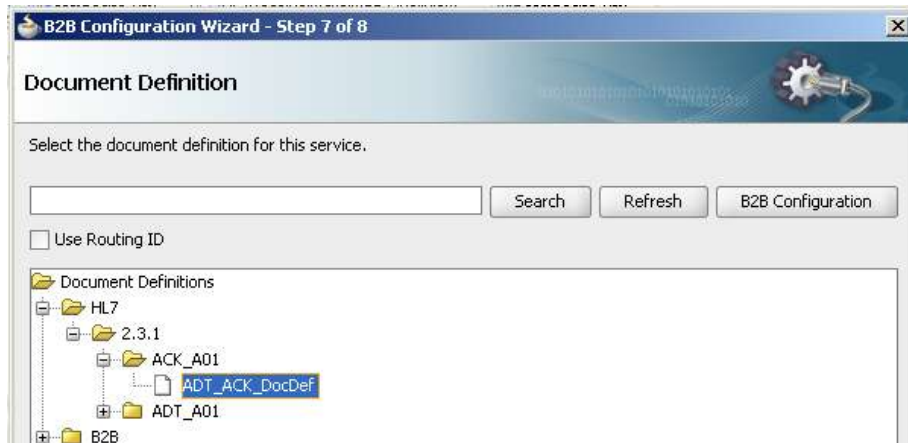
Leave B2b Integration Type as default and click Next.

Choose WebLogic_Local for AppServer Connection and Test B2B (this requires that the app server be up and running). Click Next.

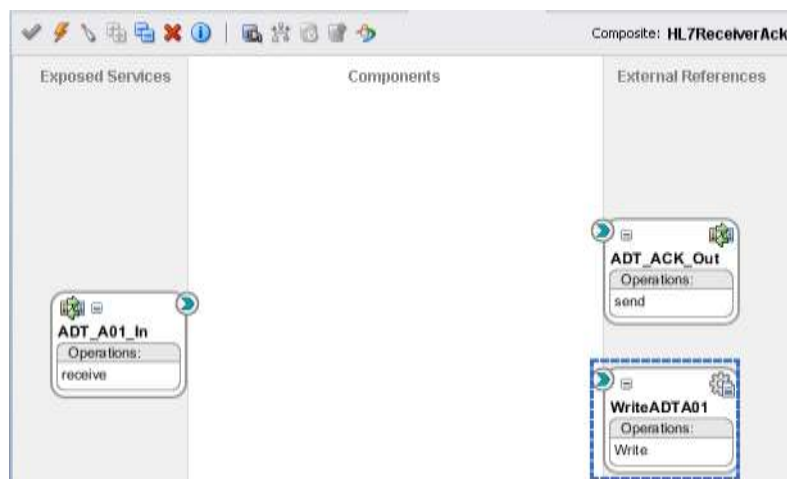
Leave the default of “Send” as Operation and click Next.

Accept default of Basic for Document Definition Handling and click Next.

Choose ADT_ACK_DocDef HL7 Document Definition and click Next, than Finish.



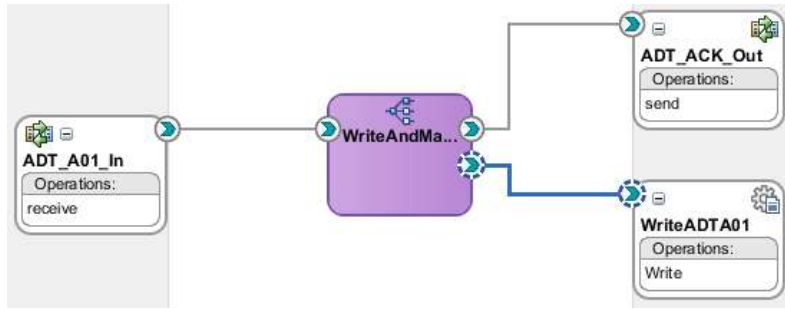
All adapters are now added and configured.



Drag the Mediator component to the Components swim line.

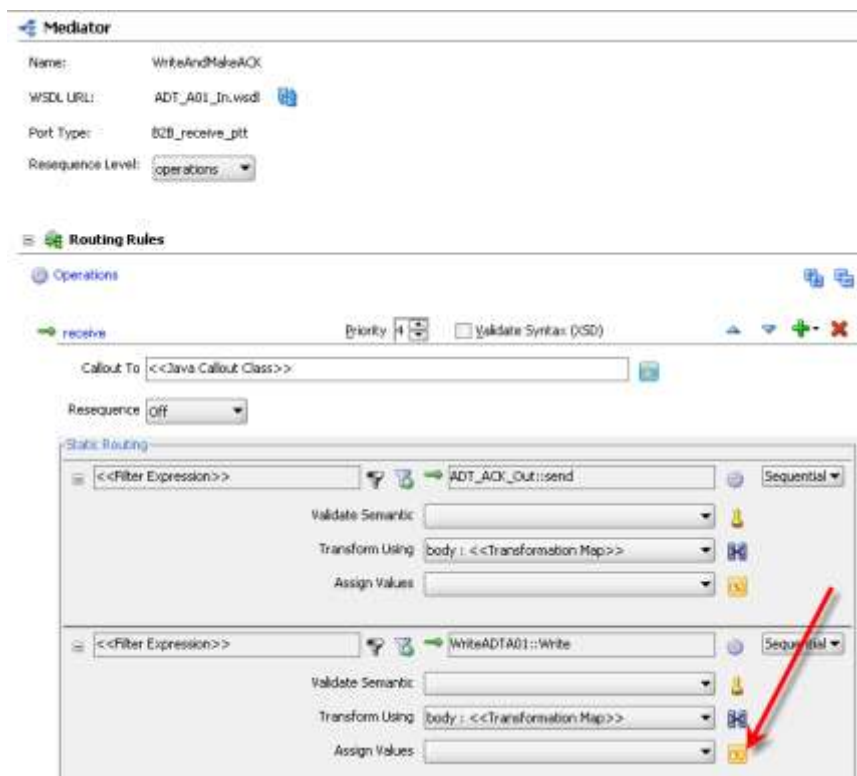
Name the Mediator component WriteAndMakeACK, and then click OK.

Wire the ADT_A01_In B2B Adapter to the Mediator and the Mediator to the ADT_ACK_Out B2B Adapter and the WriteADTA01 File Adapter.

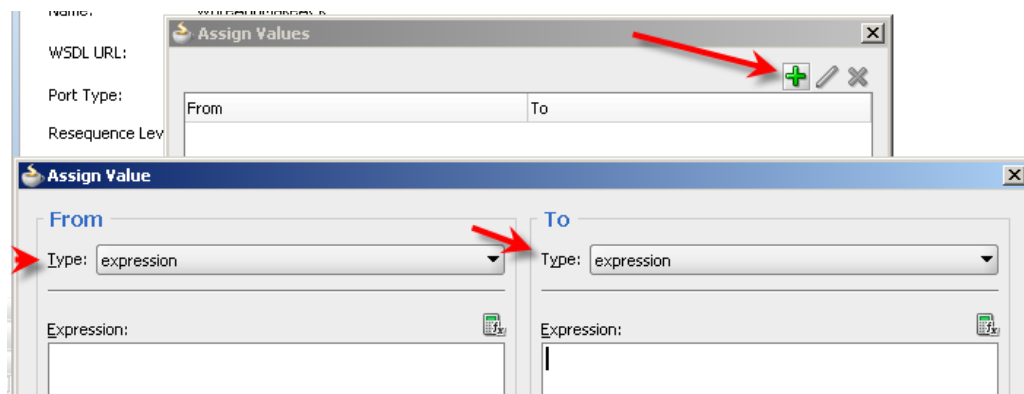


Double-click the Mediator component to open the mediator plan.

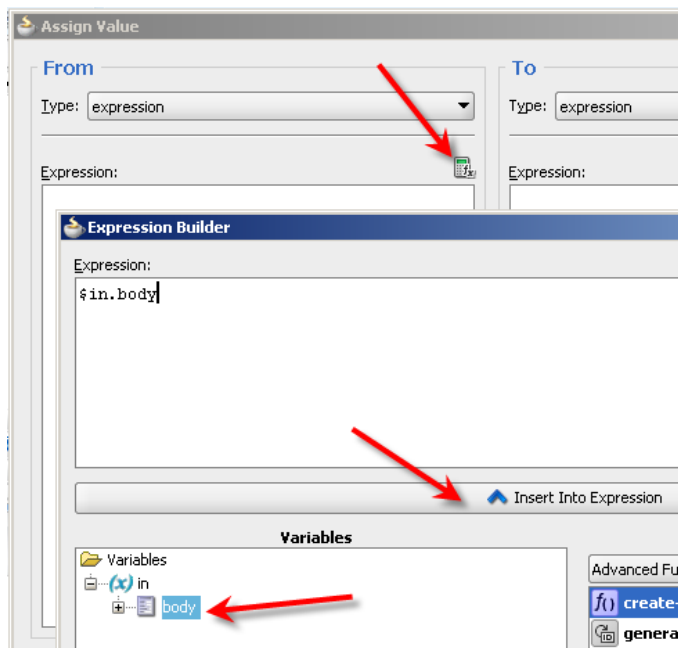
Click the Assign Value button in the WriteADTA01::Write operation routing rule.



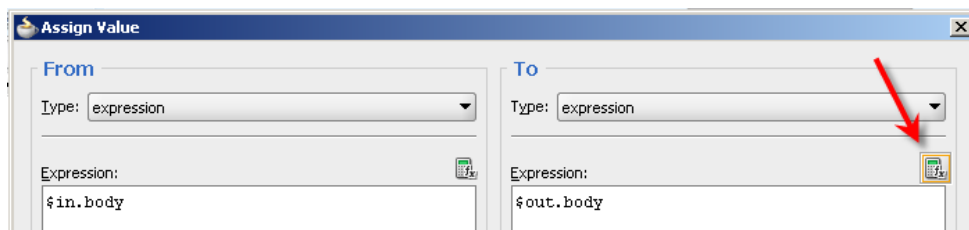
Click the Add a New Value Assignment icon (Plus sign) and choose expression for the From and the To sides.



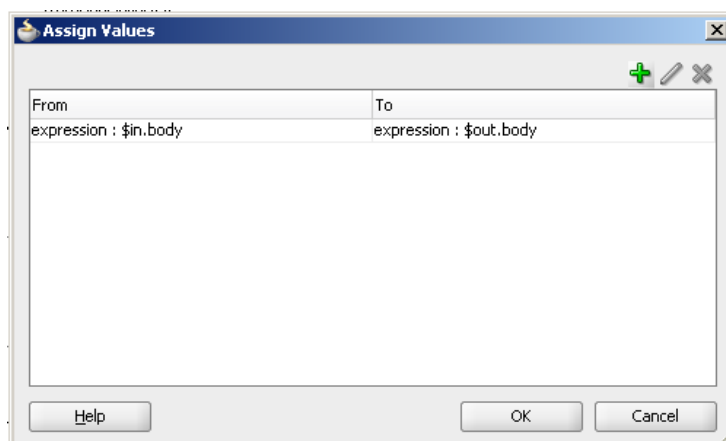
Click the “Invoke Expression Builder” on the From side, expand the variables list, choose in.body, click the wide “Insert Into Expression” button and click OK.



Repeat the process for the To side choosing out.body variable.

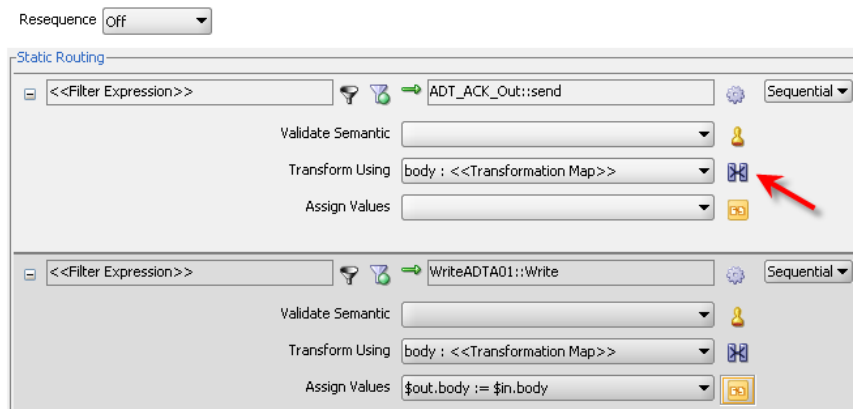


Click OK to close the Assign Values dialogue box.

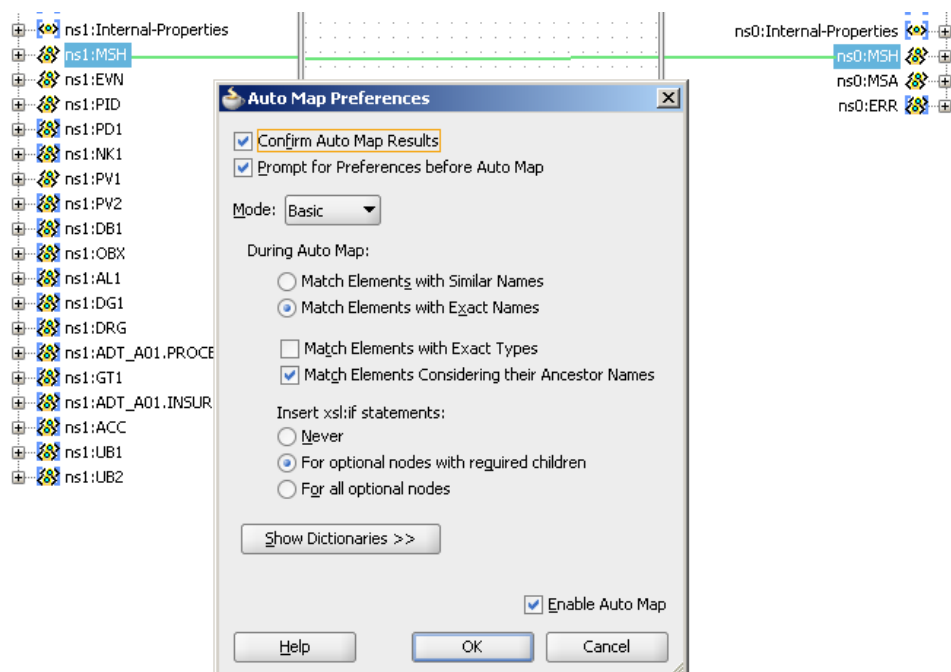


The mapping required to pass the entire ADT A01 to the File Adapter is now complete.

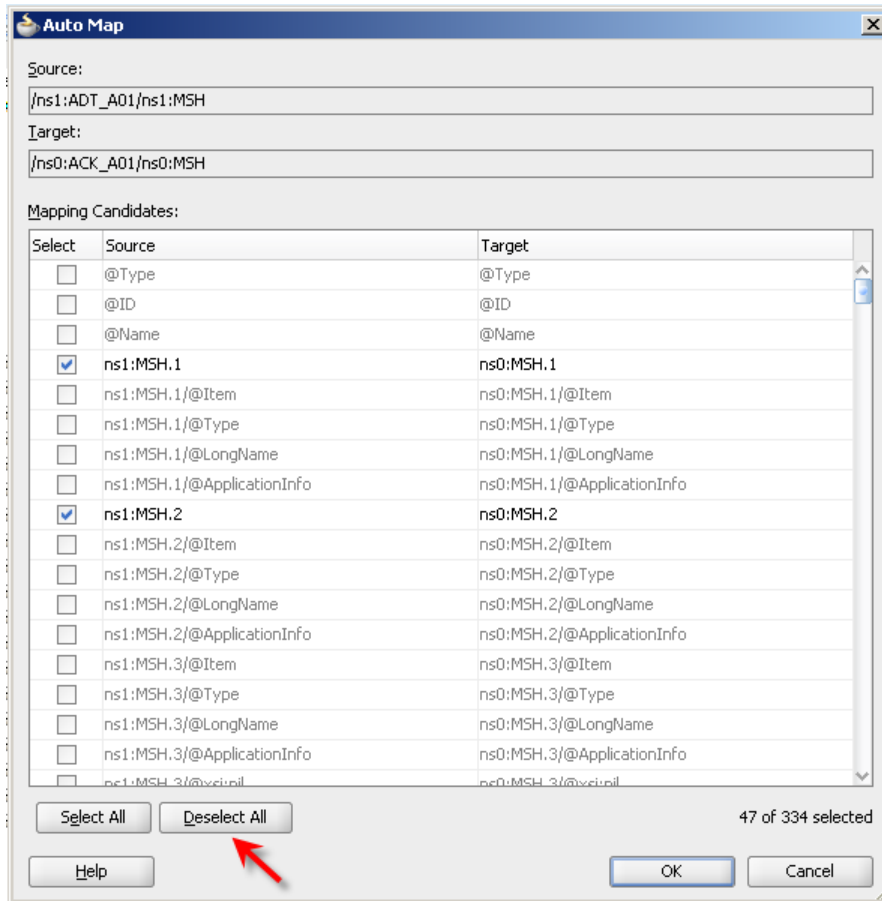
Create a New Mapper file for the ADT_ACK_Out mapping, accepting default name.



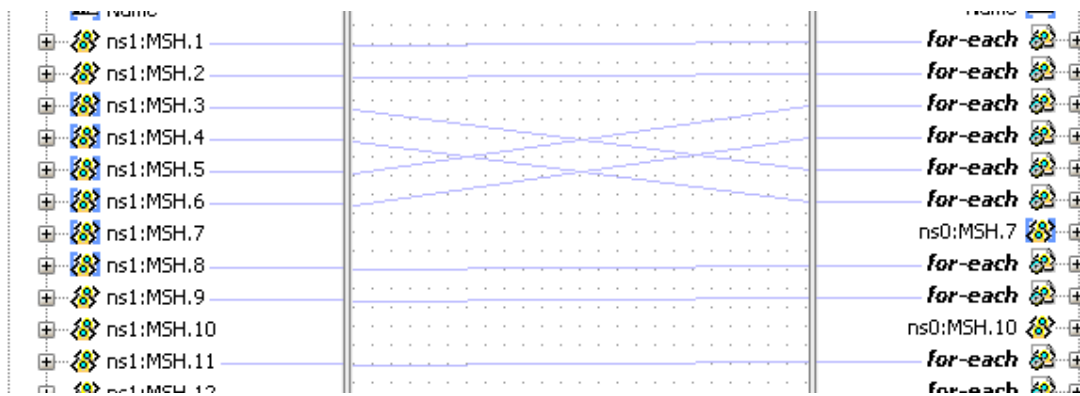
Map MSH to MSH, choosing to Match Elements with Exact Names, Match Elements Considering their Ancestor Names and For optional nodes with required children.



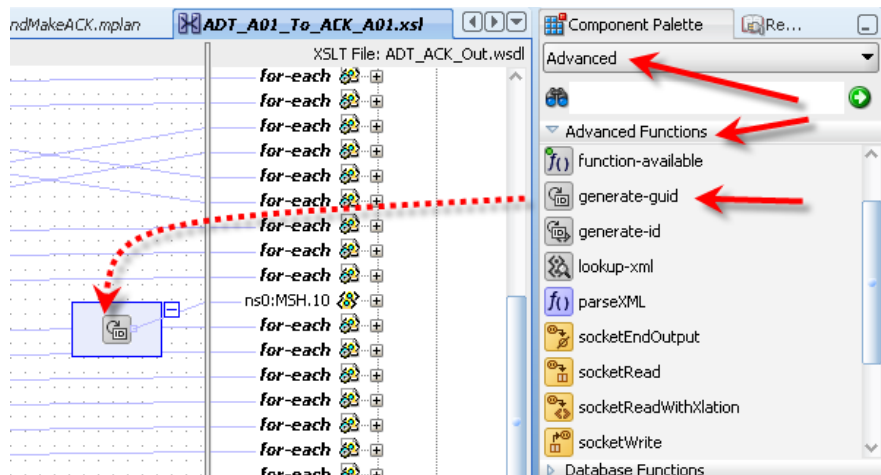
Click Deselect All button then select only the Element nodes, that is nodes whose source path does not contain the @ sign. Leave MSH3, MSH-4, MSH-5, MSH-6, MSH-9.1 and MSH-10 deselected. We will map them individually.



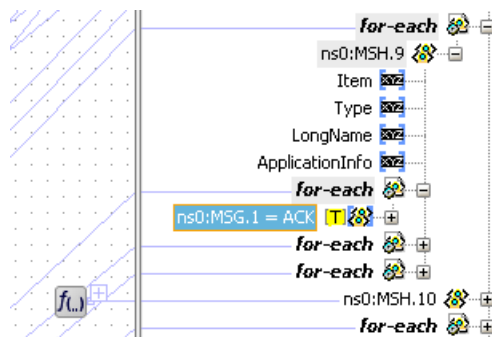
Map source MSH-3 to target MSH-5, source MSH-4 to target MSH-6, source MSH-5 to target MSH-3 and source MSH-6 to target MSH-4.



Switch to Advanced Palette, choose Advanced Functions and drag generate-guid function to the canvas. Connect it to the MSH-10 node on the target side.



Set text of MSH-9 →MSG.1 to the literal ACK.

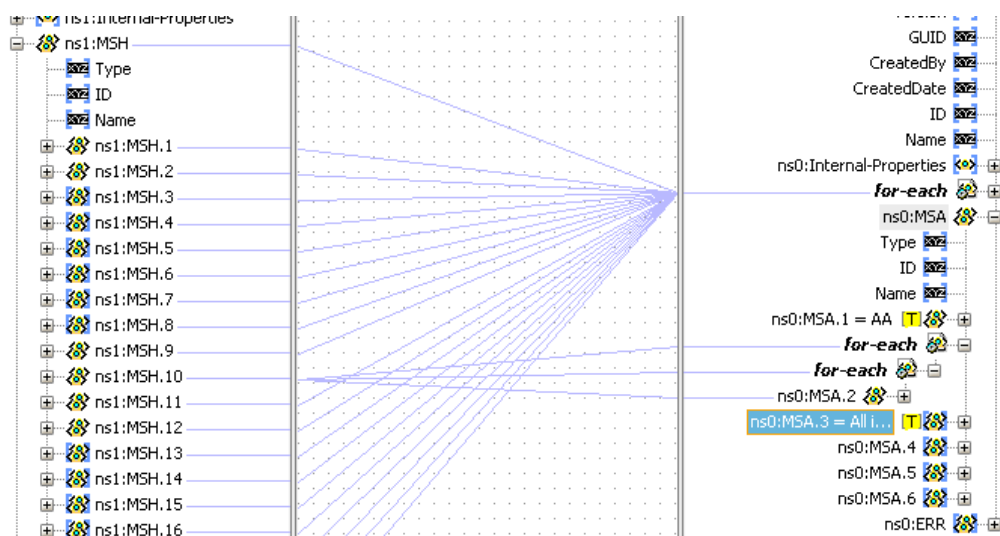


Collapse MSH structure on the target side and expand the MSA structure on the target side one level.

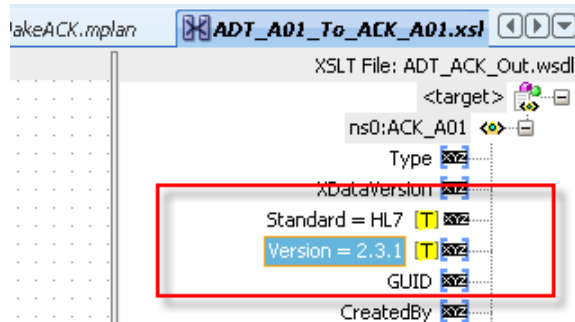
Set text “AA” in target MSA-1.

Map source MSH-10 (Message Control ID) to target MSA-2.

Set text “All is well” in target MSA-3.

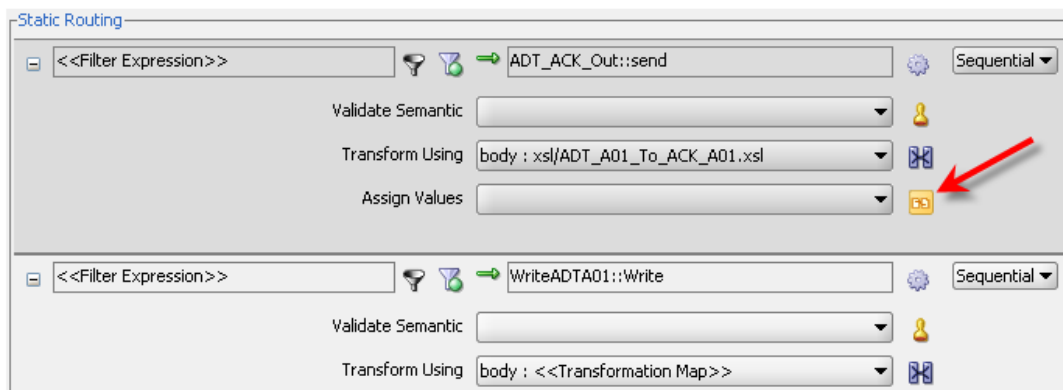


Set text HL7 in the global attribute Standard and text 2.3.1 in the global attribute Version.



Click Save All toolbar button and close the mapper.

Click the Assign Values button to set properties for the message exchange in such a way that the B2B infrastructure can identify the trading partnership agreement to use for processing the ACK and so that it can correlate this ACK with the original message.



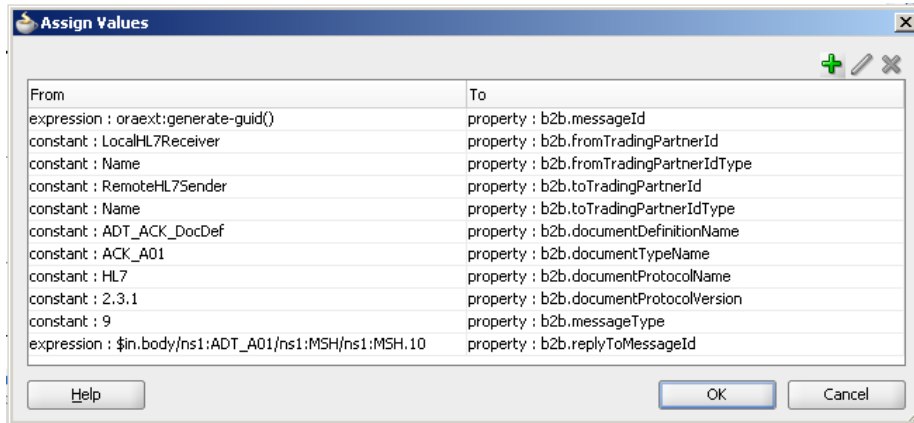
Populate the following properties as indicated in the table below. The process is illustrated in the article [Oracle SOA Suite 11g HL7 Outbound Example](#).

Expression	Property
oraext:generate-guid()	b2b.messageId
\$in.body/ns1:ADT_A01/ns1:MSH/ns1:MSH.10	b2b.replyToMessageId
Constant	Property
LocalHL7Receiver	b2b.fromTradingPartnerId
Name	b2b.fromTradingPartnerIdType
RemoteHL7Sender	b2b.toTradingPartnerId
Name	b2b.toTradingPartnerIdType
ADT_ACK_DocDef	b2b.documentDefinitionName
ACK_A01	b2b.documentTypeName
HL7	b2b.documentProtocolName
2.3.1	b2b.documentProtocolVersion
9	b2b.messageType

Message types are 1 == Request, 2 == Response, 9 == Functional Acknowledgement.

The value of the MSH-10 field, set in the b2b.replyToMessageId property, is used by the B2B infrastructure to correlate the ACK and the original request so that it can be sent as a response on the same channel and so that it is tracked together with the request.

The properties are now set.



Save All, close the Mediator plan and Deploy this composite.

Once deployment is finished close JDeveloper.

Exercise HL7 Inbound solution

In the B2B Trading Partner Manager Web Console, <http://localhost:7001/b2b>.

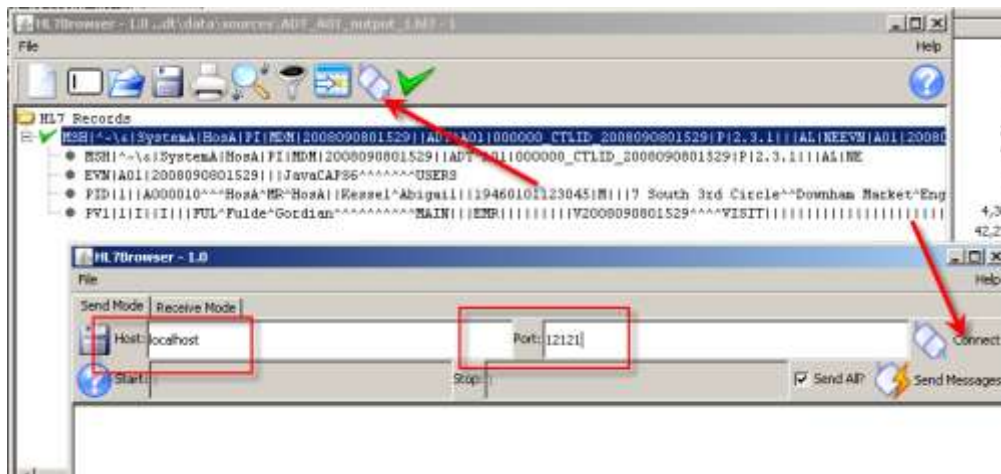
Open a command window and run the HL7Browser:

```
C:\jdk1.6.0_20\bin\java.exe -jar C:\tools\Hl7Browser.1.0\Hl7Browser.jar
```

When the UI appears click the “Open an HL7 File” button, locate the ADT A01 transaction file, C:\hl7\adt\sources\ADT_A01_output1.hl7, and open it.



Click “Run the network utility” button, provide localhost as the Host (or leave blank since this defaults to localhost) and 12121 as the Port (recall this is the port on which the B2B listener is listening) and click Connect.



Once connected, click the “Send Messages” button.

Observe message exchange - the message sent and the ACK received.

```

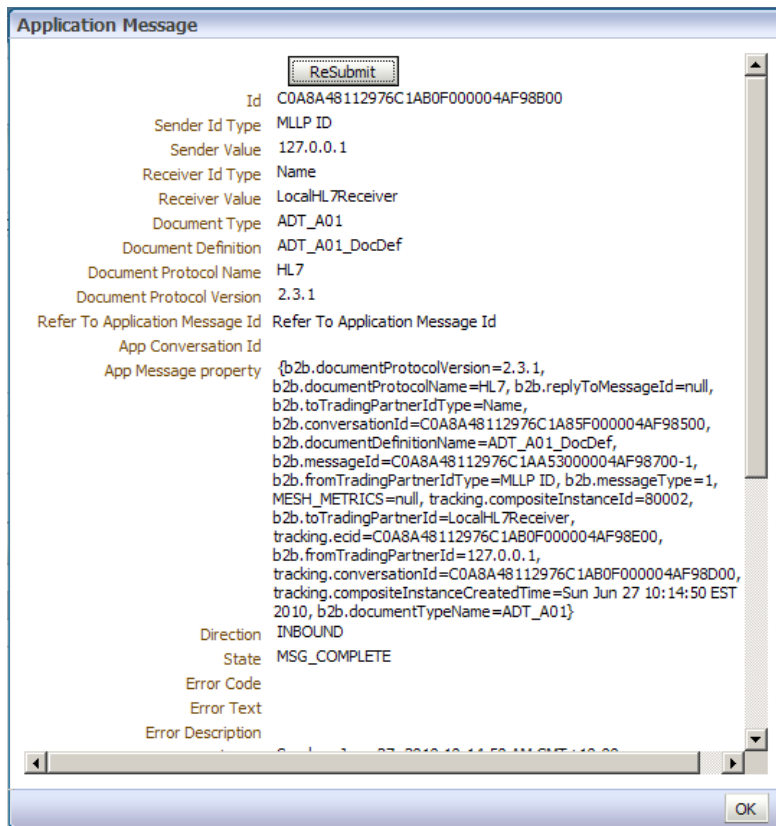
Starting at: 10:14:48
10:14:48 sending: MSH|^~^&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NEEVM|A01|2008090801529|||
JavaCAP56^~~~~~USERSPID|1||A000010^~^HosA^MR^HosA||Kessel^Abigail||19460101123045|M||7 South 3rd Circle^Downham Market^England - Nor
folk^30828^UK|||||A2008090801529PV1|1|I|I||FUL^Fulde^Gordian^~~~~~MAIN||EMR|||||V2008090801529^~^VISIT|||||2
008090801529
10:14:53 received: MSH|^~^&|PI|MDM|SystemA|HosA||ACK^A01|2d333536353343233353133363638|P|2.3.1|||AL|NEM5A|AA|000000_CTLID_200809080
1529|All is well
    
```

Inspect message tracking information.

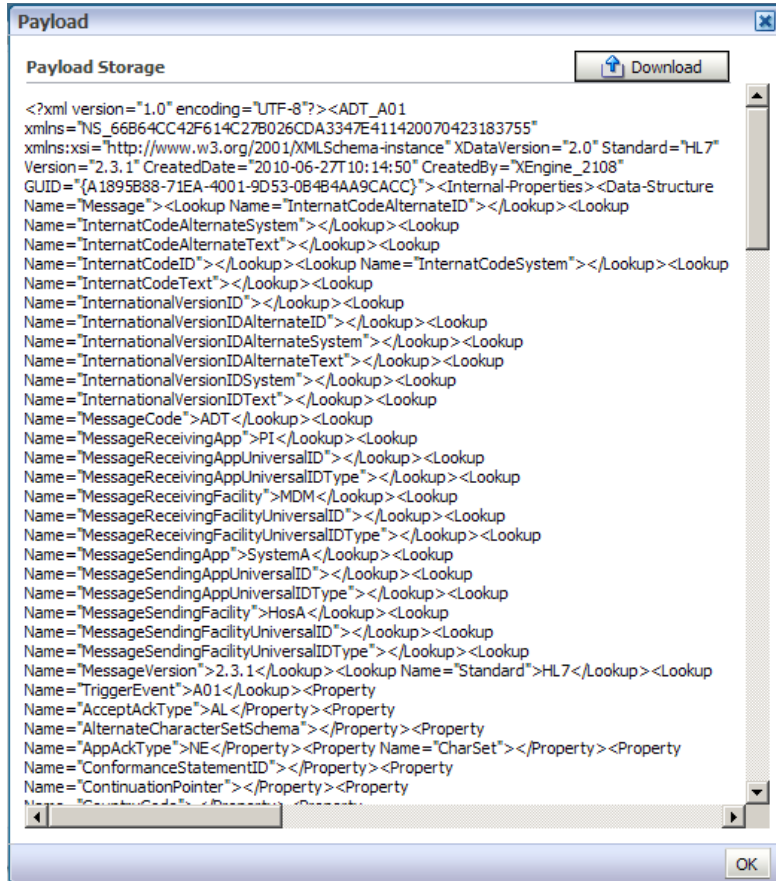
Application Message Tab.



ADT A01 Details.



Application Message Payload (XML).



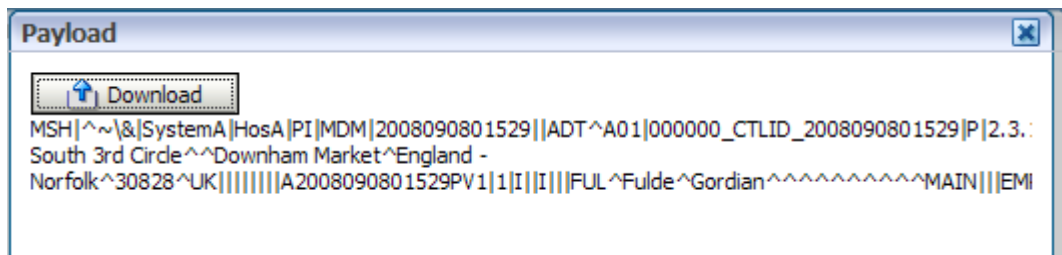
Business Message Details.

Business Message	
Id	C0A8A48112976C1AA63000004AF98800
Message Id	C0A8A48112976C1AA53000004AF98700-1
Refer To Message	Refer To Message
Sender Type	MLLP ID
Sender Value	127.0.0.1
Receiver Type	Name
Receiver Value	LocalHL7Receiver
Sender	RemoteHL7Sender
Receiver	LocalHL7Receiver
Agreement Id	HL7ReceiverTPA
Agreement	HL7ReceiverTPA_Agr
Document Type	ADT_A01
Document Protocol	HL7
Document Version	2.3.1
Message Type	REQ
Direction	INBOUND
State	MSG_COMPLETE
Acknowledgement Mode	SYNC
Response Mode	ASYNC
Send Time Stamp	Sunday, June 27, 2010 10:14:49 AM GMT+10:00
Receive Time Stamp	Sunday, June 27, 2010 10:14:49 AM GMT+10:00
Document Retry Interval	0
Document Remaining Retry	0
Native Message Size	412
Translated Message Size	10989
Business Action Name	
Business Transaction Name	
Xpath Name.1	XPathName1

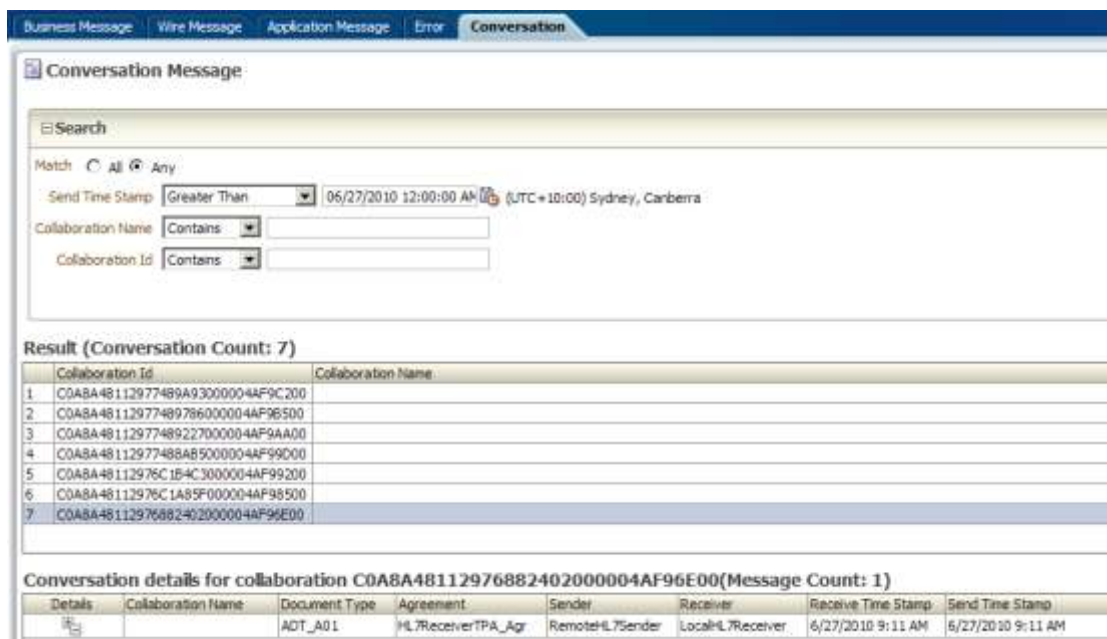
Wire Message Details.

Wire Message	
Id	C0A8A48112976C1A37D000004AF98000
Message Id	C0A8A48112976C1A37D000004AF98000
Business Message	C0A8A48112976C1AA63000004AF98800
Packed Message	Packed Message
Payload	Payload
Protocol Message Id	12775976886701975452346
Refer To Protocol Message Id	
Protocol Collaboration Id	
Protocol Transport Binding	From=localhost FromIP=127.0.0.1 MSG_RECEIVED_TIME=Sun Jun 27 10:14:48 EST 2010 protocolVersion=1.0 ToIP=127.0.0.1 Port=12121 Sequencing=false Message-Id=12775976886701975452346 DynamicIP=127.0.0.1:2582 To=localhost protocol=TCP
Message Digest	Message Digest
Digest Algorithm	
Transport Protocol	TCP
Transport Protocol Version	1.0
Url	TCP://127.0.0.1:12121
security	
Transport Headers	From=localhost FromIP=127.0.0.1 MSG_RECEIVED_TIME=Sun Jun 27 10:14:48 EST 2010 protocolVersion=1.0 ToIP=127.0.0.1 Port=12121 Sequencing=false Message-Id=12775976886701975452346 DynamicIP=127.0.0.1:2582 To=localhost protocol=TCP
certificates	certificates
State	COMPLETE
Error Code	
Error Description	
Error Text	

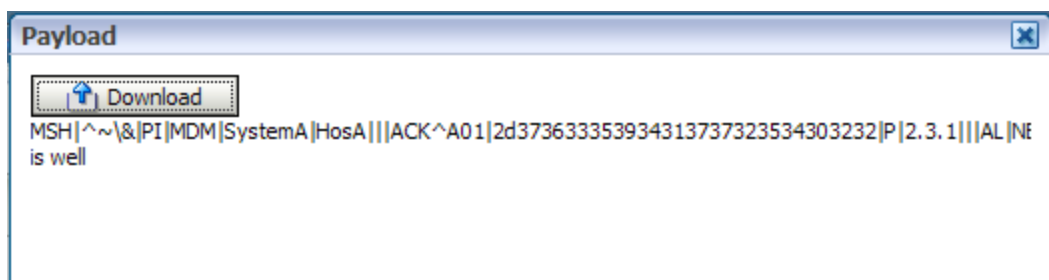
Wire Message Payload.



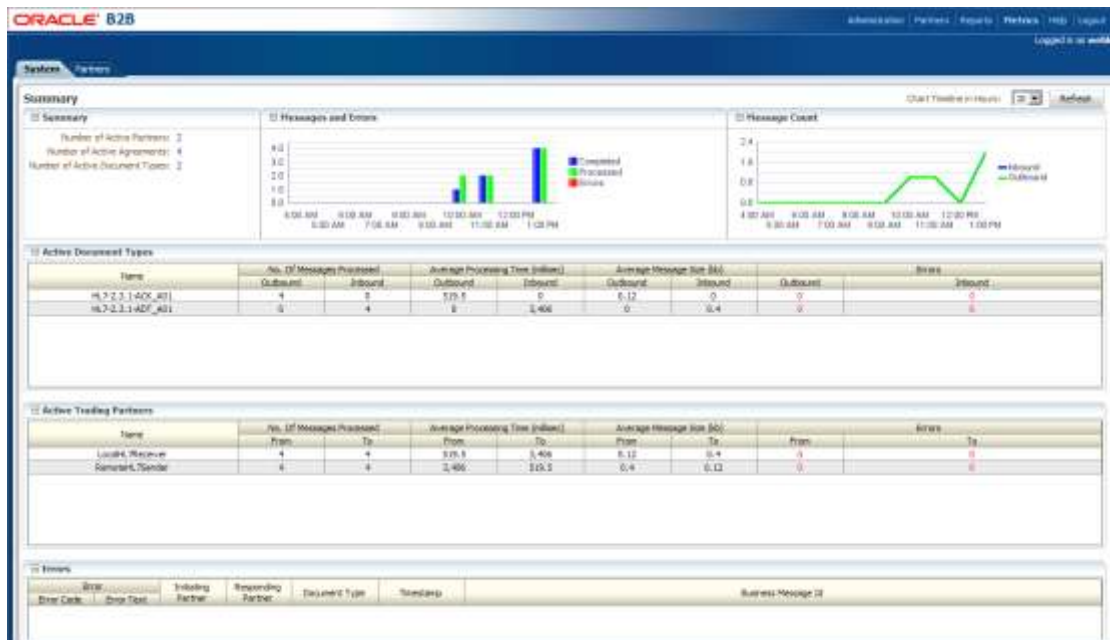
Conversations.



Acknowledgement Payload.



Messaging Metrics.



This concludes the exercise.

Summary

Oracle SOA Suite B2B component can be used to provide HL7 v2 messaging support for healthcare environments.

This article is a follow on to the “[Oracle SOA Suite 11g HL7 Inbound Example – Functional ACK Addendum](#)” article and the “[Oracle SOA Suite 11g HL7 Inbound – Customized HL7 Message Structure and Data Validation](#)” article. In these articles the B2B infrastructure was configured to return the “Functional ACK” when it validated each message. The ACK was a positive or a negative ACK depending on whether the message passed validation. The ACK was generated by the B2B Layer before the message was passed on to the SOA Layer.

In this article I expanded on the previous posts by configuring the B2B Layer to pass the message to the SOA Layer and pass the Functional ACK, generated by the SOA Layer, on to the requester. To process a message and produce the ACK we built and deployed a new SOA Composite.

References

- [1] Oracle B2B Site, Available: <http://www.oracle.com/technology/products/integration/b2b/index.html>, Accessed: 4 June 2010
- [2] Oracle B2B User’s Guide, Available: http://download.oracle.com/docs/cd/E15523_01/integration.1111/e10229.pdf, Accessed: 19 June 2010