Oracle SOA Suite 11g

Oracle SOA Suite 11g HL7 Inbound Example Functional ACK Addendum

michael@czapski.id.au
June 2010

Table of Contents

Introduction

This article is a follow on to the "Oracle SOA Suite 11g HL7 Inbound Example", at http://blogs.czapski.id.au/2010/06/oracle-soa-suite-11g-hl7-inbound-example. In that article the B2B infrastructure was configured to return the "immediate ack" as soon as it received each message. The ACK was always a positive ACK, regardless of whether the message was valid or not and whether it was successfully processed.

In this article I expand on the previous post by adding the Functional ACK, an explicit ACK message, which is returned after a message is parsed and validated, if validation is required. This means that rather than always returning an ACK the receiver will return a NAK if the message is invalid.

Pre-requisites

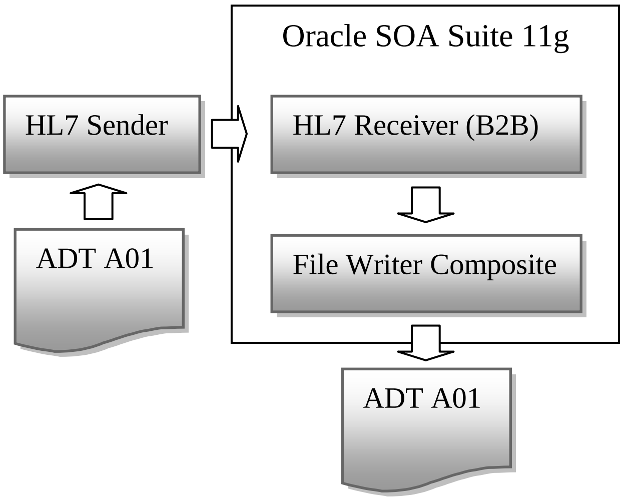
It is assumed that a Windows XP SP3 platform with the Oracle SOA Suite 11g, installed and configured as discussed in "Installing Oracle SOA Suite for HL7 Exploration", published at http://blogs.czapski.id.au/wp-content/uploads/2010/06/01_Installing_Oracle_SOA_Suite_for_HL7_exploration_v1.1.pdf, is available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

It is assumed that the article "Oracle SOA Suite 11g HL7 Inbound Example", at http://blogs.czapski.id.au/2010/06/oracle-soa-suite-11g-hl7-inbound-example, has been followed and the solution discussed in this article is available for further work.

HL7 v2 Receiver Solution

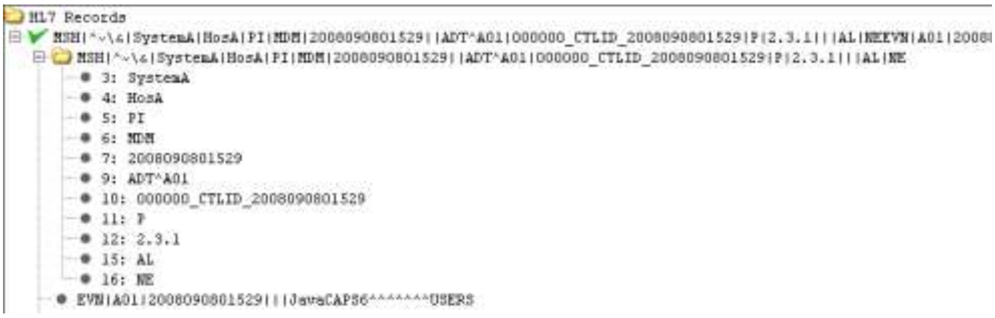
The solution that was built in the previous article consisted of a HL7 Receiver, which received v2 ADT A01 messages and wrote them to files in the file system.

The solution consists of a B2B Listener Channel, to which HL7 v2 ADT A01 messages are sent, and a SOA Composite which receives these messages and writes them to a file in the file system.



Messages in the sample message set use the following identifiers:

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



```
HL7 Records
⊟ ✓ MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NEEVN|A01|2008
  ⊟ MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NE
      ● 3: SystemA
      ● 4: HosA
      ● 5: PI
      ● 6: MDM
      ● 7: 2008090801529
      ● 9: ADT^A01
      ● 10: 000000_CTLID_2008090801529
      ● 11: P
      ● 12: 2.3.1
      ● 15: AL
      ● 16: NE
    ● EVN|A01|2008090801529|||JavaCAPS6^^^^^^^USERS
```

Inbound HL7 v2 Solution

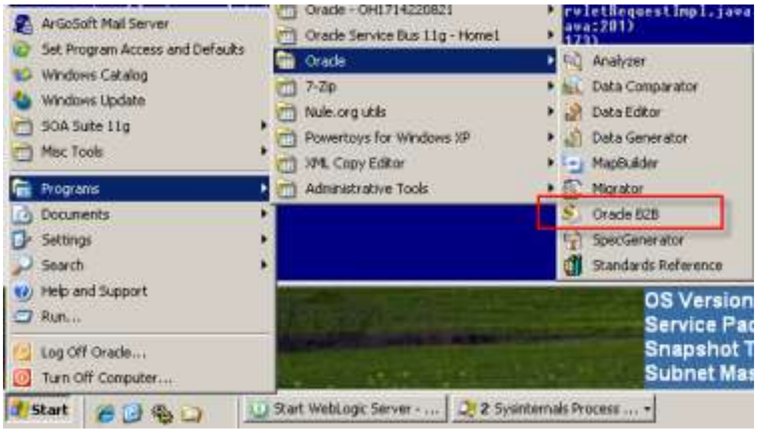
To develop the HL7 v2 inbound solution went through a number of steps. The steps were:

1. Extract HL7 message structures from the standard library
2. Configure Local side of the B2B Partnership
3. Configure Remote side of the B2B Partnership
4. Create and deploy a Partnership Agreement
5. Create and deploy a HL7 Receiver B2B Composite

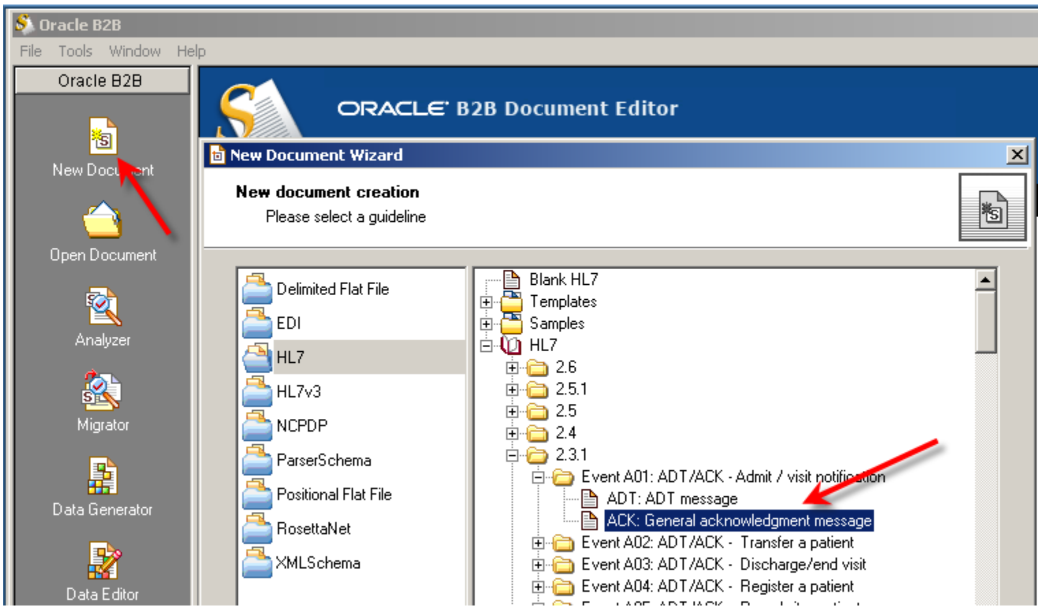
This solution is assumed to be available for extension in this article.

Extract HL7 ADT ACK Message Structures

Start the B2B Document Editor, which should be accessible through Start menu → Programs → Oracle as Oracle B2B.



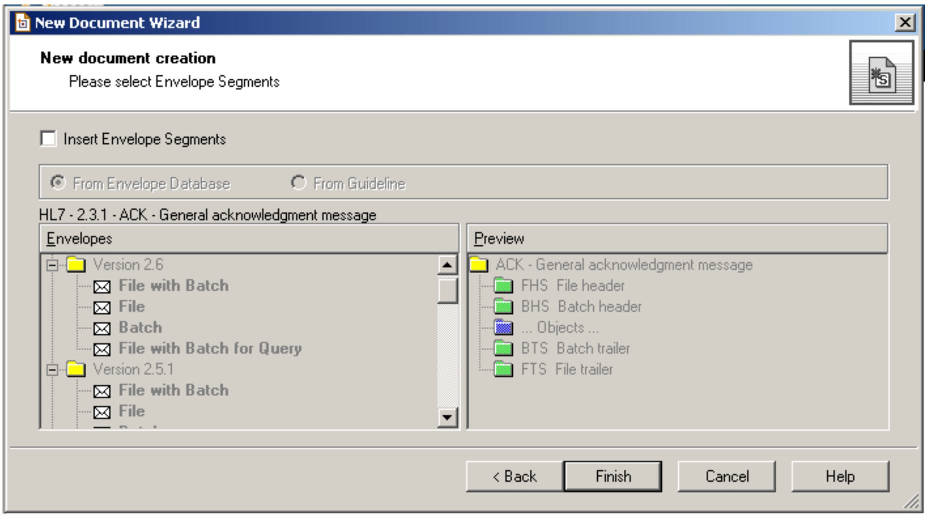
Click the “New Document” button, select HL7 from the list, expand 2.3.1, select Event A01, ACK General Acknowledgement message and click Next.



New document creation

Please select a guideline

Leave the “Insert Envelope Segments” unchecked and click Finish.



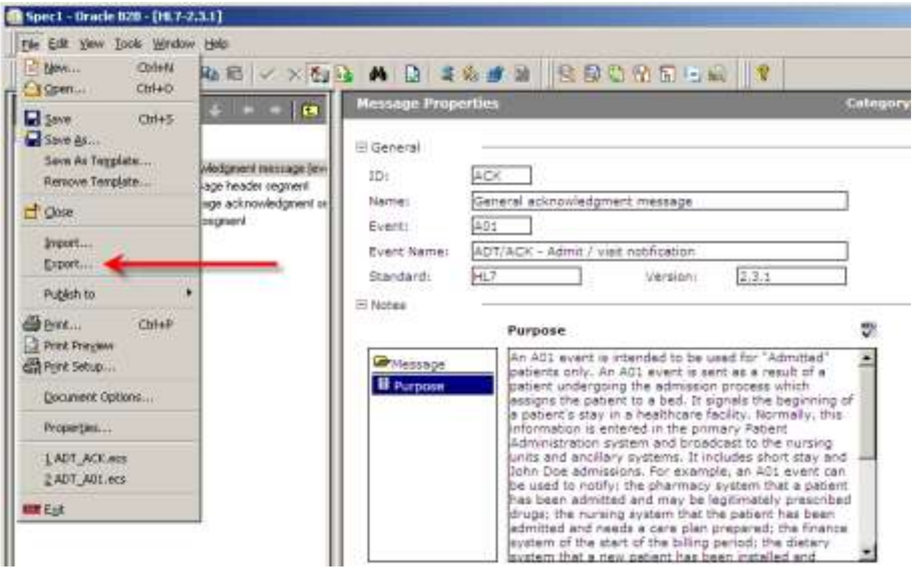
New document creation

Please select Envelope Segments

☐ Insert Envelope Segments

HL7 - 2.3.1 - ACK - General acknowledgment message

In the new window pull down the File menu and select “Export …”.



ID: ACK

Name: General acknowledgment message

Event: A01

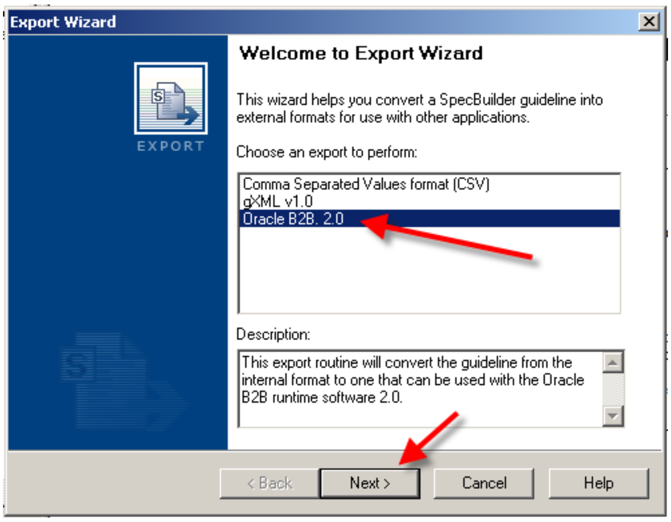
Event Name: ADT/ACK - Admit / visit notification

Standard: HL7 Version: 2.3.1

Purpose

An A01 event is intended to be used for "Admitted" patients only. An A01 event is sent as a result of a patient undergoing the admission process which assigns the patient to a bed. It signals the beginning of a patient's stay in a healthcare facility. Normally, this information is entered in the primary Patient Administration system and broadcast to the nursing units and ancillary systems. It includes short stay and John Doe admissions. For example, an A01 event can be used to notify: the pharmacy system that a patient has been admitted and may be legitimately prescribed drugs; the nursing system that the patient has been admitted and needs a care plan prepared; the finance system of the start of the billing period; the dietary system that a new patient has been installed and

Select Oracle B2B 2.0 and click Next.



Welcome to Export Wizard

This wizard helps you convert a SpecBuilder guideline into external formats for use with other applications.

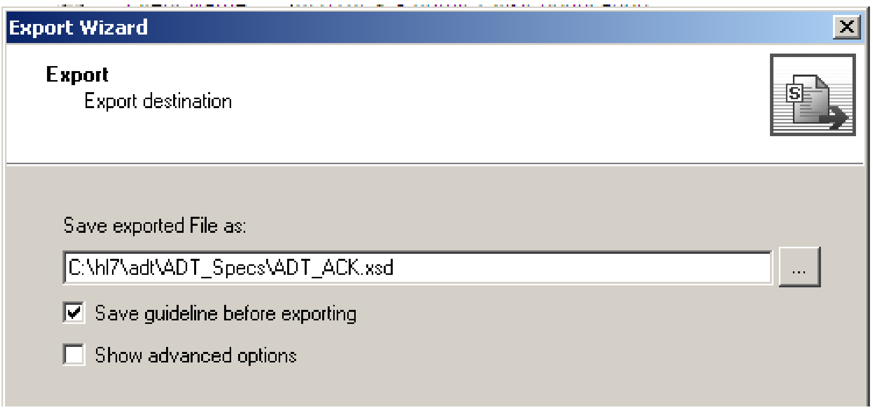
Choose an export to perform:

- Comma Separated Values format (CSV)
- gXML v1.0
- Oracle B2B. 2.0

Description:

This export routine will convert the guideline from the internal format to one that can be used with the Oracle B2B runtime software 2.0.

Enter C:\hl7\ADT_Specs\ADT_ACK into the “Save exported File as:” entry field, check the “Save guideline before exporting” checkbox and click Next.



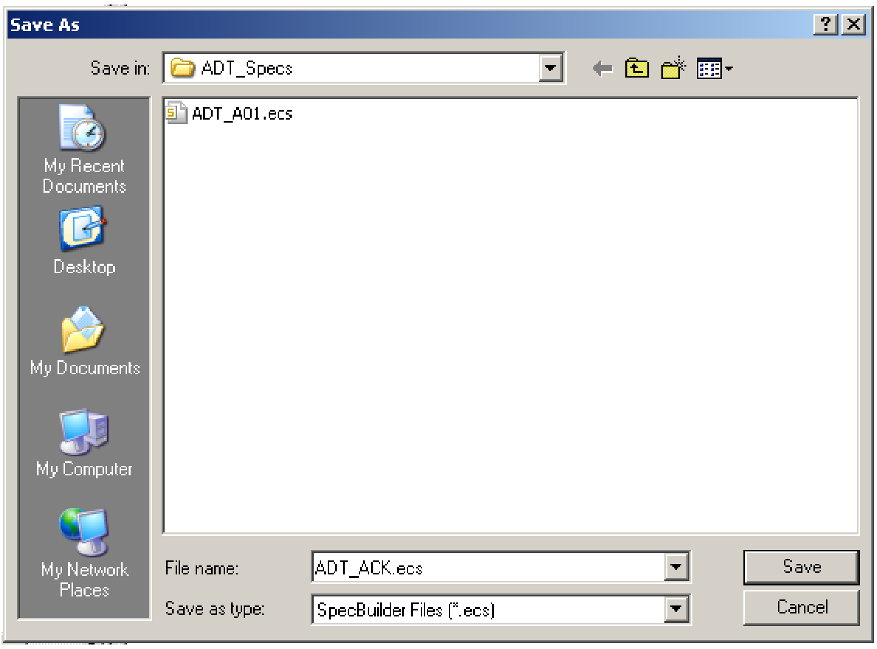
Save exported File as:

C:\hl7\adt\ADT_Specs\ADT_ACK.xsd

☑ Save guideline before exporting

☐ Show advanced options

Enter C:\hl7\ADT_Specs\ADT_ACK into the “File name:” text box and click Save.



File name: ADT_ACK.ecs

Save as type: SpecBuilder Files (*.ecs)

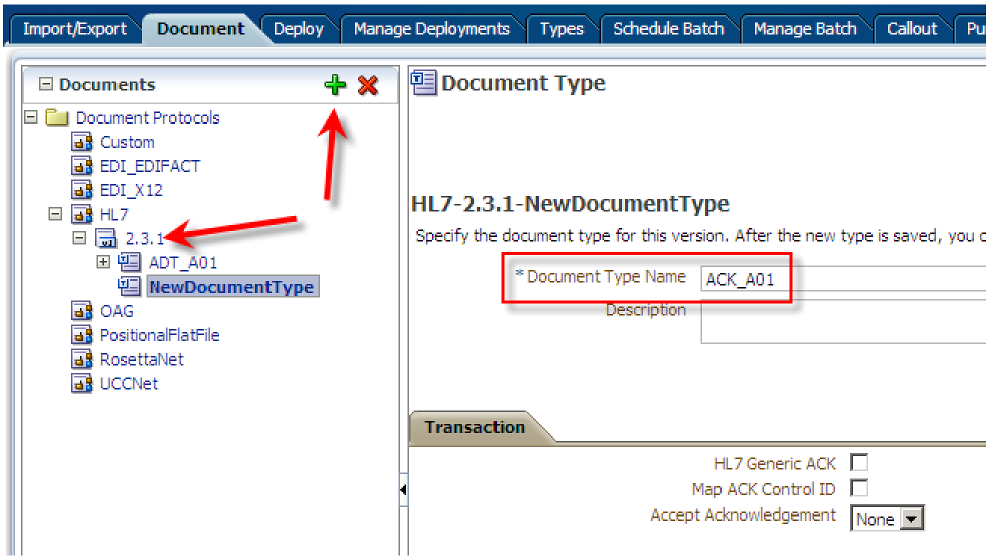
Click Finish. Close both windows. We are done with the B2B Document Editor for this exercise.

We exported the HL7 v2 ADT ACK definition in both the XML Schema form and the format used by the EDIFECS engine. The former can and will be used in the SOA Composite and the later is required to configure the B2B document for the trading partnership we will create shortly, and is used at runtime by the EDIFECS Engine.

Re-configure B2B Partnership

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, select the Document Tab, expand the HL7 node tree all the way to 2.3.1. With the 2.3.1 node selected, click the large Plus sign to create a new document under that node, name it ACK_A01 and click Save. This name is derived from the Message Type and Trigger Event in the HL7 message.



Document Type

HL7-2.3.1-NewDocumentType

Specify the document type for this version. After the new type is saved, you c

* Document Type Name: ACK_A01

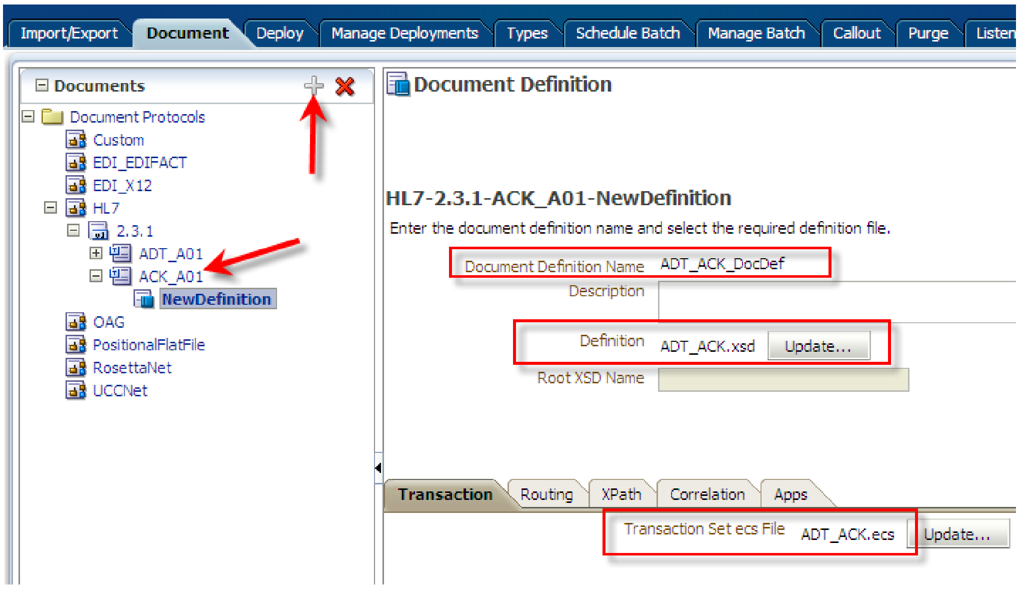
Description:

HL7 Generic ACK □

Map ACK Control ID □

Accept Acknowledgement: None

With the new node, ACK_A01, selected click the large Plus sign to add a new document under that node. Provide the name of ADT_ACK_DocDef, locate and choose c:\hl7\ADT_Specs\ADT_ACK.xsd file for document definition and c:\hl7\ADT_Specs\ADT_ACK.ecs file for Transaction Set ecs File.



Document Definition

HL7-2.3.1-ACK_A01-NewDefinition

Enter the document definition name and select the required definition file.

Document Definition Name: ADT_ACK_DocDef

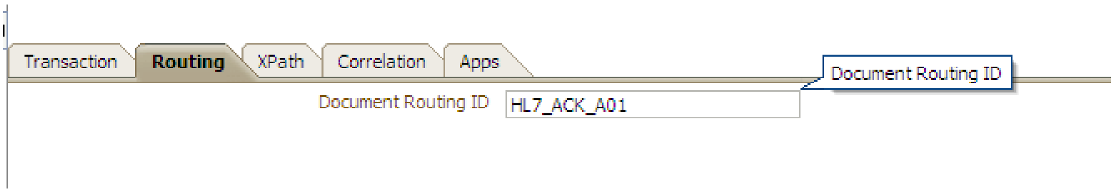
Description:

Definition: ADT_ACK.xsd

Root XSD Name:

Transaction Set ecs File: ADT_ACK.ecs

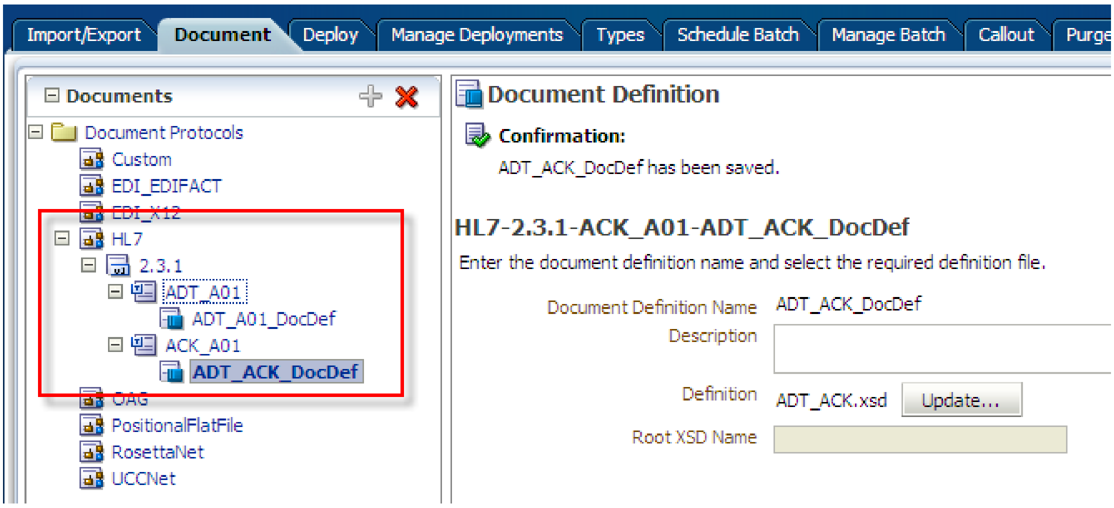
Click the Routing Tab and enter HL7_ACK_A01 into the Routing ID field.



Document Routing ID: HL7_ACK_A01

Click Save to save the changes.

A hierarchy like that shown below should be now defined.



Document Definition

Confirmation:
ADT_ACK_DocDef has been saved.

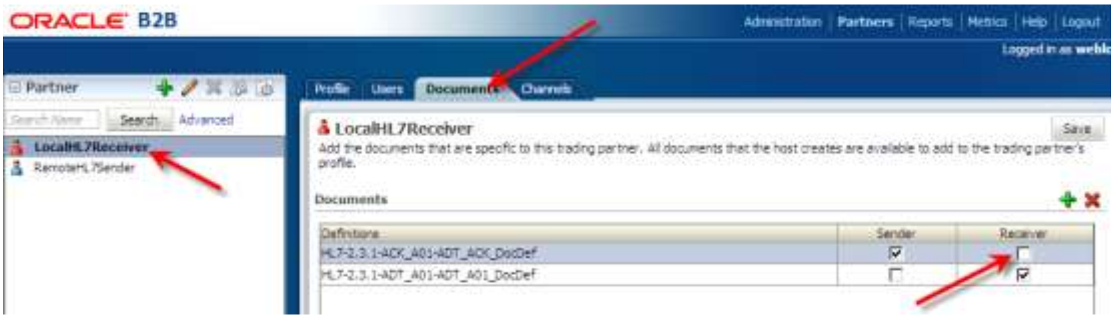
HL7-2.3.1-ACK_A01-ADT_ACK_DocDef
Enter the document definition name and select the required definition file.

Document Definition Name: ADT_ACK_DocDef
Description:
Definition: ADT_ACK.xsd
Root XSD Name:

This defines the additional document to be used for acknowledging message exchanges.

Now we need to modify configuration of the trading partners, the local partner and the remote partner.

Click the Partners navigational link, select the LocalHL7Receiver, click the Documents Tab and uncheck the Receiver checkbox associated with the ADT_ACK_DocDef document.

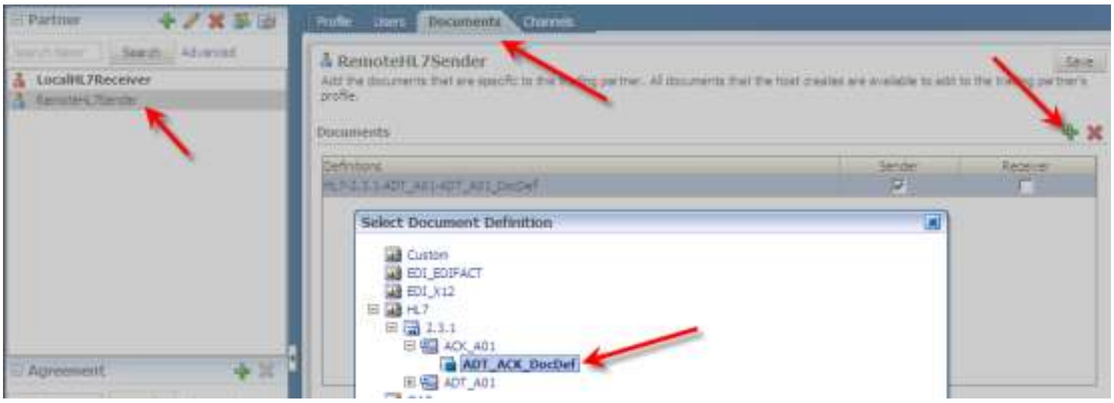


LocalHL7Receiver
Add the documents that are specific to this trading partner. All documents that the host creates are available to add to the trading partner's profile.

Documents

Definitions	Sender	Receiver
HL7-2.3.1-ACK_A01-ADT_ACK_DocDef	☑	□
HL7-2.3.1-ADT_A01-ADT_A01_DocDef	□	☑

Select the RemoteHL7Sender, click Documents Tab, click the large Plus sign to add a document, select HL7 → 2.3.1 → ACK_A01 → ADT_ACK_DocDef and click Add.

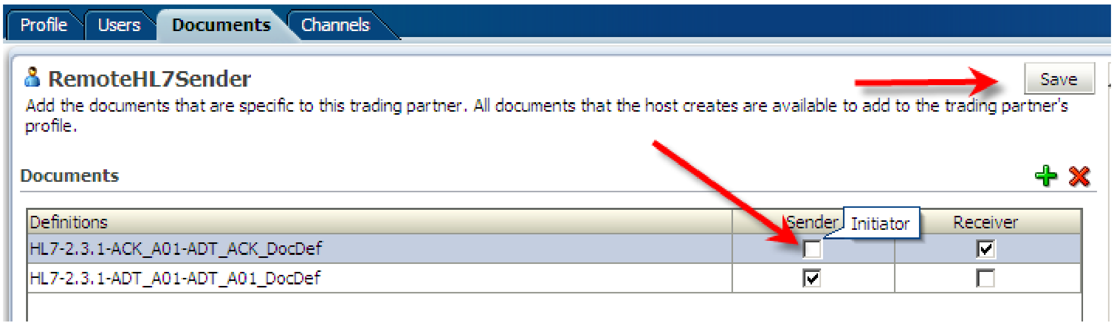


RemoteHL7Sender
Add the documents that are specific to this trading partner. All documents that the host creates are available to add to the trading partner's profile.

Documents

Definitions	Sender	Receiver
HL7-2.3.1-ADT_A01-ADT_A01_DocDef	☑	□

Uncheck the Sender checkbox and click Save.



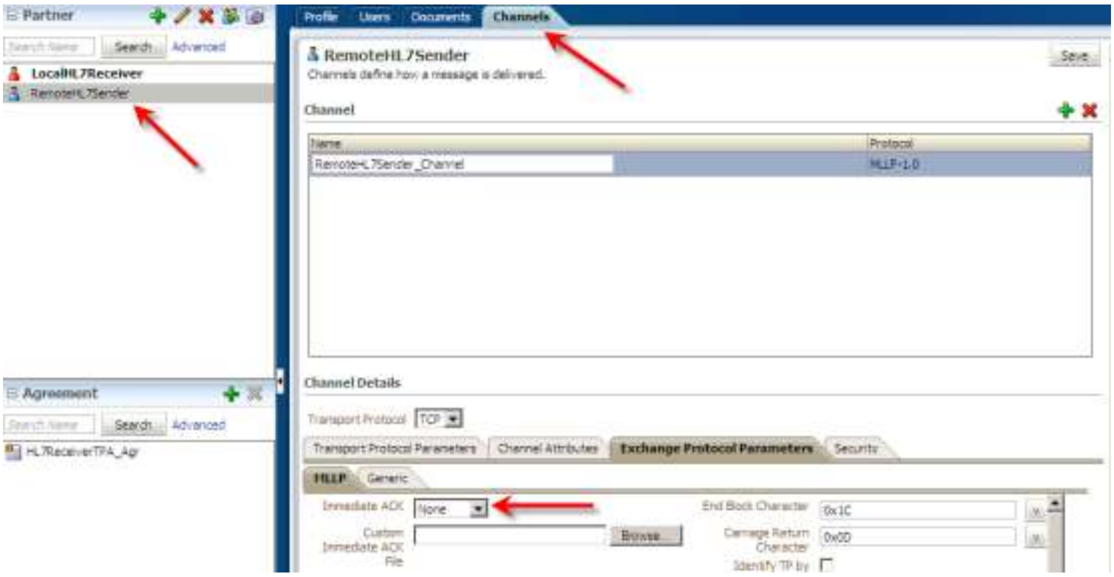
RemoteHL7Sender

Add the documents that are specific to this trading partner. All documents that the host creates are available to add to the trading partner's profile.

Documents

Definitions	Sender Initiator	Receiver
HL7-2.3.1-ACK_A01-ADT_ACK_DocDef	□	☑
HL7-2.3.1-ADT_A01-ADT_A01_DocDef	☑	□

With the RemoteHL7Sender partner selected click the Channels Tab. In the Channel Details section of the panel click the Exchange Protocol Parameters Tab. Choose “none” from the Immediate ACK drop-down and click the Save button.



RemoteHL7Sender

Channels define how a message is delivered.

Channel

Name	Protocol
RemoteHL7Sender_Channel	MLLP-1.0

Channel Details

Transport Protocol: TCP

Immediate ACK: None

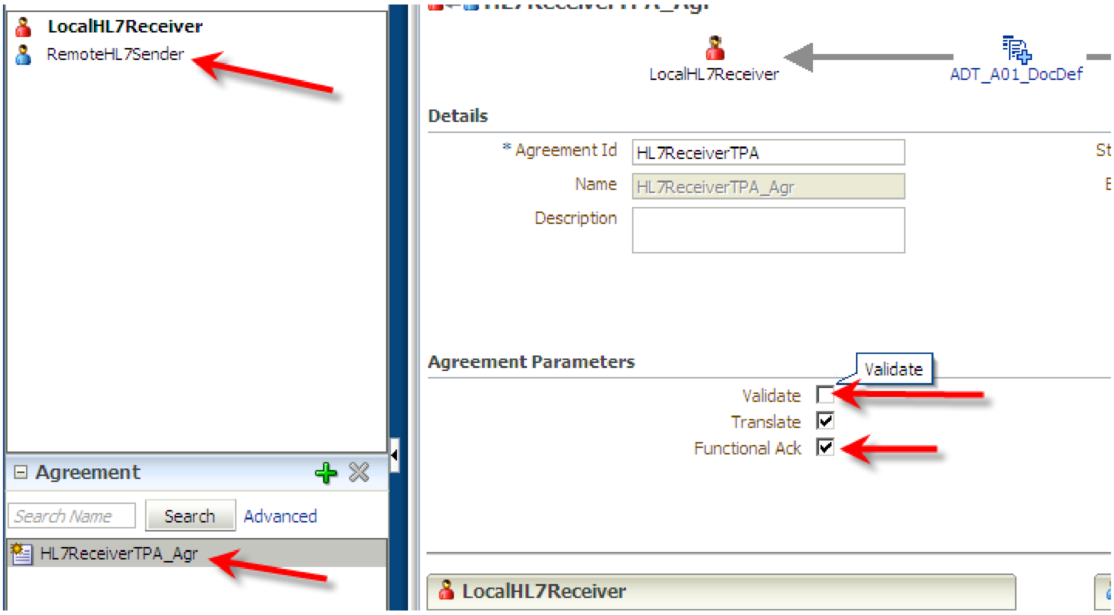
Custom Immediate ACK File:

End Block Character: 0x1C

Carriage Return Character: 0x0D

Identify TP by: □

With the RemoteHL7Sender selected click the HL7ReceiverTPA_Agr agreement, uncheck Validate and check Functional Ack then click Save abd Deploy. With Functional Ack enabled and Immediate Ack in the Trading Partner Profile disabled we are requiring acks to be generated after message parsing and validation, if any. This naturally introduces a delay in acknowledging receipt of a message which the Immediate Ack in the Trading Partner Profile was designed to avoid.



Details

* Agreement Id: HL7ReceiverTPA

Name: HL7ReceiverTPA_Agr

Description:

Agreement Parameters

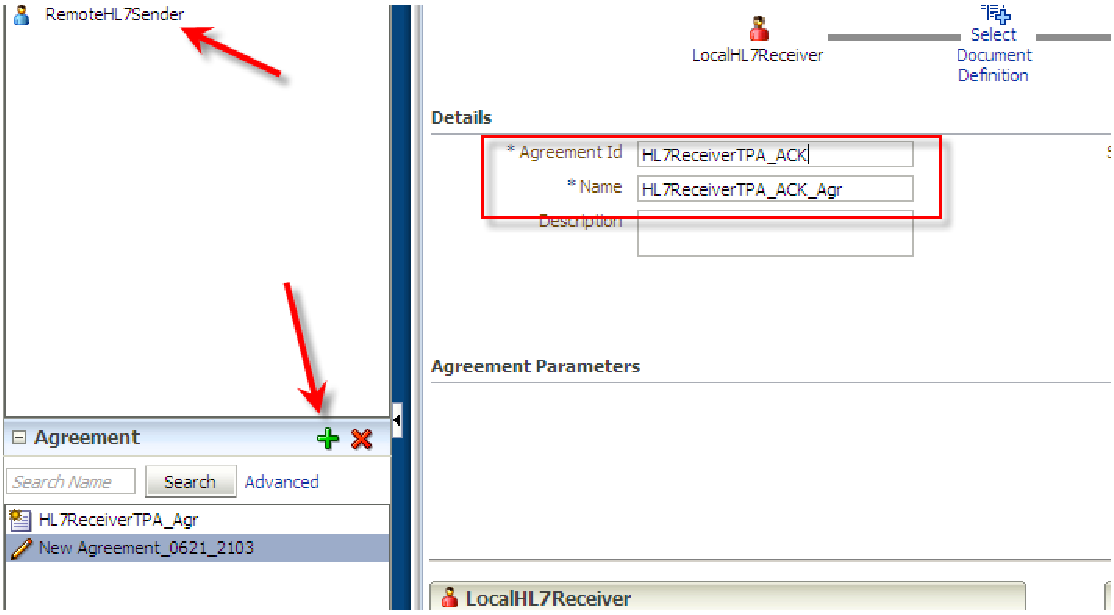
Validate □

Translate ☑

Functional Ack ☑

We now need to add an agreement for sending the ACK back to the sender.

Click the large Plus sign in the Agreements section to add a new trading partnership agreement. Provide the Agreement ID of HL7ReceiverTPA_ACK and the Name of HL7ReceiverTPA_ACK_Agr.



Details

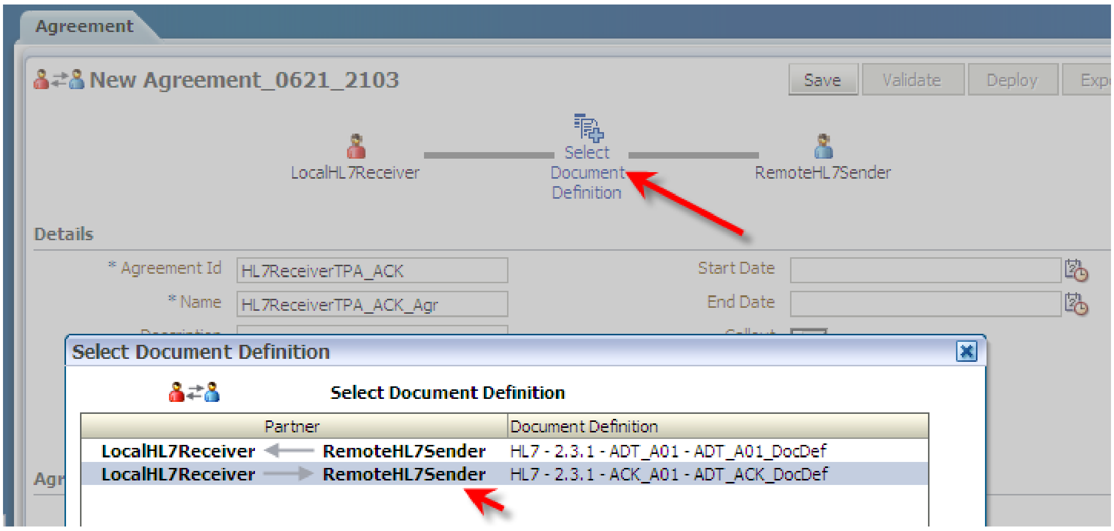
* Agreement Id: HL7ReceiverTPA_ACK

* Name: HL7ReceiverTPA_ACK_Agr

Description:

Agreement Parameters

Click the “Select Document Definition” icon in the top centre of the panel, select the definition of the document being sent from the local receiver to the remote sender and click OK.

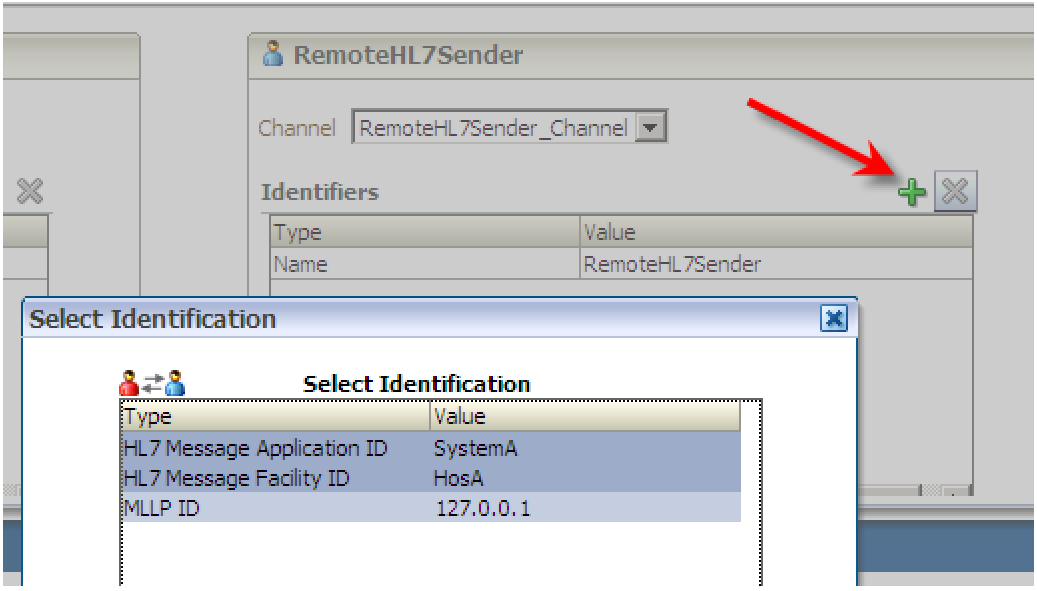


* Agreement Id: HL7ReceiverTPA_ACK
* Name: HL7ReceiverTPA_ACK_Agr
Start Date:
End Date:

Partner	Document Definition
LocalHL7Receiver ← RemoteHL7Sender	HL7 - 2.3.1 - ADT_A01 - ADT_A01_DocDef
LocalHL7Receiver → RemoteHL7Sender	HL7 - 2.3.1 - ACK_A01 - ADT_ACK_DocDef

Choose the RemoteHL7Sender_Channel from the Channel drop-down on the RemoteHL7Sender panel in the bottom right portion of the display.

In the bottom right panel click the large Plus sign to specify identifiers defined when the RemoteHL7Sender partner was created.

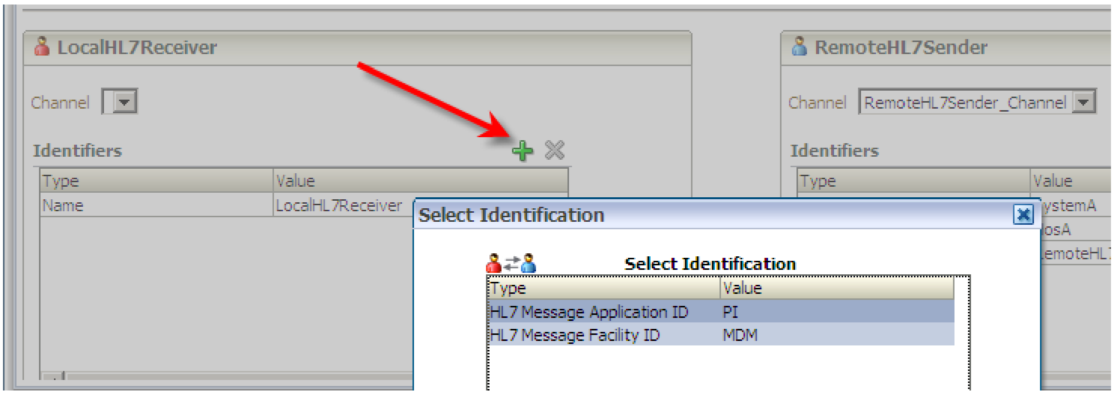


Channel: RemoteHL7Sender_Channel

Type	Value
Name	RemoteHL7Sender

Type	Value
HL7 Message Application ID	SystemA
HL7 Message Facility ID	HosA
MLLP ID	127.0.0.1

In the bottom left panel click the large Plus sign to add identifiers used by the LocalHL7Receiver partner.



LocalHL7Receiver

Channel

Identifiers

Type	Value
Name	LocalHL7Receiver

RemoteHL7Sender

Channel RemoteHL7Sender_Channel

Identifiers

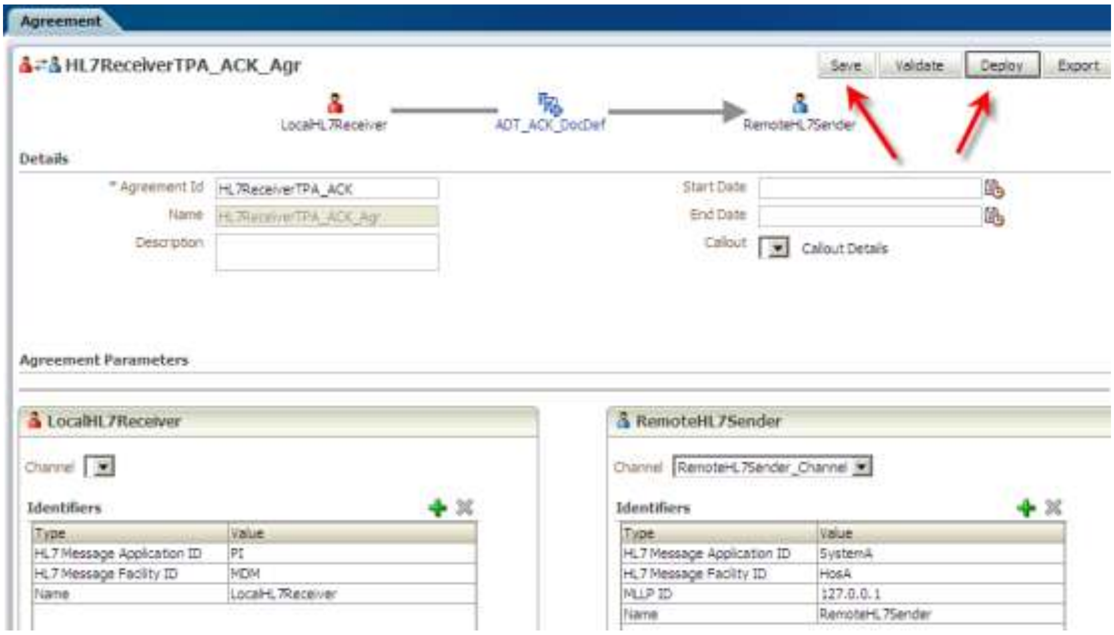
Type	Value
	ystemA
	osA
	emoteHL

Select Identification

Select Identification

Type	Value
HL7 Message Application ID	PI
HL7 Message Facility ID	MDM

Click Save to save the new trading partnership agreement and click Deploy to deploy it.



HL7ReceiverTPA_ACK_Agr

Details

* Agreement Id: HL7ReceiverTPA_ACK

Name: HL7ReceiverTPA_ACK_Agr

Description:

Start Date:

End Date:

Callout: Callout Details

Agreement Parameters

LocalHL7Receiver

Channel

Identifiers

Type	Value
HL7 Message Application ID	PI
HL7 Message Facility ID	MDM
Name	LocalHL7Receiver

RemoteHL7Sender

Channel RemoteHL7Sender_Channel

Identifiers

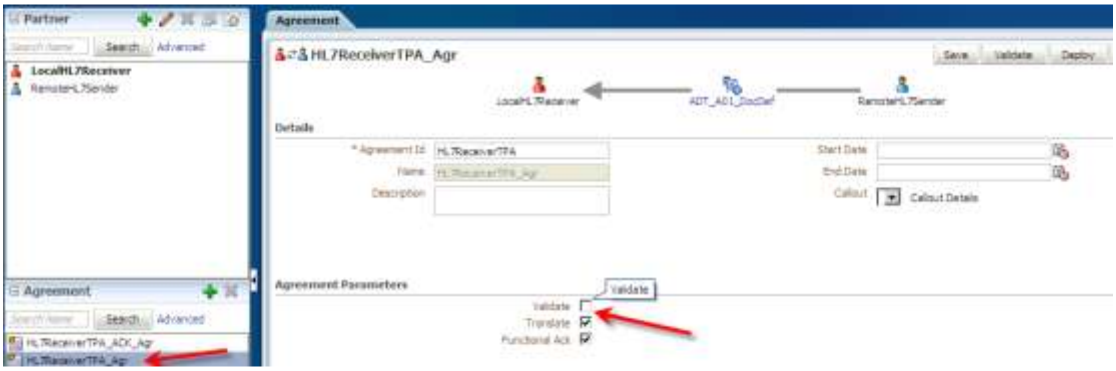
Type	Value
HL7 Message Application ID	SystemA
HL7 Message Facility ID	HosA
MLLP ID	127.0.0.1
Name	RemoteHL7Sender

No changes are required to the SOA Composite. The ACK will still be generated by the B2B but it will be generated after the message is parsed and validated, if configured, and before it is passed to the SOA Composite for application processing.

Exercise HL7 Inbound solution

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

Make sure that the Validate property of the Agreement Parameters is not checked. If it is, uncheck it, Save and Deploy the agreement.



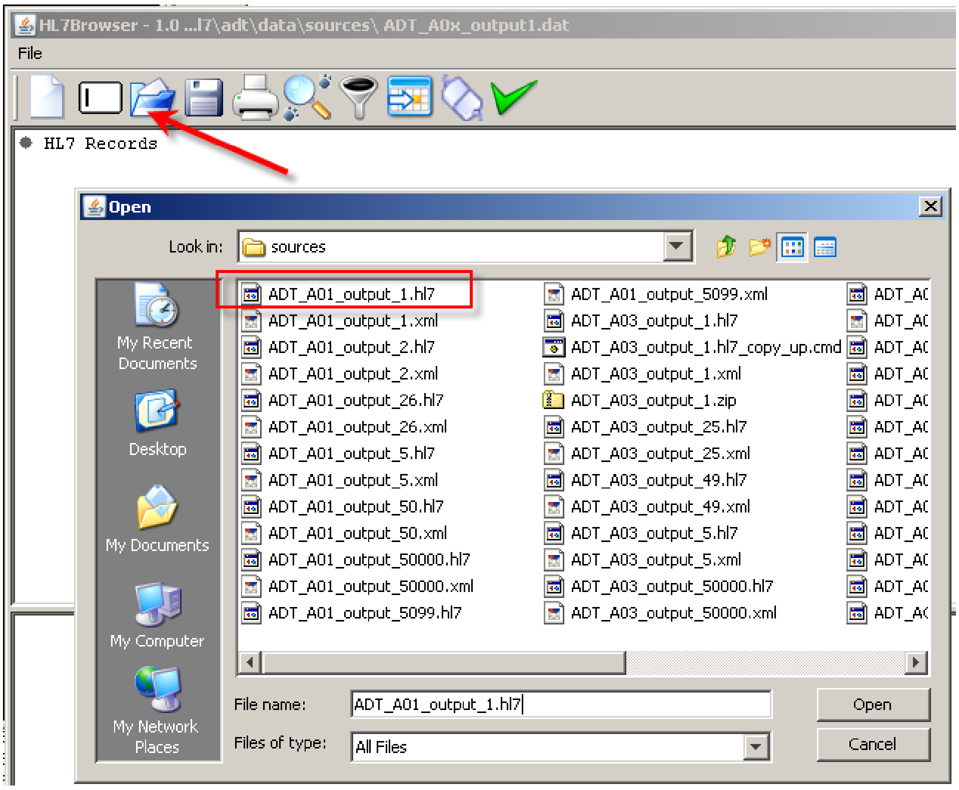
* Agreement Id: HL7ReceiverTPA
Name: HL7ReceiverTPA_Agr
Description:

Validate ☐
Translate ☑
Functional Ack ☑

Open a command window and run the HL7Browser:

```
C:\jdk1.6.0_20\bin\java.exe -jar C:\tools\Hl7Browser.1.0\HL7Browser.jar
```

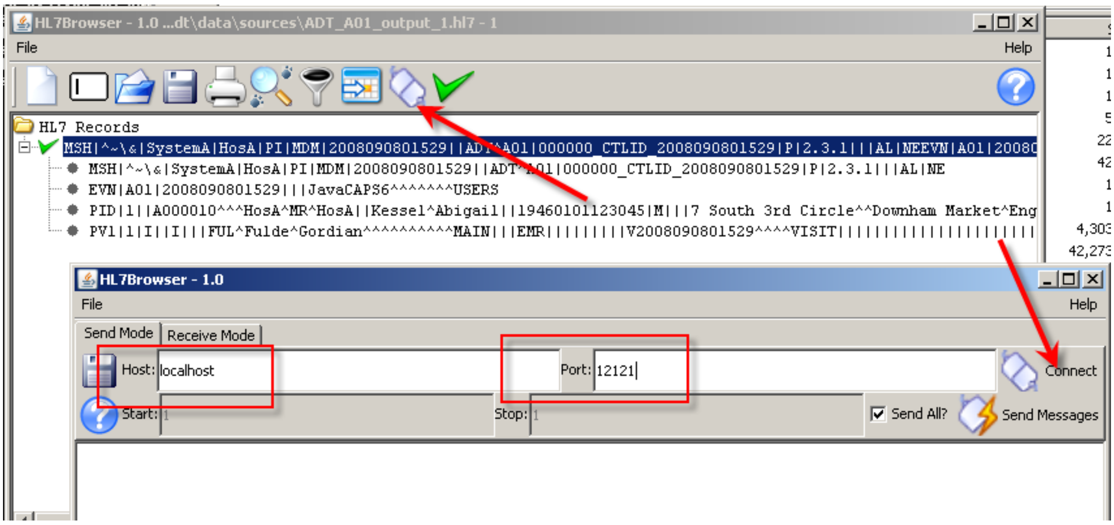
When the UI appears click the “Open an HL7 File” button, locate the ADT A01 transaction file, C:\hl7\adt\data\sources\ADT_A01_output1.hl7, and open it.



ADT_A01_output_1.hl7
ADT_A01_output_1.xml
ADT_A01_output_2.hl7
ADT_A01_output_2.xml
ADT_A01_output_26.hl7
ADT_A01_output_26.xml
ADT_A01_output_5.hl7
ADT_A01_output_5.xml
ADT_A01_output_50.hl7
ADT_A01_output_50.xml
ADT_A01_output_50000.hl7
ADT_A01_output_50000.xml
ADT_A01_output_5099.hl7
ADT_A01_output_5099.xml
ADT_A03_output_1.hl7
ADT_A03_output_1.hl7_copy_up.cmd
ADT_A03_output_1.xml
ADT_A03_output_1.zip
ADT_A03_output_25.hl7
ADT_A03_output_25.xml
ADT_A03_output_49.hl7
ADT_A03_output_49.xml
ADT_A03_output_5.hl7
ADT_A03_output_5.xml
ADT_A03_output_50000.hl7
ADT_A03_output_50000.xml

File name: ADT_A01_output_1.hl7
Files of type: All Files

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.



```
HL7 Records
MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NEEVN|A01|20080
  MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NE
  EVN|A01|2008090801529|||JavaCAPS6^^^^^^^USERS
  PID|1||A000010^^^HosA^MR^HosA||Kessel^Abigail||19460101123045|M|||7 South 3rd Circle^^Downham Market^Eng
  PV1|1|I||I|||FUL^Fulde^Gordian^^^^^^^^^^MAIN|||EMR||||||||V2008090801529^^^^VISIT||||||||||||||||||||
```

Host: localhost

Port: 12121

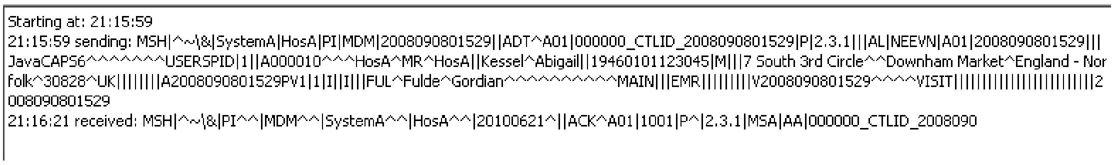
Start: 1

Stop: 1

☑ Send All?

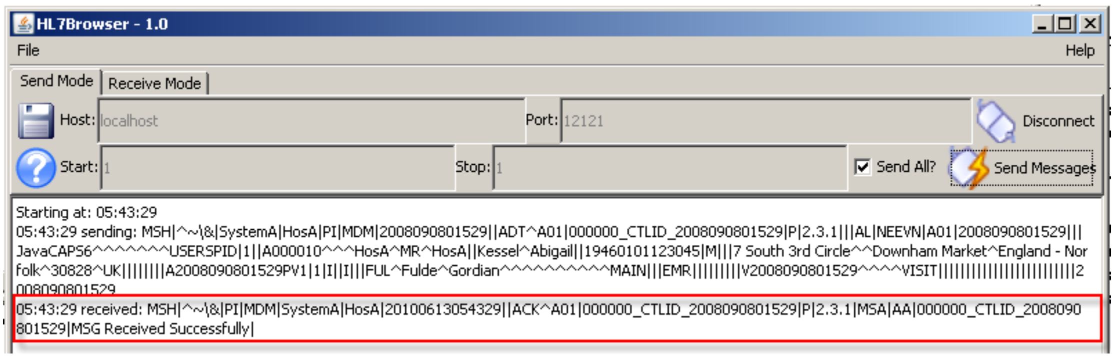
Once connected, click the “Send Messages” button.

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



```
Starting at: 21:15:59
21:15:59 sending: MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NEEVN|A01|2008090801529|||
JavaCAPS6^^^^^^^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
21:16:21 received: MSH|^~\&|PI^^|MDM^^|SystemA^^|HosA^^|20100621^||ACK^A01|1001|P^|2.3.1|MSA|AA|000000_CTLID_2008090
```

This ACK is different from the Immediate ACK, shown in the picture below.



Host: localhost

Port: 12121

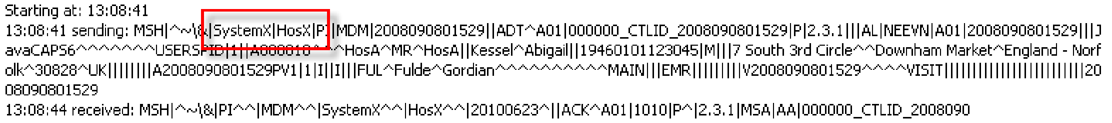
Start: 1

Stop: 1

☑ Send All?

```
Starting at: 05:43:29
05:43:29 sending: MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NEEVN|A01|2008090801529|||
JavaCAPS6^^^^^^^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
05:43:29 received: MSH|^~\&|PI|MDM|SystemA|HosA|20100613054329||ACK^A01|000000_CTLID_2008090801529|P|2.3.1|MSA|AA|000000_CTLID_2008090
801529|MSG Received Successfully|
```

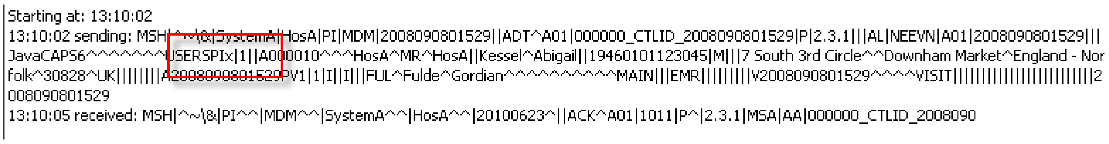
Let’s deliberately invalidate the sender, by changing the sending application and facility to SystemX and HosX, then let’s submit the modified message to see the acknowledgement.



```
Starting at: 13:08:41
13:08:41 sending: MSH|^~\&|SystemX|HosX|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NEEVN|A01|2008090801529|||J
avaCAPS6^^^^^^^USERSPID|1||A000010^^^HosA^MR^HosA||Kessel^Abigail||19460101123045|M|||7 South 3rd Circle^^Downham Market^England - Norf
olk^30828^UK|||||||A2008090801529PV1|1|I||I|||FUL^Fulde^Gordian^^^^^^^^^^MAIN|||EMR||||||||V2008090801529^^^^VISIT||||||||||||||||||||||20
08090801529
13:08:44 received: MSH|^~\&|PI^^|MDM^^|SystemX^^|HosX^^|20100623^||ACK^A01|1010|P^|2.3.1|MSA|AA|000000_CTLID_2008090
```

The B2B infrastructure returns MSA-2 AA, application accept, even though we know the sender and the receiver are not correct.

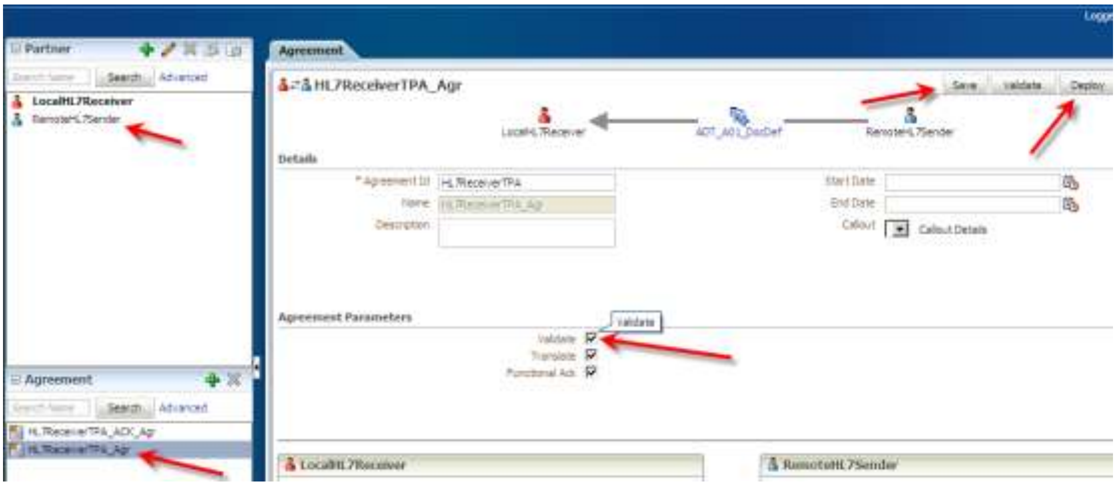
Let’s change sender parameters back to SystemA and HosA, but modify the PID segment ID to read PIx. Let’s re-submit the message and see the acknowledgement.



```
Starting at: 13:10:02
13:10:02 sending: MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NEEVN|A01|2008090801529|||
JavaCAPS6^^^^^^^^USERSPIx|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
13:10:05 received: MSH|^~\&|PI^^|MDM^^|SystemA^^|HosA^^|20100623^||ACK^A01|1011|P^|2.3.1|MSA|AA|000000_CTLID_2008090
```

Now also the acknowledgment indicates that all is well (MSA-2 == AA) even though we know this not to be the case.

Let's now enable Validation in the partnership agreement, save and deploy the agreement.

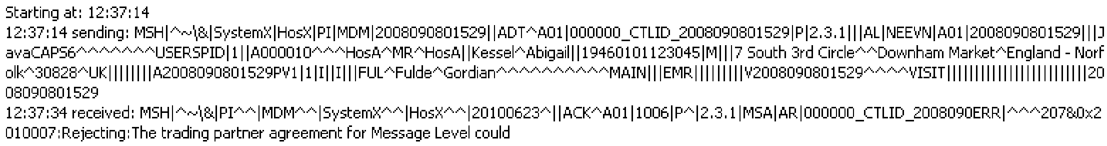


Agreement Id: HL7ReceiverTPA
Name: HL7ReceiverTPA_Agr

Agreement Parameters
Validate ☑
Translate ☑
Functional Ack ☑

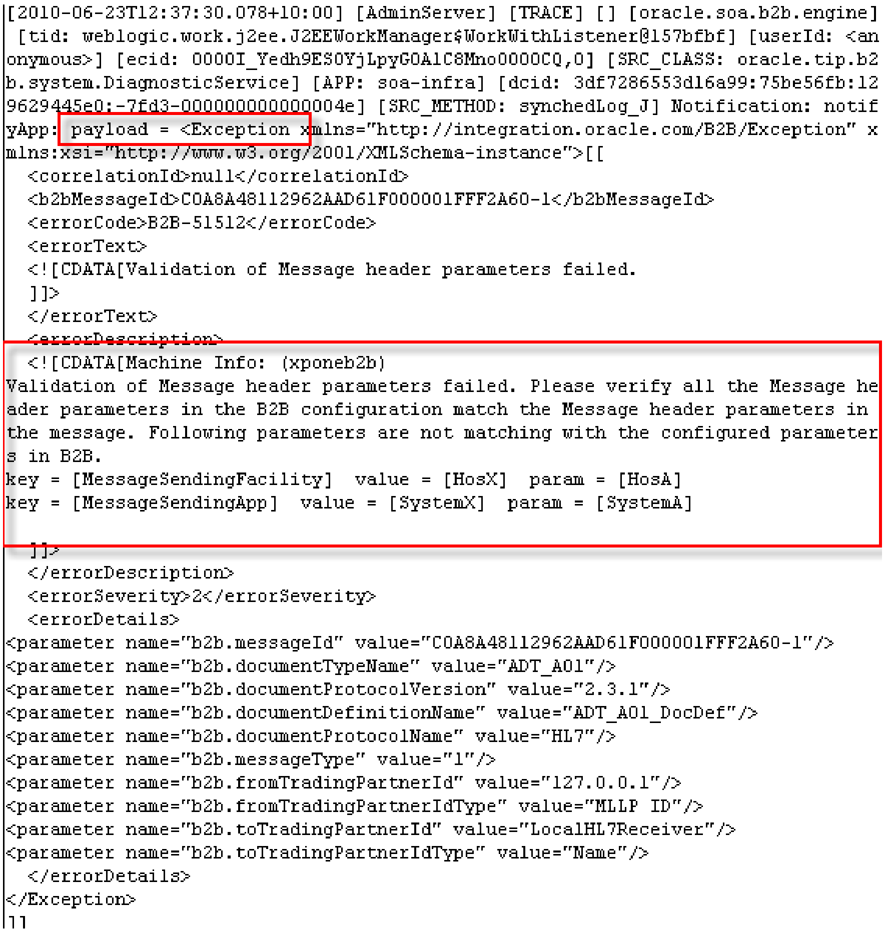
Let's again change the message such that segment IDs are correct but the sender is SystemX and HosX, then re-submit the message.

The B2B infrastructure returns a negative acknowledgement with MSA-2 set to AR – Application Rejection, with a truncated error message.



```
Starting at: 12:37:14
12:37:14 sending: MSH|^~\&|SystemX|HosX|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NEEVN|A01|2008090801529|||J
avaCAPS6^^^^^^^^USERSPID|1||A000010^^^HosA^MR^HosA||Kessel^Abigail||19460101123045|M|||7 South 3rd Circle^^Downham Market^England - Norf
olk^30828^UK|||||||A2008090801529PV1|1|I||I|||FUL^Fulde^Gordian^^^^^^^^^^^MAIN|||EMR||||||||V2008090801529^^^^VISIT|||||||||||||||||||||||||20
08090801529
12:37:34 received: MSH|^~\&|PI^^|MDM^^|SystemX^^|HosX^^|20100623^||ACK^A01|1006|P^|2.3.1|MSA|AR|000000_CTLID_2008090ERR|^^^207&0x2
010007:Rejecting:The trading partner agreement for Message Level could
```

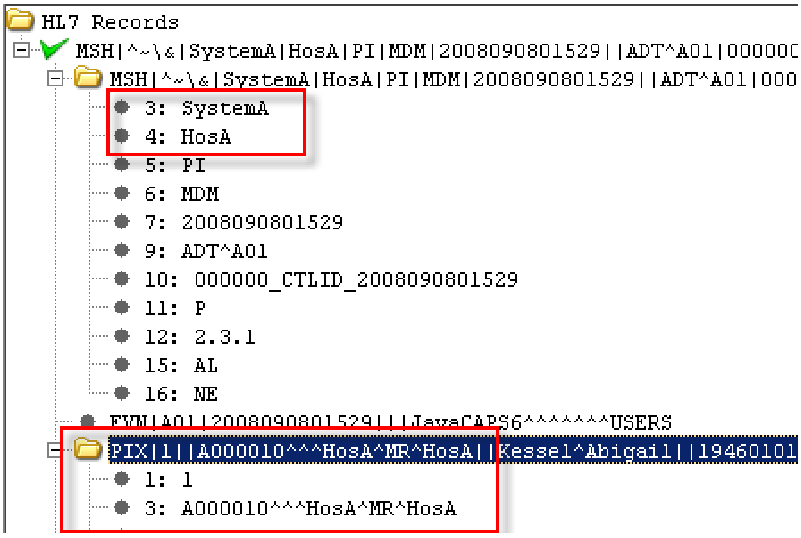
Inspection of server log, looking for expression "payload = <Exception", shows a much more intelligible message:



```
[2010-06-23T12:37:30.078+10:00] [AdminServer] [TRACE] [] [oracle.soa.b2b.engine]
 [tid: weblogic.work.j2ee.J2EEWorkManager$WorkWithListener@157bfbf] [userId: <an
onymous>] [ecid: 0000I_Yedh9ES0YjLpyG0A1C8Mno0000CQ,0] [SRC_CLASS: oracle.tip.b2
b.system.DiagnosticService] [APP: soa-infra] [dcid: 3df7286553d16a99:75be56fb:12
9629445e0:-7fd3-000000000000004e] [SRC_METHOD: synchedLog_J] Notification: notif
yApp: payload = <Exception xmlns="http://integration.oracle.com/B2B/Exception" x
mlns:xsi="http://www.w3.org/2001/XMLSchema-instance">[[
  <correlationId>null</correlationId>
  <b2bMessageId>C0A8A48112962AAD61F000001FFF2A60-1</b2bMessageId>
  <errorCode>B2B-51512</errorCode>
  <errorText>
  <![CDATA[Validation of Message header parameters failed.
  ]]>
  </errorText>
  <errorDescription>
  <![CDATA[Machine Info: (xponeb2b)
Validation of Message header parameters failed. Please verify all the Message he
ader parameters in the B2B configuration match the Message header parameters in
the message. Following parameters are not matching with the configured parameter
s in B2B.
key = [MessageSendingFacility]  value = [HosX]  param = [HosA]
key = [MessageSendingApp]  value = [SystemX]  param = [SystemA]

  ]]>
  </errorDescription>
  <errorSeverity>2</errorSeverity>
  <errorDetails>
<parameter name="b2b.messageId" value="C0A8A48112962AAD61F000001FFF2A60-1"/>
<parameter name="b2b.documentTypeName" value="ADT_A01"/>
<parameter name="b2b.documentProtocolVersion" value="2.3.1"/>
<parameter name="b2b.documentDefinitionName" value="ADT_A01_DocDef"/>
<parameter name="b2b.documentProtocolName" value="HL7"/>
<parameter name="b2b.messageType" value="1"/>
<parameter name="b2b.fromTradingPartnerId" value="127.0.0.1"/>
<parameter name="b2b.fromTradingPartnerIdType" value="MLLP ID"/>
<parameter name="b2b.toTradingPartnerId" value="LocalHL7Receiver"/>
<parameter name="b2b.toTradingPartnerIdType" value="Name"/>
  </errorDetails>
</Exception>
]]
```

Let’s fix the sender parameters but break the message structure so that the segment ID of PID is changed to PIX.



```
HL7 Records
  MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|00000C
    MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000
      3: SystemA
      4: HosA
      5: PI
      6: MDM
      7: 2008090801529
      9: ADT^A01
      10: 000000_CTLID_2008090801529
      11: P
      12: 2.3.1
      15: AL
      16: NE
    EVN|A01|2008090801529|||JavaCAPS6^^^^^^^USERS
    PIX|1||A000010^^^HosA^MR^HosA||Kessel^Abigail||19460101
      1: 1
      3: A000010^^^HosA^MR^HosA
```

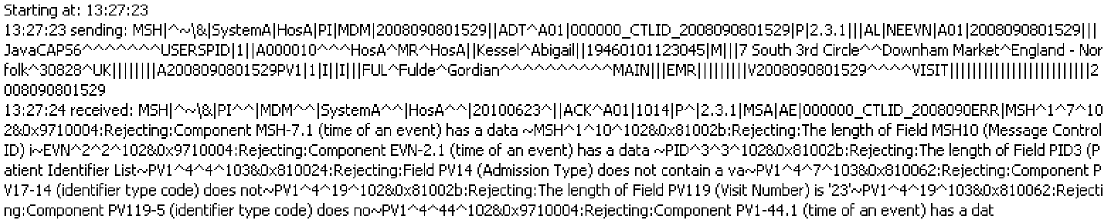
Let’s resubmit the message and inspect the acknowledgement. The negative acknowledgement this time has the MSA-2 set to AE – Application Error. The acknowledgement looks like that shown below.

```
12:55:26 received:
MSH|^~\&|PI^^|MDM^^|SystemA^^|HosA^^|20100623^||ACK^A01|1007|P^|2.3.1|MSA|AE|000000_CT
```

```
LID_2008090ERR|MSH^1^7^102&0x9710004:Rejecting:Component MSH-7.1 (time of an event)
has a data ~MSH^1^10^102&0x81002b:Rejecting:The length of Field MSH10 (Message Control
ID) i~EVN^2^2^102&0x9710004:Rejecting:Component EVN-2.1 (time of an event) has a data
~^^^100&0x810070:Rejecting:Unrecognized data was found in the data file. The
last~PID^3^^100&0x81004d:Rejecting:Segment PID (PID - patient identification
segment)~PV1^4^4^103&0x810024:Rejecting:Field PV14 (Admission Type) does not contain a
va~PV1^4^7^103&0x810062:Rejecting:Component PV17-14 (identifier type code) does
not~PV1^4^19^102&0x81002b:Rejecting:The length of Field PV119 (Visit Number) is
'23'~PV1^4^19^103&0x810062:Rejecting:Component PV119-5 (identifier type code) does
no~PV1^4^44^102&0x9710004:Rejecting:Component PV1-44.1 (time of an event) has a dat
```

Strictly speaking our message is not valid. Certain values exceed HL7-mandated data lengths and other issues are present. When we were not validating messages they were accepted as valid, even with the faults they exhibit, and forwarded to the SOA Composite.

Let's fix the message so that both the sender parameters and segment IDs are correct, then re-submit the message.



```
Starting at: 13:27:23
13:27:23 sending: MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008090801529|P|2.3.1|||AL|NEEVN|A01|2008090801529|||
JavaCAPS6^^^^^^^^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
13:27:24 received: MSH|^~\&|PI^^|MDM^^|SystemA^^|HosA^^|20100623^||ACK^A01|1014|P^|2.3.1|MSA|AE|000000_CTLID_2008090ERR|MSH^1^7^10
2&0x9710004:Rejecting:Component MSH-7.1 (time of an event) has a data ~MSH^1^10^102&0x81002b:Rejecting:The length of Field MSH10 (Message Control
ID) i~EVN^2^2^102&0x9710004:Rejecting:Component EVN-2.1 (time of an event) has a data ~PID^3^3^102&0x81002b:Rejecting:The length of Field PID3 (P
atient Identifier List~PV1^4^4^103&0x810024:Rejecting:Field PV14 (Admission Type) does not contain a va~PV1^4^7^103&0x810062:Rejecting:Component P
V17-14 (identifier type code) does not~PV1^4^19^102&0x81002b:Rejecting:The length of Field PV119 (Visit Number) is '23'~PV1^4^19^103&0x810062:Rejecti
ng:Component PV119-5 (identifier type code) does no~PV1^4^44^102&0x9710004:Rejecting:Component PV1-44.1 (time of an event) has a dat
```

The message is recognized as invalid and a negative acknowledgment is returned.

As we see the Immediate Ack setting, which we used in the exercise in the previous article, causes the infrastructure to always return success because it is sent before message parsing takes place. With the Immediate Ack turned off and the Functional Ack turned on in the Agreement the acknowledgment is produced after message parse and reflects the outcome of parsing, including rejections when sender is not recognized and appropriate for the agreement and when the message structure is invalid.

Repairing and resubmitting the message until it is valid and accepted is left as an exercise for the reader ☺

Summary

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

In this article a simple Oracle SOA Suite 11g B2B infrastructure-based HL7 v2 Receiver project for ADT A01 messages was modified to support acknowledgements sent after message processing. This solution was then exercised and behavior was observed.

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