Oracle SOA Suite 11g B2B

HL7 v2 Inbound to WebLogic JMS Queue

michael.w.czapski@gmail.com

May 2011
Rev. 1.0.0

Contents

Introduction

I notice that people used to the eGate/Java CAPS way of doing things, when looking at migrating to the SOA Suite for HL7 messaging, are trying to reproduce the pattern "HL7v2Adapter→JMS Queue". This is not necessary when using SOA Suite but can be done if one insists. This article walks through the process of implementing this pattern using Oracle SOA Suite 11g R1 PS3.

The process will follow these steps:

1. Obtain and configure the QBrowser tool for JMS browsing
2. Obtain and configure the HL7 Sender tool
3. Create two WebLogic JMS Queues to be used in the solution
4. Create and deploy a HL7 v2 Inbound Trading Partnership Agreement
5. Submit HL7 v2 messages and inspect them in the corresponding JMS Queue
6. Repeat steps 4 and 5 for another inbound stream

We will demonstrate that Oracle SOA Suite B2B HL7 infrastructure can be configured to receive message streams over multiple inbound MLLP channels and deliver each stream to a distinct JMS destination, much as eGate and Java CAPS solutions used to do.

Preliminaries

I assume the existence of a functional SOA Suite 11g installation.

I assume a clean, unused B2B environment. This can be accomplished by purging all runtime and design time data. To not lose work one can export the B2B repository to a ZIP archive for later restoration. It is perfectly reasonable to work through this article in an environment with existing objects but one will have to account for that while working and I cannot write an article which accounts for the environments other people have and various objects people have there.

I am using a SOA Suite 11g R1 PS3 for this article but PS2 or PS4 should do just fine, though in PS4 the exact steps needed to create JMS Queues using the WebLogic Admin Console may vary slightly.

I use Windows XP for all work done for this article. Unix installation would have done just as well, with directory paths suitably changed.

Obtain and configure the QBrowser tool, which will be used to inspect messages in WebLogic JMS Queues. The steps are discussed in the blog article "Using QBrowser v2 with WebLogic JMS for 10.3", at http://blogs.czapski.id.au/2011/05/using-qbrowser-v2-with-weblogic-jms-for-10-3.

Obtain and configure (unzip to a convenient directory, for example C:\tools\CMDHL7) the HL7 tools, discussed in blog article "HL7 Sender, HL7 Listener and HL7 Proxy – developer tools I always wanted" and available at http://blogs.czapski.id.au/2010/12/hl7-sender-hl7-listener-and-hl7-proxy-developer-tools-i-always-wanted. The tool we will use is the CMDHL7Sender.

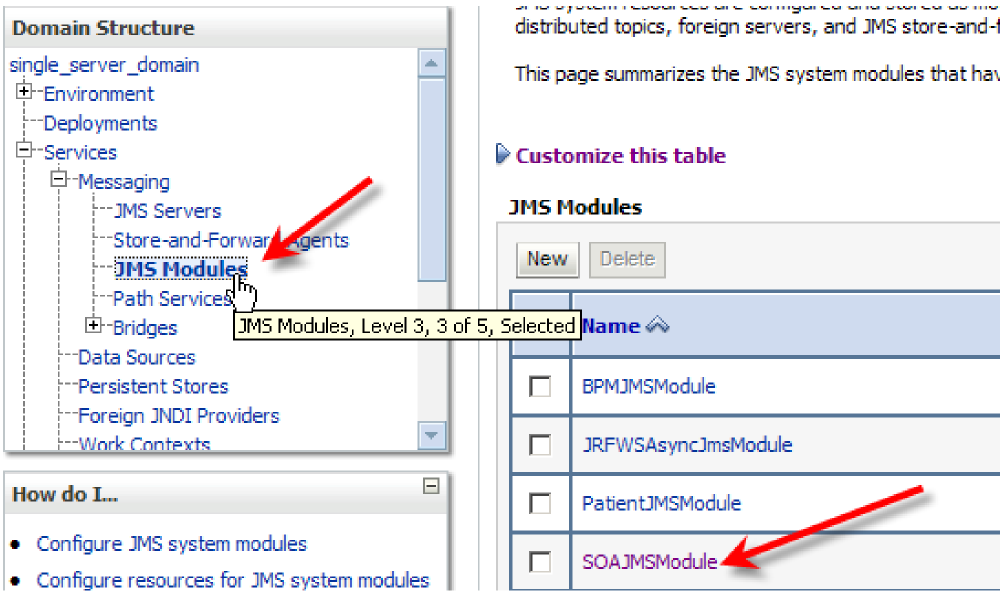
Create or obtain HL7 sample message files. I am using data from http://blogs.czapski.id.au/wp-content/uploads/2010/06/HL7_messages_sources.zip.

Configuration of an inbound requires a document definition. I will use document definition developed in the article "Healthcare Enterprise – IT Architecture Building Blocks – Canonical Message Model for a HL7 Enterprise", available at http://blogs.czapski.id.au/2010/10/healthcare-enterprise-%e2%80%93-it-architecture-building-blocks-canonical-message-model-for-a-hl7-enterprise. If you have not already done so, work your way through this article, or download the document definition archive http://blogs.czapski.id.au/wp-content/uploads/2010/12/CMM_v1.0.zip, and unzip it to a suitable directory, for example "C:\Documents and Settings\Administrator\My Documents\Oracle\Oracle B2B\Guidelines".

Create JMS Queues

Create a couple of JMS Queues, qHL7fromHosA and qHL7fromHosB, using WebLogic Administration Console.

Start WebLogic Administration Console: http://localhost:7001/console.
Navigate the hierarchy Services→Messaging. Click on "JMS Modules", then on "SOAJMSModule".



Domain Structure

single_server_domain
- Environment
- Deployments
- Services
  - Messaging
    - JMS Servers
    - Store-and-Forward Agents
    - JMS Modules
    - Path Services
    - Bridges
  - Data Sources
  - Persistent Stores
  - Foreign JNDI Providers
  - Work Contexts

JMS Modules, Level 3, 3 of 5, Selected

How do I...

- Configure JMS system modules
- Configure resources for JMS system modules

distributed topics, foreign servers, and JMS store-and-f

This page summarizes the JMS system modules that hav

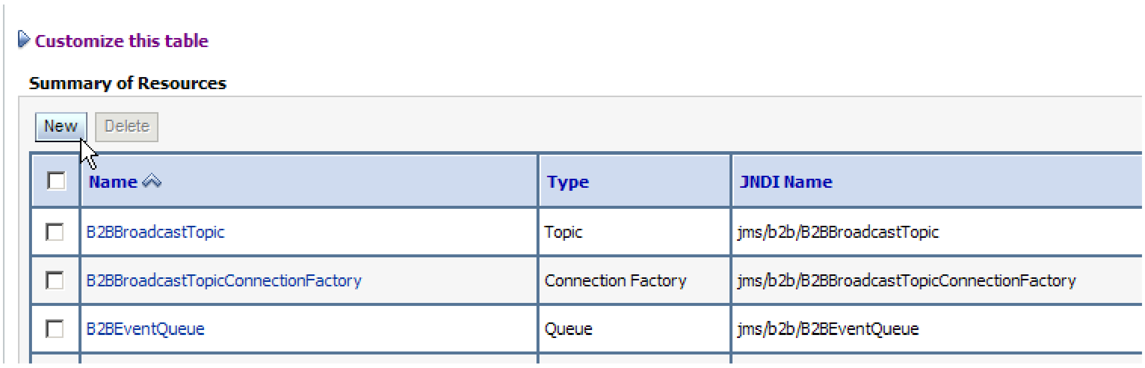
Customize this table

JMS Modules

New Delete

□	Name
□	BPMJMSModule
□	JRFWSAsyncJmsModule
□	PatientJMSModule
□	SOAJMSModule

Click "New".



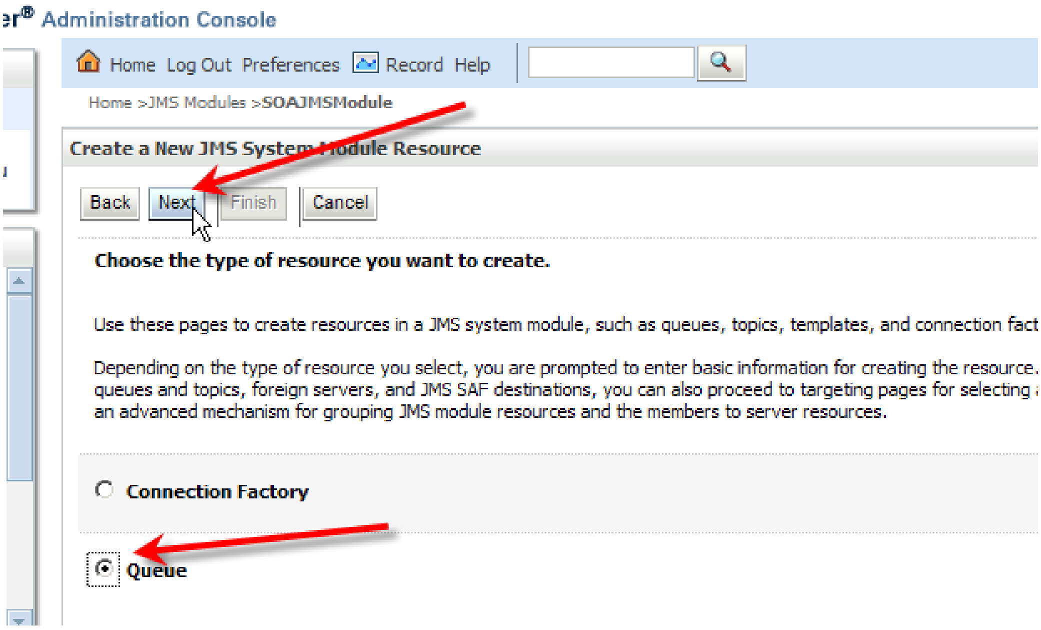
Customize this table

Summary of Resources

New Delete

□	Name	Type	JNDI Name
□	B2BBroadcastTopic	Topic	jms/b2b/B2BBroadcastTopic
□	B2BBroadcastTopicConnectionFactory	Connection Factory	jms/b2b/B2BBroadcastTopicConnectionFactory
□	B2BEventQueue	Queue	jms/b2b/B2BEventQueue

Select Queue and click Next.



er® Administration Console

Home >JMS Modules >SOAJMSModule

Create a New JMS System Module Resource

Back Next Finish Cancel

Choose the type of resource you want to create.

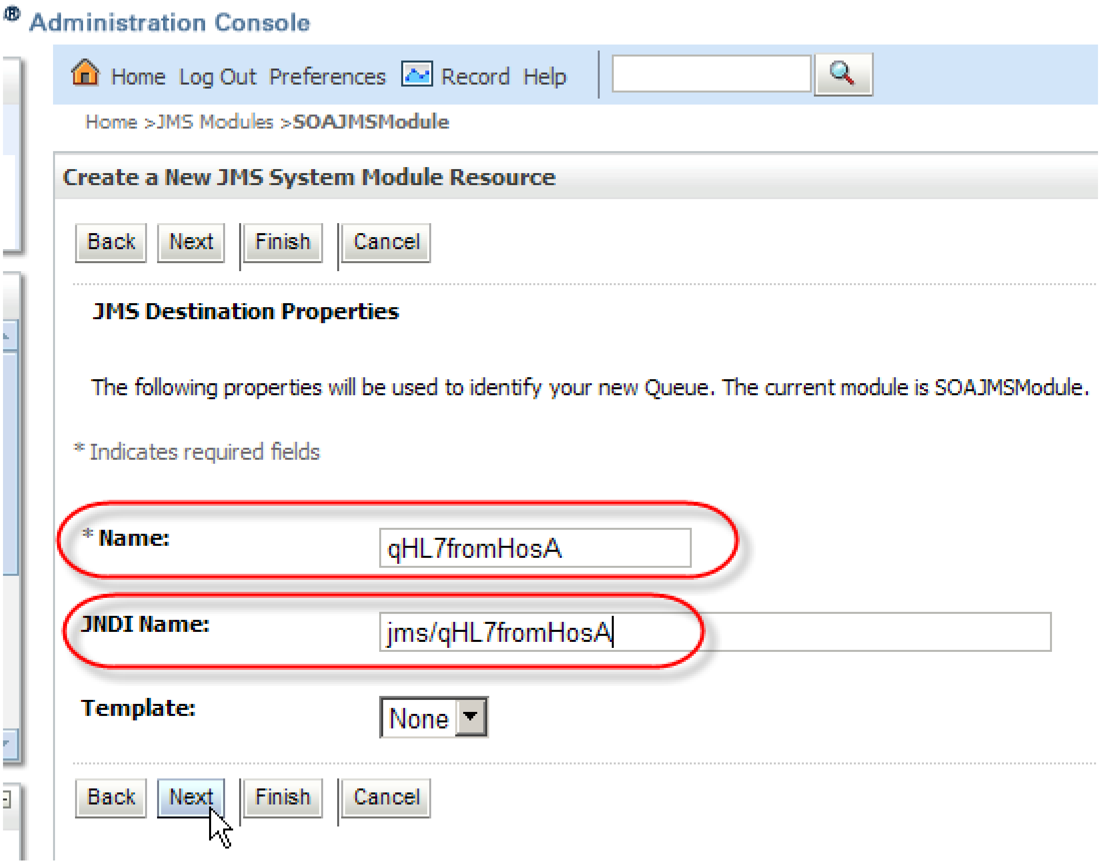
Use these pages to create resources in a JMS system module, such as queues, topics, templates, and connection fact

Depending on the type of resource you select, you are prompted to enter basic information for creating the resource. queues and topics, foreign servers, and JMS SAF destinations, you can also proceed to targeting pages for selecting an advanced mechanism for grouping JMS module resources and the members to server resources.

○ Connection Factory

● Queue

Set "qHL7fromHosA" for Name and "jms/qHL7fromHosA" for JNDI Name, then click Next.



Home >JMS Modules >**SOAJMSModule**

Create a New JMS System Module Resource

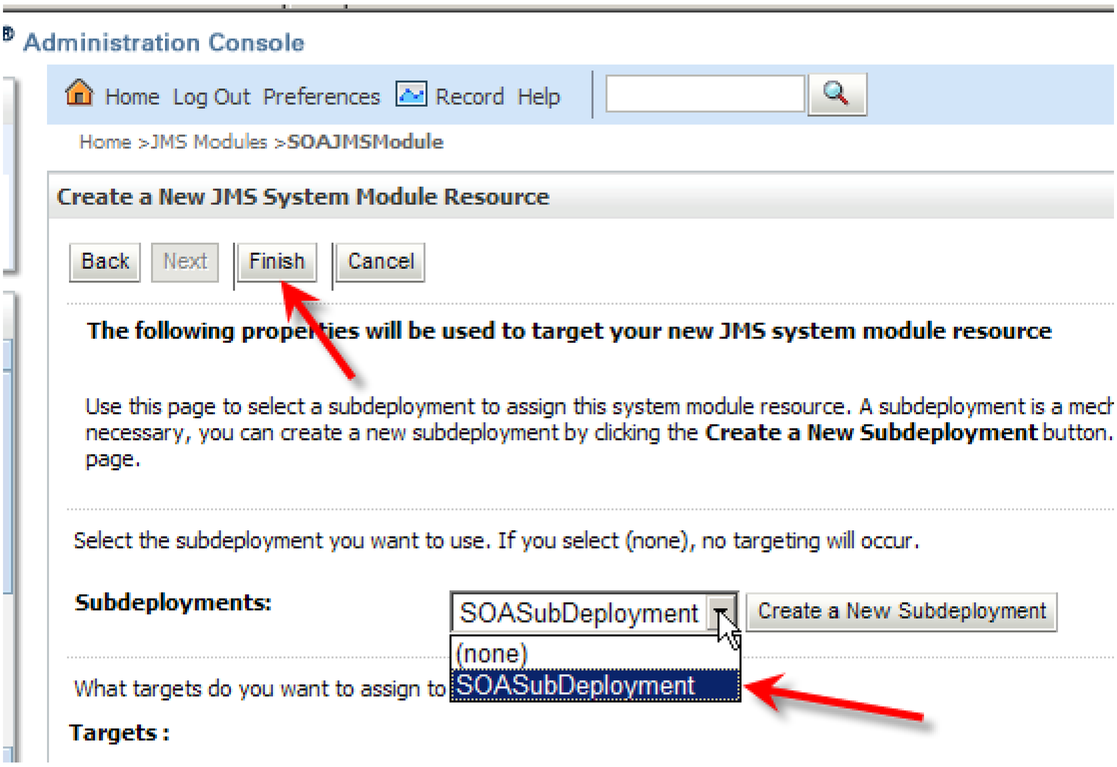
JMS Destination Properties

The following properties will be used to identify your new Queue. The current module is SOAJMSModule.

* Indicates required fields

* **Name:**	qHL7fromHosA
JNDI Name:	jms/qHL7fromHosA
Template:	None

Choose "SOASubDeployment" then click Finish.



Home >JMS Modules >**SOAJMSModule**

Create a New JMS System Module Resource

The following properties will be used to target your new JMS system module resource

Use this page to select a subdeployment to assign this system module resource. A subdeployment is a mech necessary, you can create a new subdeployment by clicking the **Create a New Subdeployment** button. page.

Select the subdeployment you want to use. If you select (none), no targeting will occur.

Subdeployments: SOASubDeployment

What targets do you want to assign to

Targets :

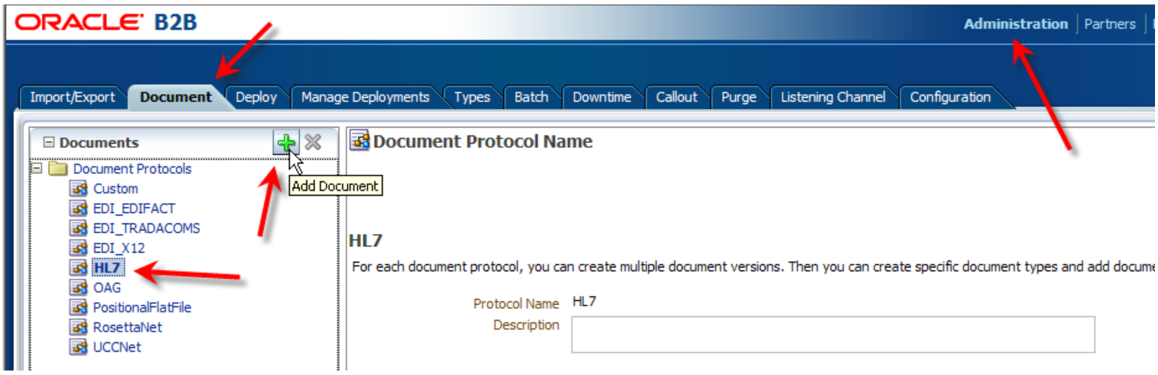
Repeat these steps to create the queue qHL7fromHosB with the JNDI Name of jms/qHL7fromHosB.

Use the QBrowser to verify the presence and accessibility of the two new queues.

Configure Inbound

Add HL7 Document

Start the B2B Web Console, http://localhost:7001/b2b, and click Administration link. Click Document tab, select HL7 node in the Document Protocols tree and click teh "Add Document" icon.



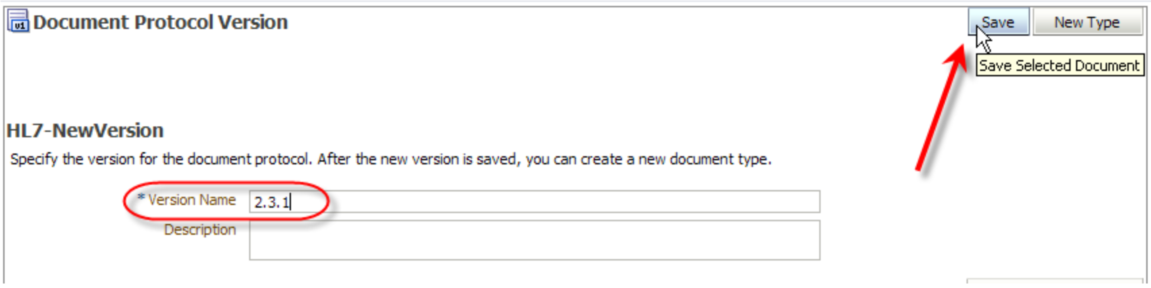
HL7

For each document protocol, you can create multiple document versions. Then you can create specific document types and add docume

Protocol Name HL7

Description

Enter "2.3.1" for version name and click "Save".



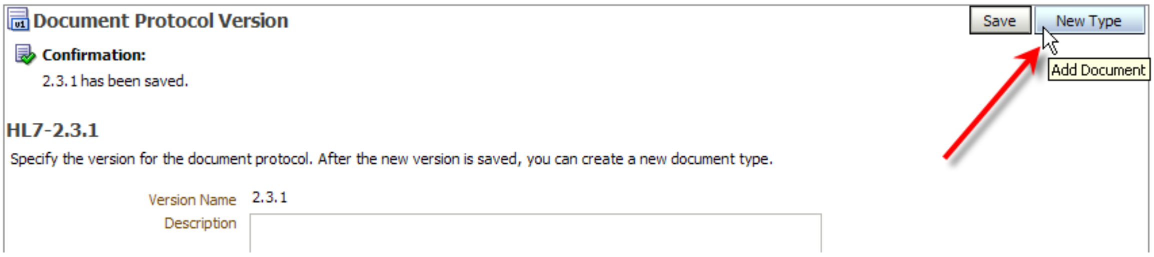
HL7-NewVersion

Specify the version for the document protocol. After the new version is saved, you can create a new document type.

* Version Name 2.3.1

Description

Click "New Type"



Confirmation:

2.3.1 has been saved.

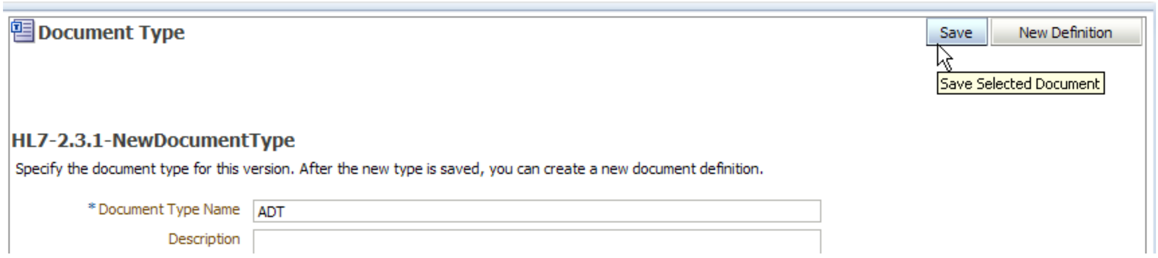
HL7-2.3.1

Specify the version for the document protocol. After the new version is saved, you can create a new document type.

Version Name 2.3.1

Description

Enter "ADT" as Document Type Name, click "Save" and click "New Definition".



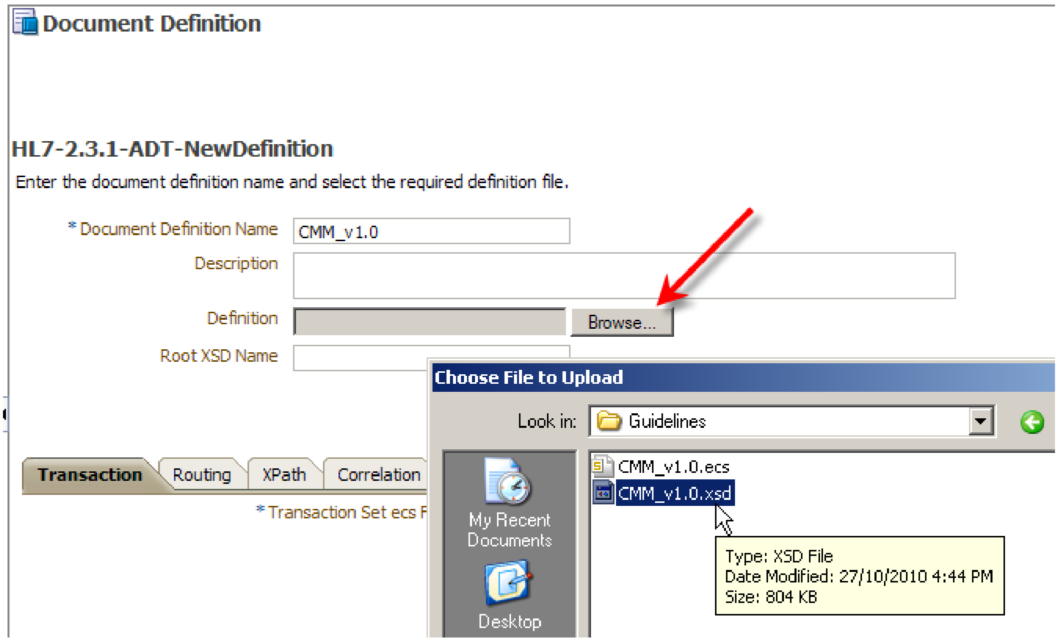
HL7-2.3.1-NewDocumentType

Specify the document type for this version. After the new type is saved, you can create a new document definition.

* Document Type Name: ADT

Description:

Enter "CMM_v1.0" for Document Definition Name, click Browse button next to the Definition box and select the CMM_v1.0.xsd definition file.



HL7-2.3.1-ADT-NewDefinition

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

* Document Definition Name: CMM_v1.0

Description:

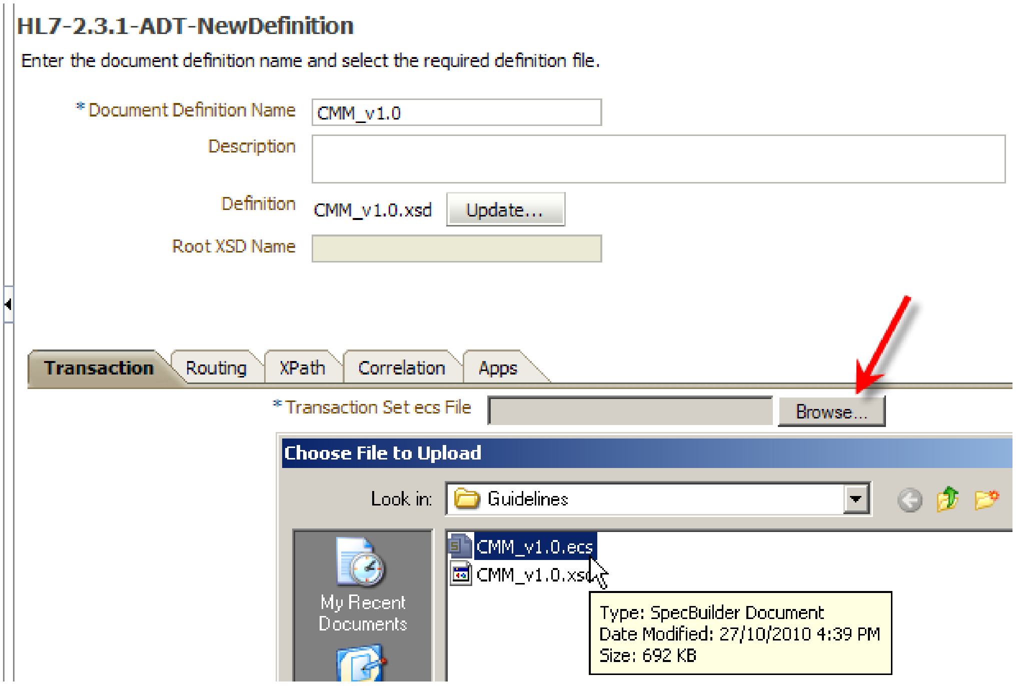
Definition:

Root XSD Name:

* Transaction Set ecs F

Type: XSD File
Date Modified: 27/10/2010 4:44 PM
Size: 804 KB

Click the Browse button next to the "Transaction Set ecs File" and choose the CMM_V1.0.ecs file.



HL7-2.3.1-ADT-NewDefinition

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

* Document Definition Name: CMM_v1.0

Description:

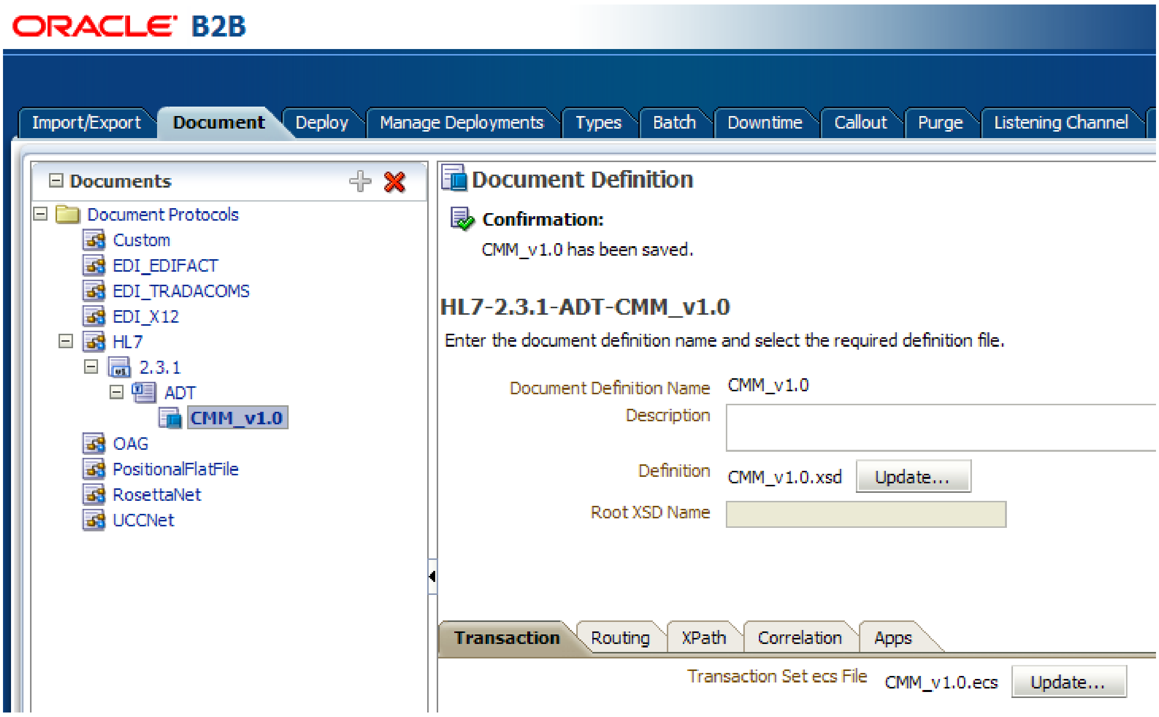
Definition: CMM_v1.0.xsd

Root XSD Name:

* Transaction Set ecs File

Type: SpecBuilder Document
Date Modified: 27/10/2010 4:39 PM
Size: 692 KB

Click the "Save" button to save changes. The document hierarchy should look like that shown below.



Document Definition

Confirmation:

CMM_v1.0 has been saved.

HL7-2.3.1-ADT-CMM_v1.0

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

Document Definition Name: CMM_v1.0

Description:

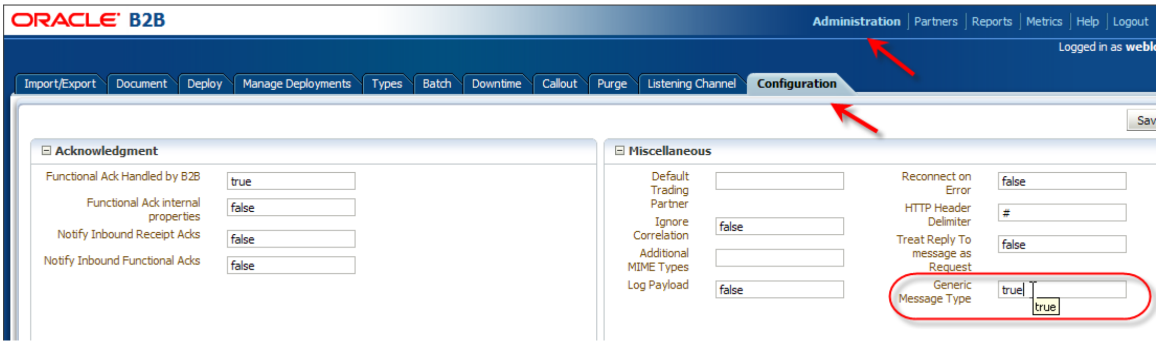
Definition: CMM_v1.0.xsd

Root XSD Name:

Transaction Set ecs File: CMM_v1.0.ecs

Configure generic document processing

Click the "Configuration" Tab, set "Generic Message Type" property to "true" and click "Save". This will allow the infrastructure to use the document definition which we configured previously to be used for all messages of type ADT.



Logged in as webl

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:

Log Payload: false

Reconnect on Error: false

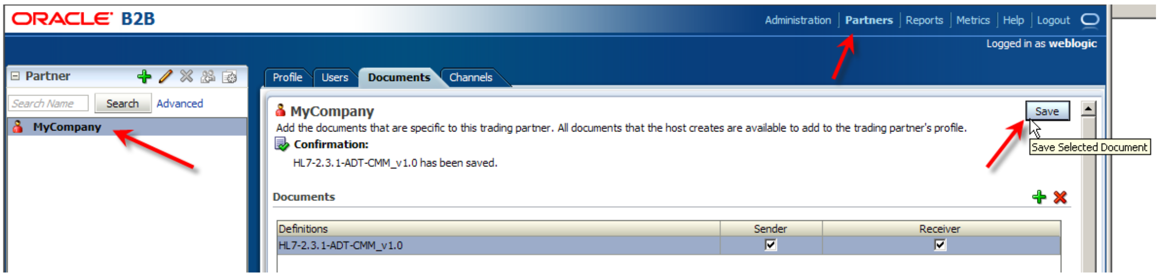
HTTP Header Delimiter: #

Treat Reply To message as Request: false

Generic Message Type: true

Configure "self" partner - MyCompany.

Click the "Partners" link, click the "MyCompany" link and click "Save". We will both receive and send documents of this type.



Logged in as weblogic

MyCompany

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.

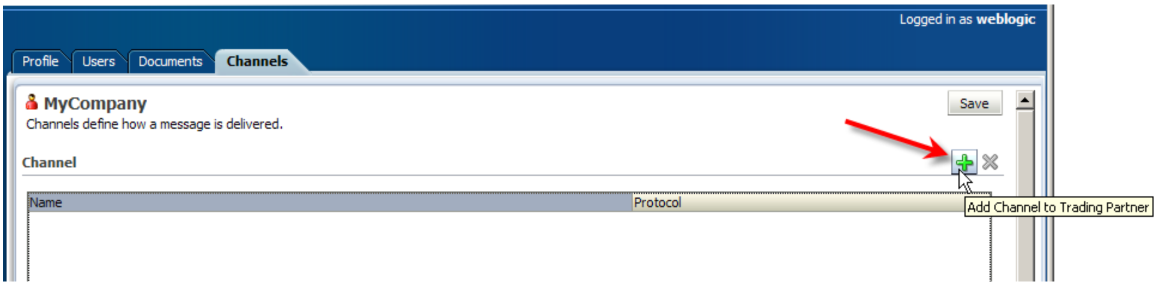
Confirmation:

HL7-2.3.1-ADT-CMM_v1.0 has been saved.

Documents

Definitions	Sender	Receiver
HL7-2.3.1-ADT-CMM_v1.0	☑	☑

Click the Channels Tab, then click the "Add Channel" button.



Logged in as **weblogic**

Profile | Users | Documents | **Channels**

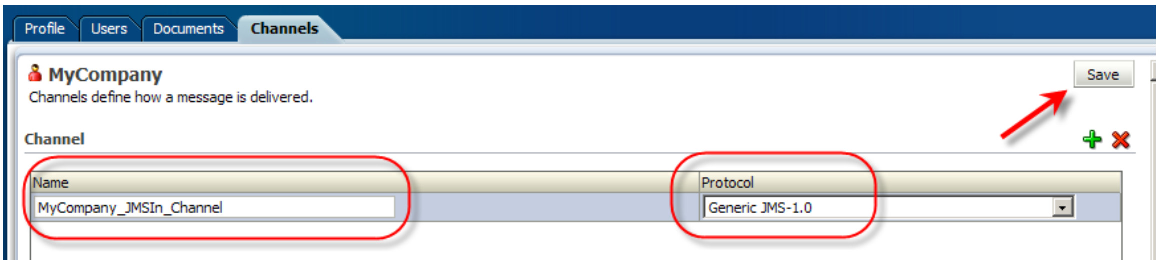
MyCompany

Channels define how a message is delivered.

Channel

Name	Protocol

Change channel name to MyCompany_JMSIn_Channel and choose "Generic JMS-1.0" protocol. Click "Save".



Profile | Users | Documents | **Channels**

MyCompany

Channels define how a message is delivered.

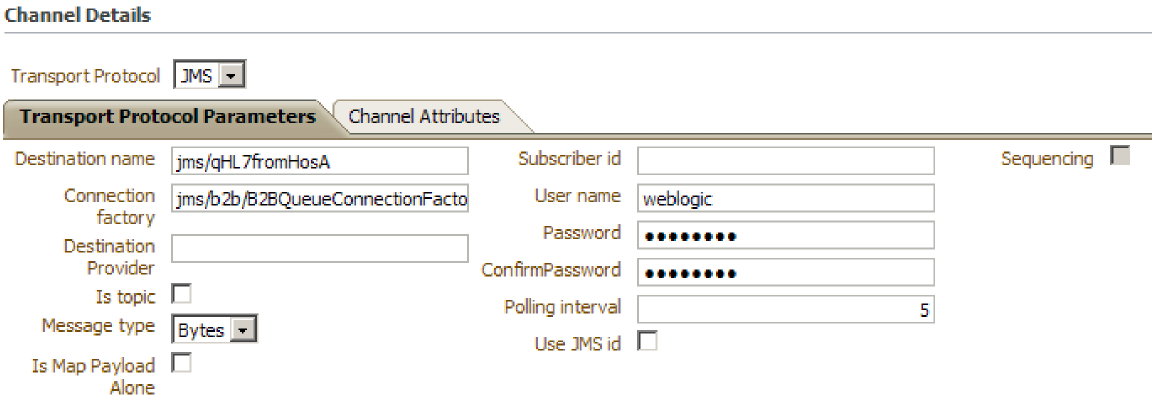
Channel

Name	Protocol
MyCompany_JMSIn_Channel	Generic JMS-1.0

Change "Transport Protocol Configuration" to set:

1. "Destination name" : "jms/qHL7fromHosA" - this is the JNDI name of the queue we created earlier
2. Set Username and Password as appropriate to your environment - they will be the same credentials as the ones you use to log into the WebLogic Admin Console.

Click "Save".



Channel Details

Transport Protocol: JMS

Transport Protocol Parameters | Channel Attributes

Destination name: jms/qHL7fromHosA

Connection factory: jms/b2b/B2BQueueConnectionFacto

Destination Provider:

Is topic: □

Message type: Bytes

Is Map Payload Alone: □

Subscriber id:

User name: weblogic

Password: ••••••••

ConfirmPassword: ••••••••

Polling interval: 5

Use JMS id: □

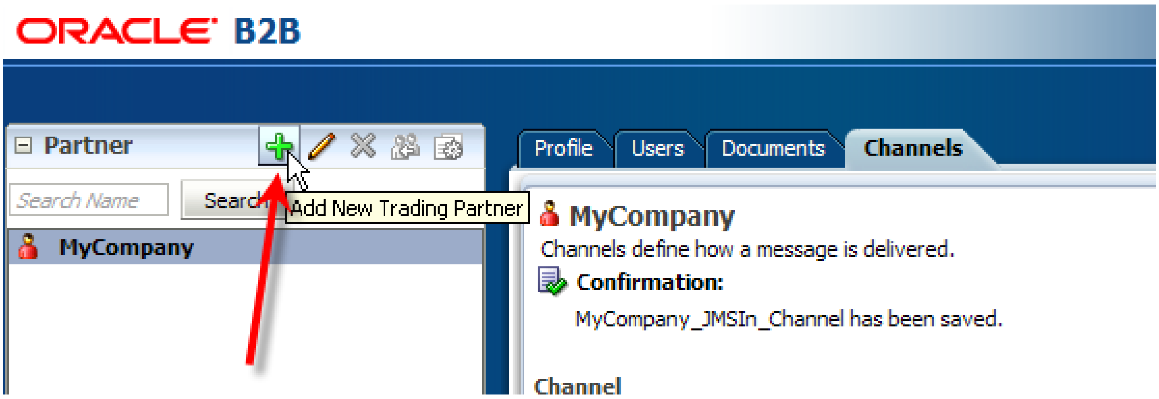
Sequencing: □

Click "Channel Attributes" tab. Ensure "Internal" checkbox is checked and click "Save".

This completes configuration of the "Self" trading partner for this article.

Configure first Inbound Partner - HosAIn

Click on the "Add New Trading Partner" button.



MyCompany

Channels define how a message is delivered.

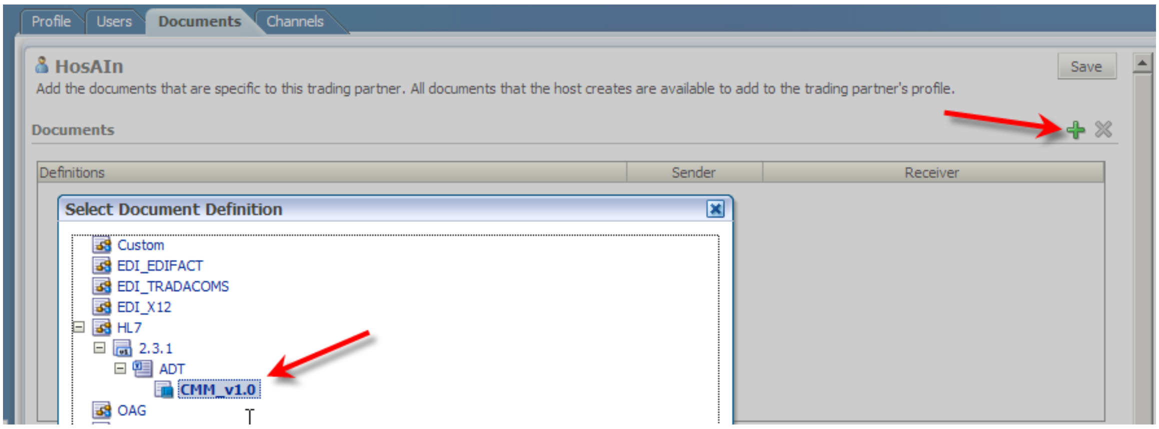
Confirmation:

MyCompany_JMSIn_Channel has been saved.

Channel

Name the trading partner HosAIn and click OK.

Click on the "Add Document Definition" button, select CMM_v1.0 document definition and click OK.

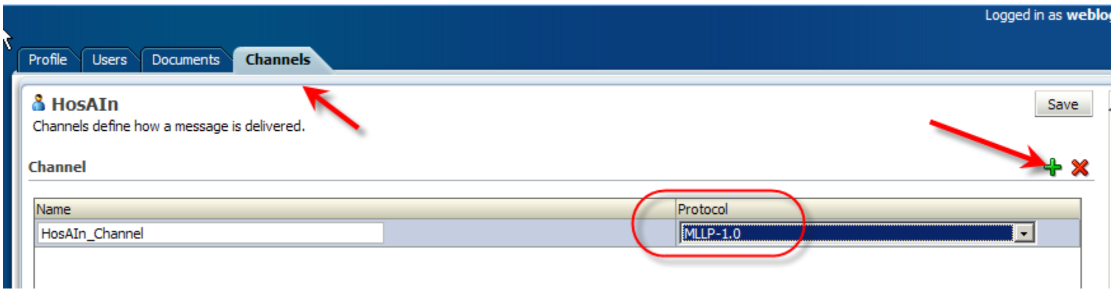


HosAIn

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.

Uncheck the "Receiver" checkbox and click "Save". This partner is the sender of these kinds of documents.

Click the "Channels" Tab. Click the "Add Channel to Trading Partner" button. Choose "MLLP 1.0" Protocol and click "Save".



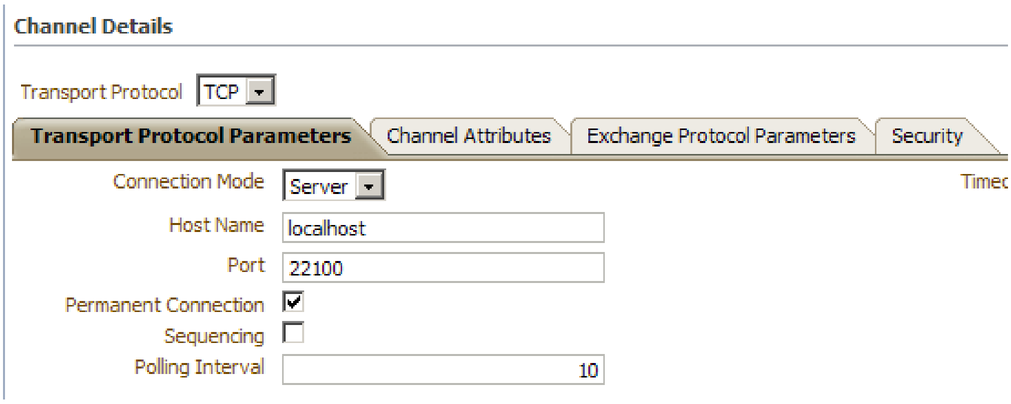
HosAIn

Channels define how a message is delivered.

Name	Protocol
HosAIn_Channel	MLLP-1.0

Configure "Channel Details".

Click "Transport Protocol Parameters" tab. Change "Connection Mode" to "Server". Enter "Host Name" as localhost. Enter suitable "Port", for example 22100. Check "Permanent Connection" check box. Click "Save".

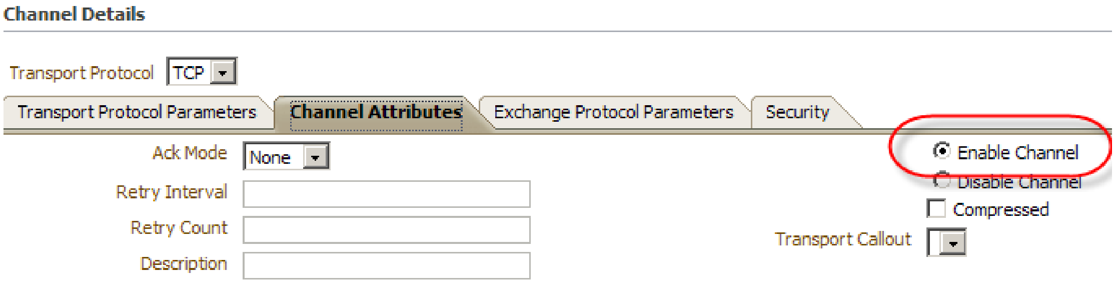


Channel Details

Transport Protocol: TCP

Connection Mode: Server
Host Name: localhost
Port: 22100
Permanent Connection: ☑
Sequencing: □
Polling Interval: 10

Click the "Channel Attributes" tab. Ensure "Enable Channel" is selected and click "Save".



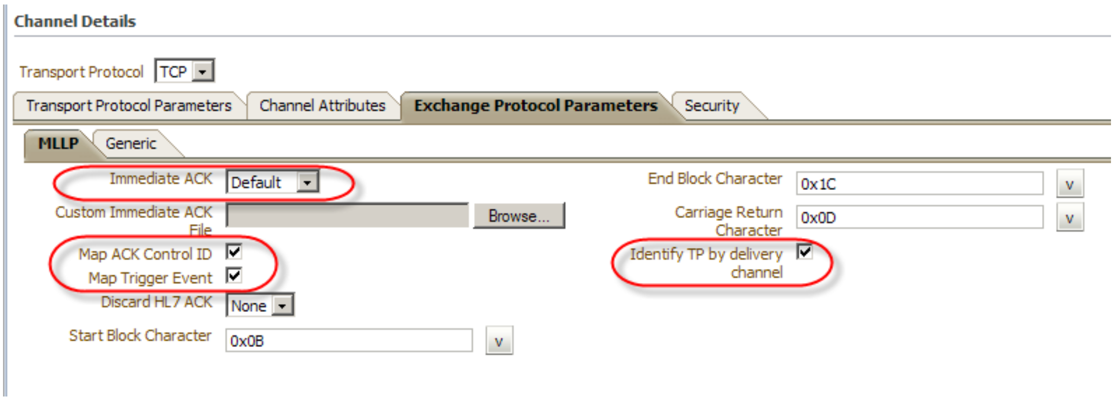
Channel Details

Transport Protocol: TCP

Ack Mode: None
Retry Interval:
Retry Count:
Description:

● Enable Channel
○ Disable Channel
□ Compressed
Transport Callout:

Click the "Exchange Protocol Parameters" tab. Choose "Default" for the "Immediate ACK". Check "Map ACK Control ID" and "Map Trigger Event" checkboxes. Check the "Identify TP by delivery channel;" checkbox. Click "Save".



Channel Details

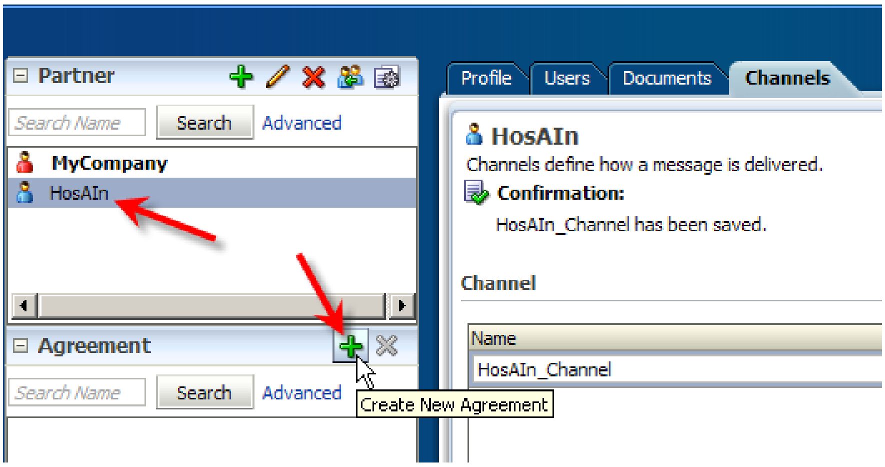
Transport Protocol: TCP

Immediate ACK: Default
Custom Immediate ACK File:
Map ACK Control ID: ☑
Map Trigger Event: ☑
Discard HL7 ACK: None
Start Block Character: 0x0B
End Block Character: 0x1C
Carriage Return Character: 0x0D
Identify TP by delivery channel: ☑

This completes configuration of the first inbound partner.

Configure first Inbound Trading Partnership Agreement

Make sure the HosIn partner is selected in the "Partner" panel and click the "Create New Agreement" button.



HosAIn

Channels define how a message is delivered.

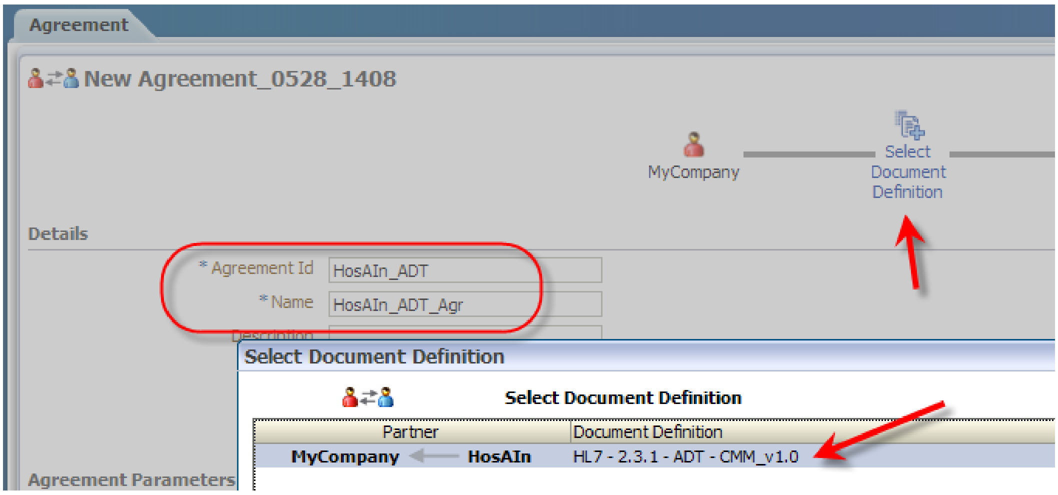
Confirmation:

HosAIn_Channel has been saved.

Channel

Name
HosAIn_Channel

Name the agreement "HosAIn_ADT_Agr" and set "Agreement Id" to "HosAIn_ADT". Click the "Select Document Definition" link and choose the one and only document definition. These are the kinds of documents that we will be expecting from this partner.



New Agreement_0528_1408

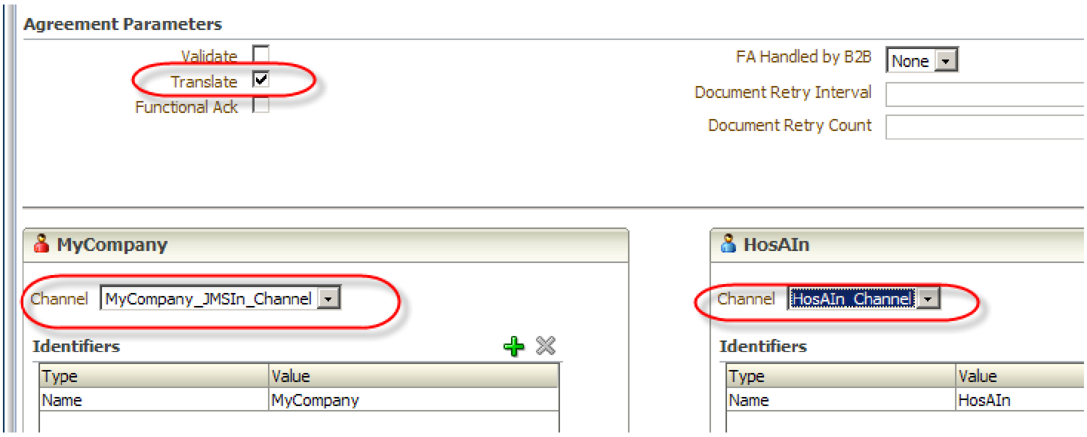
* Agreement Id	HosAIn_ADT
* Name	HosAIn_ADT_Agr
Description	

Select Document Definition

Partner		Document Definition
MyCompany ←	HosAIn	HL7 - 2.3.1 - ADT - CMM_v1.0

Leave "Translate" agreement parameter checked.

Choose "MyCompany_JMSIn_Channel" channel from the Channel dropdown at the MyCompany side. Choose "HosAIn_Channel" channel form the Channel dropdown at the HosAIn side. Click "Save" to save this agreement.


Agreement Parameters

Validate	□
Translate	☑
Functional Ack	□
FA Handled by B2B	None
Document Retry Interval	
Document Retry Count	

MyCompany

Channel: MyCompany_JMSIn_Channel

Identifiers

Type	Value
Name	MyCompany

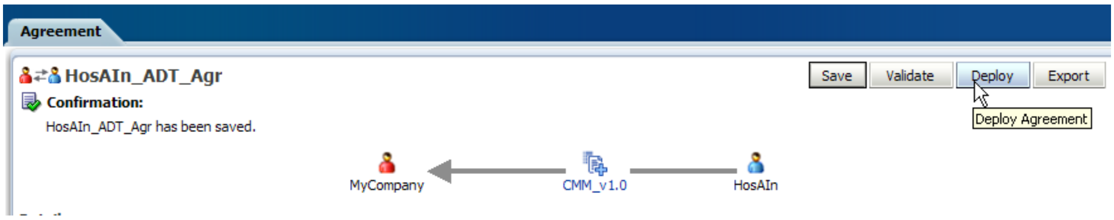
HosAIn

Channel: HosAIn_Channel

Identifiers

Type	Value
Name	HosAIn

Click "Deploy" to deploy this agreement".



HosAIn_ADT_Agr

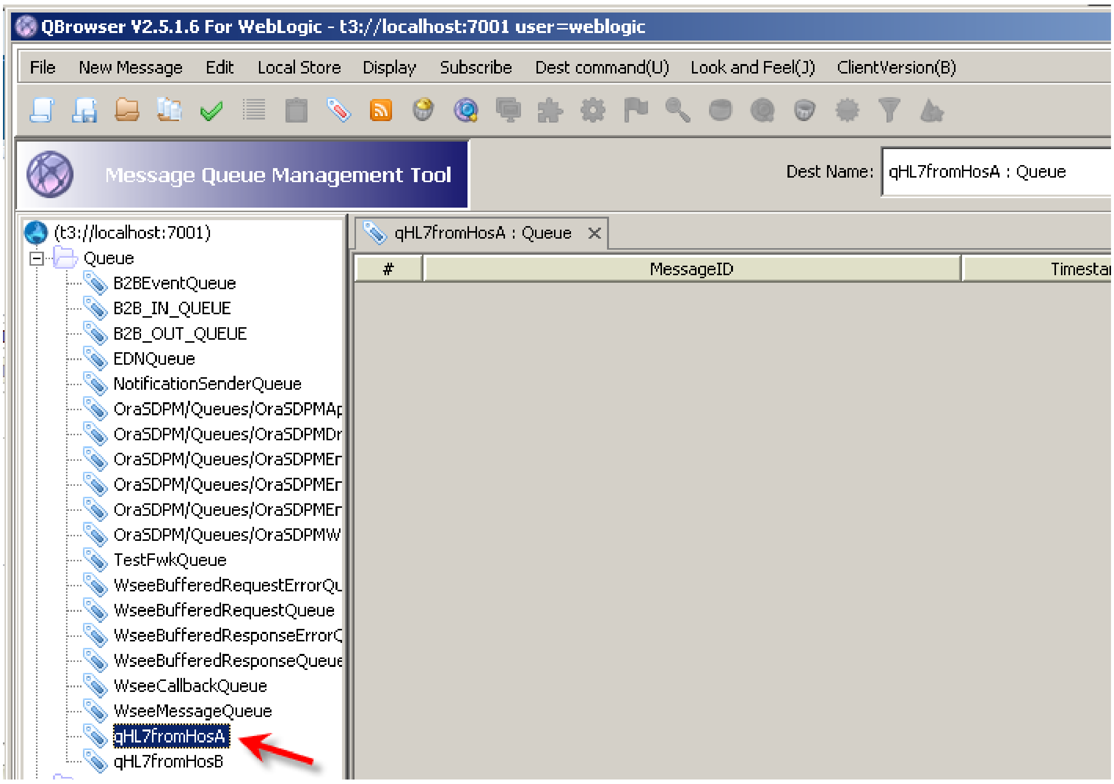
Confirmation:

HosAIn_ADT_Agr has been saved.

The partners and the agreement are now configured and the channel is ready to receive messages.

Send Test Messages from HosA

Use QBrowser to inspect qHL7fromHosA and see that there are no messages there.



Open a command window in the location where CMDHL7Sender_v0.7.jar is located and issue the following command, assuming Java 5 or Java 6 are in the path and that sample message file, ADT_A01_output_1.hl7, is located in C:\hl7\adt\data\sources. Adjust locations and names as needed. The command is a single line.

```
java -jar CMDHL7Sender_v0.7.jar -a FacA -b HosA -c ID -f
c:\hl7\adt\data\sources\ADT_A01_output_1.hl7 -h localhost -p 22100 -n 1
```

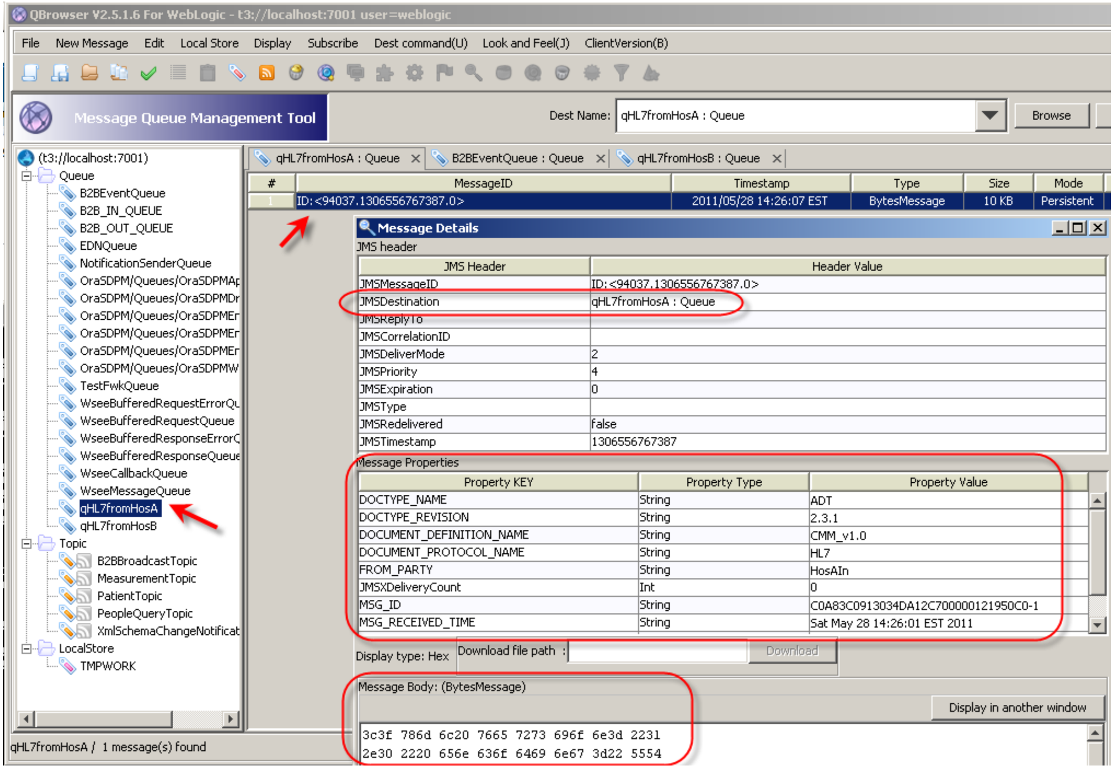
Inspect the output - mine looks like:

```
28/05/2011 2:20:27 PM au.id.czapski.hl7.CMDHL7Sender main
INFO: Host: localhost
28/05/2011 2:20:27 PM au.id.czapski.hl7.CMDHL7Sender main
INFO: Port: 22100
```

```
28/05/2011 2:20:27 PM au.id.czapski.hl7.CMDHL7Sender main
INFO: File Path: c:\hl7\adt\data\sources\ADT_A01_output_1.hl7
28/05/2011 2:20:27 PM au.id.czapski.hl7.CMDHL7Sender main
INFO: Delimiter Set:

28/05/2011 2:20:27 PM au.id.czapski.hl7.CMDHL7Sender main
INFO: Number of messages to send : 1
28/05/2011 2:20:27 PM ca.uhn.log.HapiLogImpl info
INFO: Instantiating msg of class ca.uhn.hl7v2.model.v231.message.ADT_A01
28/05/2011 2:20:27 PM au.id.czapski.hl7.CMDHL7Sender main
INFO: Read Message:
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^England - Norfolk^3
0828^UK||||||||A2008090801529
PV1|1|I||I|||FUL^Fulde^Gordian^^^^^^^^^^MAIN|||EMR|||||||||V2008090801529^^^^VISIT||||
|||||||||||||||||||||2008090801529

28/05/2011 2:20:27 PM au.id.czapski.hl7.CMDHL7Sender main
INFO: Sending Message:
MSH|^~\&|FacA|HosA|PI|MDM|2008090801529||ADT^A01|ID_0000000|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^England - Norfolk^3
0828^UK||||||||A2008090801529
PV1|1|I||I|||FUL^Fulde^Gordian^^^^^^^^^^MAIN|||EMR|||||||||V2008090801529^^^^VISIT||||
|||||||||||||||||||||2008090801529


28/05/2011 2:20:27 PM ca.uhn.log.HapiLogImpl info
PID|1||A000010^^^HosA^MR^HosA||Kessel^Abigail||19460101123045|M|||7 South 3rd
Circle^^Downham Market^England - Norfolk^3
PV1|1|I||I|||FUL^Fulde^Gordian^^^^^^^^^^MAIN|||EMR|||||||||V2008090801529^^^^VISIT||||
|||||||||||||||||||||2008090801529

28/05/2011 2:20:27 PM ca.uhn.log.HapiLogImpl info
PID|1||A000010^^^HosA^MR^HosA||Kessel^Abigail||19460101123045|M|||7 South 3rd
Circle^^Downham Market^England - Norfolk^3
PV1|1|I||I|||FUL^Fulde^Gordian^^^^^^^^^^MAIN|||EMR|||||||||V2008090801529^^^^VISIT||||
|||||||||||||||||||||2008090801529

28/05/2011 2:20:27 PM ca.uhn.log.HapiLogImpl info
MSA|AA|ID_0000000|MSG Received
Successfully|I|MDM|FacA|HosA|20110528142027||ACK^A01|ID_0000000|P|2.3.1|
28/05/2011 2:20:27 PM ca.uhn.log.HapiLogImpl info
MSA|AA|ID_0000000|MSG Received Successfully|27||ACK^A01|ID_0000000|P|2.3.1|
28/05/2011 2:20:27 PM ca.uhn.log.HapiLogImpl info
INFO: Instantiating msg of class ca.uhn.hl7v2.model.v231.message.ACK
28/05/2011 2:20:27 PM au.id.czapski.hl7.CMDHL7Sender main
INFO: Received response:
MSH|^~\&|PI|MDM|FacA|HosA|20110528142027||ACK^A01|ID_0000000|P|2.3.1
MSA|AA|ID_0000000|MSG Received Successfully


28/05/2011 2:20:27 PM ca.uhn.log.HapiLogImpl info
INFO: SocketException on read() attempt.  Socket appears to have been closed: socket
closed
28/05/2011 2:20:27 PM ca.uhn.log.HapiLogImpl info
INFO: Closing connection (no more messages available).
```

Using QBrowser inspect the queue qHL7fromHosA.

Note JMS message in the queue. Double-click the message to open its properties window.



Dest Name: qHL7fromHosA : Queue

#	MessageID	Timestamp	Type	Size	Mode
1	ID:<94037.1306556767387.0>	2011/05/28 14:26:07 EST	BytesMessage	10 KB	Persistent

JMS header

JMS Header	Header Value
JMSMessageID	ID:<94037.1306556767387.0>
JMSDestination	qHL7fromHosA : Queue
JMSReplyTo	
JMSCorrelationID	
JMSDeliverMode	2
JMSPriority	4
JMSExpiration	0
JMSType	
JMSRedelivered	false
JMSTimestamp	1306556767387

Message Properties

Property KEY	Property Type	Property Value
DOCTYPE_NAME	String	ADT
DOCTYPE_REVISION	String	2.3.1
DOCUMENT_DEFINITION_NAME	String	CMM_v1.0
DOCUMENT_PROTOCOL_NAME	String	HL7
FROM_PARTY	String	HosAIn
JMSXDeliveryCount	Int	0
MSG_ID	String	C0A83C0913034DA12C700000121950C0-1
MSG_RECEIVED_TIME	String	Sat May 28 14:26:01 EST 2011

Display type: Hex

Download file path :

Message Body: (BytesMessage)

```
3c3f 786d 6c20 7665 7273 696f 6e3d 2231
2e30 2220 656e 636f 6469 6e67 3d22 5554
```

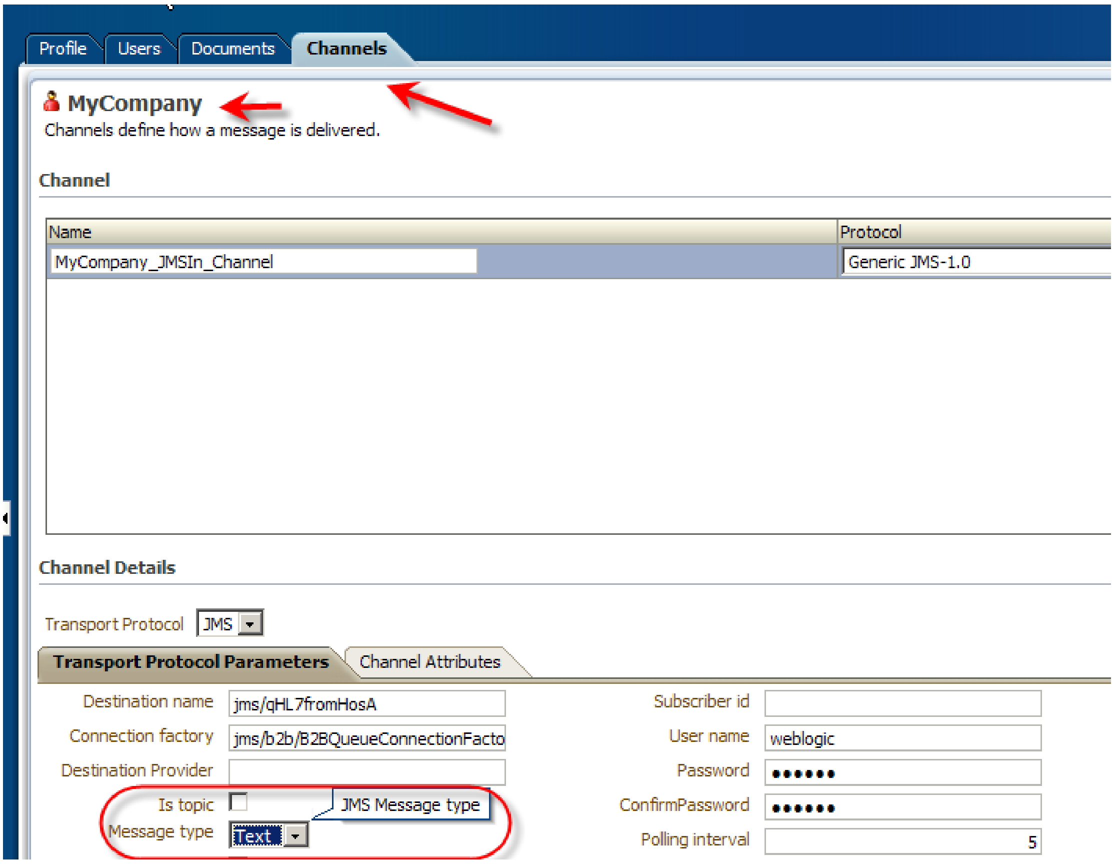
qHL7fromHosA / 1 message(s) found

Note JMSDestintion name - qHL7fromHosA.

Note B2B properties, conveyed as JMS User-defined properties. These can be used in a downstream component to process the message based on the type, revision, source, etc..

Note that we chose ByteMessage as message type. This makes the body of the message unreadable in the QBrowser.

Let's modify the internal channel configuration of the MyCmpany partner to use a TextMessage type instead and save.



MyCompany

Channels define how a message is delivered.

Channel

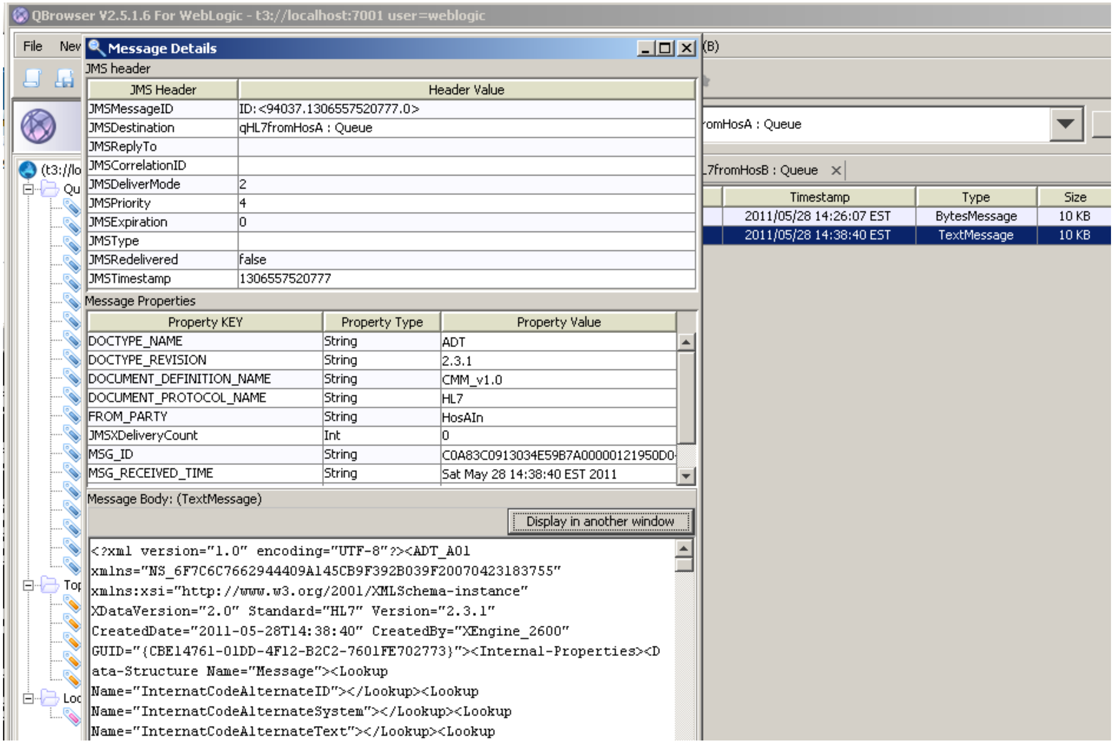
Name	Protocol
MyCompany_JMSIn_Channel	Generic JMS-1.0

Channel Details

Transport Protocol: JMS

Destination name	jms/qHL7fromHosA	Subscriber id	
Connection factory	jms/b2b/B2BQueueConnectionFacto	User name	weblogic
Destination Provider		Password	••••••
Is topic	□	ConfirmPassword	••••••
Message type	Text	Polling interval	5

Let's save and deploy the HosAIn_ADT_Agr partnership agreement, submit a new message and see the message body in the QBrowser.


QBrowser V2.5.1.6 For WebLogic - t3://localhost:7001 user=weblogic

Message Details

JMS header

JMS Header	Header Value
JMSMessageID	ID:<94037.1306557520777.0>
JMSDestination	qHL7fromHosA : Queue
JMSReplyTo	
JMSCorrelationID	
JMSDeliverMode	2
JMSPriority	4
JMSExpiration	0
JMSType	
JMSRedelivered	false
JMSTimestamp	1306557520777

Message Properties

Property KEY	Property Type	Property Value
DOCTYPE_NAME	String	ADT
DOCTYPE_REVISION	String	2.3.1
DOCUMENT_DEFINITION_NAME	String	CMM_v1.0
DOCUMENT_PROTOCOL_NAME	String	HL7
FROM_PARTY	String	HosAIn
JMSXDeliveryCount	Int	0
MSG_ID	String	C0A83C0913034E59B7A00000121950D0
MSG_RECEIVED_TIME	String	Sat May 28 14:38:40 EST 2011

Message Body: (TextMessage)

```
<?xml version="1.0" encoding="UTF-8"?><ADT_A01
xmlns="NS_6F7C6C7662944409A145CB9F392B039F20070423183755"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
XDataVersion="2.0" Standard="HL7" Version="2.3.1"
CreatedDate="2011-05-28T14:38:40" CreatedBy="XEngine_2600"
GUID="{CBE14761-01DD-4F12-B2C2-7601FE702773}"><Internal-Properties><D
ata-Structure Name="Message"><Lookup
Name="InternatCodeAlternateID"></Lookup><Lookup
Name="InternatCodeAlternateSystem"></Lookup><Lookup
Name="InternatCodeAlternateText"></Lookup><Lookup
```

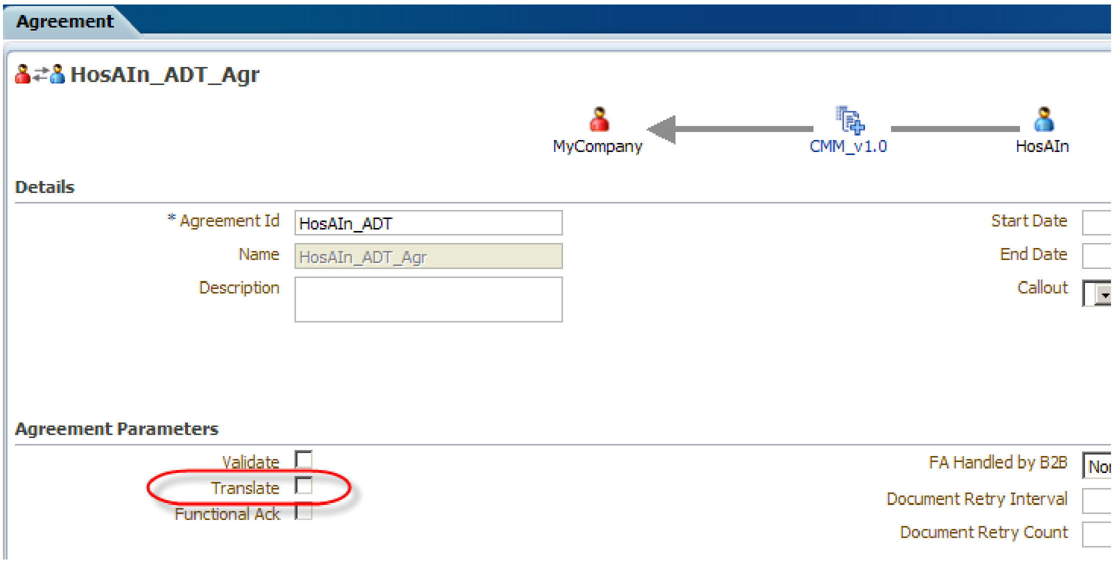
romHosA : Queue

7fromHosB : Queue

Timestamp	Type	Size
2011/05/28 14:26:07 EST	BytesMessage	10 KB
2011/05/28 14:38:40 EST	TextMessage	10 KB

This time we clearly see XML instance document. As configured, the trading partnership agreement "Translate" property causes the message to be translated to XML.

Let's uncheck the "Translate" checkbox, save and deploy the agreement.



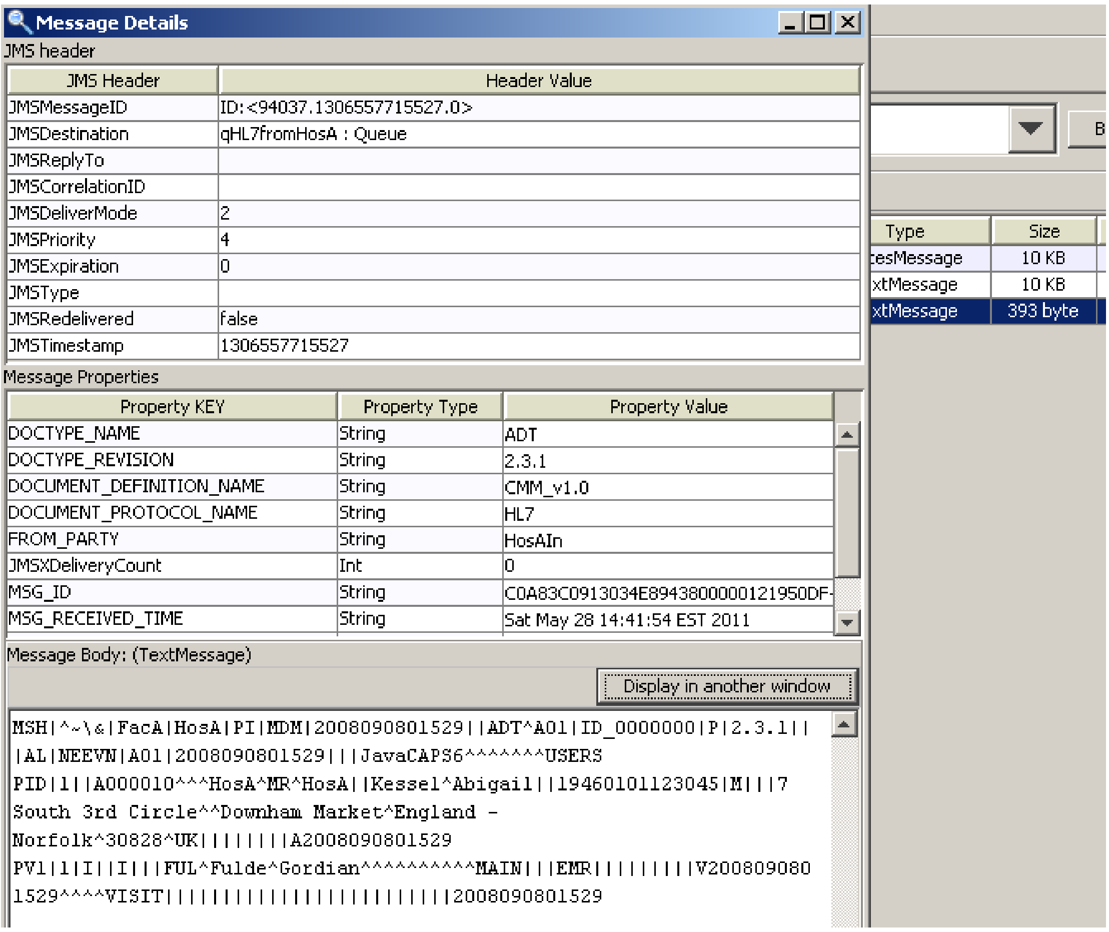
Details

* Agreement Id	HosAIn_ADT	Start Date	
Name	HosAIn_ADT_Agr	End Date	
Description		Callout	

Agreement Parameters

Validate	□	FA Handled by B2B	Nor
Translate	□	Document Retry Interval	
Functional Ack	□	Document Retry Count	

Let's send another message and inspect the message body using the QBrowser.



JMS header

JMS Header	Header Value
JMSMessageID	ID:<94037.1306557715527.0>
JMSDestination	qHL7fromHosA : Queue
JMSReplyTo	
JMSCorrelationID	
JMSDeliverMode	2
JMSPriority	4
JMSExpiration	0
JMSType	
JMSRedelivered	false
JMSTimestamp	1306557715527

Message Properties

Property KEY	Property Type	Property Value
DOCTYPE_NAME	String	ADT
DOCTYPE_REVISION	String	2.3.1
DOCUMENT_DEFINITION_NAME	String	CMM_v1.0
DOCUMENT_PROTOCOL_NAME	String	HL7
FROM_PARTY	String	HosAIn
JMSXDeliveryCount	Int	0
MSG_ID	String	C0A83C0913034E8943800000121950DF-
MSG_RECEIVED_TIME	String	Sat May 28 14:41:54 EST 2011

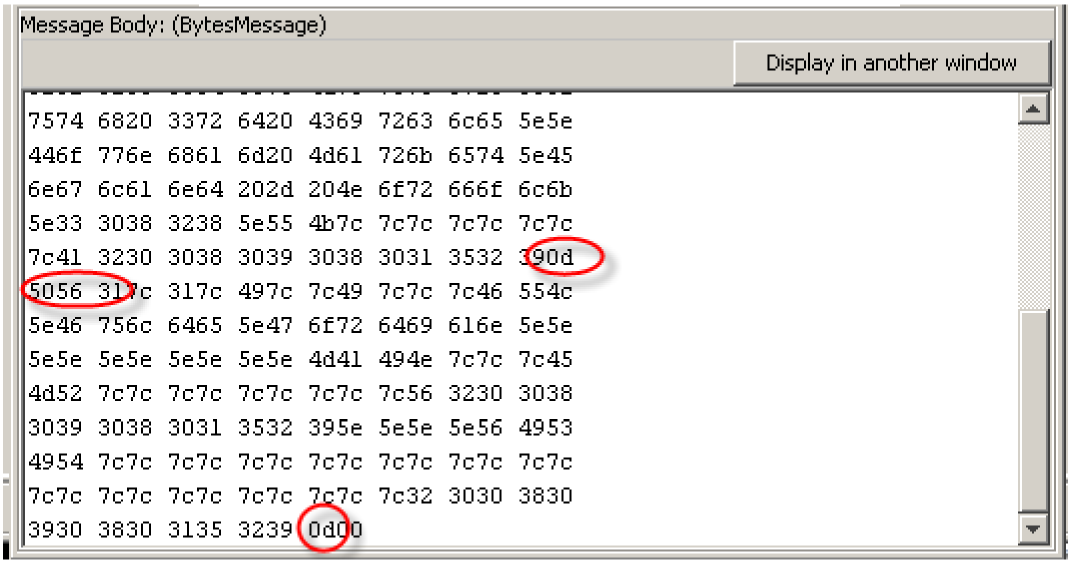
Type	Size
:esMessage	10 KB
xtMessage	10 KB
xtMessage	393 byte

Message Body: (TextMessage)

```
MSH|^~\&|FacA|HosA|PI|MDM|2008090801529||ADT^A01|ID_0000000|P|2.3.1||
|AL|NEEVN|A01|2008090801529|||JavaCAPS6^^^^^^^USERS
PID|1||A000010^^^HosA^MR^HosA||Kessel^Abigail||19460101123045|M|||7
South 3rd Circle^^Downham Market^England -
Norfolk^30828^UK|||||||A2008090801529
PV1|1|I||I|||FUL^Fulde^Gordian^^^^^^^^^^MAIN|||EMR||||||||V200809080
1529^^^^VISIT|||||||||||||||||||||||||2008090801529
```

The HL7 v2 delimited message has been received and deposited in the JMS queue of or choosing.

Now change the internal channel at the MyCompany back to Bytes, save, save and deploy the trading partnership agreement and submit another message.



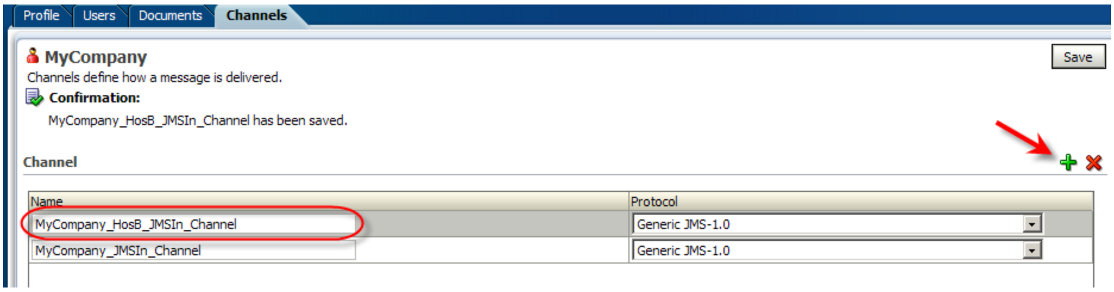
```
7574 6820 3372 6420 4369 7263 6c65 5e5e
446f 776e 6861 6d20 4d61 726b 6574 5e45
6e67 6c61 6e64 202d 204e 6f72 666f 6c6b
5e33 3038 3238 5e55 4b7c 7c7c 7c7c 7c7c
7c41 3230 3038 3039 3038 3031 3532 390d
5056 317c 317c 497c 7c49 7c7c 7c46 554c
5e46 756c 6465 5e47 6f72 6469 616e 5e5e
5e5e 5e5e 5e5e 5e5e 4d41 494e 7c7c 7c45
4d52 7c7c 7c7c 7c7c 7c7c 7c56 3230 3038
3039 3038 3031 3532 395e 5e5e 5e56 4953
4954 7c7c 7c7c 7c7c 7c7c 7c7c 7c7c 7c7c
7c7c 7c7c 7c7c 7c7c 7c7c 7c32 3030 3830
3930 3830 3135 3239 0d00
```

Note that Carriage Returns, critical to HL7, are preserved (0d just before the PV1 segment and at the end of the message). Others are preserved as well. The message was received intact, with no translation to XML.

Configure and exercise HosB Inbound

To receive messages from HosA into a separate queue we need a separate internal channel for the MyCompany partner.

Click the Channels Tab, then click the "Add Channel" button. Change channel name to MyCompany_HosB_JMSIn_Channel and choose "Generic JMS-1.0" protocol. Click "Save".



MyCompany
Channels define how a message is delivered.
Confirmation:
MyCompany_HosB_JMSIn_Channel has been saved.

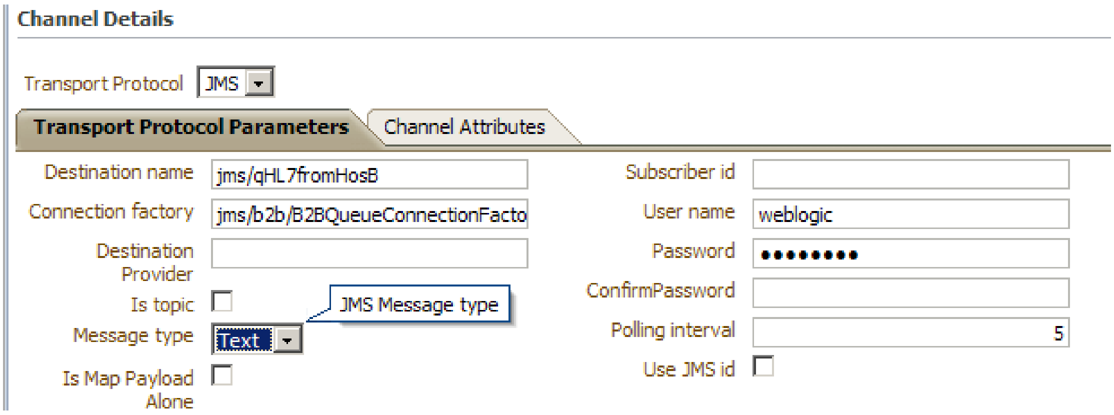
Channel

Name	Protocol
MyCompany_HosB_JMSIn_Channel	Generic JMS-1.0
MyCompany_JMSIn_Channel	Generic JMS-1.0

Change "Transport Protocol Configuration" to set:

1. "Destination name" : "jms/qHL7fromHosB" - this is the JNDI name of the queue we created earlier
2. Set Username and Password as appropriate to your environment - they will be the same credentials as the ones you use to log into the WebLogic Admin Console.

Click "Save".



Channel Details

Transport Protocol: JMS

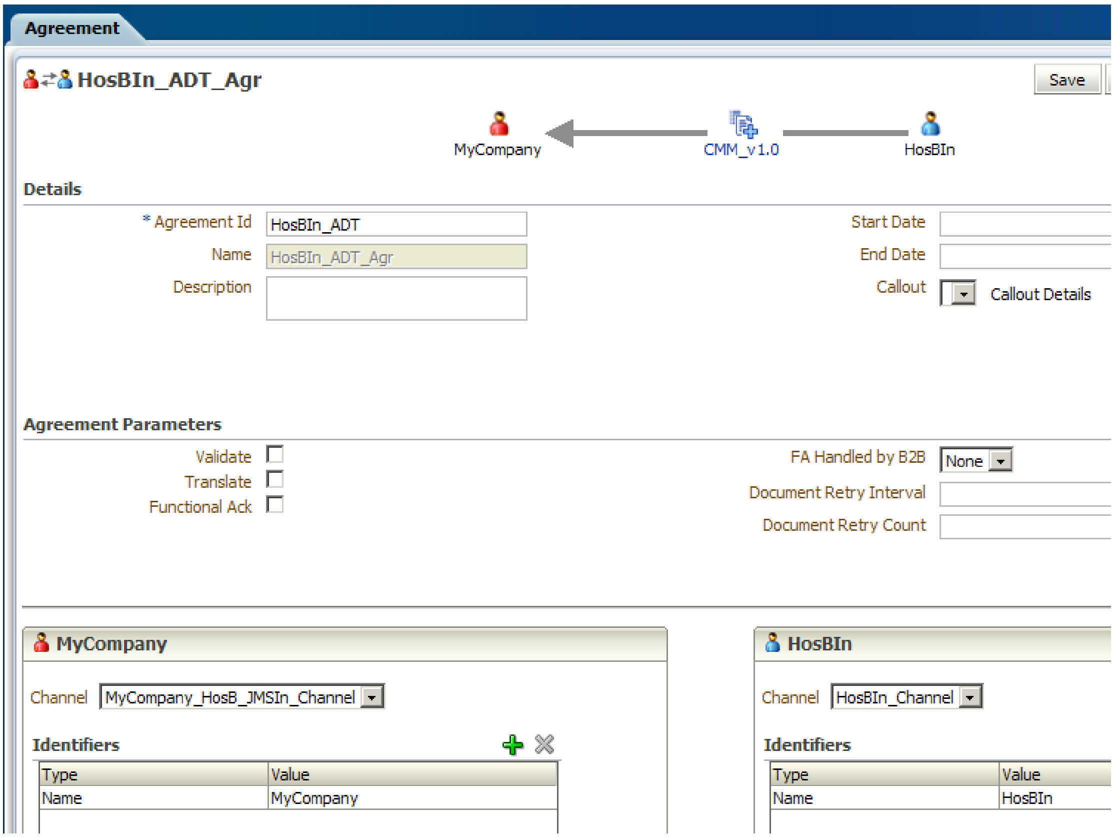
Transport Protocol Parameters | Channel Attributes

Destination name: jms/qHL7fromHosB
Connection factory: jms/b2b/B2BQueueConnectionFacto
Destination Provider:
Is topic: □
Message type: Text
Is Map Payload Alone: □

Subscriber id:
User name: weblogic
Password: ••••••••
ConfirmPassword:
Polling interval: 5
Use JMS id: □

Click "Channel Attributes" tab. Ensure "Internal" checkbox is checked and click "Save".

Now repeat the configuration and testing process discussed in sections "Configure first Inbound Partner - HosAIn", "Configure first Inbound Trading Partnership Agreement HosAIn_ADT", and "Send a Test Message from HosA", but use HosB instead of HosA and port 22200 instead of 22100 wherever they occur in the instructions and literals.



Agreement

HosBIn_ADT_Agr

Details

* Agreement Id: HosBIn_ADT
Name: HosBIn_ADT_Agr
Description:

Start Date:
End Date:
Callout: Callout Details

Agreement Parameters

Validate: □
Translate: □
Functional Ack: □

FA Handled by B2B: None
Document Retry Interval:
Document Retry Count:

MyCompany

Channel: MyCompany_HosB_JMSIn_Channel

Identifiers

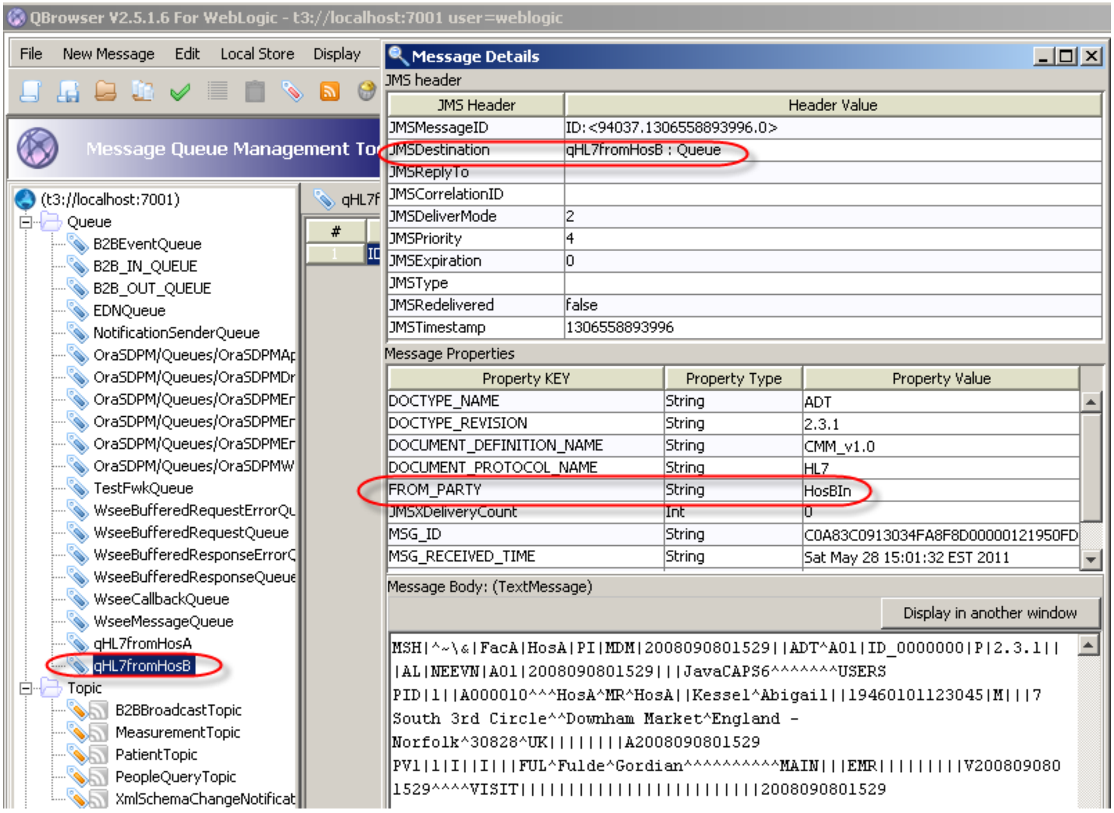
Type	Value
Name	MyCompany

HosBIn

Channel: HosBIn_Channel

Identifiers

Type	Value
Name	HosBIn



JMS header

JMS Header	Header Value
JMSMessageID	ID:<94037.1306558893996.0>
JMSDestination	qHL7fromHosB : Queue
JMSReplyTo	
JMSCorrelationID	
JMSDeliverMode	2
JMSPriority	4
JMSExpiration	0
JMSType	
JMSRedelivered	false
JMSTimestamp	1306558893996

Message Properties

Property KEY	Property Type	Property Value
DOCTYPE_NAME	String	ADT
DOCTYPE_REVISION	String	2.3.1
DOCUMENT_DEFINITION_NAME	String	CMM_v1.0
DOCUMENT_PROTOCOL_NAME	String	HL7
FROM_PARTY	String	HosBIn
JMSXDeliveryCount	Int	0
MSG_ID	String	C0A83C0913034FA8F8D00000121950FD
MSG_RECEIVED_TIME	String	Sat May 28 15:01:32 EST 2011

Message Body: (TextMessage)

```
MSH|^~\&|FacA|HosA|PI|MDM|2008090801529||ADT^A01|ID_0000000|P|2.3.1||
|AL|NEEVN|A01|2008090801529|||JavaCAPS6^^^^^^^USERS
PID|1||A000010^^^HosA^MR^HosA||Kessel^Abigail||19460101123045|M|||7
South 3rd Circle^^Downham Market^England -
Norfolk^30828^UK||||||||A2008090801529
PV1|1|I||I|||FUL^Fulde^Gordian^^^^^^^^^^MAIN|||EMR|||||||||V200809080
1529^^^^VISIT||||||||||||||||||||||||||2008090801529
```

Summary

In this article we reproduced the pattern "HL7v2Adapter→ JMS Queue", typical of eGate and Java CAPS solutions. The article walked through the process of implementing this pattern using Oracle SOA Suite 11g R1 PS3.

We demonstrated that Oracle SOA Suite B2B HL7 infrastructure can be configured to receive message streams over multiple inbound MLLP channels and deliver each stream to a distinct JMS destination, much as eGate and Java CAPS solutions used to do.