Oracle SOA Suite 11g

Oracle SOA Suite 11g HL7 Outbound Example

michael@czapski.id.au

June 2010

Table of Contents

Introduction

In this article I develop and exercise a simple Oracle SOA Suite 11g B2B infrastructure-based HL7 v2 Sender project for an ADT A01 message and use Message tracker to view messages, message states and messaging performance.

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 and Oracle SOA Suite 11g HL7 Inbound Example – Functional ACK Addendum are built and deployed. Strictly speaking it is not necessary to have the solutions deployed. All that is necessary is for the ADT A01 and A01 ACK documents, exported from the B2B Document Editor while building these solutions, to be available for use in work done which implementing the solution discussed in this article.

It is assumed that the data in the archive, HL7_messages_sources.zip, has been unpacked to C:\hl7\adt\data\. 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 the author's page at http://mac.softpedia.com/developer/Michael-Litherland-5914.html, is available. We will use the tool later to look at and receive HL7 messages.

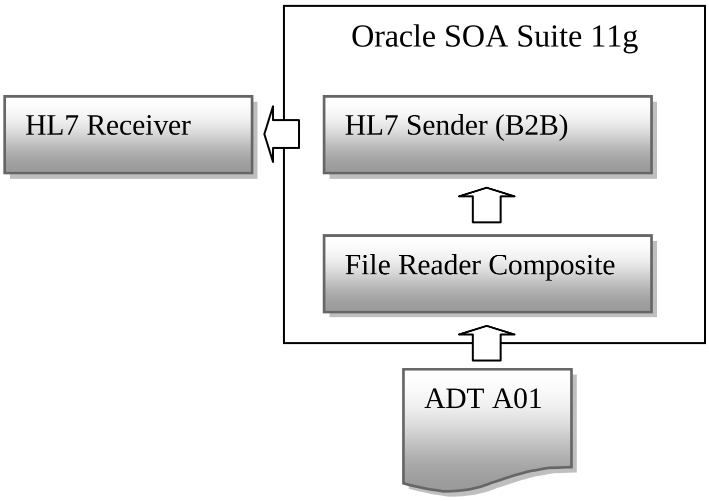
Download HL7 v2 XML Schemas from http://wiki.open-esb.java.net/attach/HL7BC/HL7v2XSDs.zip and unpack them to a convenient directory, for example c:\hl7. We will use some of them later.

HL7 v2 Sender Solution

Oracle uses the SOA Suite B2B component to provide HL7 v2 messaging support. It uses HL7 v2 message libraries and the B2B engine to provide message parsing and transformation between the native format (HL7 v2 delimited) and XML, which is used internally by the SOAP Suite. Oracle B2b User's Guide can be found at http://download.oracle.com/docs/cd/E15523_01/integration.1111/e10229.pdf.

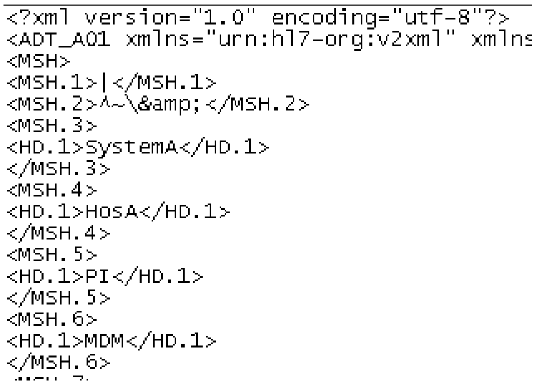
The solution we will be building is a HL7 Sender, which will send v2 ADT A01 messages, read from files in the file system, and will expect ACK messages in return.

The solution will consists of a SOA Composite which will read HL7 v2 XML messages and transform them into XML ADT messages required by the B2B infrastructure which will send HL7 v2 ADT A01 messages to the remote partner.



Messages in the sample message set use the following identifiers:

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



```
<?xml version="1.0" encoding="utf-8"?>
<ADT_A01 xmlns="urn:hl7-org:v2xml" xmlns
<MSH>
<MSH.1>|</MSH.1>
<MSH.2>^~\&amp;</MSH.2>
<MSH.3>
<HD.1>SystemA</HD.1>
</MSH.3>
<MSH.4>
<HD.1>HosA</HD.1>
</MSH.4>
<MSH.5>
<HD.1>PI</HD.1>
</MSH.5>
<MSH.6>
<HD.1>MDM</HD.1>
</MSH.6>
```

Develop HL7 v2 Outbound Solution

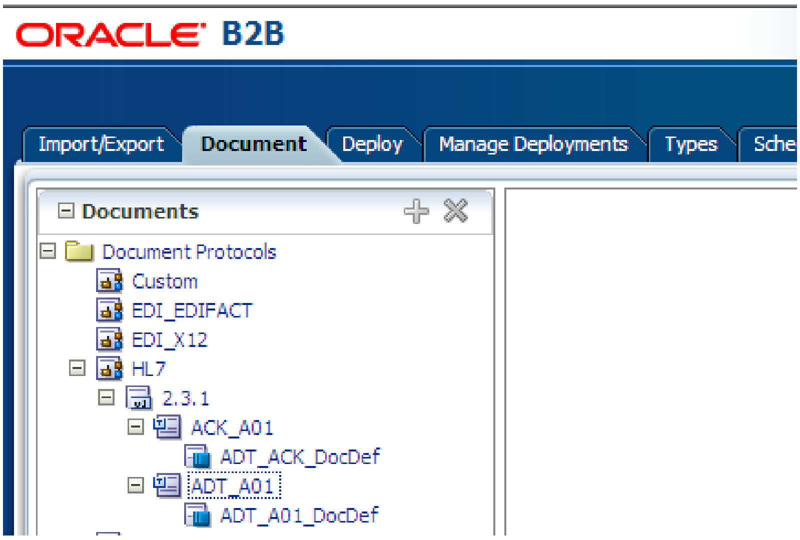
To develop the HL7 v2 outbound solution we need to go through a number of steps. The steps are:

1. *[Extract HL7 message structures from the standard library – already done in previous articles]*
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 File Reader B2B Composite

Extract HL7 Message Structures

As part of the B2B Document Editor installation we installed a number of standards libraries. One of these libraries was the HL7 v2 library.

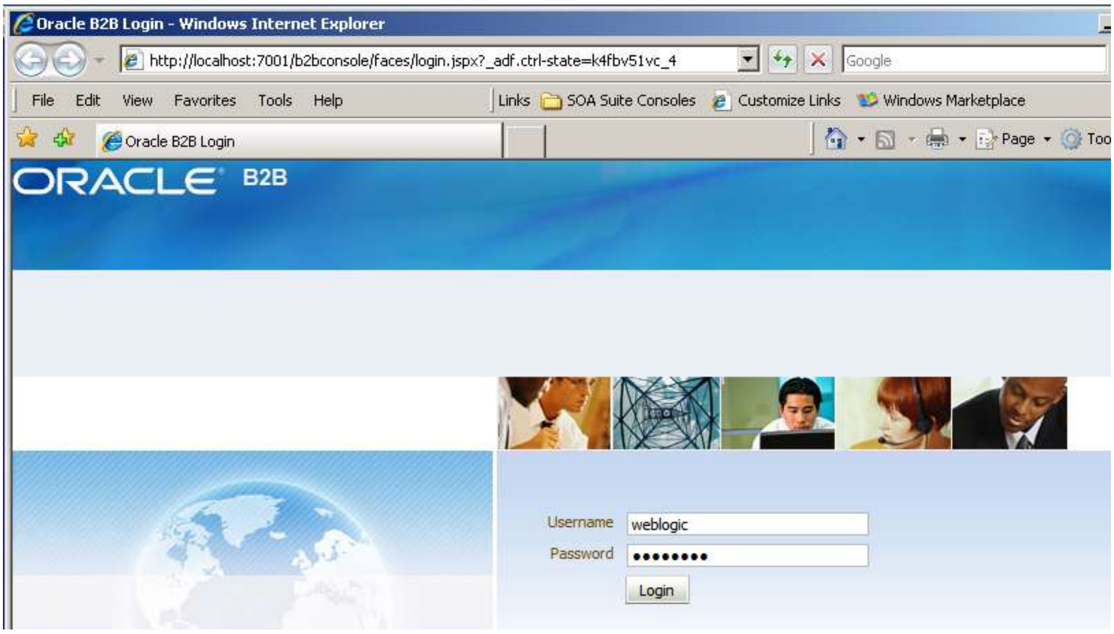
The solution developed in this article will send ADT A01 messages to the remote partner and will receive A01 ACK Messages in return. Message structures needed to implement the trading partnership agreement to support this interaction have already been exported from the message structure library in previous articles and have already been defined using the B2B Trading Partner Manager Console.



We will use these definitions in the work done in this article.

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.

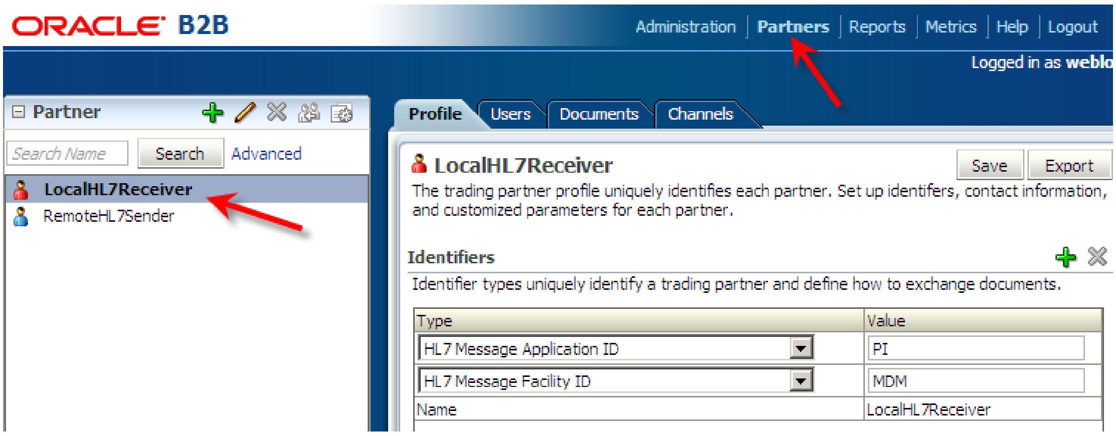


http://localhost:7001/b2bconsole/faces/login.jspx?_adf.ctrl-state=k4fbv51vc_4

Username: weblogic

Password: ••••••••

Note that the local partner, LocalHL7Receiver, already exists and has local identifiers defined. The trading partner was defined in previous articles. The identifiers associated with the local partner are the same as in the previous articles.



LocalHL7Receiver

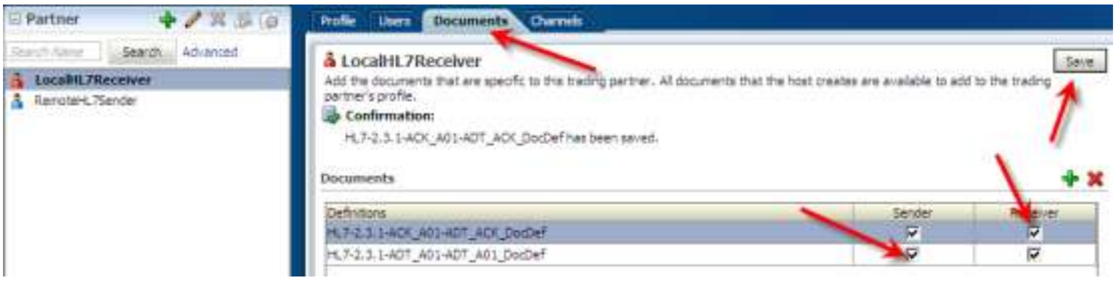
The trading partner profile uniquely identifies each partner. Set up identifers, contact information, and customized parameters for each partner.

Identifier types uniquely identify a trading partner and define how to exchange documents.

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

In the previous articles we were receiving A01s and sending ACKs. In this article we will be sending A01s and receiving ACKs. To do so we must ensure that both the sender and the receiver checkboxes are checked.

With the LocalHL7Receiver selected click the Documents Tab, check the Sender checkbox for the A01 document, check the Receiver checkbox for ACK document and click Save. We are configuring the HL7 outbound so the local role is that of a receiver of messages. To not break the receiver solution from previous articles we will leave appropriate checkboxes checked and will leave trading partner names as they are even though they no longer reflect messaging direction as they did in the first article. This should be a lesson for us on naming trading partners. Perhaps it would have been better to name the local partner Local_MDM_PI (local, Receiving Facility, Receiving Application) and Remote_HosA_SystemA. You work it out ☺



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.

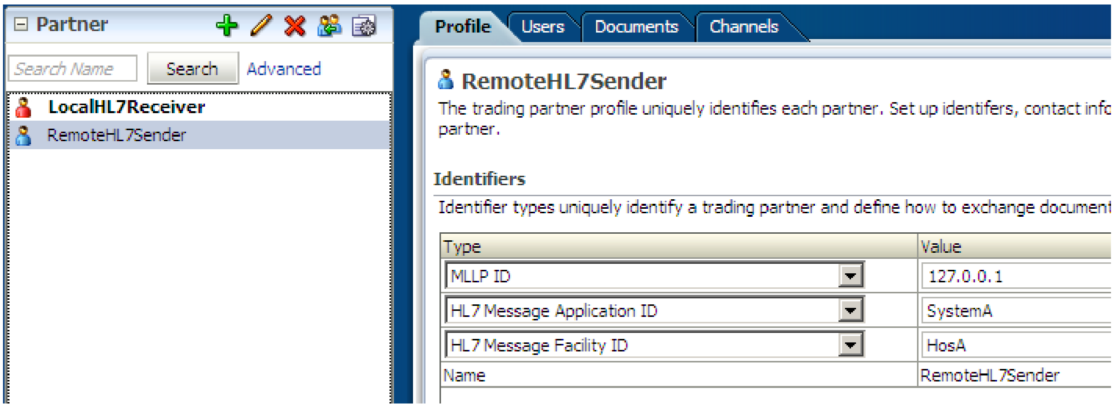
Confirmation:

HL7-2.3.1-ACK_A01-ADT_ACK_DocDef has been saved.

Documents

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

Since we will be sending A01 documents to the same external party from which we were receiving A01 documents we will reuse the RemoteHL7Sender trading partner. The identifiers with which it is configured are correct.



RemoteHL7Sender

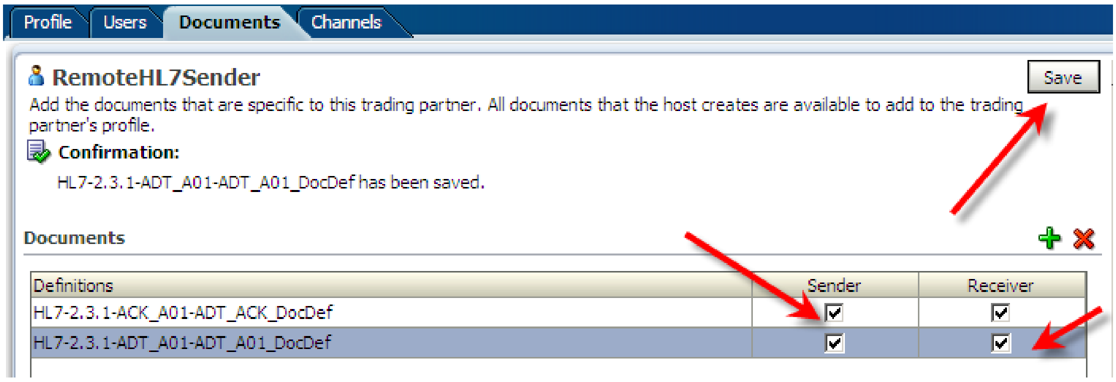
The trading partner profile uniquely identifies each partner. Set up identifers, contact info partner.

Identifiers

Identifier types uniquely identify a trading partner and define how to exchange document

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

With the RemoteHL7Sender selected, click the Documents Tab, check the Sender checkbox for the ACK document and the Receiver checkbox for the A01 document, then 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.

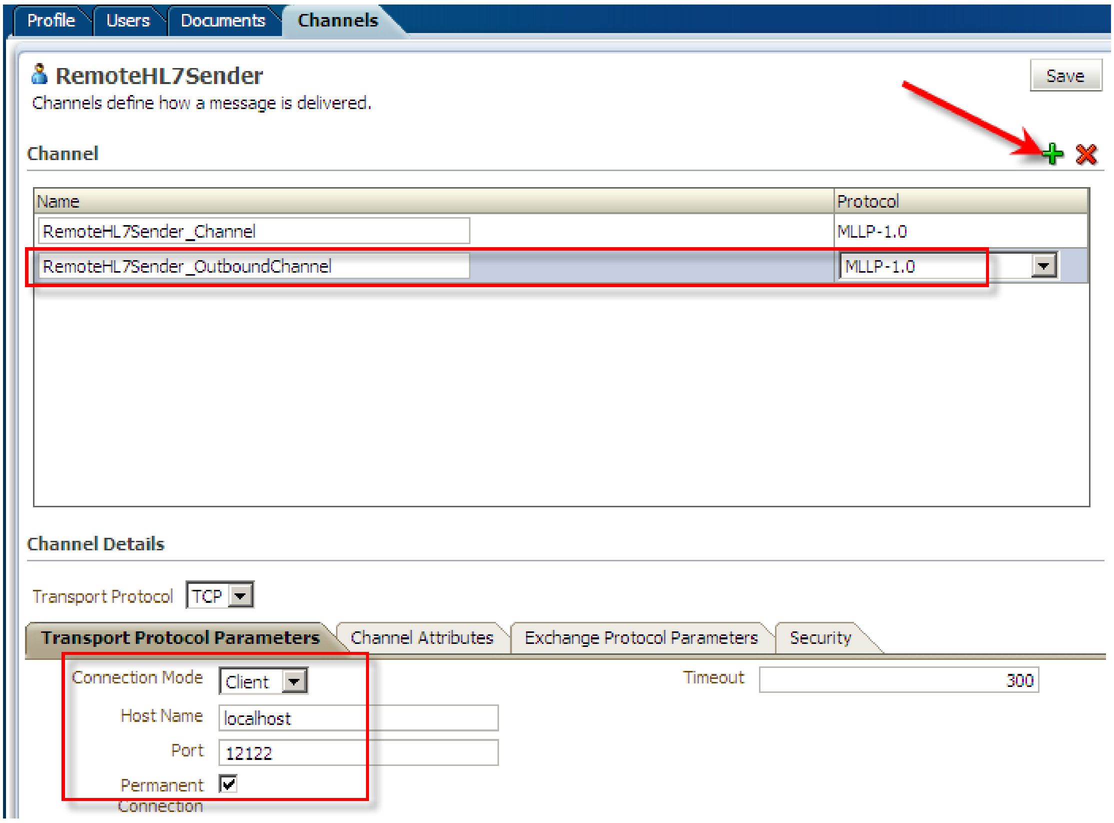
Confirmation:

HL7-2.3.1-ADT_A01-ADT_A01_DocDef has been saved.

Documents

Definitions	Sender	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, Click the large Plus sign to add a new Channel. Change the name of the channel to RemoteHL7Sender_OutboundChannel, choose MLLP-1.0 from the drop-down and configure Connection Mode: Client, Host Name: localhost, Port: 12122 and check Permanent Connection. Click Save once done.



RemoteHL7Sender

Channels define how a message is delivered.

Channel

Name	Protocol
RemoteHL7Sender_Channel	MLLP-1.0
RemoteHL7Sender_OutboundChannel	MLLP-1.0

Channel Details

Transport Protocol: TCP

Connection Mode: Client

Host Name: localhost

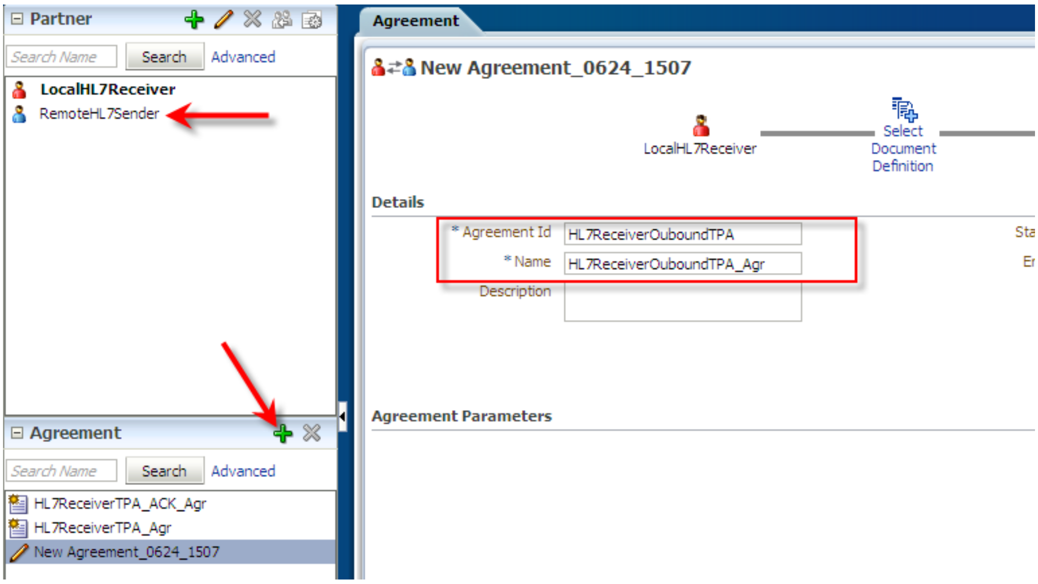
Port: 12122

Permanent Connection: ☑

Timeout: 300

We configured document types the remote partner will be receiving and configured the client we need to use on our side to send messages to that partner.

With the RemoteHL7Sender partner selected, click the large Plus sign in the Agreements section to add a new trading partnership agreement. Provide the Agreement ID of HL7ReceiverOuboundTPA and the Name of HL7ReceiverOutboundTPA_Agr.



LocalHL7Receiver

RemoteHL7Sender

HL7ReceiverTPA_ACK_Agr

HL7ReceiverTPA_Agr

New Agreement_0624_1507

New Agreement_0624_1507

LocalHL7Receiver

Details

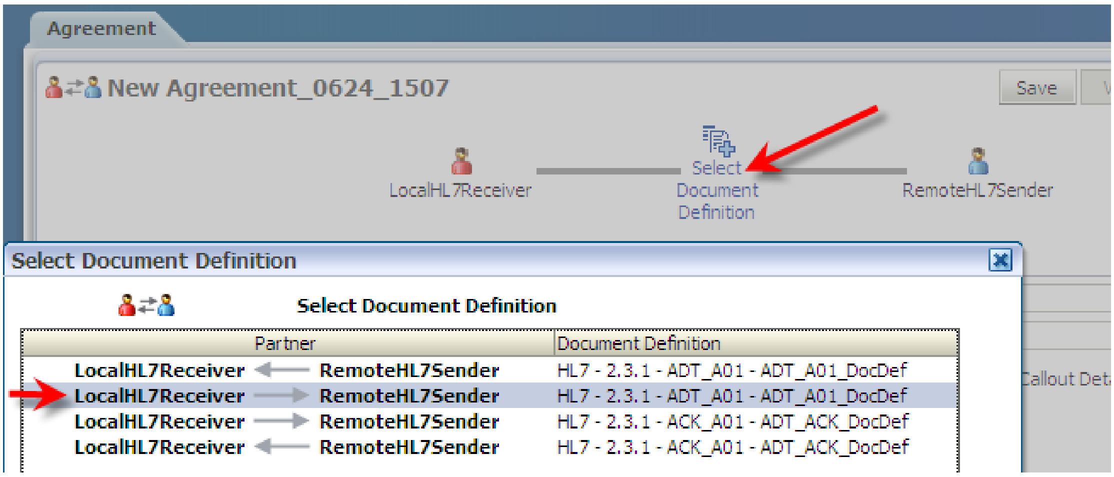
* Agreement Id: HL7ReceiverOuboundTPA

* Name: HL7ReceiverOuboundTPA_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 sender to remote receiver and click OK.



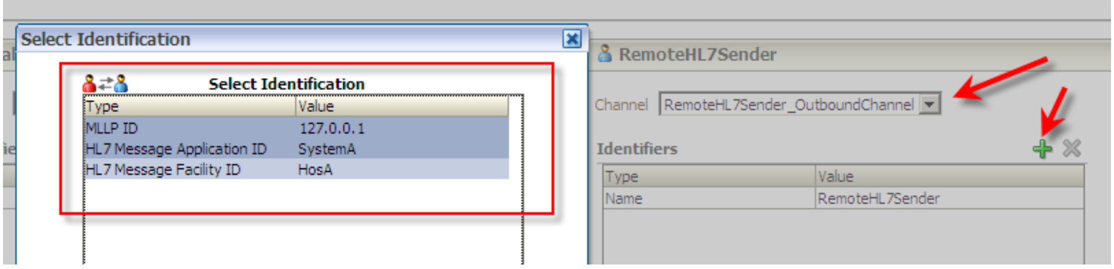
Select Document Definition

Select Document Definition

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

Choose the RemoteHL7Sender_OutboundChannel 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.



Select Identification

Select Identification

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

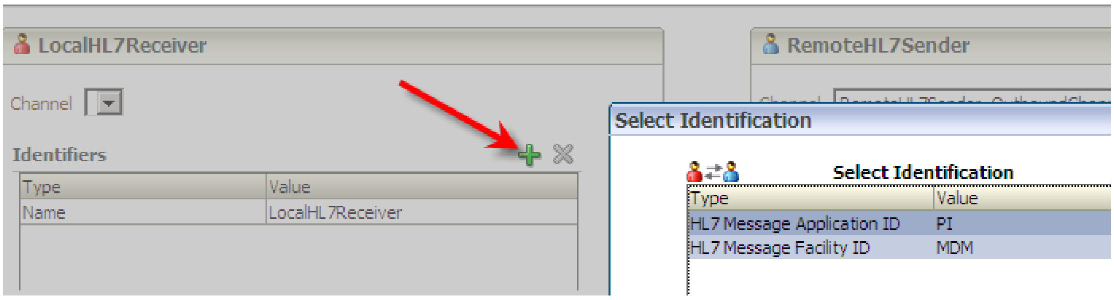
RemoteHL7Sender

Channel: RemoteHL7Sender_OutboundChannel

Identifiers

Type	Value
Name	RemoteHL7Sender

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

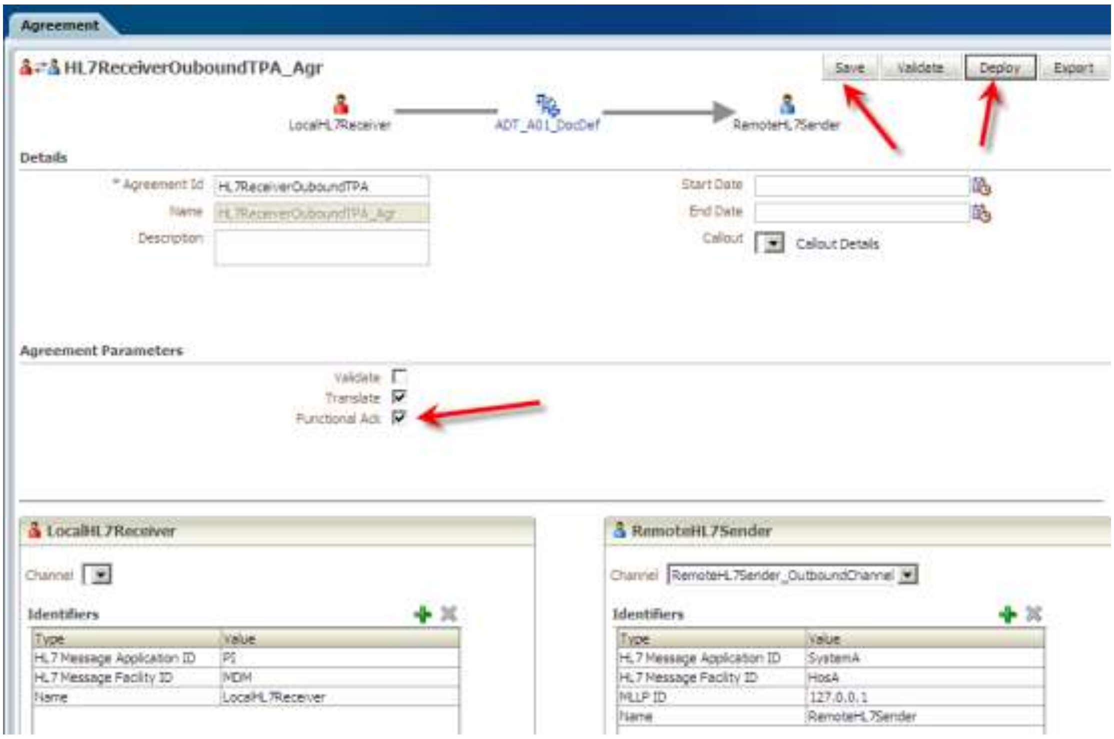
RemoteHL7Sender

Select Identification

Select Identification

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

Check the Functional Ack checkbox, click Save to save the new trading partnership agreement, then click Deploy to deploy it.



HL7ReceiverOuboundTPA_Agr

Details

* Agreement Id: HL7ReceiverOuboundTPA

Name: HL7ReceiverOuboundTPA_Agr

Description:

Start Date:

End Date:

Callout: Callout Details

Agreement Parameters

Validate ☐

Translate ☑

Functional Ack ☑

LocalHL7Receiver

Channel:

Identifiers

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

RemoteHL7Sender

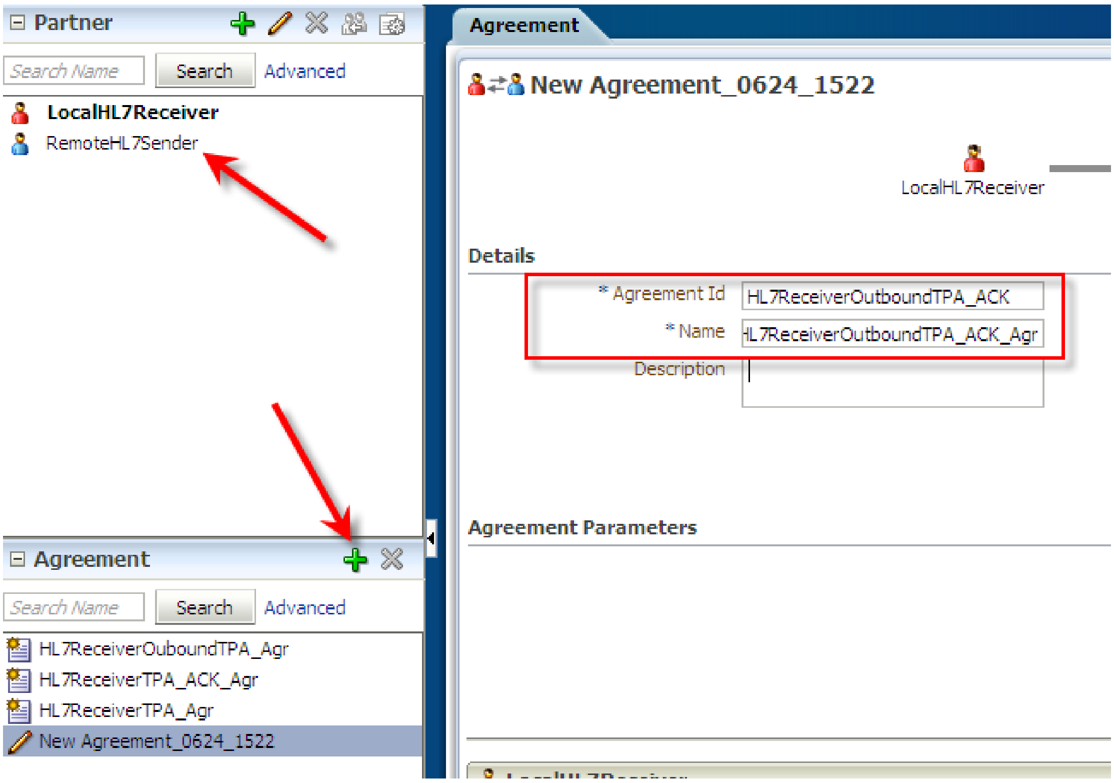
Channel: RemoteHL7Sender_OutboundChannel

Identifiers

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

We now need to add an agreement for receiving the ACK back from the receiver.

Click the large Plus sign in the Agreements section to add a new trading partnership agreement. Provide the Agreement ID of HL7ReceiverOutboundTPA_ACK and the Name of HL7ReceiverOutboundTPA_ACK_Agr.



HL7ReceiverOutboundTPA_Agr

HL7ReceiverTPA_ACK_Agr

HL7ReceiverTPA_Agr

New Agreement_0624_1522

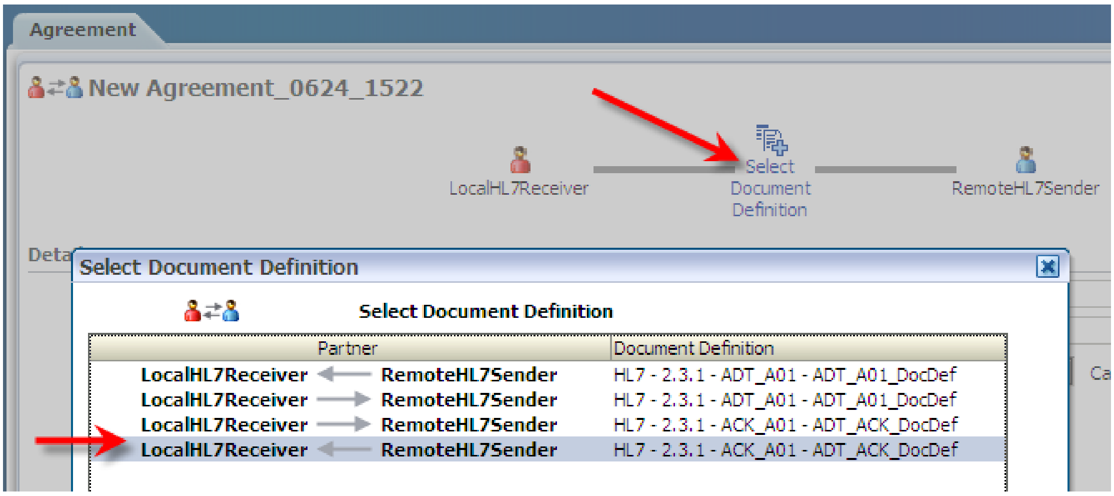
New Agreement_0624_1522

* Agreement Id: HL7ReceiverOutboundTPA_ACK

* Name: HL7ReceiverOutboundTPA_ACK_Agr

Description:

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

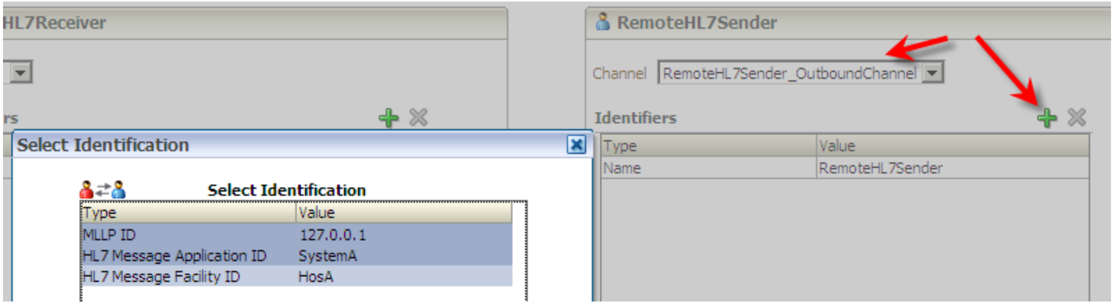


Select Document Definition

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

Choose the RemoteHL7Sender_OuboundChannel 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.



RemoteHL7Sender

Channel RemoteHL7Sender_OutboundChannel

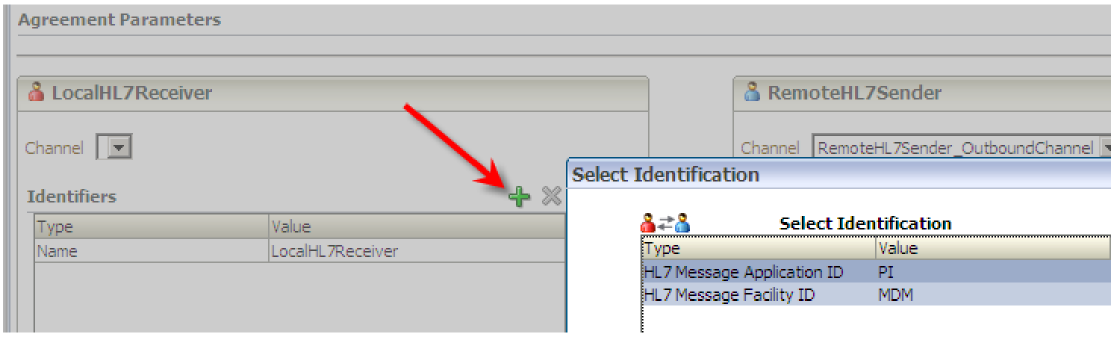
Identifiers

Type	Value
Name	RemoteHL7Sender

Select Identification

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

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

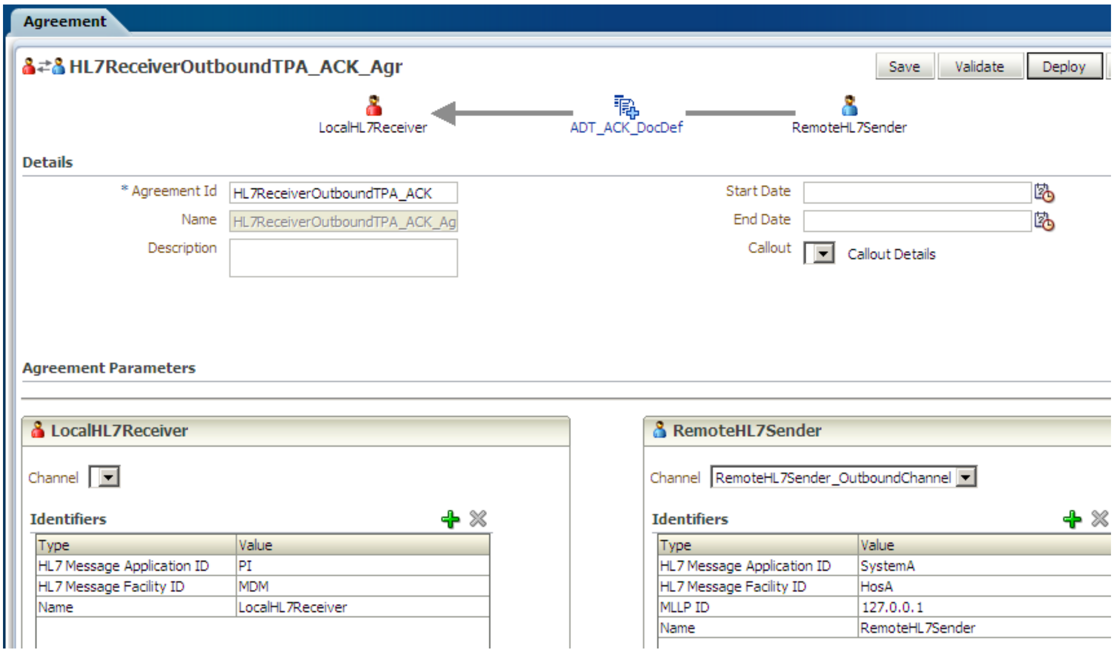
RemoteHL7Sender

Channel RemoteHL7Sender_OutboundChannel

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.



HL7ReceiverOutboundTPA_ACK_Agr

Details

* Agreement Id: HL7ReceiverOutboundTPA_ACK
Name: HL7ReceiverOutboundTPA_ACK_Ag
Description:
Start Date:
End Date:
Callout:

Agreement Parameters

LocalHL7Receiver

Channel:

Identifiers

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

RemoteHL7Sender

Channel: RemoteHL7Sender_OutboundChannel

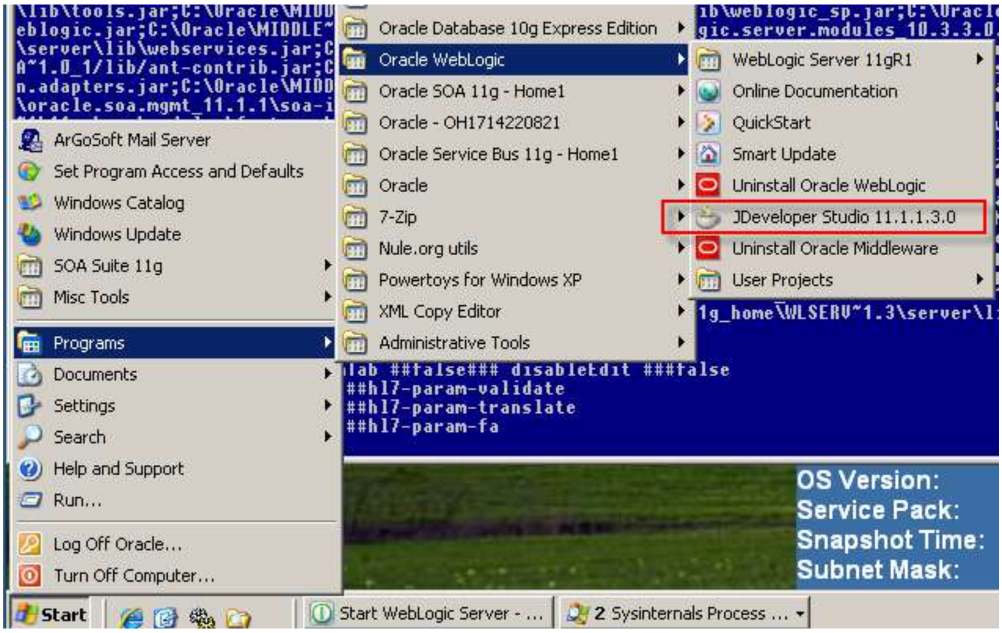
Identifiers

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

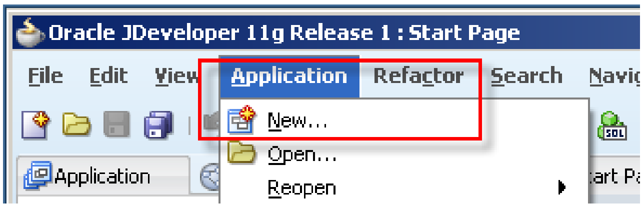
We are finished with the trading partner management console for the time being. The partnerships have been defined and deployed. Now we need the runtime solution that will read messages, and send them to the remote partner according to the trading partnership agreement. This solution, which will be a SOA Suite Composite, will be developed using the JDeveloper IDE and will be deployed to the WebLogic Application Server.

Develop File Reader Solution

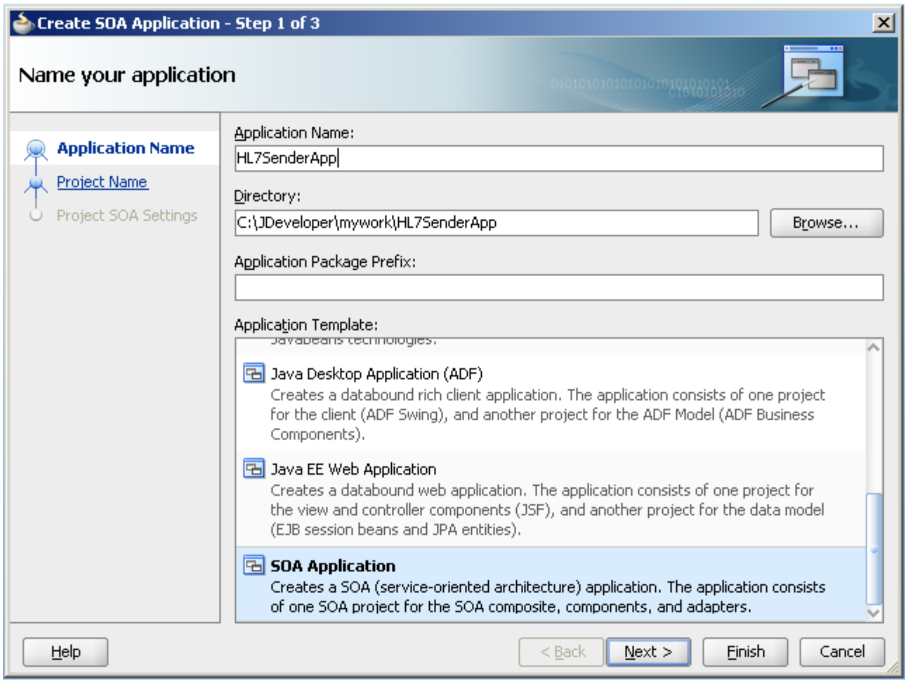
Start JDevelope IDE, perhaps by following Start menu → Programs → Oracle WebLogic → JDeveloper Studio 11.1.1.3.0 path through the menus.


\lib\tools.jar;C:\Oracle\MIDD
eblogic.jar;C:\Oracle\MIDDLE~
\server\lib\webservices.jar;C
A~1.0_1/lib/ant-contrib.jar;C
n.adapters.jar;C:\Oracle\MIDD
\oracle.soa.mgmt_11.1.1\soa-i
ArGoSoft Mail Server
Set Program Access and Defaults
Windows Catalog
Windows Update
SOA Suite 11g
Misc Tools
Programs
Documents
Settings
Search
Help and Support
Run...
Log Off Oracle...
Turn Off Computer...
Oracle Database 10g Express Edition
Oracle WebLogic
Oracle SOA 11g - Home1
Oracle - OH1714220821
Oracle Service Bus 11g - Home1
Oracle
7-Zip
Nule.org utils
Powertoys for Windows XP
XML Copy Editor
Administrative Tools
ib\weblogic_sp.jar;C:\Oracl
gic.server.modules_10.3.3.0
WebLogic Server 11gR1
Online Documentation
QuickStart
Smart Update
Uninstall Oracle WebLogic
JDeveloper Studio 11.1.1.3.0
Uninstall Oracle Middleware
User Projects
1g_home\WLSERV~1.3\server\l
lab ##false### disableEdit ###false
##hl7-param-validate
##hl7-param-translate
##hl7-param-fa
OS Version:
Service Pack:
Snapshot Time:
Subnet Mask:
Start
Start WebLogic Server - ...
2 Sysinternals Process ...

Once JDeveloper is up create a new SOA Application.



Name this application HL7SenderApp, choose SOA Application template and click Next


Create SOA Application - Step 1 of 3

Name your application

- Application Name
- Project Name
- Project SOA Settings

Application Name: HL7SenderApp

Directory: C:\JDeveloper\mywork\HL7SenderApp

Application Package Prefix:

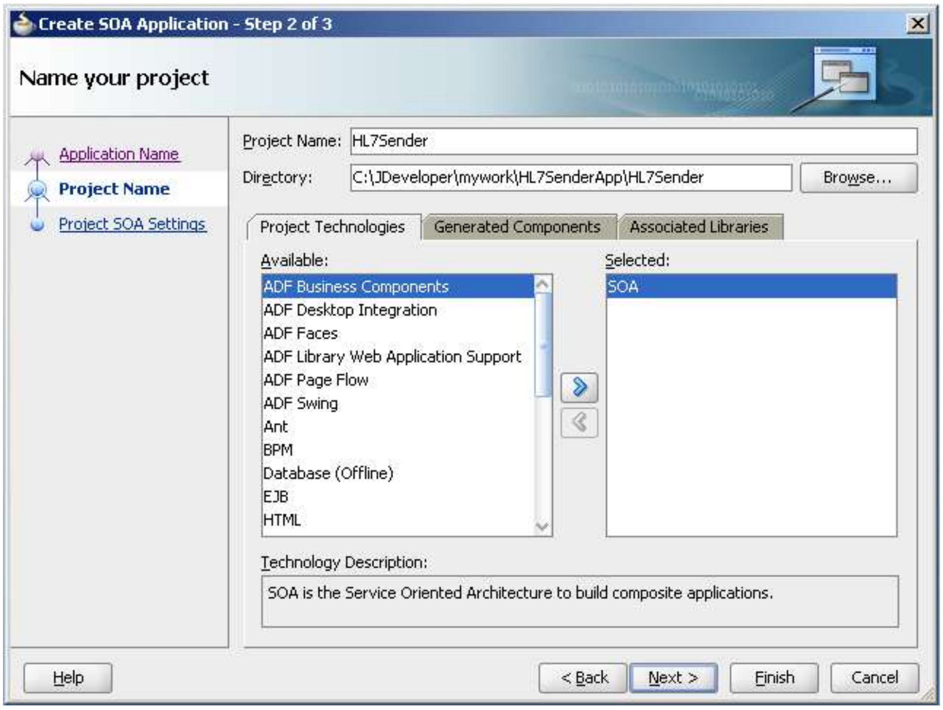
Application Template:

Java Desktop Application (ADF)
Creates a databound rich client application. The application consists of one project for the client (ADF Swing), and another project for the ADF Model (ADF Business Components).

Java EE Web Application
Creates a databound web application. The application consists of one project for the view and controller components (JSF), and another project for the data model (EJB session beans and JPA entities).

SOA Application
Creates a SOA (service-oriented architecture) application. The application consists of one SOA project for the SOA composite, components, and adapters.

Name the project HL7Sender and click Next.


Create SOA Application - Step 2 of 3

Name your project

- Application Name
- Project Name
- Project SOA Settings

Project Name: HL7Sender

Directory: C:\JDeveloper\mywork\HL7SenderApp\HL7Sender

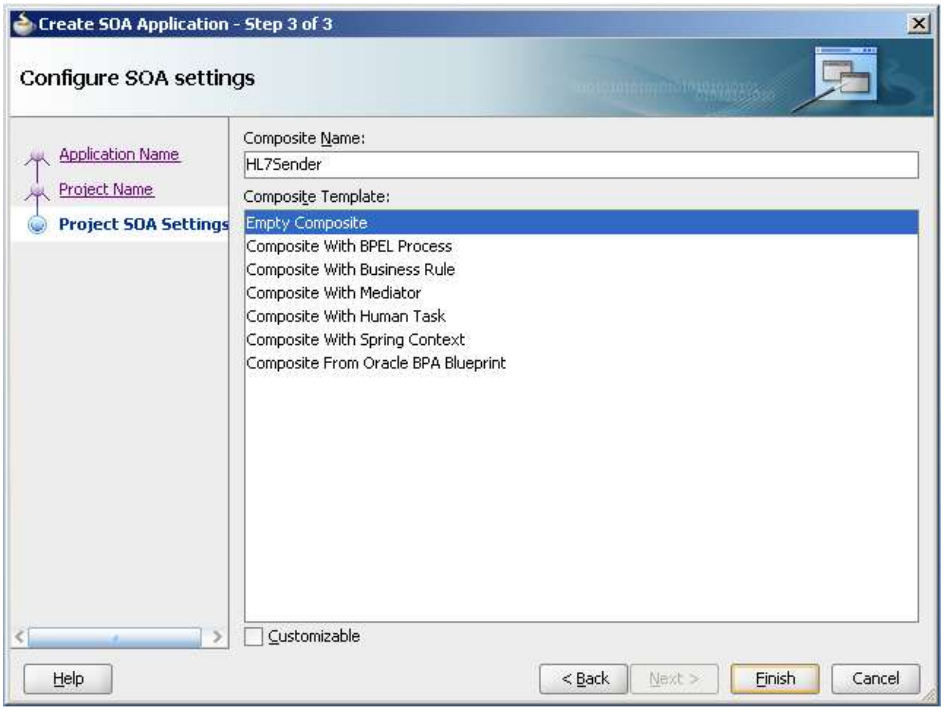
Project Technologies | Generated Components | Associated Libraries

Available:
ADF Business Components
ADF Desktop Integration
ADF Faces
ADF Library Web Application Support
ADF Page Flow
ADF Swing
Ant
BPM
Database (Offline)
EJB
HTML

Selected:
SOA

Technology Description:
SOA is the Service Oriented Architecture to build composite applications.

Accept the default Composite Name of HL7Sender and default template Empty Composite, and click Finish.



Create SOA Application - Step 3 of 3

Configure SOA settings

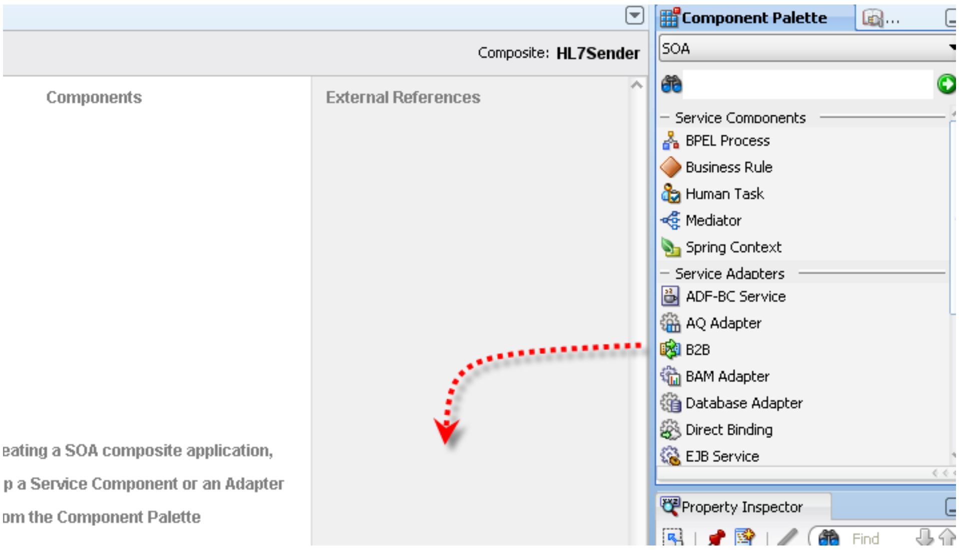
Composite Name: HL7Sender

Composite Template: Empty Composite

☐ Customizable

Application Name
Project Name
Project SOA Settings
Empty Composite
Composite With BPEL Process
Composite With Business Rule
Composite With Mediator
Composite With Human Task
Composite With Spring Context
Composite From Oracle BPA Blueprint
Help
< Back
Next >
Finish
Cancel

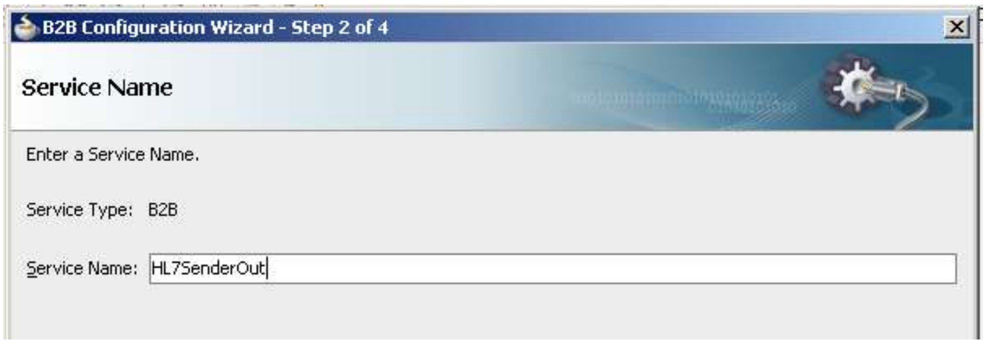
Drag the B2B Service Adapter from the list of service adapters in the SOA components palette onto the External References swim line.



Composite: HL7Sender

Click Next to the “Welcome to B2B Configuration Wizard” dialogue window.

Set the Service Name to HL7SenderOut and click Next.



B2B Configuration Wizard - Step 2 of 4

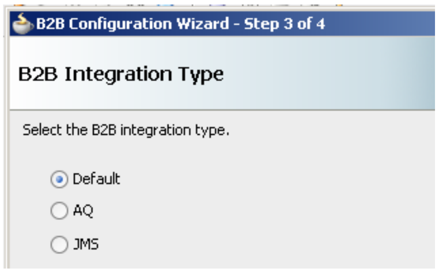
Service Name

Enter a Service Name.

Service Type: B2B

Service Name: HL7SenderOut

Accept default “Default” and the B2B Integration Type and click Next.



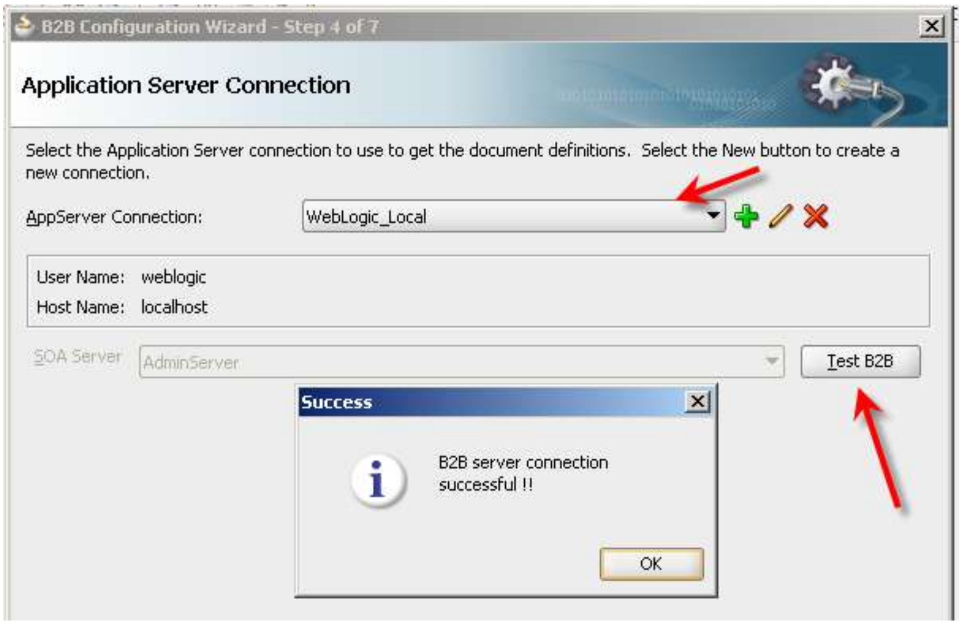
B2B Configuration Wizard - Step 3 of 4

B2B Integration Type

Select the B2B integration type.

● Default
○ AQ
○ JMS

Choose WebLogic_Local connection name, click B2B Test and click OK to dismiss the success dialogue box. Click Next.



B2B Configuration Wizard - Step 4 of 7

Application Server Connection

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

AppServer Connection: WebLogic_Local

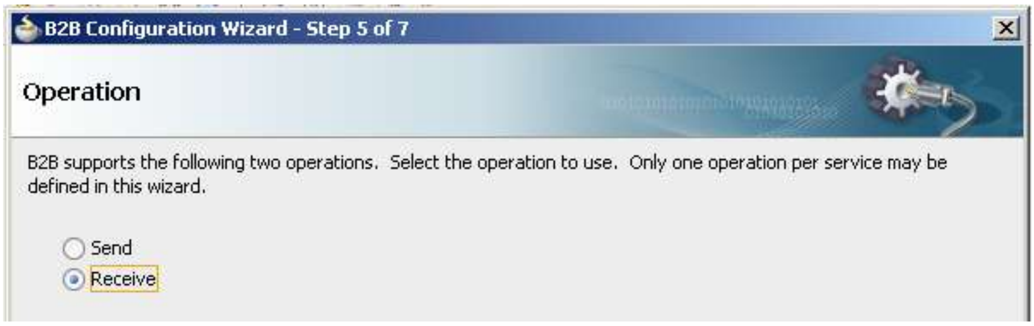
User Name: weblogic
Host Name: localhost

SOA Server: AdminServer

Success

B2B server connection successful !!

Choose the Send operation and click Next.



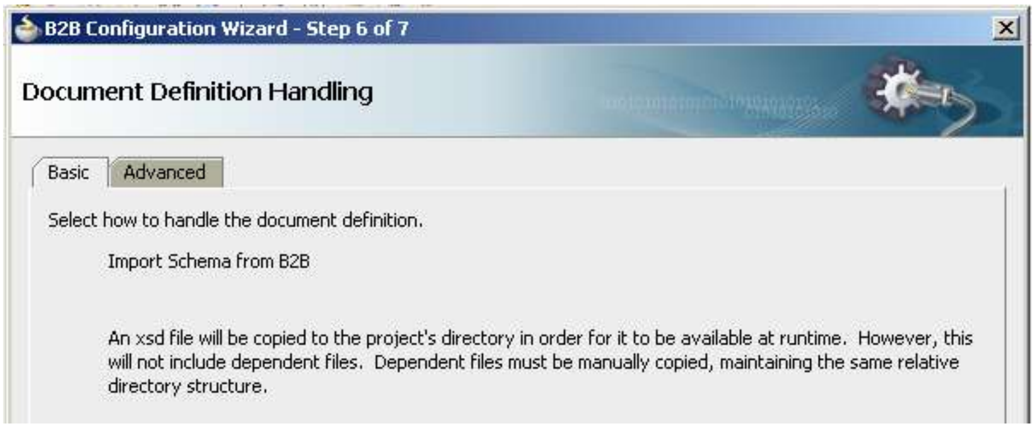
B2B Configuration Wizard - Step 5 of 7

Operation

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

○ Send
● Receive

Keep the Document Definition Handling at Basic, the default, and click Next.



B2B Configuration Wizard - Step 6 of 7

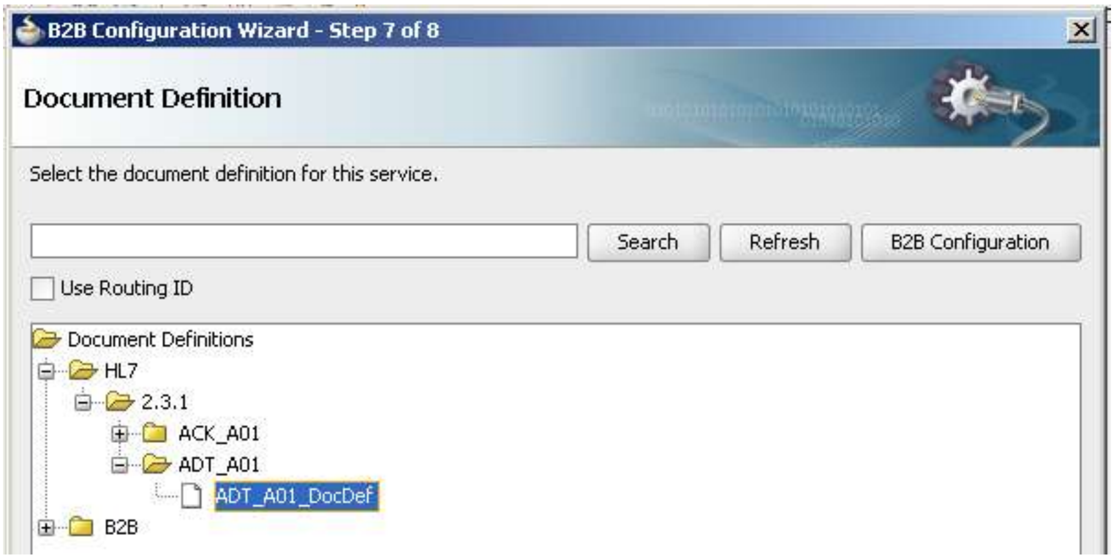
Document Definition Handling

Select how to handle the document definition.

Import Schema from B2B

An xsd file will be copied to the project's directory in order for it to be available at runtime. However, this will not include dependent files. Dependent files must be manually copied, maintaining the same relative directory structure.

Choose the HL7 2.3.1 ADT A01 document definition and click Next.



B2B Configuration Wizard - Step 7 of 8

Document Definition

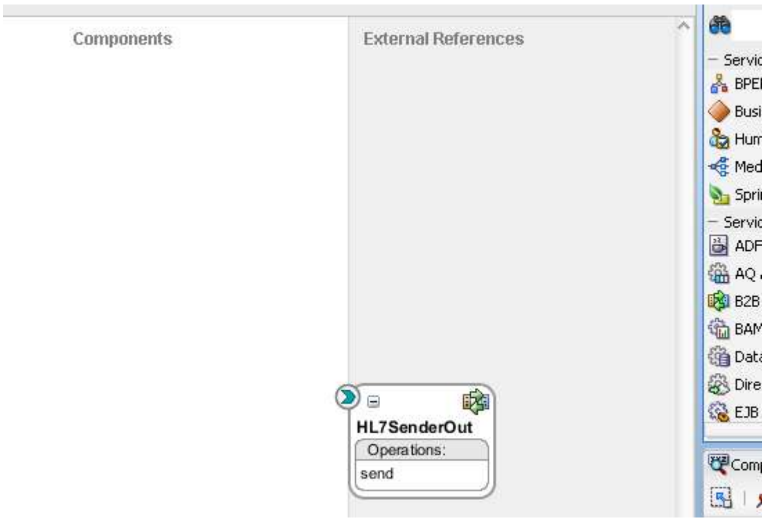
Select the document definition for this service.

□ Use Routing ID

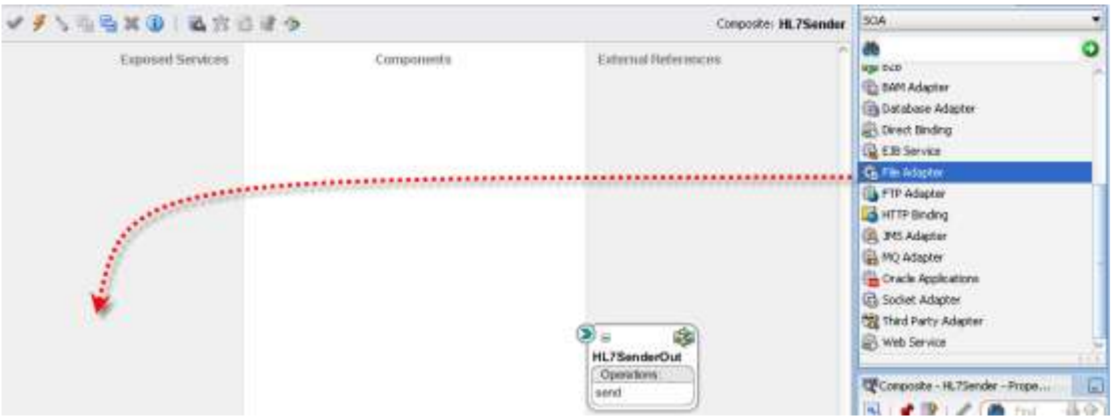
Document Definitions
- HL7
  - 2.3.1
    - ACK_A01
    - ADT_A01
      - ADT_A01_DocDef
- B2B

Click Finish to complete the wizard.

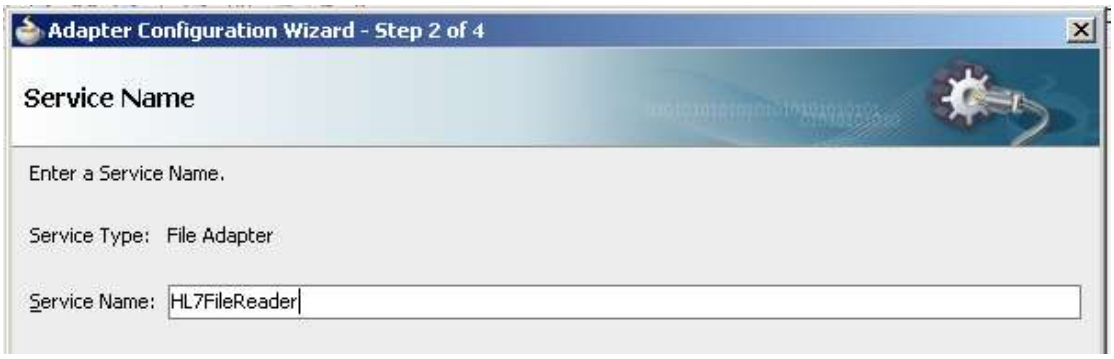
The HL7SenderOut will appear in the composite in the “External References” swim line.



Drag the File Adapter from the Service Adapters component palette to the Exposed Services swim line.



Click Next to the Welcome dialogue, enter HL7FileReader as service name and click Next.


Adapter Configuration Wizard - Step 2 of 4

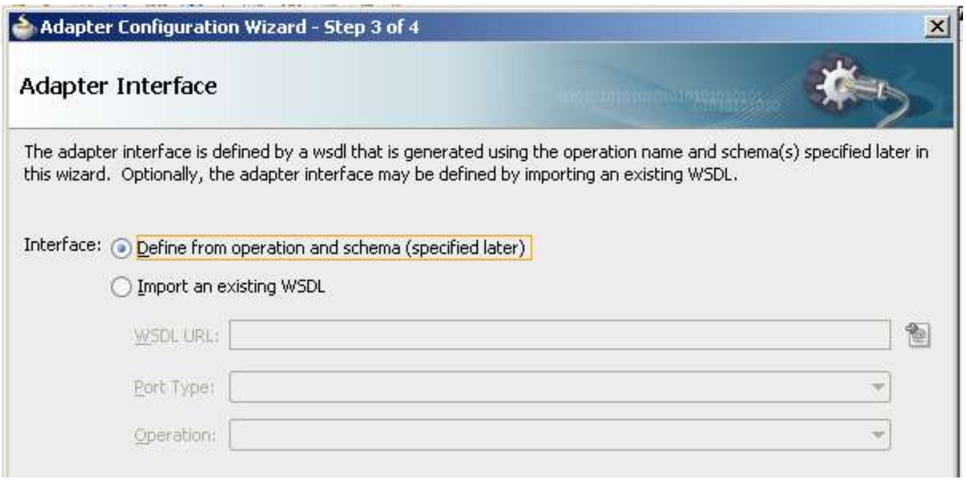
Service Name

Enter a Service Name.

Service Type: File Adapter

Service Name: HL7FileReader

Accept the defaults for Adapter Interface and click Next.



The adapter interface is defined by a wsdl that is generated using the operation name and schema(s) specified later in this wizard. Optionally, the adapter interface may be defined by importing an existing WSDL.

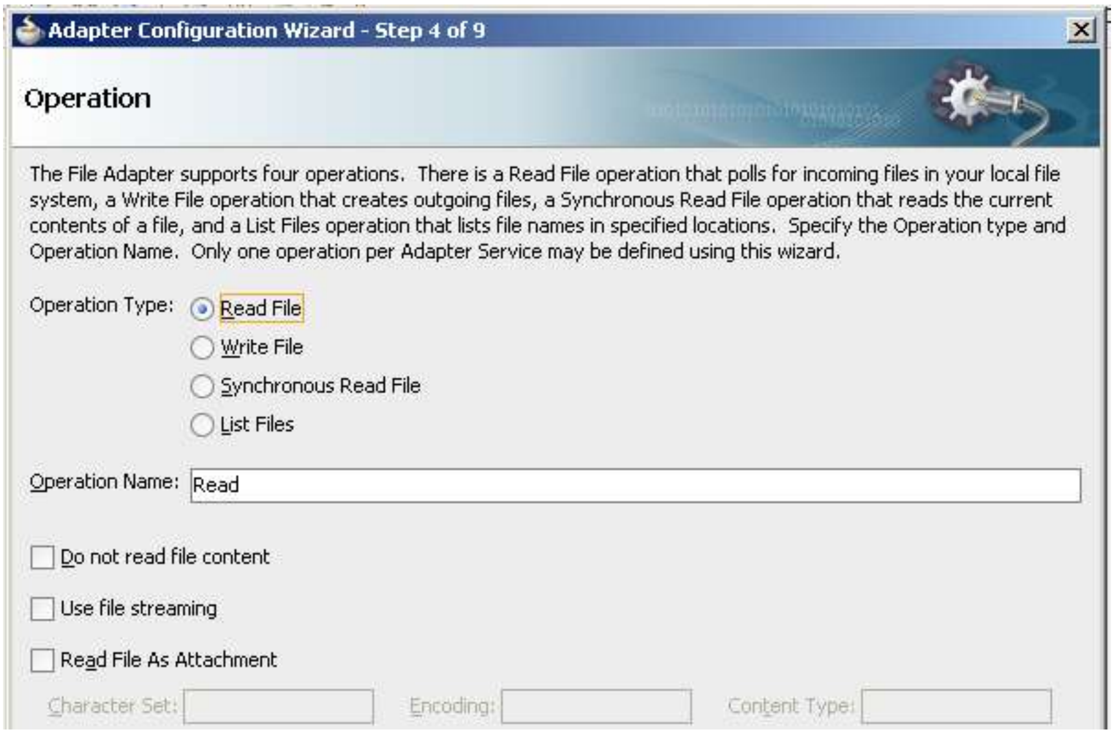
Interface: ● Define from operation and schema (specified later)
○ Import an existing WSDL

WSDL URL:

Port Type:

Operation:

Choose Read File operation type, accept default operation name and other defaults and click Next.



The File Adapter supports four operations. There is a Read File operation that polls for incoming files in your local file system, a Write File operation that creates outgoing files, a Synchronous Read File operation that reads the current contents of a file, and a List Files operation that lists file names in specified locations. Specify the Operation type and Operation Name. Only one operation per Adapter Service may be defined using this wizard.

Operation Type: ● Read File
○ Write File
○ Synchronous Read File
○ List Files

Operation Name: Read

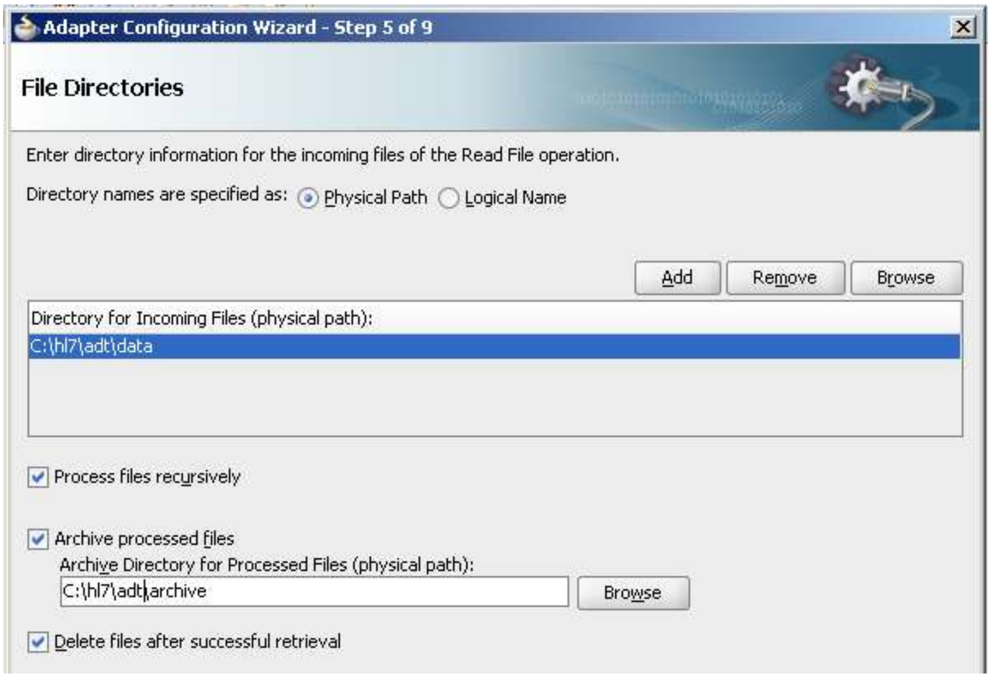
□ Do not read file content

□ Use file streaming

□ Read File As Attachment

Character Set: Encoding: Content Type:

Specify C:\hl7\adt\data as the physical path to the directory from which the files will be read and C:\hl7\adt\archive as the archive directory to which read files will be moved. Make sure both directories exist.



Adapter Configuration Wizard - Step 5 of 9

File Directories

Enter directory information for the incoming files of the Read File operation.

Directory names are specified as: ● Physical Path ○ Logical Name

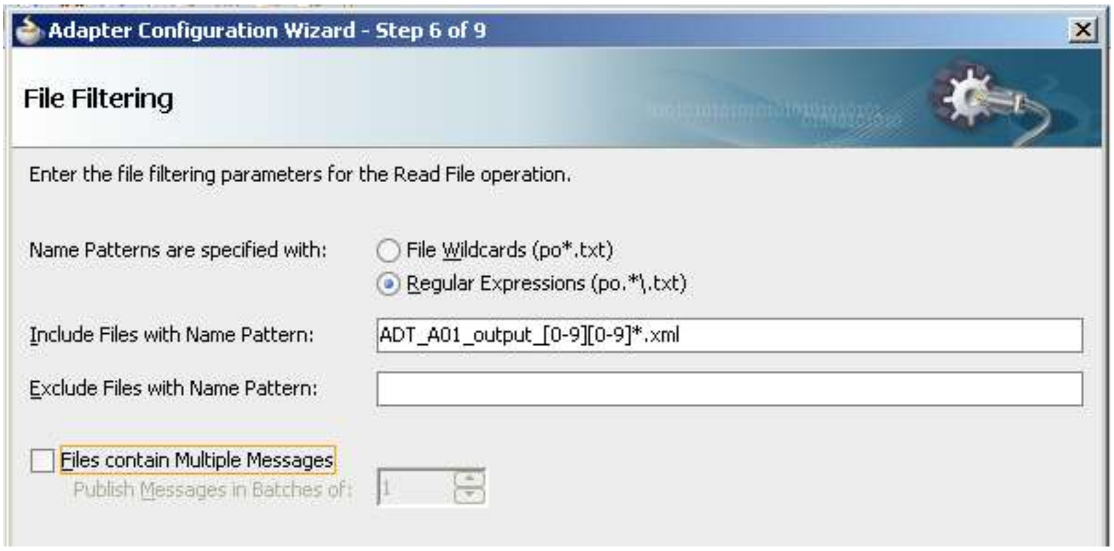
Directory for Incoming Files (physical path):
C:\hl7\adt\data

☑ Process files recursively

☑ Archive processed files
Archive Directory for Processed Files (physical path):
C:\hl7\adt\archive

☑ Delete files after successful retrieval

Click the “Regular Expressions …” radio button and specify ADT_A01_output_[0-9]*.xml as the regular expression. Make sure “File contains Multiple Messages” is not checked. Click Next.



Adapter Configuration Wizard - Step 6 of 9

File Filtering

Enter the file filtering parameters for the Read File operation.

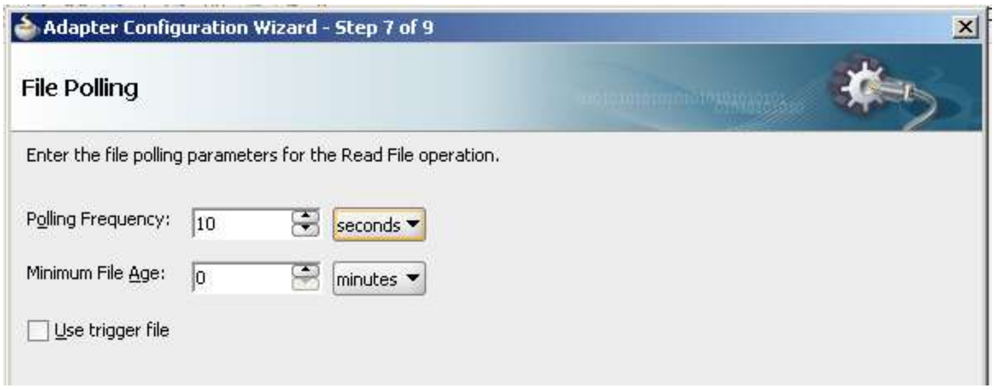
Name Patterns are specified with: ○ File Wildcards (po*.txt) ● Regular Expressions (po.*\.txt)

Include Files with Name Pattern: ADT_A01_output_[0-9][0-9]*.xml

Exclude Files with Name Pattern:

□ Files contain Multiple Messages
Publish Messages in Batches of: 1

Change polling frequency to 10 seconds and click Next.



Adapter Configuration Wizard - Step 7 of 9

File Polling

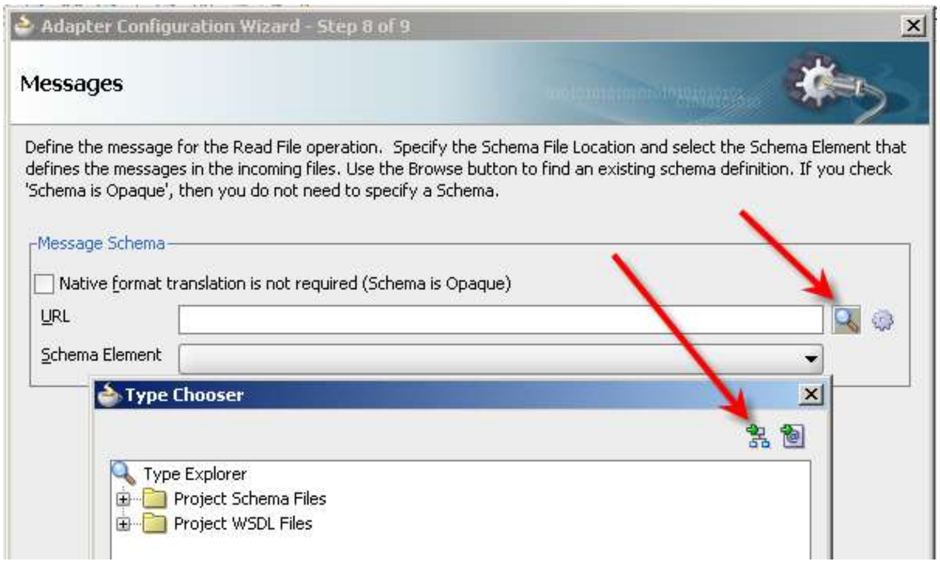
Enter the file polling parameters for the Read File operation.

Polling Frequency: 10 seconds

Minimum File Age: 0 minutes

□ Use trigger file

Click “Browse for Schema File” and Import Schema File.



Adapter Configuration Wizard - Step 8 of 9

Messages

Define the message for the Read File operation. Specify the Schema File Location and select the Schema Element that defines the messages in the incoming files. Use the Browse button to find an existing schema definition. If you check 'Schema is Opaque', then you do not need to specify a Schema.

Message Schema

□ Native format translation is not required (Schema is Opaque)

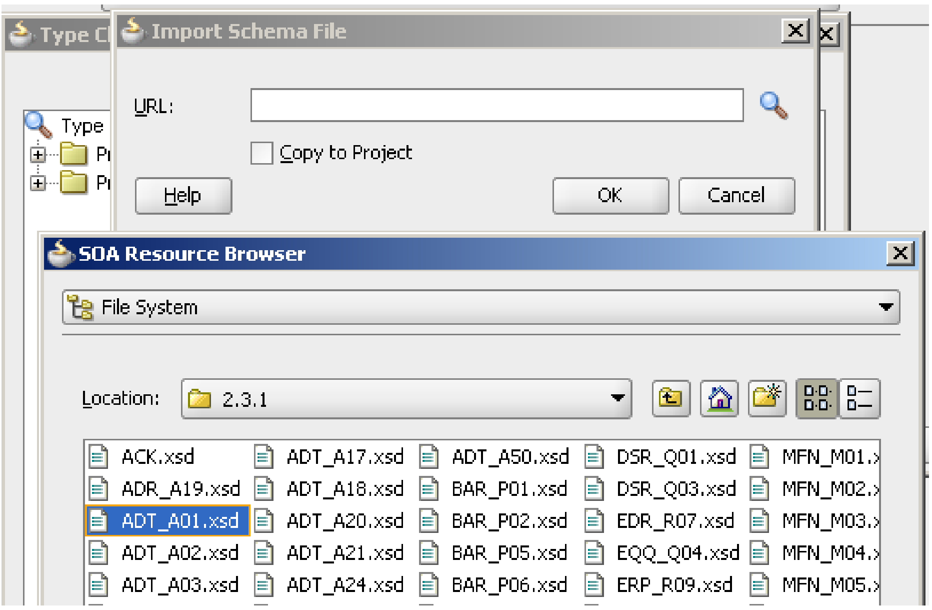
URL

Schema Element

Type Chooser

Type Explorer
- Project Schema Files
- Project WSDL Files

Locate ADT A01 schema in the 2.3.1 version directory of XML Schemas we downloaded earlier – C:\hl7\HL7v2XSDs\2.3.1.



Import Schema File

URL:

□ Copy to Project

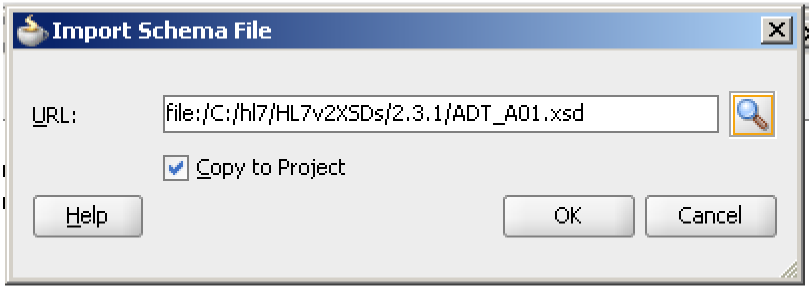
SOA Resource Browser

File System

Location: 2.3.1

ACK.xsd
ADR_A19.xsd
ADT_A01.xsd
ADT_A02.xsd
ADT_A03.xsd
ADT_A17.xsd
ADT_A18.xsd
ADT_A20.xsd
ADT_A21.xsd
ADT_A24.xsd
ADT_A50.xsd
BAR_P01.xsd
BAR_P02.xsd
BAR_P05.xsd
BAR_P06.xsd
DSR_Q01.xsd
DSR_Q03.xsd
EDR_R07.xsd
EQQ_Q04.xsd
ERP_R09.xsd
MFN_M01.x
MFN_M02.x
MFN_M03.x
MFN_M04.x
MFN_M05.x

Make sure “Copy to project” is checked and click OK.

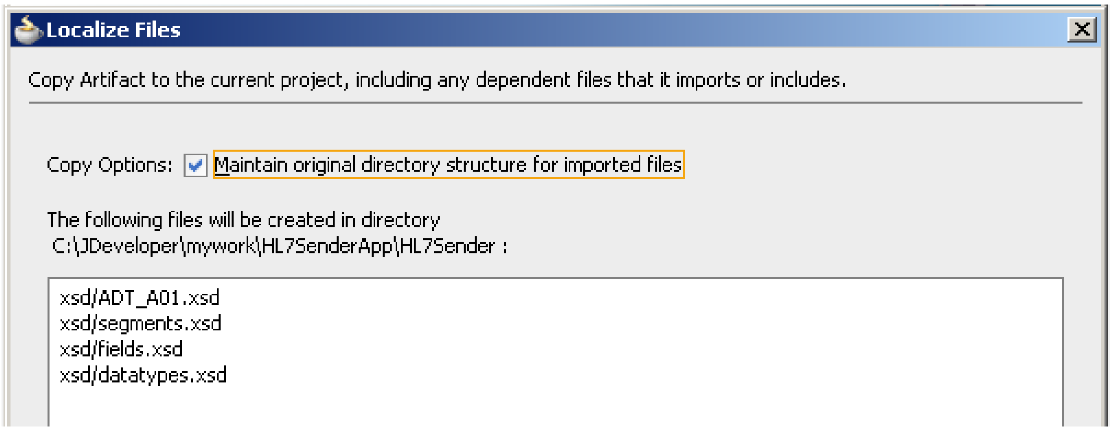


Import Schema File

URL: file:/C:/hl7/HL7v2XSDs/2.3.1/ADT_A01.xsd

☑ Copy to Project

Click OK to accept what the dependencies dialogue box says.



Localize Files

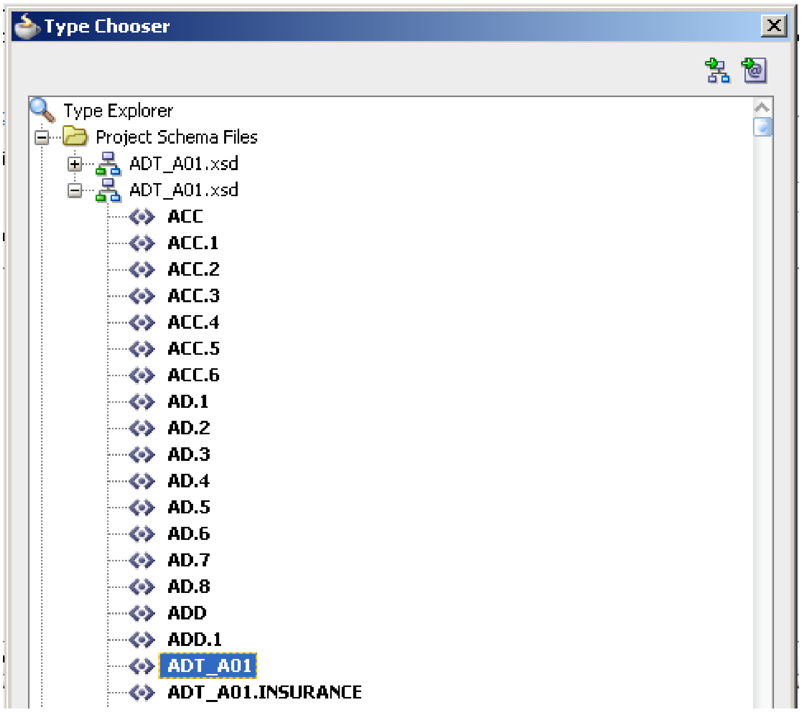
Copy Artifact to the current project, including any dependent files that it imports or includes.

Copy Options: ☑ Maintain original directory structure for imported files

The following files will be created in directory C:\JDeveloper\mywork\HL7SenderApp\HL7Sender :

```
xsd/ADT_A01.xsd
xsd/segments.xsd
xsd/fields.xsd
xsd/datatypes.xsd
```

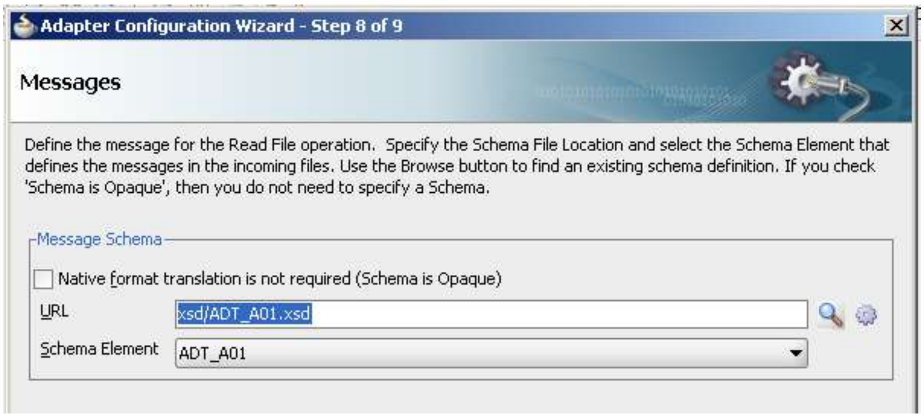
Choose ADT_A01 and click OK.



Type Chooser

```
Type Explorer
  Project Schema Files
    ADT_A01.xsd
    ADT_A01.xsd
      ACC
      ACC.1
      ACC.2
      ACC.3
      ACC.4
      ACC.5
      ACC.6
      AD.1
      AD.2
      AD.3
      AD.4
      AD.5
      AD.6
      AD.7
      AD.8
      ADD
      ADD.1
      ADT_A01
      ADT_A01.INSURANCE
```

Confirm the schema element, click Next and click Finish.



Adapter Configuration Wizard - Step 8 of 9

Messages

Define the message for the Read File operation. Specify the Schema File Location and select the Schema Element that defines the messages in the incoming files. Use the Browse button to find an existing schema definition. If you check 'Schema is Opaque', then you do not need to specify a Schema.

Message Schema

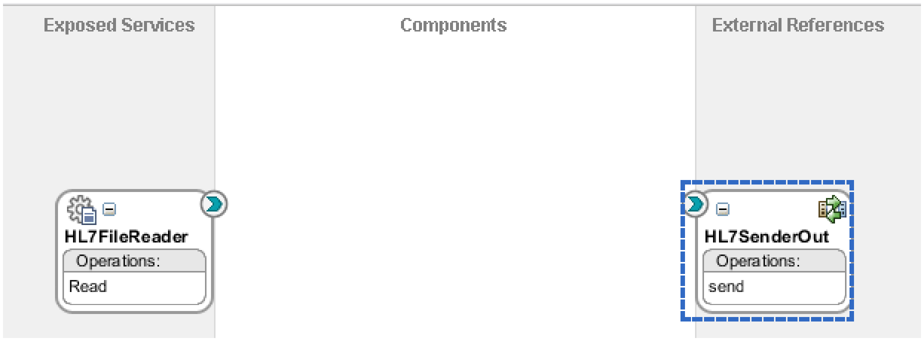
□ Native format translation is not required (Schema is Opaque)

URL: xsd/ADT_A01.xsd

Schema Element: ADT_A01

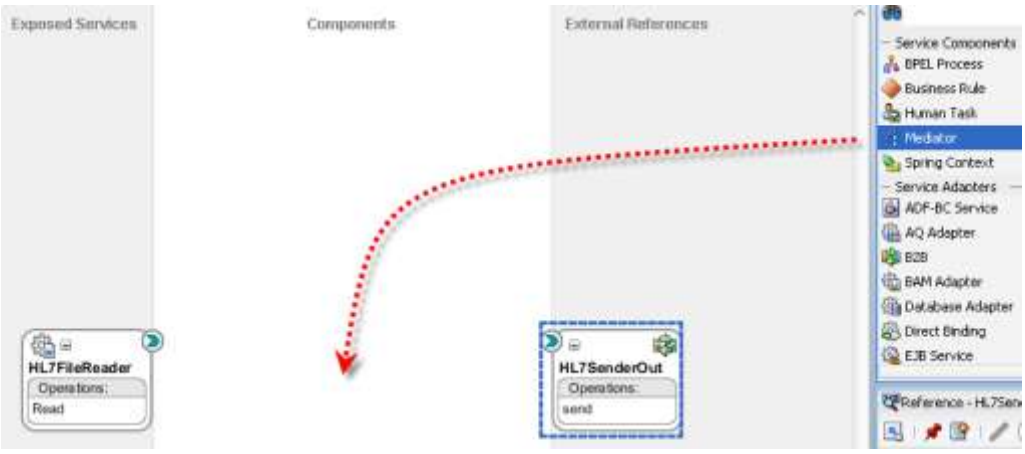
The structure of the HL7 message which will be read by the File Adapter will conform to this schema element.

Our composite has an inbound (File) and an outbound (B2B) adapter.

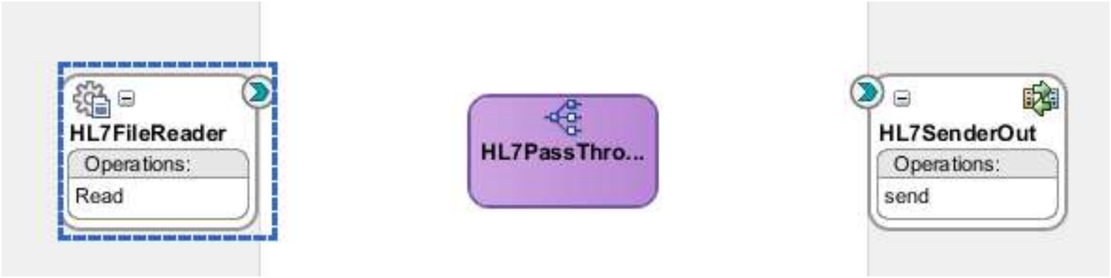


Now we need a piece of logic to copy the message from one adapter to the other, potentially transforming it if needed. We will not need to do much of transforming in this example since all we care about is to read a message from a file and send it out to the partner, but we will need to populate metadata required by the B2B infrastructure.

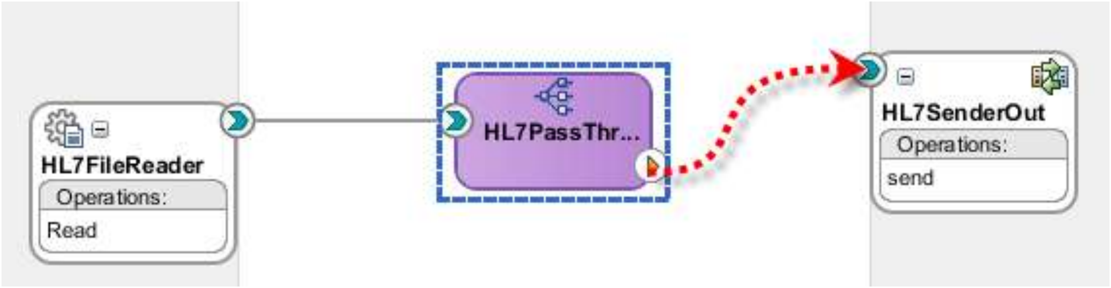
Let’s choose the Mediator to develop the simple logic we need. Drag the Mediator component from the Service Components palette onto the Components swim line.



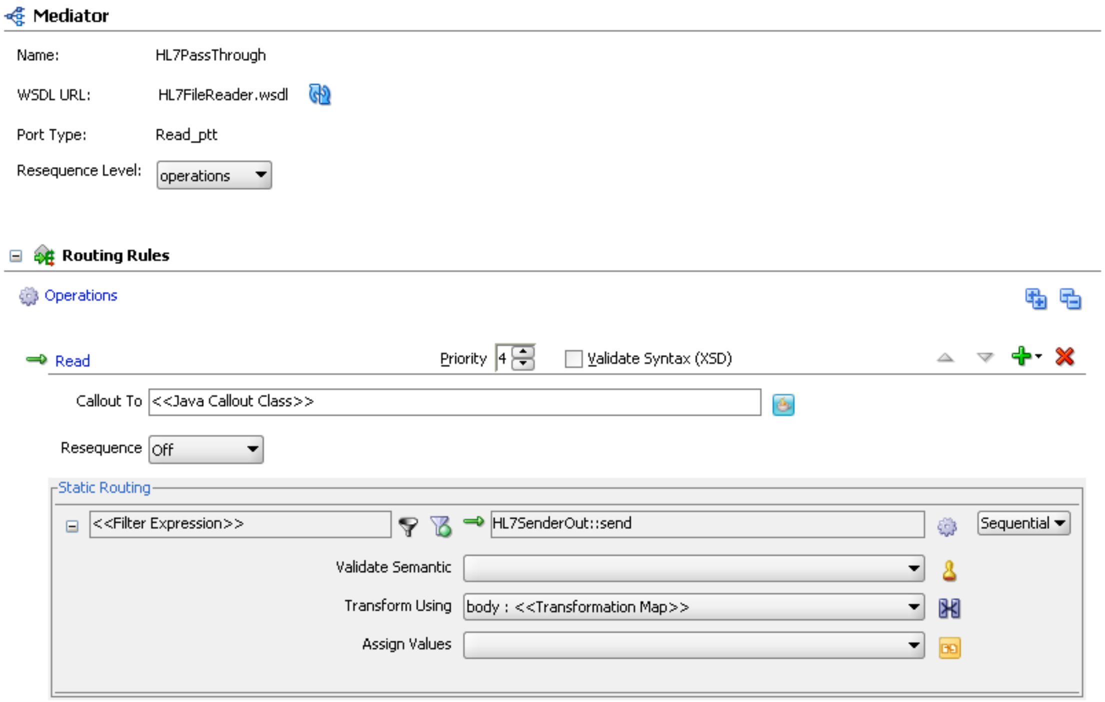
Name the component HL7PassThrough and click OK. The Mediator component appears is the composite.



“Wire” components together by dragging from the chevron icon of the component at the left to the “corresponding” chevron icon at the component to the right.



Double-click the Mediator component to open its plan and configure it.



Mediator

Name: HL7PassThrough

WSDL URL: HL7FileReader.wsdl

Port Type: Read_ptt

Resequence Level: operations

Routing Rules

Operations

Read — Priority 4 — □ Validate Syntax (XSD)

Callout To: <<Java Callout Class>>

Resequence: Off

Static Routing

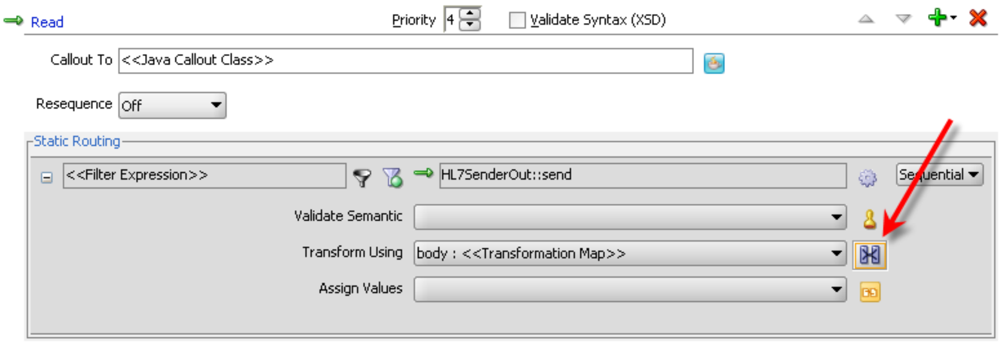
<<Filter Expression>> → HL7SenderOut::send — Sequential

Validate Semantic:

Transform Using: body : <<Transformation Map>>

Assign Values:

Click the Mapper icon to start defining the mapping between the File Adapter and the B2B Adapter.



Read — Priority 4 — □ Validate Syntax (XSD)

Callout To: <<Java Callout Class>>

Resequence: Off

Static Routing

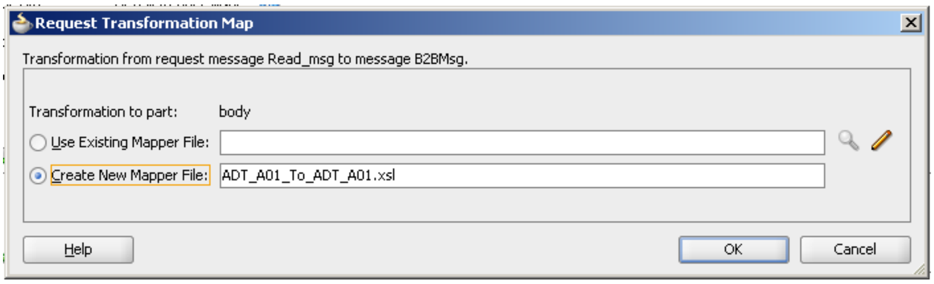
<<Filter Expression>> → HL7SenderOut::send — Sequential

Validate Semantic:

Transform Using: body : <<Transformation Map>>

Assign Values:

Choose “Create New Mapper File” and click OK.



Request Transformation Map

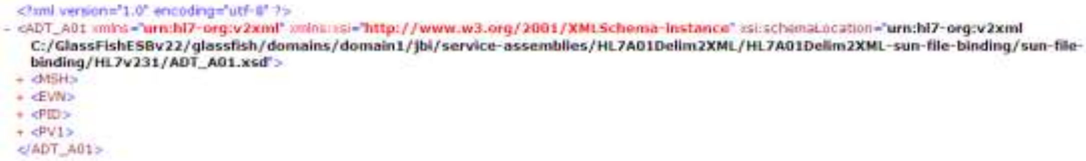
Transformation from request message Read_msg to message B2BMsg.

Transformation to part: body

○ Use Existing Mapper File:

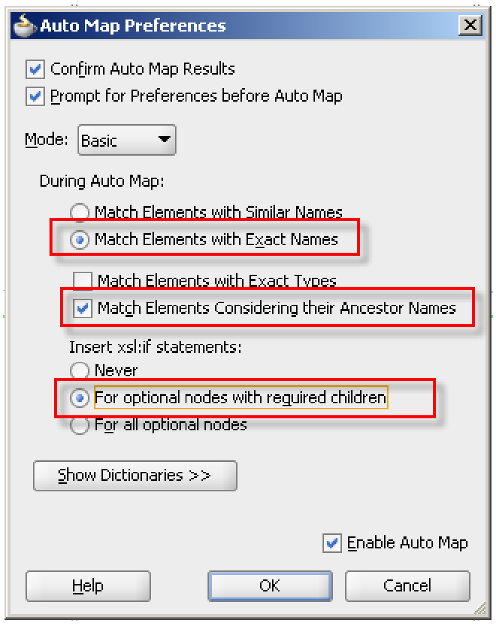
● Create New Mapper File: ADT_A01_To_ADT_A01.xsl

Based on your knowledge of the outgoing message, map segments and fields as appropriate. In the case of our sample message we have only MSH, ENV, PID and PV1 segments.



```
  <?xml version="1.0" encoding="utf-8" ?>
- <ADT_A01 xmlns="urn:hl7-org:v2xml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v2xml
    C:/GlassFishESBv22/glassfish/domains/domain1/jbi/service-assemblies/HL7A01Delim2XML/HL7A01Delim2XML-sun-file-binding/sun-file-
    binding/HL7v231/ADT_A01.xsd">
  + <MSH>
  + <EVN>
  + <PID>
  + <PV1>
  </ADT_A01>
```

Map MSH to MSH, EVN to EVN, PID to PID and PV1 to PV1. Ensure the confirmation dialogue box options are correctly set.



Auto Map Preferences

☑ Confirm Auto Map Results
☑ Prompt for Preferences before Auto Map

Mode: Basic

During Auto Map:

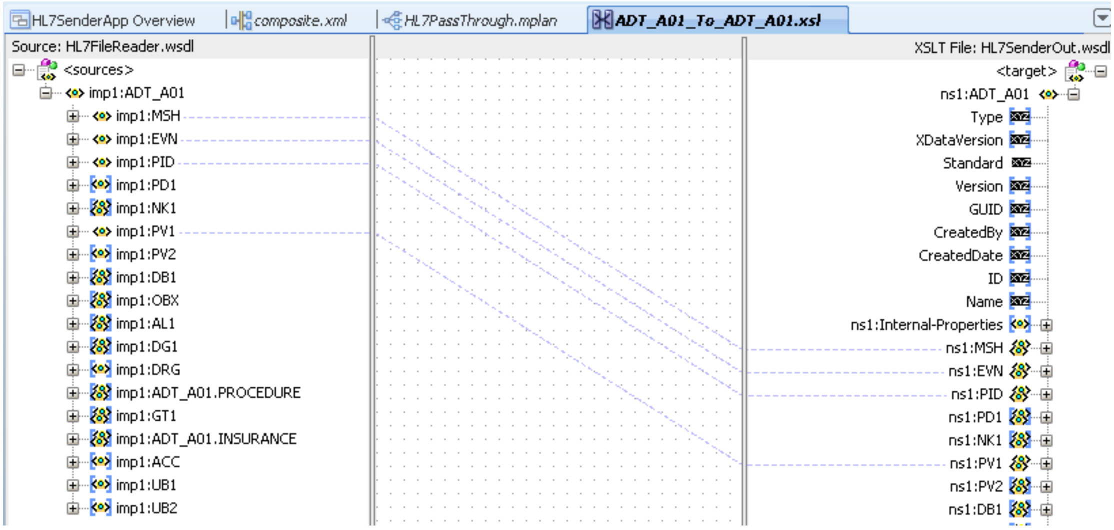
○ Match Elements with Similar Names
● Match Elements with Exact Names

□ Match Elements with Exact Types
☑ Match Elements Considering their Ancestor Names

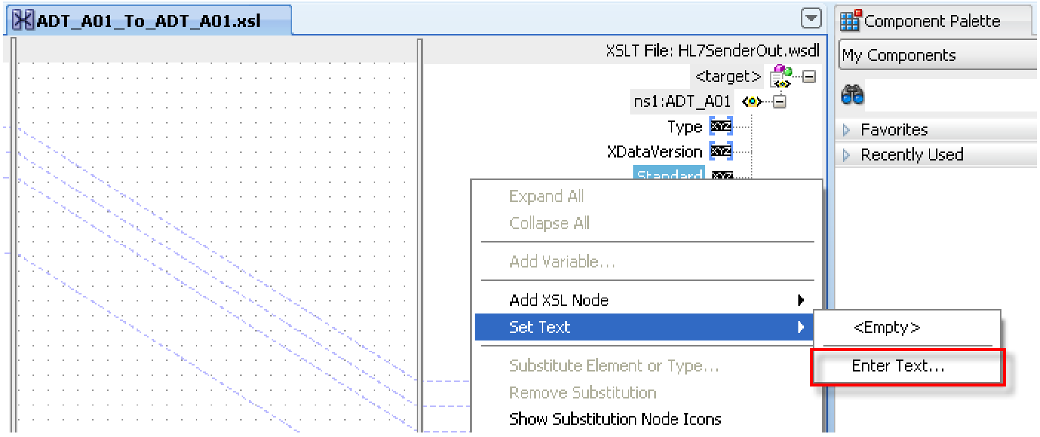
Insert xsl:if statements:

○ Never
● For optional nodes with required children
○ For all optional nodes

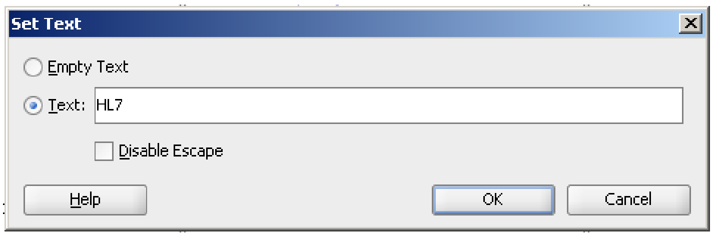
☑ Enable Auto Map



Right-click on the Standard attribute name in the right-hand structure, choose Set Text and Enter Text.



Provide text “HL7” and click OK.

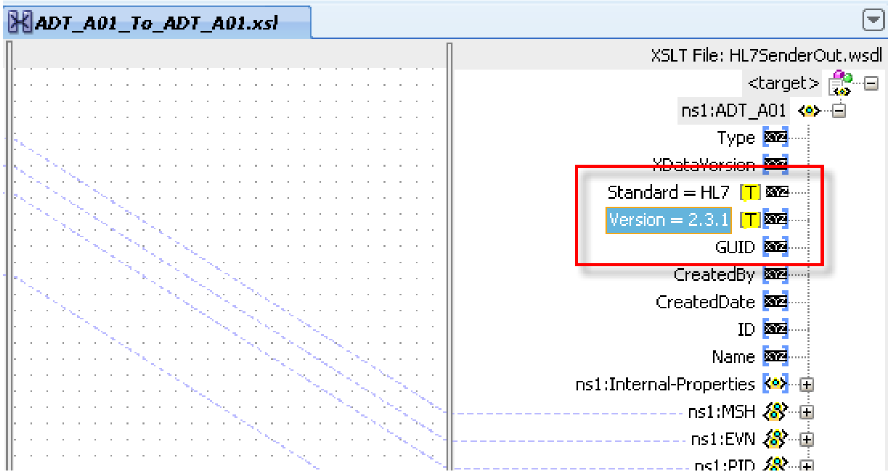

Set Text

○ Empty Text

● Text: HL7

□ Disable Escape

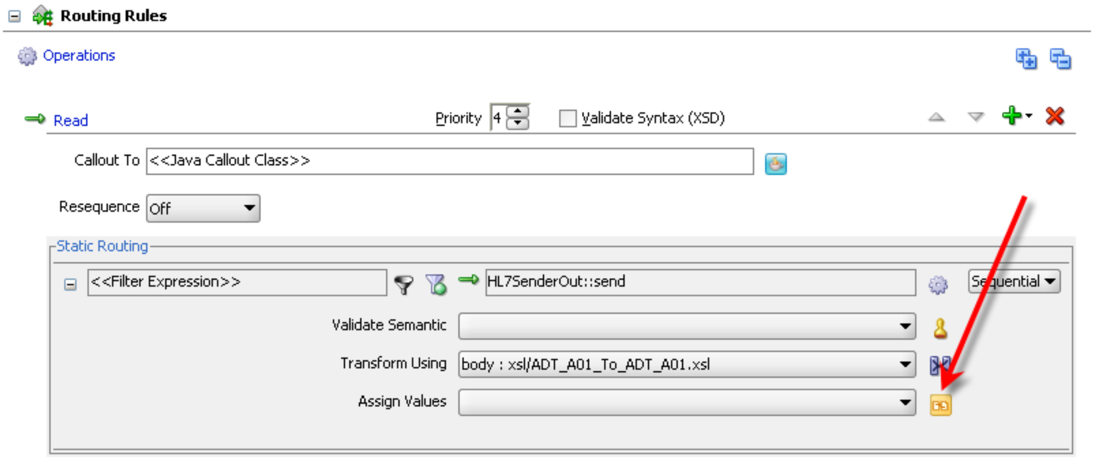
Set Text “2.3.1” for attribute Version.



Click the Save All button in the JDeveloper tool bar. Close the Mapper.

This produces a translation between the source message and the target message. Since the target message will be handed over to the B2B infrastructure for sending out we must specify the trading partner parameters so that the B2B infrastructure finds the correct partnership agreement and consequently sends the message to the correct partner.

Click the Assign Values button in the Mediator Plan.



Priority 4

□ Validate Syntax (XSD)

Callout To <<Java Callout Class>>

Resequence Off

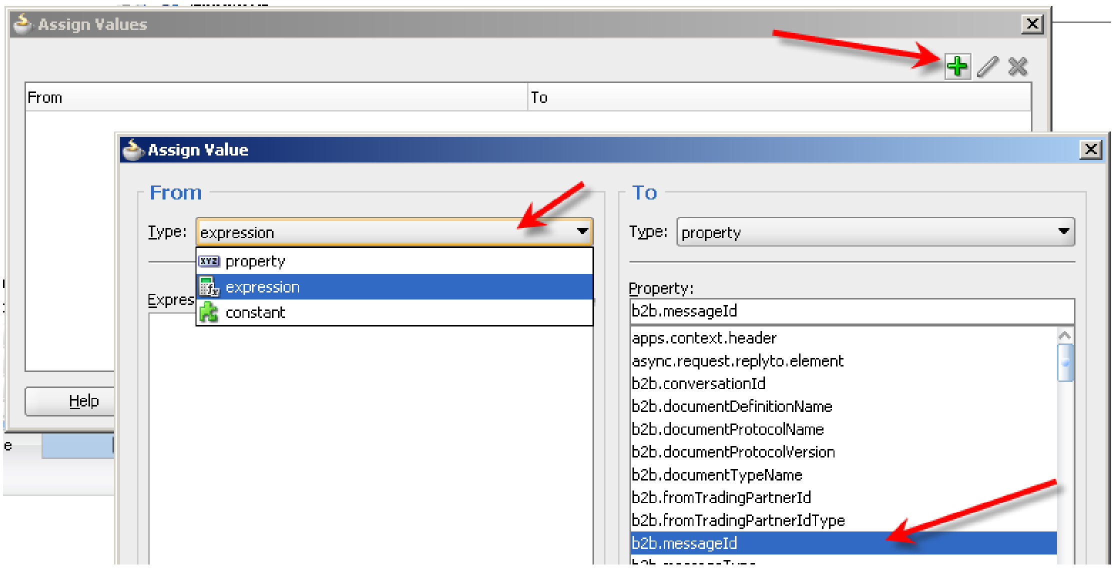
<<Filter Expression>> HL7SenderOut::send Sequential

Validate Semantic

Transform Using body : xsl/ADT_A01_To_ADT_A01.xsl

Assign Values

Click the large Plus sign to Add a New Value Assignment, select expression from the drop-down box on the From side and b2b.messageIf from the property list on the To side.



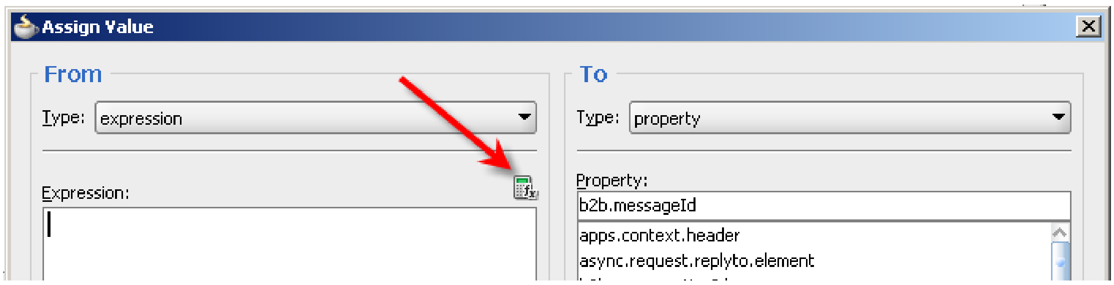
Type: expression

Type: property

Property: b2b.messageId

```
apps.context.header
async.request.replyto.element
b2b.conversationId
b2b.documentDefinitionName
b2b.documentProtocolName
b2b.documentProtocolVersion
b2b.documentTypeName
b2b.fromTradingPartnerId
b2b.fromTradingPartnerIdType
b2b.messageId
```

Click “Invoke Expression Builder” button.



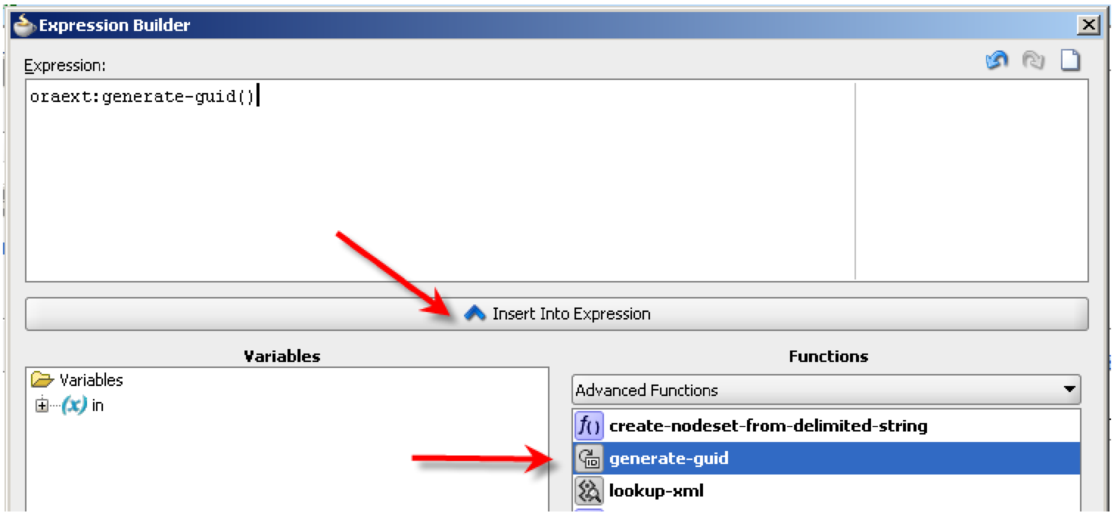
Type: expression

Type: property

Property: b2b.messageId

```
apps.context.header
async.request.replyto.element
```

Select generate-guid from the list of advanced functions and click the large/wide “Insert into expression” button. Click OK to close the Expression Builder and OK again to close the Assign Value dialogue box. This will provide unique message ID for each outgoing message.

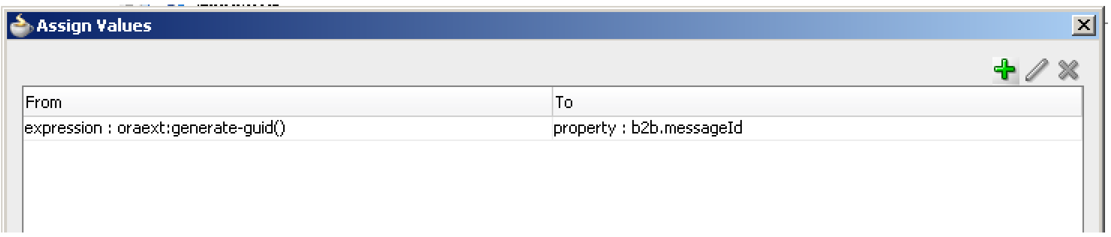


Expression:

```
oraext:generate-guid()
```

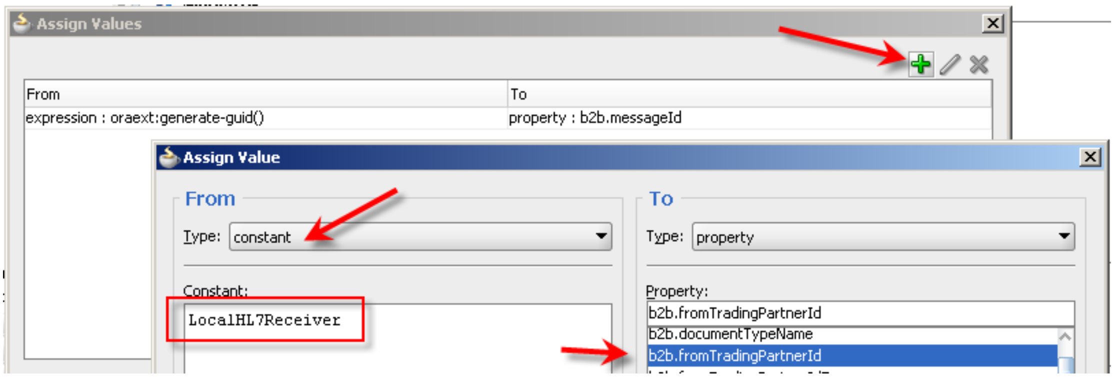
Functions: Advanced Functions

The expression gets added to the Assign Values list.



From	To
expression : oraext:generate-guid()	property : b2b.messageId

Click the large Plus sign to Add a New Value Assignment, select “constant” from the drop-down box on the From side and b2b.fromTradingPartnerId from the property list on the To side. Click OK to add the constant assignment.



From	To
expression : oraext:generate-guid()	property : b2b.messageId

From — Type: constant

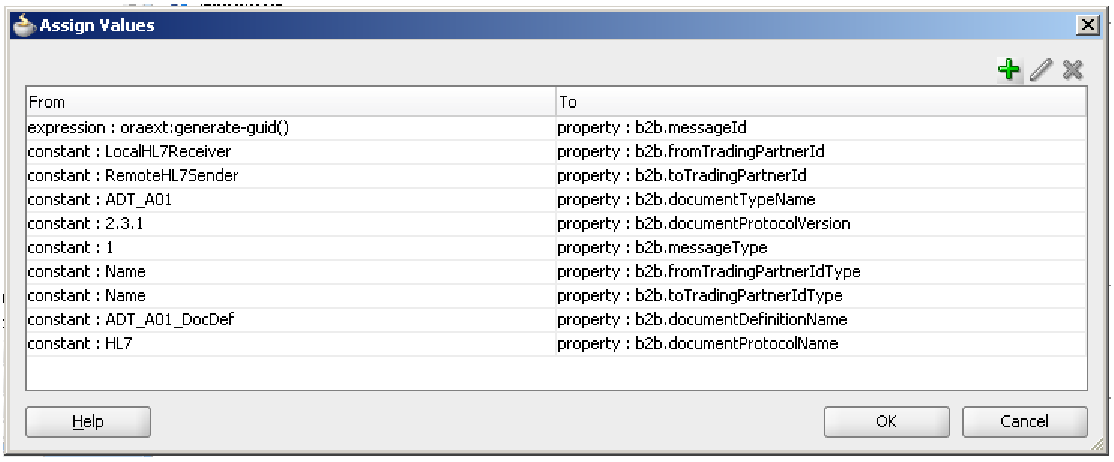
Constant: LocalHL7Receiver

To — Type: property

Property: b2b.fromTradingPartnerId

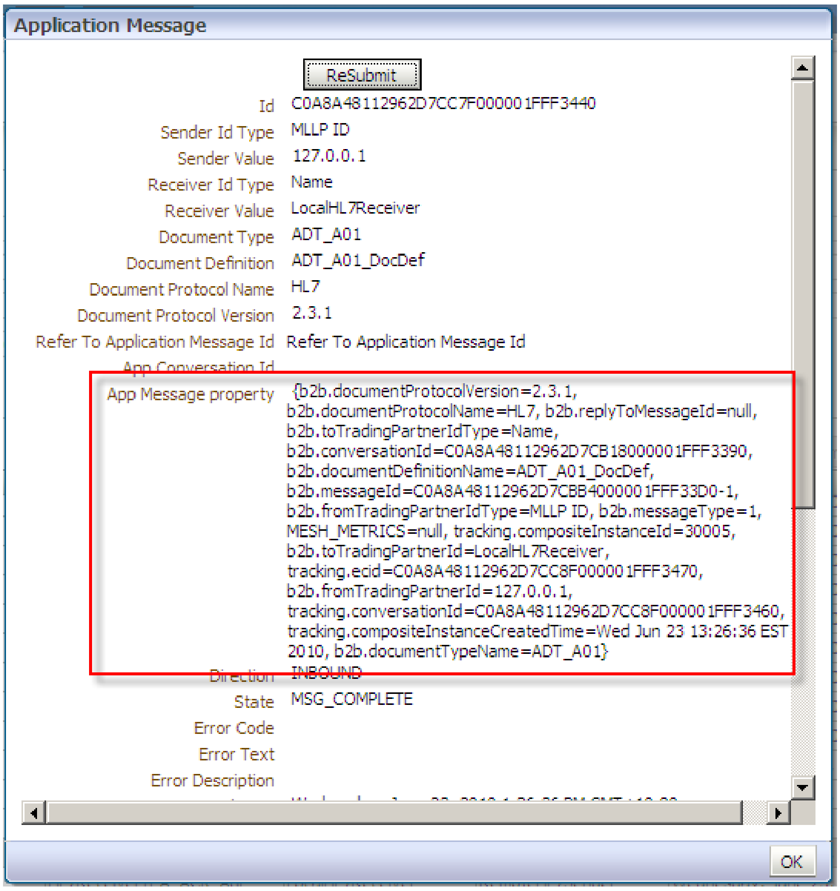
Repeat the steps to add constants in the following list.

Constant	Property
Name	b2b.fromTradingPartnerIdType
Name	b2b.toTradingPartnerIdType
RemoteHL7Sender	b2b.toTradingPartnerId
ADT_A01_DocDef	b2b.documentDefinitionName
ADT_A01	b2b.documentTypeName
HL7	b2b.documentProtocolName
2.3.1	b2b.documentProtocolVersion
1	b2b.messageType



From	To
expression : oraext:generate-guid()	property : b2b.messageId
constant : LocalHL7Receiver	property : b2b.fromTradingPartnerId
constant : RemoteHL7Sender	property : b2b.toTradingPartnerId
constant : ADT_A01	property : b2b.documentTypeName
constant : 2.3.1	property : b2b.documentProtocolVersion
constant : 1	property : b2b.messageType
constant : Name	property : b2b.fromTradingPartnerIdType
constant : Name	property : b2b.toTradingPartnerIdType
constant : ADT_A01_DocDef	property : b2b.documentDefinitionName
constant : HL7	property : b2b.documentProtocolName

It should be clear where these values come from. If in doubt, go to the B2B Message Tracker and open any successfully received Application Message.



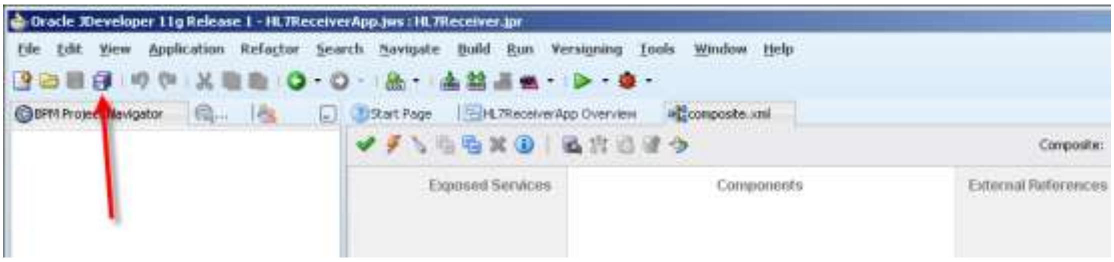
Application Message

ReSubmit

Id	C0A8A48112962D7CC7F000001FFF3440
Sender Id Type	MLLP ID
Sender Value	127.0.0.1
Receiver Id Type	Name
Receiver Value	LocalHL7Receiver
Document Type	ADT_A01
Document Definition	ADT_A01_DocDef
Document Protocol Name	HL7
Document Protocol Version	2.3.1
Refer To Application Message Id	Refer To Application Message Id
App Conversation Id	
App Message property	{b2b.documentProtocolVersion=2.3.1, b2b.documentProtocolName=HL7, b2b.replyToMessageId=null, b2b.toTradingPartnerIdType=Name, b2b.conversationId=C0A8A48112962D7CB18000001FFF3390, b2b.documentDefinitionName=ADT_A01_DocDef, b2b.messageId=C0A8A48112962D7CBB400001FFF33D0-1, b2b.fromTradingPartnerIdType=MLLP ID, b2b.messageType=1, MESH_METRICS=null, tracking.compositeInstanceId=30005, b2b.toTradingPartnerId=LocalHL7Receiver, tracking.ecid=C0A8A48112962D7CC8F000001FFF3470, b2b.fromTradingPartnerId=127.0.0.1, tracking.conversationId=C0A8A48112962D7CC8F000001FFF3460, tracking.compositeInstanceCreatedTime=Wed Jun 23 13:26:36 EST 2010, b2b.documentTypeName=ADT_A01}
Direction	INBOUND
State	MSG_COMPLETE
Error Code	
Error Text	
Error Description	

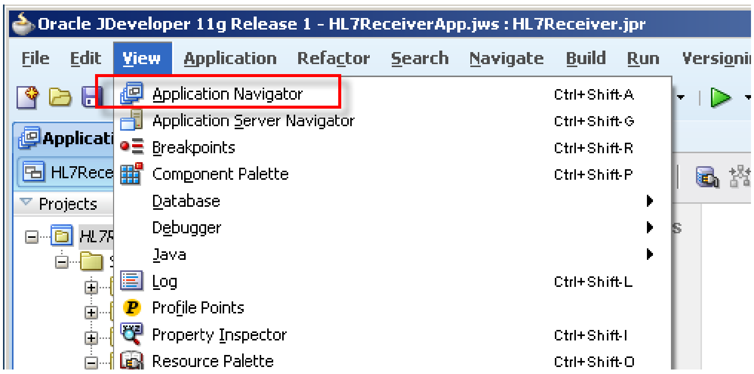
OK

Click Save All toolbar button, then close the Mediator Plan and Mapper Tabs.

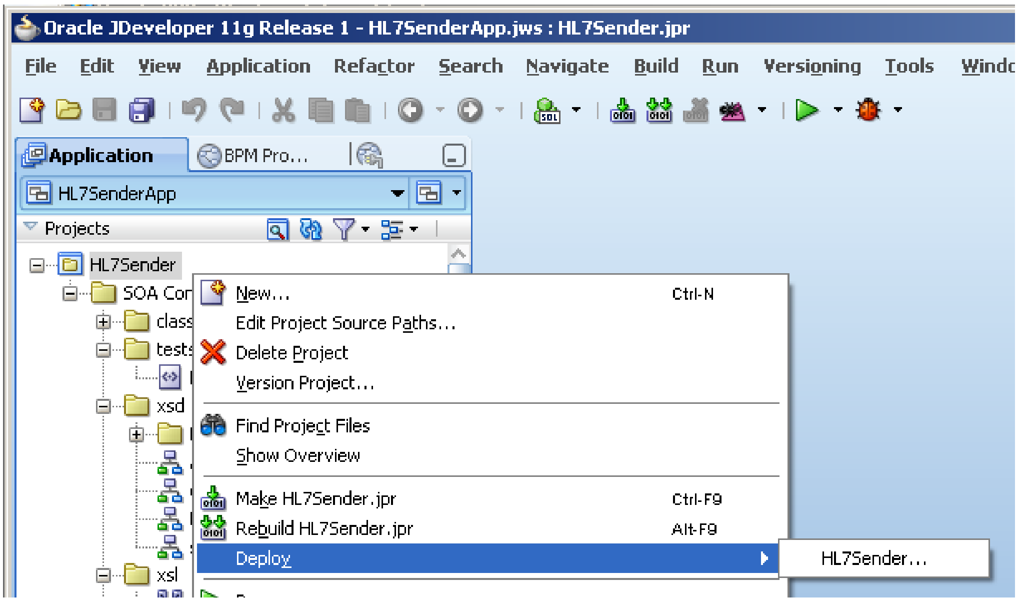

Oracle JDeveloper 11g Release 1 - HL7ReceiverApp.jws : HL7Receiver.jpr
File Edit View Application Refactor Search Navigate Build Run Versioning Tools Window Help
BPM Project Navigator
Start Page
HL7ReceiverApp Overview
composite.xml
Composite:
Exposed Services
Components
External References

We are ready to build and deploy this application.

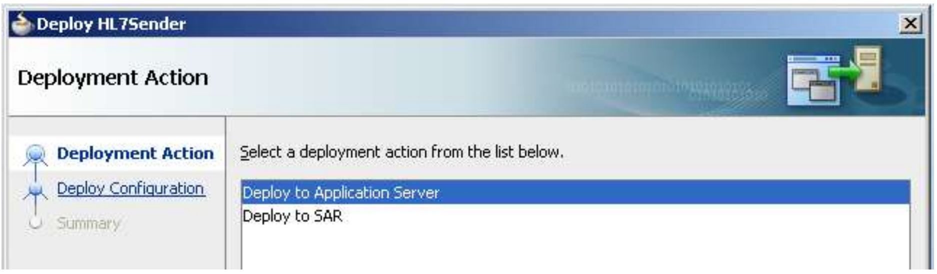
Make sure the Application Navigator panel is visible.


Oracle JDeveloper 11g Release 1 - HL7ReceiverApp.jws : HL7Receiver.jpr
File Edit View Application Refactor Search Navigate Build Run Versioni
Applicati
HL7Rece
Projects
HL7R
Application Navigator Ctrl+Shift-A
Application Server Navigator Ctrl+Shift-G
Breakpoints Ctrl+Shift-R
Component Palette Ctrl+Shift-P
Database
Debugger
Java
Log Ctrl+Shift-L
Profile Points
Property Inspector Ctrl+Shift-I
Resource Palette Ctrl+Shift-O

Right-click on the name of the project HL7Receiver, choose Deploy and HL7Sender…


Oracle JDeveloper 11g Release 1 - HL7SenderApp.jws : HL7Sender.jpr
File Edit View Application Refactor Search Navigate Build Run Versioning Tools Windo
Application
BPM Pro…
HL7SenderApp
Projects
HL7Sender
SOA Cor
class
tests
xsd
xsl
New… Ctrl-N
Edit Project Source Paths…
Delete Project
Version Project…
Find Project Files
Show Overview
Make HL7Sender.jpr Ctrl-F9
Rebuild HL7Sender.jpr Alt-F9
Deploy
HL7Sender…

Accept default and click Next.

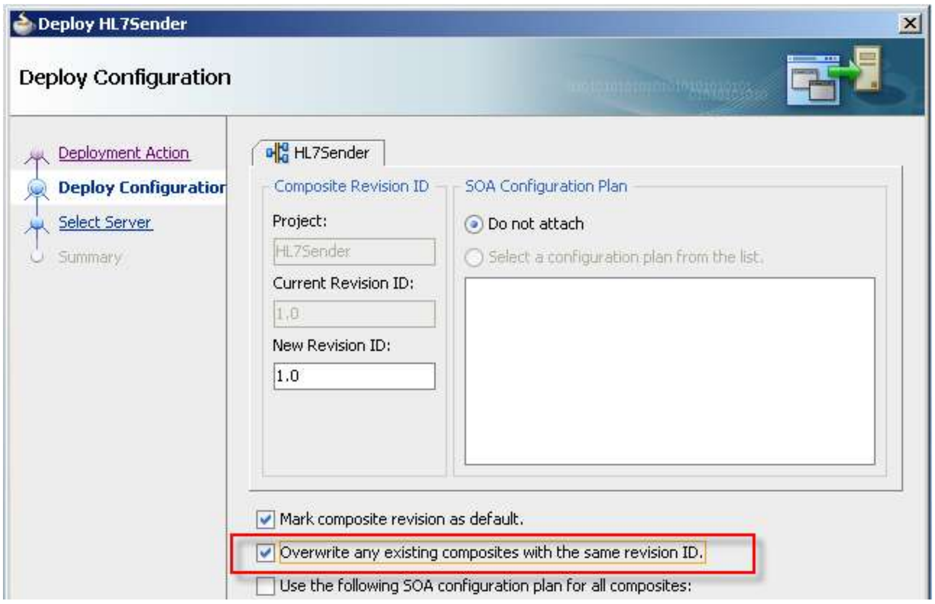


Select a deployment action from the list below.

Deploy to Application Server

Deploy to SAR

Check the “Overwrite any existing composites with the same revision ID” and click Next.



Composite Revision ID

Project: HL7Sender

Current Revision ID: 1.0

New Revision ID: 1.0

SOA Configuration Plan

● Do not attach

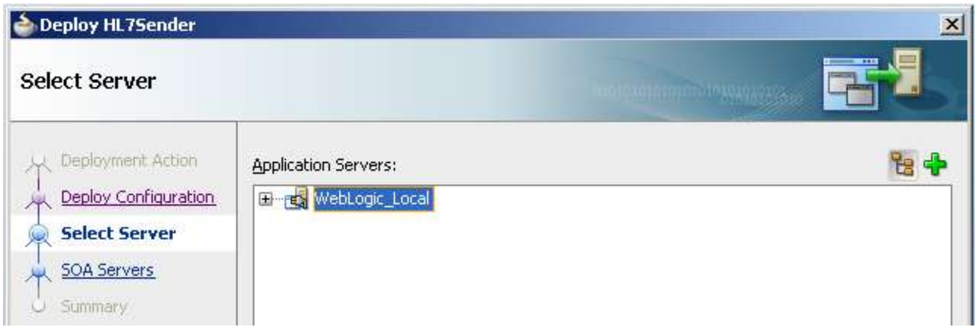
○ Select a configuration plan from the list.

☑ Mark composite revision as default.

☑ Overwrite any existing composites with the same revision ID.

□ Use the following SOA configuration plan for all composites:

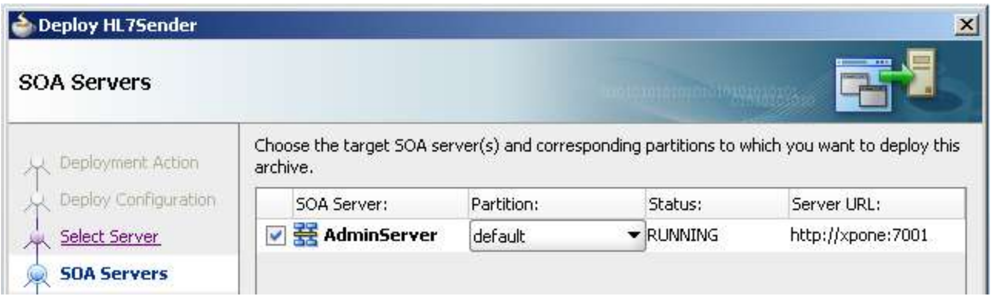
Accept the default WebLogic_Local application server and click Next.



Application Servers:

WebLogic_Local

Accept the default SOA Server and click Next.

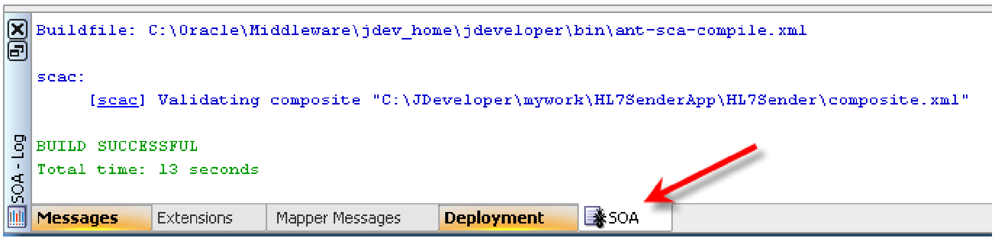


Choose the target SOA server(s) and corresponding partitions to which you want to deploy this archive.

	SOA Server:	Partition:	Status:	Server URL:
☑	AdminServer	default	RUNNING	http://xpone:7001

Click Finish.

Observer SOA Log window.

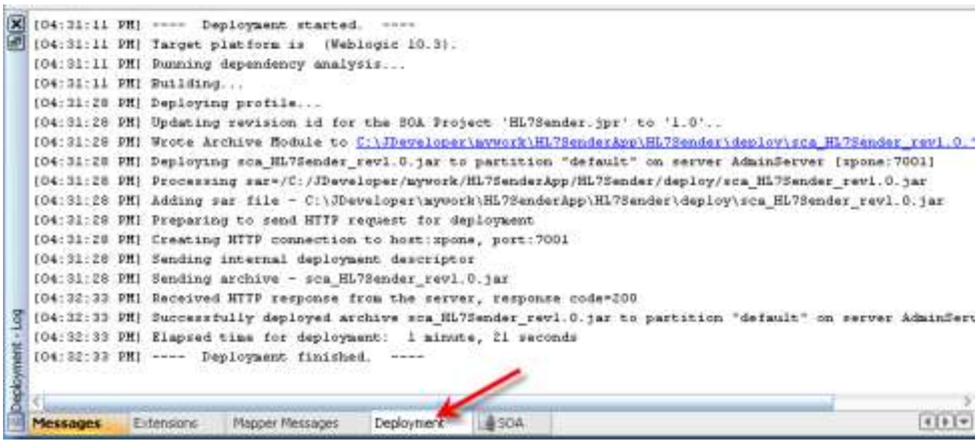


```
Buildfile: C:\Oracle\Middleware\jdev_home\jdeveloper\bin\ant-sca-compile.xml

scac:
     [scac] Validating composite "C:\JDeveloper\mywork\HL7SenderApp\HL7Sender\composite.xml"

BUILD SUCCESSFUL
Total time: 13 seconds
```

Observe Deployment Log window.



```
[04:31:11 PM] ----  Deployment started.  ----
[04:31:11 PM] Target platform is  (Weblogic 10.3).
[04:31:11 PM] Running dependency analysis...
[04:31:11 PM] Building...
[04:31:28 PM] Deploying profile...
[04:31:28 PM] Updating revision id for the SOA Project 'HL7Sender.jpr' to '1.0'..
[04:31:28 PM] Wrote Archive Module to C:\JDeveloper\mywork\HL7SenderApp\HL7Sender\deploy\sca_HL7Sender_rev1.0.'
[04:31:28 PM] Deploying sca_HL7Sender_rev1.0.jar to partition "default" on server AdminServer [xpone:7001]
[04:31:28 PM] Processing sar=/C:/JDeveloper/mywork/HL7SenderApp/HL7Sender/deploy/sca_HL7Sender_rev1.0.jar
[04:31:28 PM] Adding sar file - C:\JDeveloper\mywork\HL7SenderApp\HL7Sender\deploy\sca_HL7Sender_rev1.0.jar
[04:31:28 PM] Preparing to send HTTP request for deployment
[04:31:28 PM] Creating HTTP connection to host:xpone, port:7001
[04:31:28 PM] Sending internal deployment descriptor
[04:31:28 PM] Sending archive - sca_HL7Sender_rev1.0.jar
[04:32:33 PM] Received HTTP response from the server, response code=200
[04:32:33 PM] Successfully deployed archive sca_HL7Sender_rev1.0.jar to partition "default" on server AdminServ
[04:32:33 PM] Elapsed time for deployment:  1 minute, 21 seconds
[04:32:33 PM] ----  Deployment finished.  ----
```

Click Save All toolbar button and exit JDeveloper Studio.

Exercise HL7 Outbound solution

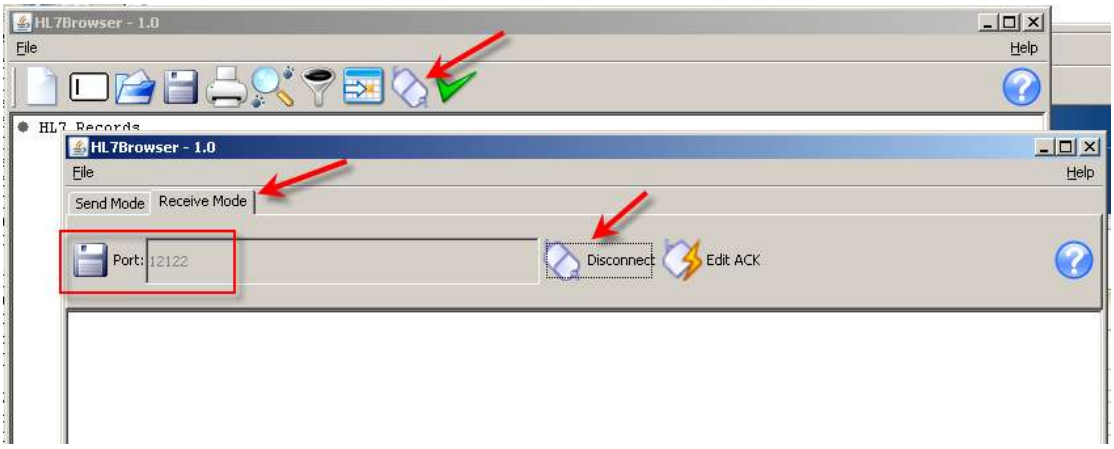
To see the interaction and message exchange we need to increase logging level for the appropriate parts of the infrastructure. It is assumed that this has already been done as discussed at this point in the previous article.

Start the B2B Trading Partner Manager Web Console, http://localhost:7001/b2b, log in and click the Reports link. Note that there are no messages shown unless you run messages through a different B2b solution to the one on which we are currently working.

Open a command window and run the HL7Browser:

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

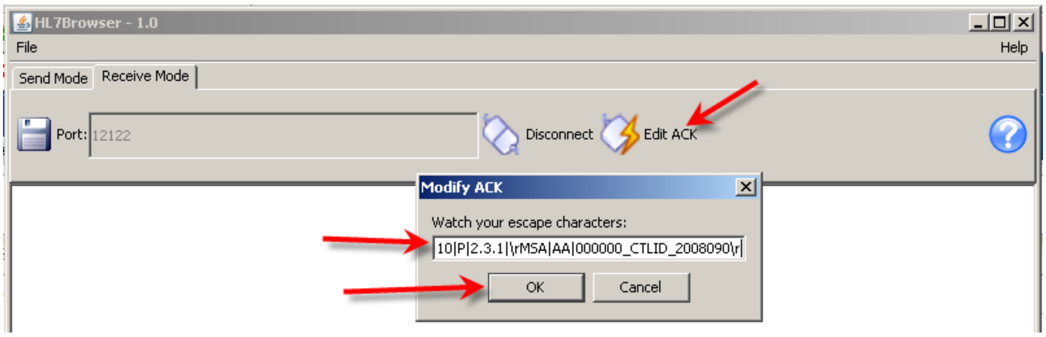
Click the “Run network utility” button, click the “Receive Mode” Tab, specify 12122 as Port.



Port: 12122

Click the Edit ACK button and paste the following text into the text box, making sure it is all on a single line:

```
MSH|^~\&|SystemA|HosA|PI|MDM|20100623||ACK^A01|1010|P|2.3.1|\rMSA|AA|000000_CTLID_2008
090\r
```



Port: 12122

Watch your escape characters:

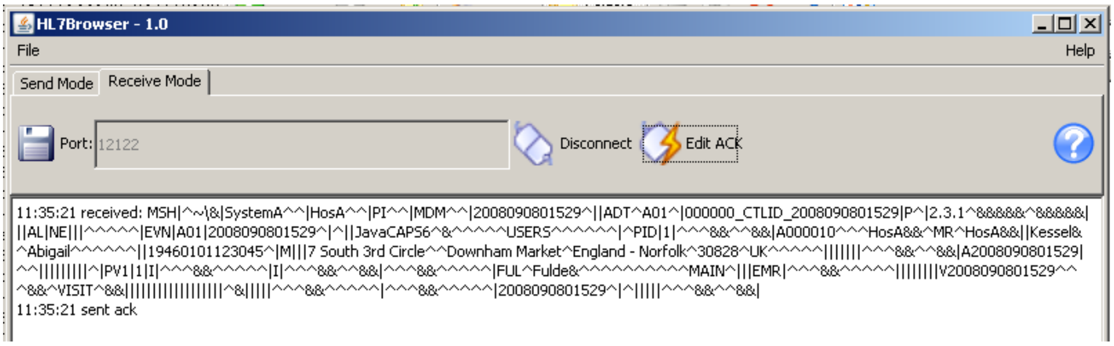
10|P|2.3.1|\rMSA|AA|000000_CTLID_2008090\r|

Click Connect. The HL7 receiver is ready to receive messages from the SOA Suite HL7 sender.

Be sure to move the c:\hl7\adt\data\sources directory out of the data directory to, for example, c:\hl7\adt directory. The File adapter scans directories recursively and would process all the XML transaction files that match the pattern in the sources directory. This is definitely not what we want at this point in time.

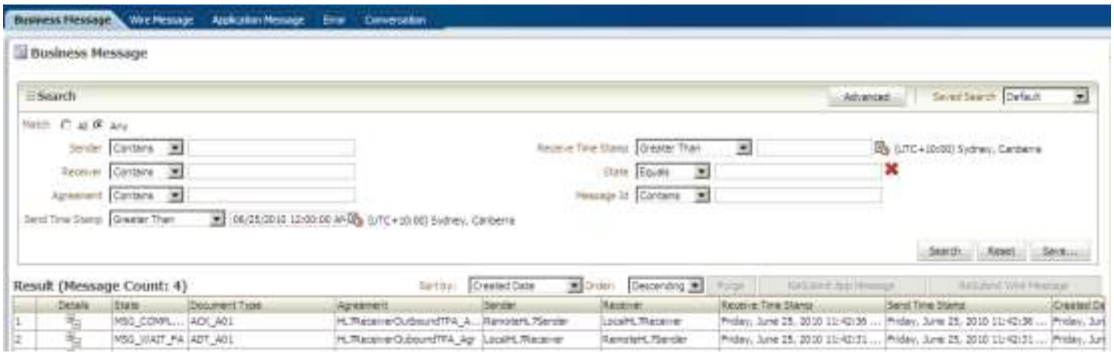
Locate the ADT A01 XML transaction file, C:\hl7\adt\data\sources\ADT_A01_output_1.xml, and copy it to the C:\hl7\adt\data directory.

In dues course the file is picked up and processed. The HL7 Browser Receive Mode window shows the incoming message and indicates that the ACK has been sent,



```
11:35:21 received: 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^30828^UK^^^^^|||||||^^^&&^^&&|A2008090801529|
^^|||||||||^|PV1|1|I|^^^&&^^^^^|I|^^^&&^^&&|^^^&&^^^^^|FUL^Fulde&^^^^^^^^^^^MAIN^|||EMR|^^^&&^^^^^||||||||V2008090801529^^
^&&^VISIT^&&|||||||||||||||||^&|||||^^^&&^^^^^|^^^&&^^^^^|2008090801529^|^|||||^^^&&^^&&|
11:35:21 sent ack
```

Switch to the B2B Message Tracker (B2B Trading Partner Manager Web Console, Reports Tab) and search for recent messages.



Sender: Contains
Receiver: Contains
Agreement: Contains
Send Time Stamp: Greater Than 06/25/2010 12:00:00 AM (UTC+10:00) Sydney, Canberra
Receive Time Stamp: Greater Than (UTC+10:00) Sydney, Canberra
State: Equals
Message Id: Contains

Result (Message Count: 4)

Sort by: Created Date Order: Descending

	Details	State	Document Type	Agreement	Sender	Receiver	Receive Time Stamp	Send Time Stamp	Created Da
1		MSG_COMPL...	ACK_A01	HL7ReceiverOutboundTPA_A...	RemoteHL7Sender	LocalHL7Receiver	Friday, June 25, 2010 11:42:36 ...	Friday, June 25, 2010 11:42:36 ...	Friday, Jun
2		MSG_WAIT_FA	ADT_A01	HL7ReceiverOutboundTPA_Agr	LocalHL7Receiver	RemoteHL7Sender	Friday, June 25, 2010 11:42:31 ...	Friday, June 25, 2010 11:42:31 ...	Friday, Jun

Explore Wire messages, Application Messages and Conversations.

This concludes the exercise.

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 Sender project for ADT A01 messages was developed and exercised. Message Tracker was used to view messages, message states and messaging performance.

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