

***Tutorial #002 :  
Building HL7 Transactions with  
Oracle B2B 11g***

---

## Contents

<i>Introduction</i> .....	4
<i>Building HL7 transactions in Oracle B2B</i> .....	5
<b>Configuring HL7 protocol</b> .....	5
Create Document Definitions .....	5
Create Document Protocols Version.....	5
Create Document Types.....	6
Create Document Definition .....	7
Add Document definitions for Initiator .....	9
Add Documents definitions for Responder.....	9
<b>Outbound HL7</b> .....	11
<b>Inbound HL7</b> .....	11
Agreement Identification .....	11
<b>Various Types of Acknowledgments supported in B2B</b> .....	12
ACK handled by B2B .....	12
Document level Acknowledgment / Functional Acknowledgement .....	12
Immediate Ack.....	13
Generic Ack .....	13
<b>Exchange layer</b> .....	14
MLLP .....	14
Generic Exchange .....	14
<i>Advanced HL7/MLLP features</i> .....	15
<b>Generic Message Type (Without Trigger event)</b> .....	15
<b>MSH.15 for ACK generation</b> .....	15
<b>Map control number to ACK</b> .....	15
<b>Broadcast</b> .....	15
<b>Dynamic IP</b> .....	16
<b>Sequencing</b> .....	16
Outbound message sequencing.....	16
Inbound message sequencing.....	17
Sequencing without Dispatching .....	17
<b>Generic Support for TCP</b> .....	18
<b>Custom header for callout</b> .....	19
<b>Overriding the connection mode</b> .....	20
<i>Best Practices</i> .....	20
<i>FAQ</i> .....	20
<i>Reference</i> :.....	21



---

## Introduction

HL7 (Health Level Seven) is one of the American National Standards Institute (ANSI) - accredited Standards Developing Organizations (SDOs). It is operating in the health care arena for exchanging clinical and administrative data between medical applications. In general, HL7 is a standard that is being used to exchange the data in the health care world.

### Purpose

The main purpose of this technote is to describe how to model HL7 transactions in Oracle B2B 11g.  
After reading this technote, user should be able to model different HL7 transactions in Oracle B2B 11g.

### Audience

HL7 technote is intended for B2B users / HL7 users who want to exchange HL7 messages and would like to understand various best practices.

### Prerequisite

Basic knowledge of HL7.

### Assumptions

This document is based on Oracle Fusion Middleware B2B 11g.

---

## Building HL7 transactions in Oracle B2B

Oracle B2B supports all the message type of version HL7.

Below is the supported exchange mechanism

- MLLP
- Generic

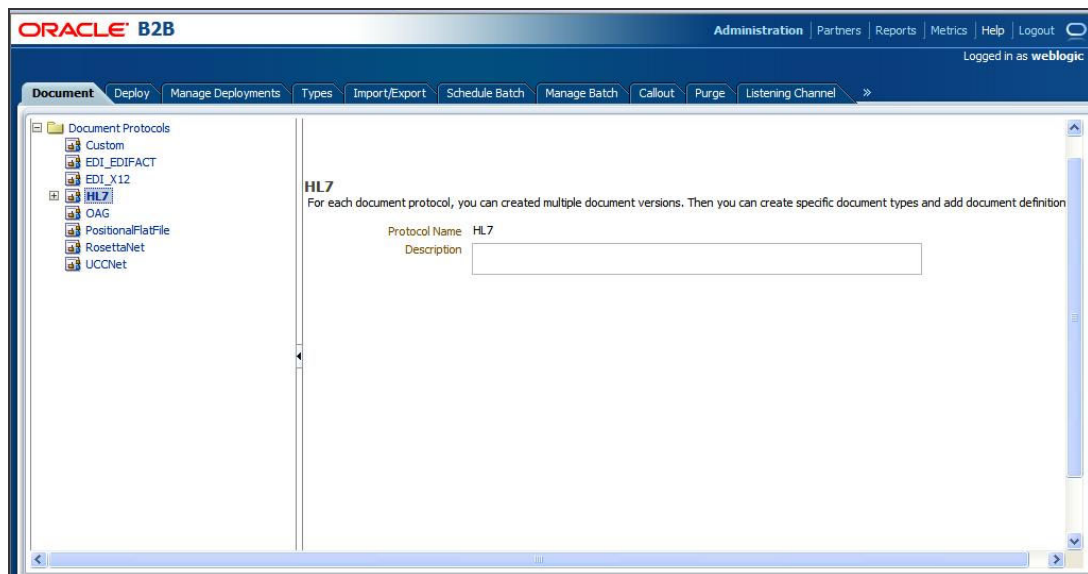
---

### Configuring HL7 protocol

#### Create Document Definitions

The required document type will be created by admin. Trading partners can make use of the same as a supported document definition.

→ Administration → Select Document → Select Document Protocols → Select HL7

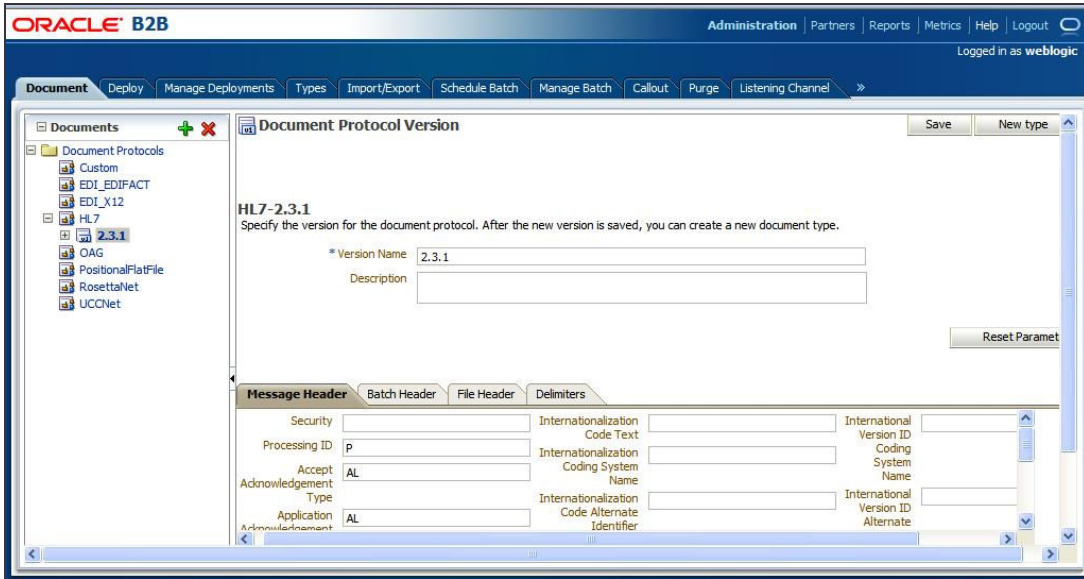


#### Create Document Protocols Version

→ Provide the **Version name (2.3.1)**

Field	Value
Version Name	2.3.1

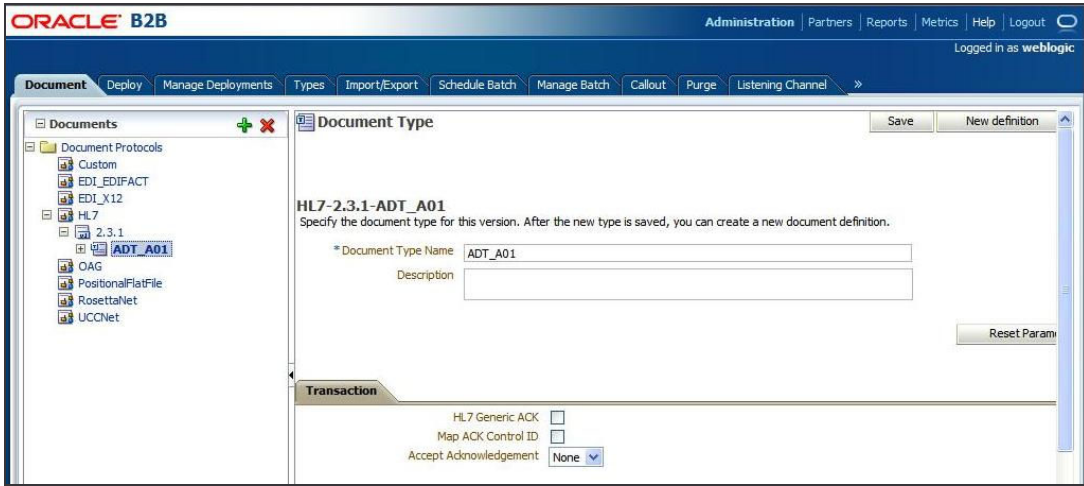
- Save the newly created Document Protocol version
- Observe the confirmation message after save activity



⚡ **Note:** Depending on the requirement you can edit the values in the remaining fields or you can choose the default. This scenario uses messages with only “MSH” segment hence choose the default values in File and Batch headers

## Create Document Types

### Create Document Type: ADT\_A01



- Select newly created **Document Protocol Version (2.3.1)** in Document Protocol pane
- Click **(+)** icon to Add Document Type
- Provide the **Document Type Name (ADT\_A01)**

Enter the following information

Field	Value
Document Type Name	ADT_A01
Document Type Description	

→ Optionally, if the ACK message is not based on the trigger event such as A01, A02, **select the option for Check HL7 Generic ACK.**

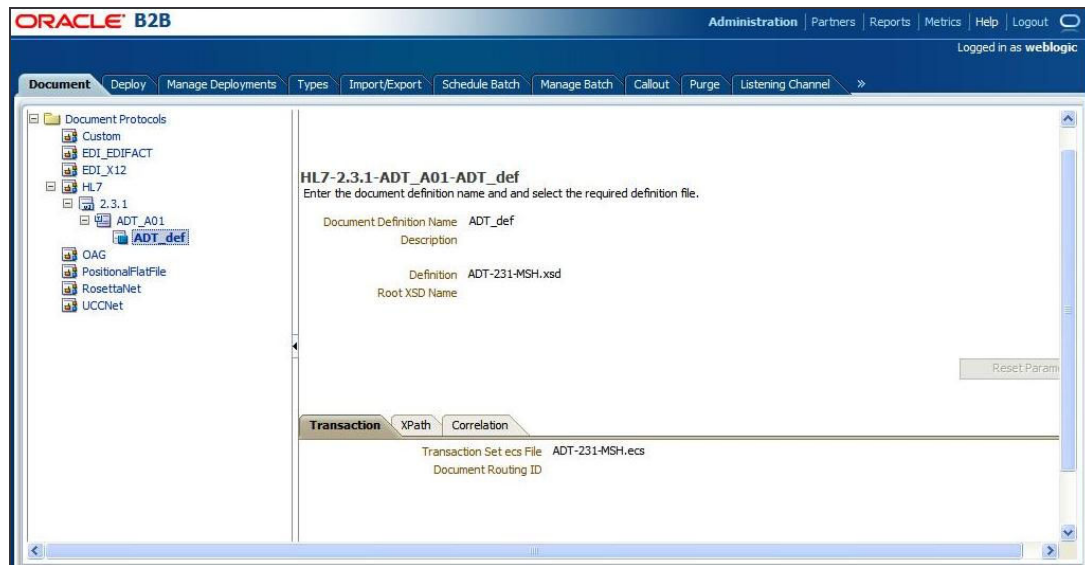
→ Save the newly created Document Type

## Create Document Definition

→ Select newly created **Document Type (ADT\_A01)** in Document Protocol pane

→ Click **(+)** icon to Add Document Definition

→ Provide the **Document Definition Name**



→ You have the option of providing an xsd corresponding to the transaction set ecs , which can be used in the back end application such as BPEL for required transformation to edifecs.xml.

→ Provide the **Transaction set ecs file**, under “**Transaction**” tab. This is mandatory. It is best practice to create the ecs file using the document editor 6.6.1 However, both B2B and document is backward compatible.

Field	Value
Document Definition Name	ADT_def – a name of user choice
Definition	ADT-231-MSH.xsd The xsd file can be created using the Spec Builder 6.6.1 by exporting the ecs file in Oracle B2B 2.0 format
Transaction Set File	ADT-231-MSH.ecs

**⚡Note:** The steps to create the ecs file are mentioned in the document <TODO : Mention the location>

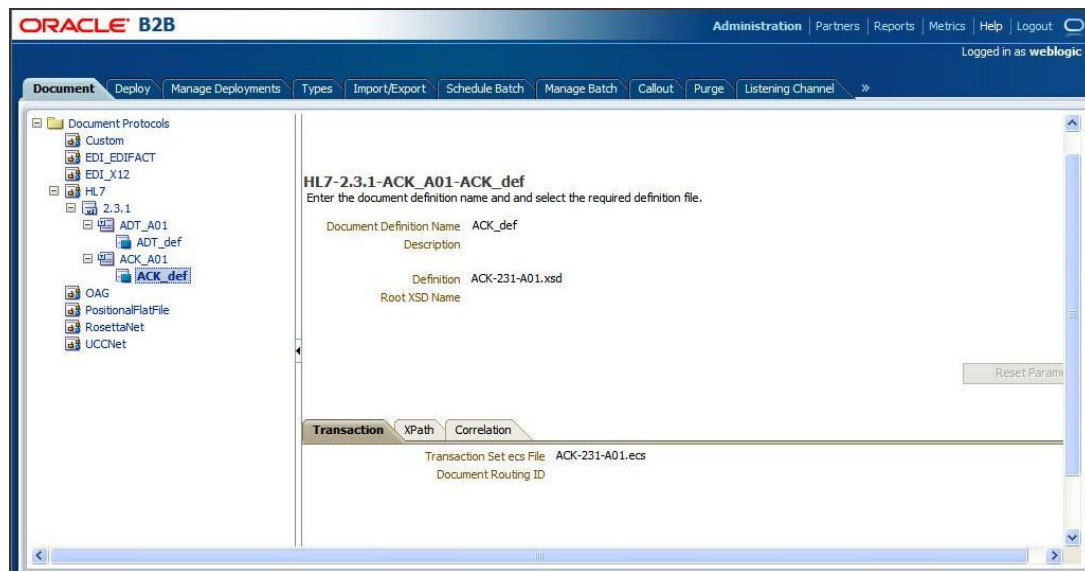
✈ **Note:** If it is required to process the ACK message create another document definition for the document type ACK\_A01 with the following information

Field	Value
Document Definition Name	ACK_def – a name of user choice

→ Observe the xsd and ecs files are already being uploaded. As this is a common document type, it has been made part of default configuration. However, it is also possible to upload the ecs and xsd files for ACK\_A01 by clicking  button with below mentioned

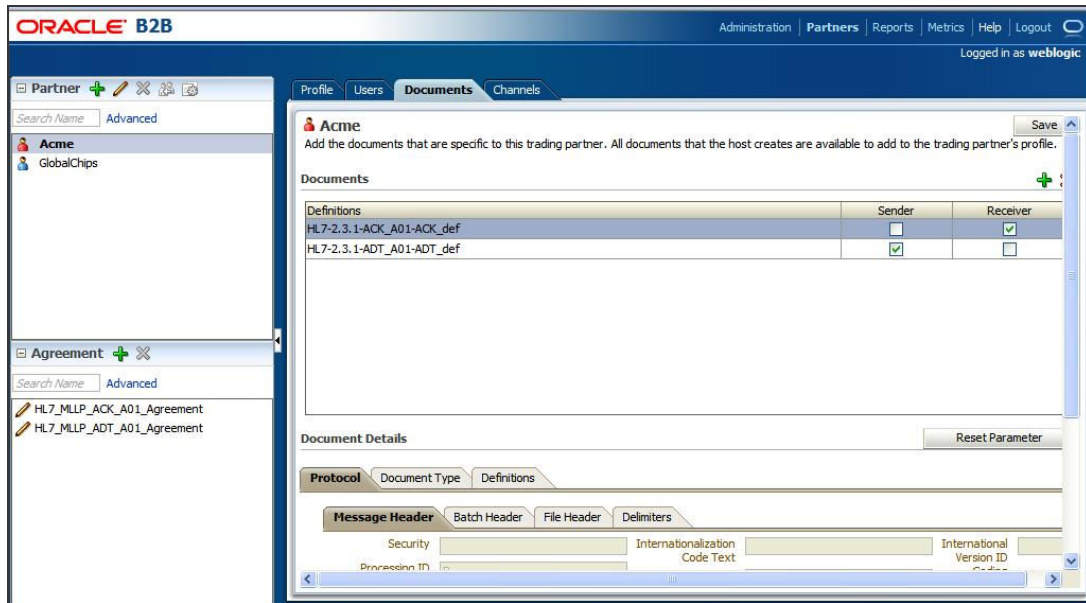
Field	Value
Definition	ACK-231-MSH.xsd The xsd file can be created using the Spec Builder 6.6.1 by exporting the ecs file in Oracle B2B 2.0 format
Transaction set ecs file	ACK-231-MSH.ecs

→ Save the newly created Document Definition



## Add Document definitions for Initiator

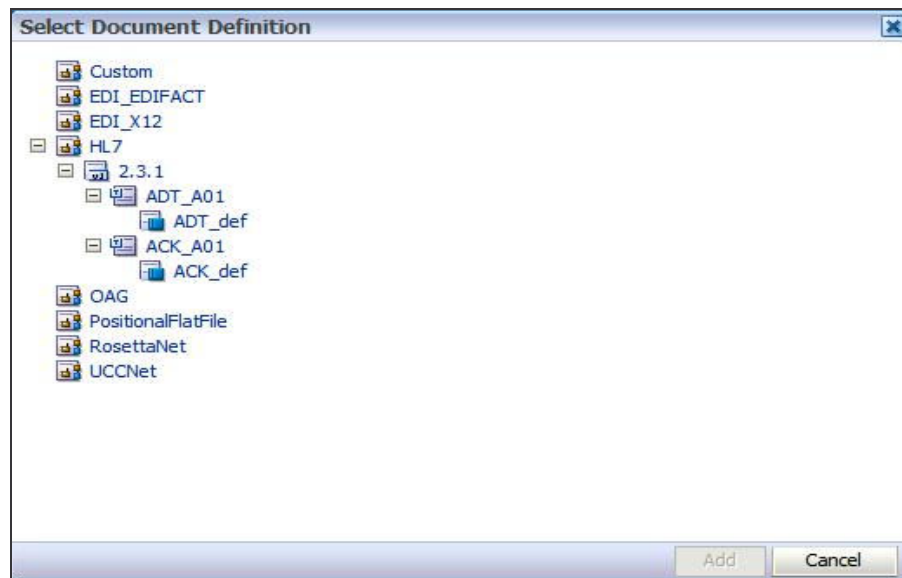
The Document definition is available to the host by default. Hence in this step it is required to choose only the role “sender/receiver”.



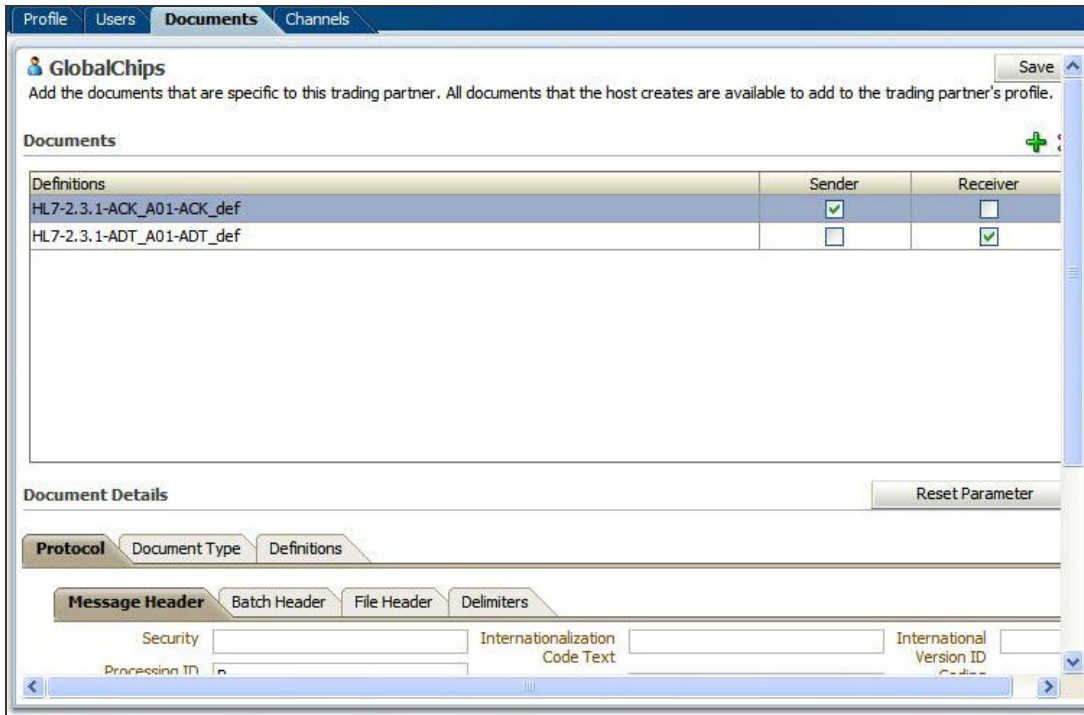
- Select appropriate role of the Initiator (sender/receiver) depending on the business usecase
- Save the updated roles

## Add Documents definitions for Responder

→ Click (+) on Document window to add Document definition to the trading Partner as shown below.



→ Select the required document for business use case and add as part of to the trading partner configuration



- Select appropriate roles (sender/receiver) of the Responder depending on the business use case
- Save the updated roles for the documents

Repeat the above steps for all the required document definitions

**Note:** It is required to validate and redeploy the agreement for any change in Document parameters.

---

## Outbound HL7

B2B can receive 2 types of documents from Back end Application.

- **Native HL7:**  
B2B acts as a gateway doing the validation of the document and forwards it to the respective trading partner using designated exchange protocol. This approach does not provide typical of the HL7 features such as generation of headers, correlation with acknowledgement.

This approach requires disabling the translation flag while creating the agreement, as B2B acts as a gateway, forwarding the document to the TP.

- **XML:** This is the most common use case, where in the backend application gets the data from downstream application and transform it to xml as defined by the schema corresponds to HL7 Document. Upon receiving the XML document, B2B translates it to Native HL7

Back end application is required to send only one transaction set at a time to B2B as B2B generates the envelope on a need basis using the pre-seeded envelop details

✈ **Note:** For samples, please refer to <TODO: To mention the location of HL7 samples>

---

## Inbound HL7

Depending on the translation flag, B2B translate the inbound native HL7 documents to Edifecs XML to be consumed by the back end application. The Edifecs XML generated also has all the Document Protocol Parameters mapped to Internal Properties.

### Agreement Identification

B2B uses 4 parameters for identifying the agreement From Party, To Party, document type and document protocol version

### Trading Partner Identification

Trading partner Identification can be done in four different ways and order of identification. and in the below mentioned order.

- **Dynamic IP**  
While sending the outbound hl7 business message to trading partner, If the message is sent using the Dynamic IP, the dynamic IP address will be cached along with corresponding trading partner name. When inbound message arrives, B2B checks the cache with Dynamic IP address as the key, to retrieve the trading partner name.
- **Delivery Channel**  
When it is required to identify the trading partner based on the delivery channel, user can enable the flag as shown below. This feature can be used as anonymous trading partner. i.e., when there is no unique way to identify the trading partner based on the ip address or dynamic ip or document , one can make use of this feature.

The screenshot shows the 'Channel Details' configuration window with the 'Exchange Protocol Parameters' tab selected. The 'Transport Protocol' is set to 'TCP'. Under 'Exchange Protocol Parameters', the following settings are visible:

- Immediate ACK: None
- End Block Character: 0x1C
- Carriage Return Character: 0x0D
- Start Block Character: 0x0B
- Map ACK Control ID:
- Map Trigger Event:
- Identify TP by delivery channel:

- **MLLP ID**  
IP address or hostname will be used to identify the trading partner
- **Document Based Identification:**  
HL7 has unique values in the MSH segment headers that can be used to identify the Trading Partner.
  - HL7 Message sending Application ID
  - HL7 Message sending Facility ID

### Document Identification

Oracle B2B extracts the Document Type with/without trigger event and revision from the MSH segment of the incoming HL7 Document. It is required that the document to be well formed for the successful extraction of Document Type and Revision.

### Validation and Translation

It is required to get the necessary ecs for translation and validation. Document protocol parameters of the document are compared against the payload.

⚡ **Note:** For samples, please refer to <TODO: To mention the location of HL7 samples>

---

## Various Types of Acknowledgments supported in B2B

There are different ways of handling ack in the HL7 Oracle B2B AS11 and is mentioned below.

### ACK handled by B2B

#### Document level Acknowledgment / Functional Acknowledgement

To enable this document layer acknowledgement or functional ack there is a field “**functional ACK**” in the agreement that needs to be true. This acknowledgement generates after processing of the message and then sends back to the trading partner. Xengine generates this functional ack as per the HL7 standard.

When the “Functional Ack handled by B2B” is set to true, B2B generates the functional acknowledgement and send it to trading partner. When set to false, back end application is expected to generate the functional acknowledgement.

Acknowledgment	
Functional Ack Handled by B2B	<input type="text" value="true"/>
Functional Ack internal properties	<input type="text" value="false"/>
Notify Inbound Receipt Acks	<input type="text" value="false"/>
Notify Inbound Functional Acks	<input type="text" value="false"/>

## Immediate Ack

The turn around time of Functional Acknowledgement is considered to be little high for some business critical healthcare application as it captures both translation, validation error and is generated by Xengine. An alternative in which case is Immediate Ack, which is generated and transmitted in the TCP transport layer instead of the Document layer.

Oracle AS B2B can send immediate acknowledgment in three modes.

**Default** : B2B will parse the incoming HL7 message and generate ack from it. In this mode b2b can send the ack to the sending application with correlation details (ex. Control number from the incoming Message, Sending App.. etc) Hence trading partner Application can correlate the incoming ACK message.

**Simple** : B2B sends a very basic ack which only tells about receiving the message. This simple ack is pre-defined.

**Custom**: In this mode, a file specified in the "Custom Immediate ACK File" field will be sent as an immediate ACK

**Negative**: When received data is not as per the defined mllp parameter or data is received properly then Negative ack is sent to the trading partner from the transport layer.

## Generic Ack

**Typically Acknowledgement carries the message type and trigger event of the original message in MSH.9.**

Some of the HL7 System has a Generic Acknowledgement implementation for all event types i.e which expects only message type in MSH.9 and not any event type.

To enable this, set the "Generic ACK" in document type parameters

### How to configure Oracle B2B for sending and Receiving GenericAck

1. Edit the Acknowledgement ecs file in spec builder.
2. Go to MSH 9, select Event type under Trigger event and delete the same.
3. Save the ecs file and export xsd.
4. Use the above ecs while modeling the flow of Acknowledgment.
5. Generic ACK in document type parameters

For processing the inbound generic ACK, please refer to [Generic Message Type \(Without Trigger event\)](#)

---

## Exchange layer

There are different ways of exchanging the hl7 messages. Most preferred option is to exchange the message via MLLP protocol.

### MLLP

The Minimal Lower Layer Protocol (MLLP) is the most common mechanism for exchanging the HL7 data. MLLP uses the TCP/IP protocol to transfer the data in continuous stream of bytes. MLLP delimiters is used to recognize the start and the end of message,

The typical structure of an MLLP message is given below

**<SB> + <Message> + <EB> + <CR>**

**<SB>** = Start Block. Messages are prefixed with <SB>

**<Message>** = HL7 Message

**<EB>** = End Block. Messages are terminated with <EB>

**<CR>** = Carriage Return. Segments are terminated with <CR>

**⚡ Note:** In Oracle B2B, these parameters are available as part of Document Exchange parameter. MLLP parameter can be overridden as per the endpoint configuration of the delivery channel. These values must be given in hexadecimal format.

Default hexadecimal values of MLLP delimiters in Oracle B2B

**<SB> = 0x0B**

**<EB> = 0x0D**

**<CR> = 0x1C**

Along with the standard MLLP offering, oracle B2B offers generic solution for TCP with the following features.

1. Data with Length of the message, with or with out Start Block and End Block
2. Data with Start Block and End Block
3. Data with Start Block and Length of the message
4. Data with Custom header

For more details refer to [Generic support for TCP](#)

### Generic Exchange

Generic Exchange is also a preferred approach to exchange the message.

There are many ways to exchange the data in generic Exchange

FILE	Exchanging the message using file transport protocol
FTP	Exchanging the message using FTP transport protocol. Where the message can be send and receive from FTP site
SFTP	Exchanging the message using SFTP transport protocol. Where the message can be send and receive from SFTP site with secure communication
JMS	Exchanging the message JMS queues. Where the message can be send and receive from JMS queue

---

## Advanced HL7/MLLP features

---

### Generic Message Type (Without Trigger event)

It is based on the configuration parameter "Generic Message Type". This property can contain the value true/false. If this property is enabled, B2B finds the agreement for the specific message type first then the generic message type as mentioned in the below example.

If the inbound message getting processed is ADT^A01, ADT^A03, ADT^<any event> and the user wants to treat all the message type as generic except ADT^A03. In this case, user has to define two doc types in B2B, one as ADT doc type for generic message type and another for ADT\_A03 doc type for specific message type. When an inbound message arrives at B2B, B2B queries for the agreement first based on ADT\_A03, if it fails it queries for generic message type.

---

### MSH.15 for ACK generation

Acknowledgment generation is dependent on the value in MSH.15 of the business message.

Select "Accept Acknowledgement " document type parameters in HL7, if it is required to generate functional acknowledgment when MSH.15 has no value.

Select None to take no action.

Select AL (always) to generate the acknowledgment under any conditions.

Select ER (error/reject) to generate the acknowledgment when the message errors or is rejected.

Select SU (successful completion) to generate the acknowledgment when the message is successfully processed.

---

### Map control number to ACK

Check "Map ACK Control ID" flag in hl7 document type parameters for mapping the MSH.10 of the business message to the MSH.10 of the acknowledgment.

---

### Broadcast

Broadcast feature of Oracle AS B2B enables the users to send the same message to multiple trading partners. This gives the flexibility to the users to create their own groups of trading partners specific to the business needs. The back-end application can then send a message to the group and B2B will send the message to all the members of the group

The salient points of the feature are:

1. Prevent backend to send multiple files there by reducing the traffic between backend - B2B.
2. Bring the concept of trading partner groups in B2B and use it for broadcasting. This will prevent backend user to know about trading partner details as well as the Group as this is B2B specific meta data
3. This is an event driven approach to handle the multiple messages to various trading partner

This is an optional feature which can used on a need basis by specifying the group name as part of "eventName" attribute. The "eventName" attribute would look like:

```
eventName = Grouping:<name of the group>
```

**Note:** eventName and actionName can be used interchangeably.

---

## Dynamic IP

When it is required to override the end point details configured in the delivery channel transport parameter, we can make use of dynamic end point feature. The dynamic IP feature of MLLP provides flexibility to deliver a message to dynamic endpoints using a single delivery channel for a trading partner. This is set in the message enqueue header property file.

For example:

```
eventName=DynamicIP:GC:140.87.154.102:4444
```

This feature is also available in B2B composites (as part of the SOA Service Component Architecture (SCA) assembly model) using the following syntax:

```
b2b.toDynamicIP=GlobalChips:152.68.196.52:5555
```

The b2b.toDynamicIP property is set in a normalized message property that is sent to B2B.

---

## Sequencing

In healthcare vertical, it is a common practice to exchange the messages in sequence to ensure the business significance.

It is always a challenge to manage the sequencing in a multi-threaded system wherein a message produced first does not mean it will arrive at the destination first.

To ensure the sequencing B2B uses sequencer to sequence the message based on the arrival time, and dispatcher to dispatch the sequenced message.

The sequencing of message can be handled in both inbound and outbound direction.

## Outbound message sequencing

### Sequencer Configuration

To enable the sequencing for outbound message, the message should be enqueued with the below key-pair value in the actionName for AQ/JMS delivery channel.

```
eventName = TARGET:<sequence target name>
```

For fabric composite

```
b2b.sequencingTarget = <sequence target name>
```

### Dispatcher configuration

To dispatch the sequenced message, it is required to configure the Outbound Dispatcher, which has a value of zero by default. For a use case of sequencing without dispatching (Stacking), the dispatcher value is set to zero. Depending on the message load it is possible to configure the Dispatcher to the appropriate value.

Dispatcher can be configured in **admin → configuration page**.

**Miscellaneous(continued)**

Generic Message Type	<input type="text" value="false"/>	Auto Stack Handler Interval	<input type="text" value="1"/>
Outbound Dispatcher Count	<input type="text" value="0"/>		
Inbound Dispatcher Count	<input type="text" value="0"/>		
Auto Stack Handler	<input type="text" value="false"/>		

## Inbound message sequencing

### Sequencer Configuration

To enable the sequencing for inbound message it is required to check the “**sequence**” flag in the MLLP delivery channel.

**Channel Details**

Transport Protocol: TCP

**Transport Protocol Parameters** | Channel Attributes | Exchange Protocol Parameters | Security

Connection Mode: Server | Timeout: 300

Host Name: localhost

Port: 12345

Permanent Connection:  Sequence

Sequence:

Polling Interval: 10

Dispatcher configuration follows a similar pattern as in outbound sequencing.

## Sequencing without Dispatching

In a traditional world the Trading partner downtime is handled by stacking the messages in back end application, which requires the entire message processing in B2B after the down time. This leads to underutilizing the B2B Application during downtime and overloading when the Trading partner comes up. This affects the regular message flow, as there is a surge in message processing.

Upon Trading partner delivery failure, the corresponding messages are marked not to be picked up by Dispatcher resulting in stacking the messages in B2B instead of the back end application.

To process such messages it is required to set the below property, which can be configured at Dispatcher can be configured in **admin → configuration page**

```
Auto Stack Handler = true
Auto Stack Handler Interval = <interval>
```

When set to **true**, the stacked message will be made eligible for delivery by dispatcher during appropriate interval. It is also possible to specify the variable interval with a comma-separated value to Auto Stack Handler Interval.

---

## Generic Support for TCP

MLLP Recipient uses SB, EB and CR to interpret the message. Given this functionality as per standard there is always a challenge to interpret the message with

- Length of the data
- Start String and End String encapsulated message instead of the SB, EB

To address this, Oracle B2B has come up with Generic solution for TCP for all the document protocols with the following features.

5. Data with Length of the message, with or with out Start Block and End Block
6. Data with Start Block and End Block
7. Data with Start Block and Length of the message
8. Data with Custom header

The generic solutions for tcp features are explained below.

### 1. Configuring start block and end block

This feature is used to send or receive data with start block and end block. In this case, the data sent and received based on start block and end block. Hence it is required to define the start block and end block in generic tab of mllp exchange parameters.

### 2. Send or Receive based on data length

This feature is used to send or receive data with data size. For this, define Message Length Index and header size in the generic tab of mllp exchange parameters.

### 3. Define start block and data length before data

This feature is used to send or receive data with start block and data size. For this, define start block, Message Length Index and header size.

Note : In this case start block is the part of header and minimum Message length index should be more than start block size

### 4. Send data without any header, start block or end block to TP

This feature is used to send data without any header, start block and end block. In case of outbound, data will be sent to Trading Partner without any modification of header, start block or end block. For this, Retain header property should to be true.

### 5. Send data with header, start block or end block to the engine

In this feature, data received from TP, is sent to engine (or callout) without removing header, start block or end block. means data is sent with header, start block or end block as per the configuration. For this, Retain header property should to be true.

Following are the various Exchange parameters added

**Start Block:** This property is used to indicate the beginning of the message. Generally, Start block sent before the message sent to TP .

**End Block:** This property is used to indicate the end of the message. Generally End block sent after the message sent to TP.

**Header Length:** This property defines the header size, which is prefixed to the actual data. (This includes the start block, message length and padded header).

**Message Length Index:** This property indicates data size in available in the header. Start index to end index defines the message size. Eg: If the data length is first four bytes then header size will be 4 and “Message Length Index” will be 1-4.

**Retain Header:** It will retain header while sending message to TP (incase of outbound) or to Engine (Incase of Inbound).

---

## Custom header for callout

### Outbound

In this case user can add an additional property in actionName properties. This property will be available in callout as a parameter and the name of the parameter is CUSTOM\_HEADER. User can retrieve this property and the same can be used for subsequent processing in the action name header.

Ex:

```
eventName= CUSTOM_HEADER:AAAAAAA
```

For fabric

```
b2b.customHeader= AAAAAAA
```

**Inbound:**

In this case, if user wants to send the custom header to back end application, user can extract the header from payload and it can be set as CUSTOM\_HEADER which is callout parameter.

---

**Overriding the connection mode**

This property can be used to override the connection mode specified in the delivery channel. For permanent connection, user can set.

```
eventName= CONNMODE:Permanent
```

For Fabric :

```
b2b.connMode : Permanent
```

---

**Best Practices**

Following are some of the best Practice for HL7 Document Processing.

1. Validate the Edifecs xml document against the respective XSD in the back end application such as BPEL. This will ensure the validity of the document before sending it to BPEL.
2. Turn on the Validation for both inbound and Outbound HL7 Message for getting appropriate error message related to the validity of the document.
3. It is needed to either to enable validation/Translation for generating FA in B2B.
4. Make sure to use the same Document routing ID in B2B and BPEL for respective Document Definition.
5. Make sure to use the proper line delimiters in the document, also to define the same in the B2B configuration

---

**FAQ**

1. Can I generate a generate a ACK without trigger event?  
Yes, refer to document type parameters
2. What happens if the message length is greater than specified in message length of generic TCP?  
An error message will be logged at the transport level.
3. Can I use File/Batch header?  
Yes
4. Can I override the File/Batch ecs files?  
Yes, it can be done at the document protocol version parameters
5. What is the state of the message when the sequencing is enabled and dispatcher is turned off?  
MSG\_WAIT\_TRANSMIT for outbound  
MSG\_CONTINUE\_PROCESS for Inbound
6. Is sequencing specific to a channel.  
Yes, only for inbound messages. Where as for outbound it is specific to a message.

7. Can I exchange a message without sequencing, when the dispatcher is ON  
Yes, for outbound message don't set the attribute "TARGET" in action name and for Inbound use different delivery with "sequence" turned off
8. Can I use both Sequencing and broadcasting for a message  
Yes, we have to set the below parameters in actionName  
TARGET:HIS;GROUPING:ADTGroup

---

## Reference:

1. [www.hl7.org](http://www.hl7.org)

Chief Editor:

Prasanna Chandrashekhara  
Ramesh Anantharamaiah  
Nitesh Jain

Contributors:

Jeffrey Hutchins  
Rakesh Saha  
Sundararaman Shenbagam  
Ted Hong  
Dheeraj Kumar Madai

Last Updated: February 9, 2010

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[www.oracle.com](http://www.oracle.com)

Oracle Corporation provides the software  
that powers the Internet.

Oracle is a registered trademark of Oracle Corporation. Various  
product and service names referenced herein may be trademarks  
of Oracle Corporation. All other product and service names  
mentioned may be trademarks of their respective owners.

Copyright © 2008 Oracle Corporation  
All rights reserved.

**ORACLE®**