Sun Adapter for TCP/IP HL7 User's Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 821–0377–10
October 2009

091015@23031

Contents

Sun Adapter for TCP/IP HL7 User's Guide

This document provides information and instructions for working with the Sun Adapter for TCP/IP HL7. It is divided into the topics listed below.

What You Need to Know

These topics provide information that is useful to know before you start working with the TCP/IP HL7 Adapter:

- "Sun Adapter for TCP/IP HL7 Overview" on page 8
- "About TCP/IP HL7 V2 Collaborations" on page 107
- "About TCP/IP HL7 V3 Collaborations" on page 120
- "MLLP V2 Content Exchange Model" on page 144
- "Standard Inbound HL7 V2 Collaboration Overview over MLLPV2" on page 146

What You Need to Do

These links provide information and instructions for working with the TCP/IP HL7 Adapter in Java CAPS projects:

- "Adding a TCP/IP HL7 External Application to a Connectivity Map" on page 28
- "Modifying the TCP/IP HL7 Adapter Properties in the Connectivity Map" on page 29
- "Configuring TCP/IP HL7 Adapter Environment Properties" on page 91
- "Creating a Copy of an HL7 Sample Project" on page 98
- "Customizing Predefined Collaborations for HL7" on page 102
- "Creating Copies of HL7 Collaborations" on page 102
- "Adding an HL7 Message Library to an Existing Collaboration" on page 104
- "Creating and Configuring the MLLP V2.0 Database" on page 140

Additional Information

These links provide additional information that is useful to know when working with the TCP/IP HL7 Adapter:

- "TCP/IP HL7 V2 Adapter Inbound Connectivity Map Properties" on page 30
- "TCP/IP HL7 V2 Adapter Outbound Connectivity Map Properties" on page 49
- "TCP/IP HL7 V3 Adapter Inbound Connectivity Map Properties" on page 64
- "TCP/IP HL7 V3 Adapter Outbound Connectivity Map Properties" on page 79
- "TCP/IP HL7 Inbound Adapter Environment Properties" on page 92
- "TCP/IP HL7 Inbound Adapter Environment Properties" on page 94

Sun Adapter for TCP/IP HL7 Overview

The following topics provide information about HL7 and the TCP/IP HL7 Adapter:

- "About Sun Adapter for TCP/IP HL7" on page 8
- "About HL7" on page 10
- "The TCP/IP HL7 Adapter Architecture" on page 10
- "Modes and Roles" on page 12
- "Inbound Functionality" on page 14
- "Outbound Functionality" on page 17
- "General Functionality" on page 20
- "TCP/IP HL7 Adapter Operation" on page 22
- "Monitoring the HL7 Adapter" on page 24
- "Schematron Support in the HL7 Adapter" on page 25

About Sun Adapter for TCP/IP HL7

The Sun Adapter for TCP/IP HL7 is a component of the Sun Java Composite Application Suite (Java CAPS) that enables the Java CAPS ESB system to exchange data with an external TCP/IP application using the HL7 data protocol. The Sun Java CAPS ESB with the TCP/IP HL7 Adapter provides:

- Macro functionality, providing ease of use and productivity.
- Prebuilt standards-compliant inbound and outbound template Collaborations that you can use as is or that you can modify for your specific needs.
- A complete set of configurable properties that allow you to customize the functionality of the Adapter. The functions can further be customized by modifying the Collaborations.
- Journaling and error messaging to JMS queues and topics. This is in addition to Sun Java CAPS ESB's standard alert and debug logging.
- Support for HL7 Standard versions 2.1, 2.2, 2.3, 2.3.1, 2.4, 2.5, 2.5.1, 2.6 and V3.

Note – Throughout this document the term "JMS queue" is used in the generic sense and actually denotes JMS queues or topics.

TCP/IP HL7 Features

The TCP/IP HL7 Adapter includes the following features:

- Bidirectional processing, including client or server mode in either direction (to or from Sun Java CAPS ESB).
- Handles both HL7 HLLP and MLLP protocols and envelopes.
- Provides a wide variety of recourse action configurations.
- Non-blocking I/O.
- Recovery and retry logic.
- Debug levels and error logging.
- Journaling of HL7 messages and associated acknowledgements.
- HL7 acknowledgement levels.
- Fully supports the HL7 sequence numbering protocol.
- Full support for HL7 ACK and NAK generation and validation.
- Supports delayed ACK in both directions.

TCP/IP HL7 Adapter Components

The TCP/IP HL7 Adapter incorporates three components:

- The HL7 TCP/IP Resource Adapter that implements the lower layer HL7 protocol over TCP/IP.
- Default inbound and outbound Collaborations that implement the HL7 messaging protocol, sequence numbering, and recourse actions.
- Generic HL7 Message Libraries that provide the structures necessary to parse and create the data messages and ACKs used by the protocol.

The TCP/IP HL7 Message Library, also known as an Object Type Definition (OTD) Library, enables the creation of HL7 interfaces capable of running over TCP/IP, and also utilizes the common Adapter services available in Java CAPS. The TCP/IP HL7 Adapter works hand in hand with the Sun Java CAPS HL7 Message Libraries, versions 2.1 through 2.5.1.

The TCP/IP HL7 Adapter properties allow the user to easily configure the operation of the TCP/IP HL7 Adapter. The Adapter includes a set of properties that are configured in the Connectivity Map and only apply to that Adapter in the Project. It also includes a set of properties that are configured in the Environment and apply to all TCP/IP HL7 Adapters in the Project. These properties are adopted into the Message Library's functions.

The Message Library handles all of the lower-layer protocol. The Message Library's behavior is customized using the Adapter configuration properties. These Adapter properties are used by the resource adapter, but are also accessed and used by the prebuilt Collaborations.

About HL7

HL7 is a standard for exchanging information between medical applications and is an abbreviation of Health Level Seven. *Level Seven* refers to the seventh OSI layer protocol for the health environment. HL7 defines the format and the content of the messages that applications must use when exchanging data with each other under various circumstances.

Hospitals and other medical institutions typically use many different types of systems to communicate with one another. Everything, from patient records to billing information, is tracked and recorded in computer systems. In order for these different types of systems to communicate with each other, they use a standard like HL7.

Note – In the computer world, a protocol is a formal, well-defined standard for exchanging information between computer applications.

An important part of the HL7 standard is the *ACKnowledgment protocol*, also known as an ACK. Every time an application accepts a message and consumes the data, it is expected to send an ACKnowledgment message back to the sending application. The sending application is expected to keep on sending a message until it has received an ACK message.

The TCP/IP HL7 Adapter Architecture

The TCP/IP HL7 Adapter's functionality comes from a combination of the TCP/IP HL7 Resource Adapter (RA), the predefined inbound and outbound HL7 Collaborations, and the generic HL7 Message Libraries.

TCP/IP HL7 Resource Adapter

The TCP/IP HL7 Resource Adapter communicates with external HL7 systems, establishes and maintains the TCP/IP socket, manages message enveloping, maintains the sequence numbering file, and provides the HL7 protocol state to the Collaboration. The RA (Resource Adapter) is configured from the Adapter Properties Editor.

HL7 Collaborations

The inbound and outbound HL7 Collaborations provide message validation, sequence numbering, ACK and NAK generation, and recourse actions. The predefined HL7 Collaborations are designed to implement the HL7 standard protocol and inter-operate with similar standard compliant systems by simply changing the Adapter property configuration.

If a system does not conform to the HL7 specification, the Collaborations can be modified for that transaction by changing the Java code using the Collaboration Editor. The Collaborations Java code is designed to be available and “transparent” so you can easily reference the predefined Java code to see how it currently handles HL7 transactions and make the appropriate modifications.

The Collaboration Editor allows you to create and edit Java code to modify a Collaboration for your specific needs. In many cases this code can be created graphically (drag and drop), using the Collaboration Editor’s Business Rules Designer. If you need to change the code of a prebuilt Collaboration, you should duplicate the Collaboration first and then modify it.

The Collaborations are designed to target one unit of work at a time, meaning the resolution of one message at a time. Once the current message transaction is resolved, the Collaboration is free to process the next HL7 message. The general form of the Collaborations is a state machine. Based on the state of the connection, the Collaboration performs the appropriate action. Additional Collaborations can be added to a Project to increase message flow.

Generic HL7 Message Libraries

The generic HL7 Message Libraries are version-agnostic structures used to send and receive HL7 messages, acknowledgements, and negative acknowledgements (NAKs). They provide the Collaboration with only the essential fields for implementing the HL7 protocol. If you need to perform functions that are specific to a version or message type, you can add the appropriate Message Library to the Collaboration.

Sun Java Composite Application Suite Functionality

The TCP/IP HL7 Adapter takes advantage of the Java CAPS facilities to provide journaling and error messaging, as well as monitoring, alerting, and logging. journaling and error messages are sent to JMS queues, which allow flexibility for postprocessing. For example, invalid messages and their negative acknowledgements (NAKs) are sent to a JMS queue. This JMS queue can then be set up to allow the invalid HL7 messages to be viewed, corrected, and resubmitted automatically to the same Adapter.

- **Error Queues**

  Each Collaboration automatically sends invalid messages that have incorrect data, have invalid formatting, or are deemed unacceptable to a JMS error queue. The error generated by the message and, if appropriate, its associated NAK, are written as JMS properties of that message for later processing. You can send errors to one common queue, to a specified queue for each Adapter, or to a combination of both.
- **Journaling Queues**

  When journaling is enabled, the HL7 message and its related acknowledgement (ACK) are written to a JMS journal queue. You determine the number of JMS error queues or journal queues used by a Project. From the JMS queue, these messages can be accessed and written to file, sent to a database, sent to a Web application for processing, and so forth.

- **Monitoring**

  The Enterprise Manager provides a real-time picture of the Adapter's state and status. The monitoring facilities display the following information:

  - Adapter up or down
  - Connected to external
  - Current sequence number
  - Date and time of the last transaction
  - Adapter properties
- **Alerts**

  Alerts are sent from the RA and the Collaborations when conditions are identified that endanger or stop the interface. You can add your own custom alert messages to the Collaborations.
- **Logging**

  Log messages are written from both the RA and the Collaborations. You can also configure your own log messages. The level is set in the Enterprise Manager. For more information on monitoring, alerting and logging, see *Using Enterprise Manager Management Application in Java CAPS* and *Alert Codes for Java CAPS Adapters*.

Modes and Roles

The TCP/IP HL7 Adapter can operate in two modes: standard and delayed ACK. Standard mode is the typical message exchange in HL7 where an HL7 message is sent and an HL7 ACK is received, or the other way around. In delayed ACK mode, the exchange of a message requires two acknowledgements: one to confirm the message was received and the other from the external system that actually received the message to verify that it was received.

In these two modes, the Adapter and the ESB have a number of roles they play within certain scenarios; that is, certain components can fulfill different responsibilities within a protocol. For example, the outbound Collaboration can fulfill two roles in the delayed ACK mode: one as the “sender of messages” that expects two delayed ACKS, and another as the “forwarder of ACKS” from the external system.

Note – Delayed ACK mode is deprecated as of HL7 version 2.2 and was removed from the HL7 standard as of version 2.5.

Standard Mode

In standard mode, the HL7 Adapter can assume two roles, sender and receiver. These are implemented in the outbound and inbound Collaborations respectively.

HL7 Adapter Sender Role

The outbound Collaboration is the implementation of the Sender and the RA is configured for an outbound data flow. A Java CAPS Service forwards the data to the Adapter, which in turn forwards the data to the accepting HL7 external system. The External System responds with an ACK or NAK response.

HL7 Adapter Receiver Role

The inbound Collaboration is the implementation of the Receiver role in conjunction with the RA being configured for inbound direction. The Adapter accepts a message from the sending HL7 external system, forwards the data to a Java CAPS Service, and responds to the sending system with an ACK or a NAK response.

Delayed ACK Mode

Delayed ACK mode is an extension to the basic HL7 message exchange mode, where there is some middleware component between the Sender and the Receiver. The sending system expects to receive the ACK from the receiving system in addition to the middleware component. In this mode, the ESB can assume two roles in the protocol: as the Sender and as the Receiver.

ESB Sender Role

In this role, the ESB acts as the Sender in the exchange. The HL7 RA is configured for the outbound direction, and the HL7 outbound Collaboration is configured so the Sends App Ack property is set to `True`. This parameter is used even though the ESB technically does not send an application ACK. Rather, it receives the application ACK and provides the compliment behavior to the ACK sent by the inbound configured Adapter. The two are related in the protocol.

The purpose of this role is for the ESB to act as if it is a system that requires the Delayed ACKs so that it can communicate with a system that operates in the Delayed ACK Receiver role. An example implementation of this role is available in the HL7Outbound sample Project.

ESB Receiver and Forwarder Role

To perform this role, the ESB is configured with two instances of the HL7 Adapter , each performing its own role as Receiver or Forwarder.

Note – For Delayed ACK, the Receiver and Forwarder must be on the same integration server.

The following steps convey the steps taken in this scenario.

1. The Receiver accepts the HL7 message. It then returns the first ACK, with an MSH - 5, value “D” (for the Delayed ACK), to the Sender.

2. The Receiver sends the message to the Forwarder.
3. The Forwarder sends the message to the receiving External System.
4. The receiving External System sends an ACK to the Forwarder.
5. The Forwarder receives the ACK from the External and forwards it on to the Receiver.
6. The Receiver then forwards the ACK, with an MSH - 5, value "F" indicating that it is the receiving External System's ACK, to the Sender.

In the Receiver role, the HL7 RA is configured for inbound, and the inbound Collaboration is used with the parameter `Sends App Ack` set to **`true`**. In the Forwarder role, the RA is set to outbound, the parameter `Forward External Acks to eGate`, is set to **`true`**, and the outbound Collaboration is selected.

This configuration table presents the necessary parameters used to configure the Adapter to assume the Delayed ACK roles

TABLE 1 Adapter Delayed ACK Configuration

Role	Direction	Sends App Ack Property	Forward External Acks
Sender	Out	True	N/A
Receiver	In	True	N/A
Forwarder	Out	N/A	True

An example of the receiver role is provided in the prjHL7Inbound sample Project. The prjHL7Outbound Project provides a sample implementation of the Forwarder role.

Inbound Functionality

The inbound TCP/IP HL7 Adapter Project, prjHL7Inbound, provides a sample implementation of an inbound flow using the Adapter. It can be configured for standard inbound mode or for forward message mode.

Inbound Adapter Data Flow

The inbound TCP/IP HL7 Adapter Project receives HL7 messages from an external system, sends an acknowledgement of the message to the external, provides sequence numbering, writes the HL7 message to a JMS data queue, and also writes the HL7 message and ACK to a JMS journal queue. Any error messages and NAKs are sent to a JMS error queue.

The HL7 data is processed so all the fields in the MSH segment of the message are stored in an internal structure to generate an HL7 response. Non-HL7 data, including HL7 acknowledgments, automatically generate warnings in the Adapter's log file and send an HL7 NAK to the external system.

Standard Inbound Message Mode Data Flow and Architecture

The following steps describe the flow of data for an inbound Adapter:

1. The external system sends the HL7 message to the Adapter.
2. The Collaboration receives the HL7 message.
3. The Collaboration validates the message (if validate is enabled). If it fails, the Collaboration takes the configured recourse action. If the recourse action is stripped and the maximum number of retries has been exceeded, the message and error are written to the error queue.
4. The Collaboration writes the message to the data queue.
5. The Collaboration then creates the appropriate ACK and sends it to the RA.
6. The RA envelopes the ACK and sends it to the External System.
7. If journaling is enabled, the message and its ACK are written to the journal queue.

Inbound Receiver Message Mode

The Inbound Receiver Message mode is used when the Delayed ACK is configured to fulfill the role of the Receiver in the Delayed ACK scenario. It accepts the message and acknowledges the External and then forwards the message to the component fulfilling the Forwarder role. It then accepts the ACK from the Forwarder and passes it on to the External that sent the message.

The following steps describe the Inbound Forward Message Role:

1. The Sender External, sends an HL7 message to the Inbound Adapter, which is configured as a Receiver (`Sends App Acks` is enabled).
2. The Inbound Adapter receives the HL7 message and returns the first Acknowledgement to the External with an MSA - 5, value “D” for Delayed Acknowledgement. The External receives the ACK, validates the ACK (verifying that it is a Delayed ACK), and waits for another ACK.
3. The Inbound Adapter creates a JMS message with the HL7 message as the payload, creates a “reply to” destination, and forwards the HL7 message to the Outbound Forwarder (to a JMS destination).
4. The Outbound Forwarder gets the HL7 message and forwards the message to the External System.
5. The External System receives the HL7 message and returns the HL7 ACK message to the Outbound Forwarder.
6. The Outbound Forwarder gets the HL7 ACK message and sends it to the Inbound Receiver Adapter using the “reply to” destination.
7. The Receiver External reads the HL7 ACK message and forwards the second HL7 ACK message with an MSA - 5, value “F”to the Sender External. The Sender External then takes the appropriate action: for example, journaling the HL7 message and the HL7 ACK.

Message Verification

Message verification begins with reading the message from the external system. The message is expected to match the MLLP envelope, since both HLLP and MLLP envelopes have the Start of Block (SOB), End of Data (EOD), and a Carriage Return (CR) in common.

If a message fails the read verification, it is considered bad data. If read by an inbound Adapter, this failure causes the Adapter to generate a Canned HL7 NAK. An outbound Adapter ignores the message and logs a warning, reporting the nature of the problem to the log file.

An HLLP envelope needs further verification as to whether it is data or a NAK, as well as the Block Checksum and Block Size. The Adapter behaves as described above if the HLLP envelope verification fails.

After stripping the message envelope, the RA hands the de-enveloped message to the inbound Collaboration where it is parsed into the generic event Message Library. This ensures that the general form and MSH segment are valid. If the MSH property is set, the Collaboration verifies that the fields specified in the HL7 segment section are the same as those of the received MSH, otherwise, a NAK is returned.

Acknowledgment Processing

- **Adapter Generates HL7 Acknowledgment**

  In this scenario, the Adapter generates an HL7 ACK after receiving and successfully storing the message in a queue; otherwise, it generates an HL7 NAK. The HL7 ACK or NAK is placed in the proper envelope and sent to the external system.
- **ESB Sends HL7 Acknowledgement**

  In this scenario, the Adapter acts as a receiver in a Delayed ACK scenario, as described in "Inbound Receiver Message Mode" on page 15.
- **Canned HL7 NAK**

  A canned HL7 NAK is created when a read error occurs or when an message cannot be identified as an HL7 message. The initial test ensures that the message conforms to the lower-layer protocol. The Resource Adapter uses the MSH section parameters to create an appropriate NAK.

Recourse Actions

Recourse actions can be configured for an inbound Adapter for the following conditions. For more information, see "Recourse Actions" on page 21.

- The empty read limit is reached.
- The maximum number of NAKs are received by the Adapter.
- The maximum number of NAKs are sent by the Adapter.
- The maximum number of response timeouts is reached.
- A NAK response is received.

- No response is received after a message is sent the maximum number of times.

Outbound Functionality

The outbound TCP/IP HL7 Adapter Project, prjHL7Outbound, can be implemented in standard outbound mode or in two forward message modes: outbound delayed ACK or outbound forwarder.

Outbound Adapter Data Flow

In outbound mode, the Adapter receives HL7 messages from a JMS queue. Each message is verified to ensure it contains HL7 data only. Legitimate HL7 data is enveloped into its configured format and sent to the external system.

A message in the JMS queue triggers the outbound Collaboration. The outbound Collaboration is provided with an HL7 message to send to the external system

The Adapter waits for a configurable number of milliseconds for an incoming HL7 ACK or NAK from the external system. After receiving an HL7 response from the external system, the Adapter strips the message from its envelope and verifies its integrity.

Any non-HL7 acknowledgment received from the external system causes the Adapter to resend the same message. If the incoming response is an HL7 ACK or NAK, the Adapter might do either of the following, as dictated by its configuration:

- Recourse action on NAK received.
- Recourse action on Max NAK received.

If journaling is set, these messages and their ACKs are placed in the journal file.

Outbound Standard Messaging Mode

The following steps describe the process for the Outbound Standard Message Mode:

1. An HL7 message triggers the Collaboration. The outbound Collaboration is designed to accept the HL7 messages.
2. The Collaboration maps the received message into the Generic Event Message Library and validates the MSH segment. If validation is enabled, the Collaboration checks the MSH segment of the outbound messages against MSH values configured in the Adapter properties file. If the validation fails or the message cannot be parsed, the message and its error are written to the error queue. Note that the HL7 message is always checked for structural correctness.
3. The Collaboration sends the message to the RA.
4. The RA envelopes the message and sends it to the External System and waits for an ACK.

5. The Collaboration receives and validates the ACK, and then journals the ACK and the HL7 message (if journaling is enabled). If the Collaboration receives a NAK, the NAK and the HL7 message are sent to the error queue.
6. Finally, the Collaboration commits the JMS receive.

Outbound Adapter Roles for Delayed ACK Scenarios

The outbound Adapter can fulfill two roles in a delayed ACK scenario. The outbound delayed acknowledgement mode is used to communicate with an external system that is configured to receive messages in a delayed ACK way; that is, it receives two ACKs. One confirms the message was received, and the other is from the application that accepts the message. For delayed ACK mode, the process is similar to that of the standard outbound mode, except that it receives two ACKs. The initial ACK comes from the receiving system.

Outbound Delayed ACK Role

The following steps describe the outbound delayed acknowledgement role process displayed:

1. The outbound Adapter, which is configured as Delayed Acknowledgement role, receives a message from JMS, and sends the message to the External System.
2. The External System receives the message and returns the first Acknowledgement to the outbound Adapter with an MSA - 5, value “D” for Delayed Acknowledgement. The outbound Adapter receives the ACK, validates the ACK (verifying that it is a Delayed ACK), and waits for another ACK.
3. The outbound Adapter receives another HL7 ACK message (the second) and validates that the second HL7 ACK message is an MSA - 5, with a value of “F.” If the second ACK is valid, the Adapter commits the message, otherwise it resends the message.

Outbound Forwarder Role

The Outbound Forward Message role is used in conjunction with the with the inbound Adapter, which is also configured to handle delayed ACKs. No validation is preformed: the Adapter acts as a “pass-through.”

The following steps describe the Outbound Forwarder Role processing:

1. Data is received by the Collaboration, from the JMS queue.
2. The Collaboration extracts the JMS property “reply to” destination from the message, but does no validation, and sends the message to the External System.
3. The Adapter receives the ACK from the External System.
4. The Collaboration sends the ACK to the temporary topic that was contained in the “reply to.”

Note – For Delayed ACK, the Receiver and Forwarder must be on the same integration server.

Message Verification

The only verification that the outbound Adapter does is to ensure that the message parses into the generic Event Message Library, and that the MSH uses the correct fields. The acknowledge is verified to ensure that the sent message is valid.

Acknowledgment Processing

- **Adapter Generates HL7 Acknowledgment**

  In this scenario, the Adapter generates an HL7 ACK after receiving and successfully storing the message in a queue; otherwise, it generates an HL7 NAK. The HL7 ACK or NAK is placed in the proper envelope and sent to the external system.
- **ESB Sends HL7 Acknowledgement**

  In this scenario, the Adapter acts as a sender in a Delayed ACK scenario, as described in "Delayed ACK Mode" on page 13.
- **Canned HL7 NAK**

  A Canned HL7 NAK is created when a read error occurs, or when an message cannot be identified as an HL7 message. The initial test ensures that the message conforms to the lower-layer protocol. The Resource Adapter uses the MSH section parameters to create an appropriate NAK.

Recourse Actions

Recourse actions can be configured for the outbound Adapter for the following conditions:

- The Adapter sends the maximum number of canned negative acknowledgments.
- The Adapter attempts to read data the maximum number of times from the external system after a read or receive operation returns nothing.
- The Adapter receives the maximum number of negative acknowledgments.
- HL7 message validation fails prior to the sending of the HL7 message to the external system.
- The Adapter reaches the maximum number of response timeouts while waiting for data from the external system.
- The Adapter receives an HL7 Application NAK from the external system.
- The Adapter waits for a response from the external system for the configured amount of time (in milliseconds).

  For more information on the available recourse actions, see "Recourse Actions" on page 21.

Note – The TCP/IP HL7 Adapter includes internal counters that keep track of all error conditions.

General Functionality

This section explains the Adapter's general functions and features. It includes the following topics:

- "Non-blocking I/O" on page 20
- "HL7 Sequence Numbering Protocol" on page 20
- "Failed Message Handling" on page 21
- "Recourse Actions" on page 21

Non-blocking I/O

The non-blocking I/O feature prevents the Adapter from locking up when attempting to read or write data blocks, allowing the Adapter to continue its operation in case of any communication errors. If the read attempt fails for a configurable number of times, the Adapter exits or resets its connection to the external system, depending on its configuration. In the event of a failed write, the Adapter can resume its write operation to pick up where it previously left off until the entire message is successfully sent.

Without this feature, the Adapter might lock up when a read or write failure occurs and be unresponsive to all external messages, including requests from the user or the Enterprise Monitor (for status).

HL7 Sequence Numbering Protocol

The Adapter can be configured to use HL7 sequence numbering. The negotiation and incrementation of this number is automatically performed by the Adapter. For more details on HL7 sequence numbering, refer to Appendix C (Lower Layer Protocols) of the HL7 Standard for the HL7 version you are using.

When the Adapter is configured for HL7 sequence numbering, the sequence number file opens when the Adapter starts up. If the sequence number file does not exist, one is created and populated with a zero sequence number. The sequence number file is updated on the inbound Adapter when the Adapter generates the HL7 ACK (this process is transparent to the user), and when the outbound Adapter receives the HL7 ACK from the external system.

If you want to change the sequence number at runtime, you need to suspend the Adapter, edit and save the sequence number file, and reactivate the Adapter. To force the Adapter to resynchronize its sequence number with the external system, you need to suspend the Adapter, edit the file so it contains a "-1", and then reactivate the Adapter.

The minimum HL7 sequence number is 1. The maximum HL7 sequence number is 2 billion. A sequence number of “0“ is used to start a session. If the sequence numbers between the Adapter and the external cannot be reconciled during start or when exchanging messages, the Adapter shuts down and wait for human intervention as dictated by the HL7 Standard.

Failed Message Handling

The Adapter can be configured to send failed or skipped messages (destined for the external system) to a JMS-based error queue. Messages that fail validation are also written to the error queue. Note that the inbound mode of the Adapter will not write messages that fail the MLLP and HLLP validation. These are automatically NAKed and not passed to the Collaboration, but are logged to the Adapter’s log file.

The failed or skipped message is written to the JMS queue and the error type and message are written as the JMS properties:

- Error: the actual error message or NAK
- Error Type: the type of error, such as HL7_NAK_error or HL7_Validation_error.

Skipped messages are those which are continuously NAKed by the external system and thus are skipped if the Adapter is configured accordingly. If the Adapter is configured for any other recourse action other than skip, the message remains in the queue.

Recourse Actions

The TCP/IP HL7 Adapter recourse actions include Reset, Resend, Skip Message, and Exit.

- On Reset, the Adapter drops its connection and then attempts to reconnect.
- On Resend, the sequence number file and journal file are opened again (provided the newly loaded configuration parameters are set for sequence numbering and journaling).
- On Skip Message, the Adapter remains connected, but writes the message to an error queue.
- On Exit, the Adapter closes its journal file and sequence number file (provided these were configured for use). The Adapter terminates its connection with the external system and shuts down. This allows you to modify these files and resolve any errors. Once the corrections are made, the Adapter can be reactivated from the Enterprise Manager.

Stopping the Collaboration with a Fatal Alert

When the Exit recourse action is triggered it logs the error that caused the action. It also shuts down the Collaboration, which in turn causes the HL7 message to roll back, and then sends an alert to the Enterprise Manager.

The Exit Recourse Action calls the fatal alerter in the Collaboration:

```
alerter.fatal(”error message”,”HL7”);
```

The argument `error message` is the user-configured alert message. The argument `HL7` is the source component (this must be "HL7").

Note – The `alerter.fatal("error msg", "HL7")` method is only applicable to the packaged TCP/IP HL7 Collaborations.

The Exit recourse action should be applied to any error condition that requires human intervention to correct an error. Once the error condition is resolved, the Collaboration can be restarted from the Enterprise Manager.

TCP/IP HL7 Adapter Operation

The following topics explain the basic elements of the TCP/IP HL7 Adapter's general operation:

Direction

The TCP/IP HL7 Adapter can be configured as either HL7 inbound or HL7 outbound. This option is determined automatically by the Adapter's binding (link) in the Connectivity Map.

Connection Type

The connection type indicates how the Adapter establishes a TCP/IP connection. The role can be as a Client, where the RA connects to the external, or as a Server, where the RA waits for a connection.

- **Connected as a TCP/IP HL7 Client**

  As a TCP/IP HL7 client, the Adapter connects to external server (host/port) and establishes a connection (in active mode).

- **Connected as a TCP/IP HL7 Server**

  The Adapter waits and listens to a specific port for incoming connection requests from an external client. Once a request is received, the Adapter accepts the request and establishes a connection (in passive mode).

Lower Layer Protocol

This section describes the two supported envelope types used in the HL7 protocol:

- HLLP (Hybrid Lower Layer Protocol)
- MLLP (Minimal Lower Layer Protocol)

Both envelope types use the following configuration parameters. For more information on these parameters, see "Lower Layer Protocol — TCP/IP HL7 V2 Inbound Adapter" on page 40 or "Lower Layer Protocol — TCP/IP HL7 V2 Outbound Adapter" on page 55.

- Start Block Character
- End Data Character
- End Block Character

MLLP

The MLLP envelope consists of a Start of Block component, a Data component, an End of Data component, and an End of Block component. The size of the HL7 Data field is determined by the length of the data (number of bytes between start and end), with a maximum size of 99999 Bytes.

HLLP

The HLLP envelope consists of a Start of Block component, a "D' (Data) or "N' (NAK) indicator, an HL7 Version component, a Carriage Return, a Data component, a Block Size component, a Block Checksum component, an End of Data component, and an End of Block component. The size of the HL7 Data field is determined by the length of the data (number of bytes between start and end), with a maximum size of 99999 Bytes.

HL7 Acknowledgment Level

The Adapter supports sending and receiving both HL7 acknowledgement types:

- Application acknowledgment: This acknowledgement is sent when the message is successfully received.
- Commit (accept) acknowledgment: This acknowledgement is sent after the message is successfully and functionally processed by one receiving system.

Journaling

The Adapter provides the option to journal successfully received or sent messages and their corresponding ACKs. The messages are sent to a JMS queue or topic, depending on how you configure the Adapter, and the ACKs are stored as a JMS property, HL7_ACK, of that message.

It is expected that, when enabled, the journal queue has one or more subscribers that process the contents of the queue so that it remains manageable. For example, the Batch Adapter or a database Adapter could periodically consume the messages by writing them to a file or a database.

Error Queues

The Adapter provides a mechanism to store failed or stripped messages in a JMS queue or topic. The advantage of this is that the messages are then saved in a form readily usable by the other data flows, that can automatically process these messages or make them available to some type of human intervention or message repair, using tools like the JMS queue editor or an eVision application.

Alerts and Monitoring

If the Adapter loses the connection to the external system in any direction or connection type, due to a crash, shutdown, or suspension (including recourse actions), an alert is generated. The monitor's status of that Adapter is changed to “down” and the Adapter's icon is encased in a red warning box. The monitor also displays the number of messages it has processed along with the date and time of the last message sent.

Support for HL7 Version 2.5 SFT Segments

HL7 version 2.5 adds a new SFT segment to every message. The Adapter not only sends and receives messages with the new segment, it can automatically create and populate them, using information from the Adapter properties, for the outbound message and the ACK sent from the inbound mode. This feature is only available when the Version ID property is set to 2.5 or later.

Delayed Acknowledgements

The Adapter supports delayed acknowledgements in either direction and in a number of roles. This functionality is described in detail in “Outbound Adapter Roles for Delayed ACK Scenarios” on page 18.

Monitoring the HL7 Adapter

You can monitor the status of the HL7 Adapter in the deployed Projects that include the adapter. This includes viewing alerts and log messages, checking connector details, and monitoring external connections. This is done on the Enterprise Manager. For more information about using the Enterprise Manager Monitor, see *Using Enterprise Manager Management Application in Java CAPS*.

For outbound HL7 Adapters, periodic monitoring for external connections is performed. The time period is based on the value defined in the HL7 Adapter web application's deployment descriptor file, `web.xml`. Below is an excerpt from the file defining the time period.

```
<web-app>
...
...
<!--  Default monitoring period used in monitoring the
external system connection  -->
<context-param>
<param-name>monitorperiod</param-name>
<param-value>2000</param-value>
</context-param>
...
</web-app>
```

Schematron Support in the HL7 Adapter

Schematron is supported for HL7 V3 Message Libraries. The Schematron uses the concept of finding tree patterns in the parsed document rather than the grammar. This approach allows representation of numerous structures that are inconvenient and difficult in grammar-based schema languages.

For example, the following file defines a Person element that includes a Name field and a Gender field:

```
<?xml version="1.0" encoding="UTF-8"?>
<Person>
<Name>Eddie</Name>
<Gender>Male</Gender>
<Person>
```

The above XML document can be validated against the below schematron, which defines a test for a Title field, a test for Name and Gender, and a test for the order of fields:

```
<?xml version="1.0" encoding="UTF-8"?>
<sch:schema xmlns:sch="http://www.ascc.net/xml/schematron">
<sch:pattern name="Check structure">
<sch:rule context="Person">
<sch:assert test="@Title">The element Person must have a Title attribute<sch:assert>
<sch:assert test="count(*) = 2 and count(Name) = 1 and count(Gender) = 1">The element
 Person should have the child elements Name and Gender.<sch:assert>
<sch:assert test="*[1] = Name">The element Name must appear before element
Gender.</sch:assert>
</sch:rule>
</sch:pattern>
</sch:schema>
```

In the HL7 Adapter, this schematron is useful for validating an HL7 V3 document against predefined schematron schemas that you write. You can also obtain schemas from organizations such as NHS and HL7.org. For example, NHS provides schemas for CDA documents.

Schematron Configuration in HL7 Adapter

You configure the schematron validation from the Connectivity Map Properties Editor. The Properties Editor includes two properties to support schematron validation:

- **Schematron Validation**: Selecting **true** enables schematron validation. You then need to enter an LDAP reference.
- **Schematron Files**: The list of schematron validation files. Use commas to separate multiple files.

API for Schematron Validation

The HL7 Adapter includes an API specific to schematron validation. This API is a wrapper of the Open source XSLT-based API available at http://xml.ascc.net/schematron/1.5. The API is an XSL file called metastylesheet (skeleton1-5.xsl). Applying the metastylesheet to the schematron XML document generates another XSL file. This XSL file can be applied to the input XML document to validate, which produces the output XML document that contains the results of the validation. This document can be embedded inside the V3 acknowledgement and can be sent to the original sender.

The metastylesheet can be extended and overridden so that you can customize the output XML document.

Example,

The following is an example of an output document generated after invoking the API using the XML input document and the schematron validation document described above.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<schematron-output phase="#ALL" schemaVersion="" title="" xmlns:
sch="http://www.ascc.net/xml/schematron">
<active-pattern name="Check structure"/>
<fired-rule context="Person" id="" role=""/>
<failed-assert id="" role="" test="@Title" location="/@Person[1]">
<text>The element Person must have a Title attribute</text>
</failed-assert>
<schematron-output>
```

Using the Schematron API

Perform the following to invoke the schematron API from a Java Collaboration:

- Obtaining the Factory Object
- Obtaining the Validator Object
- Performing the Validation

Obtaining the Factory Object

Below is a sample call to the `getSchematronValidatorFactory` method.

```
com.stc.connector.hl7.schematron.SchematronValidatorFactory
  factory = com.stc.connector.hl7.schematron.SchematronValidatorFactory.
  getSchematronValidatorFactory();
```

Obtaining the Validator Object

Below is a sample call to the `getDefaultValidator` method.

```
com.stc.connector.hl7.schematron.SchematronValidator
validator = factory.getDefaultValidator( domSource );
```

In the above example, `domSource` is the DOMSource object of the schematron XML.

Performing the Validation

Below is a sample call to the `validate` method.

```
com.stc.connector.hl7.schematron.ValidationOutput
   output = validator.validate( dataSrc );
```

In the above example, `dataSrc` is the source of the payload. The payload can be an entire V3 XML document or a CDA document.

The `ValidationOutput` object contains the resulting XML document as well as a method `isValid()`, which returns values when the validation has passed or failed.

Adding and Configuring a TCP/IP HL7 Adapter in a Connectivity Map

All Adapters contain a set of properties that are unique to that Adapter type. When you add a TCP/IP HL7 Adapter to a Connectivity Map, you can modify the properties for that specific Adapter. When you add a TCP/IP HL7 External System in the Project’s Environment, you can modify the properties for that Adapter type for all Projects that use that Environment.

You can configure the TCP/IP HL7 Adapter properties in the following locations:

- **Connectivity Map**: These properties most commonly apply to a specific component Adapter, and may vary from other Adapters of the same type in the Project.
- **Environment Explorer**: These properties are commonly global, applying to all Adapters of the same type in the Project. The properties are shared by all Adapters in the TCP/IP HL7 External System window.
- **Collaboration**: Many TCP/IP HL7 Adapter properties can also be set from a Collaboration, in which case the settings override the corresponding properties in the Adapter's configuration file. Any properties that are not overridden retain their configured default settings.

Adding a TCP/IP HL7 External Application to a Connectivity Map

To create a TCP/IP HL7 Adapter you must first add a TCP/IP HL7 External Application to the Connectivity Map. A TCP/IP HL7 Adapter is automatically created when you link a TCP/IP HL7 External Application and a Service. Services are containers for Java Collaborations, Business Processes, Data Integrator processes, and so on.

▼ To Add a TCP/IP HL7 External Application

1. **Create a Connectivity Map for the Project, and add a Service to the Connectivity Map.**

2. **On the Connectivity Map toolbar, click the External Applications icon.**

3. **Select HL7 External Application from the menu.**

   A TCP/IP HL7 External Application icon appears on the Connectivity Map toolbar.

4. **Drag the new HL7 External Application icon from the toolbar onto the Connectivity Map canvas.**

5. **To bind the External Application with the Service, do one of the following:**

   - **If messages are entering from the HL7 system, drag a link from the HL7 External Application to the Service.**
   - **If messages are being sent from the Service to the HL7 system, drag a link from the Service to the HL7 External Application.**

     The TCP/IP HL7 Adapter appears on the link.

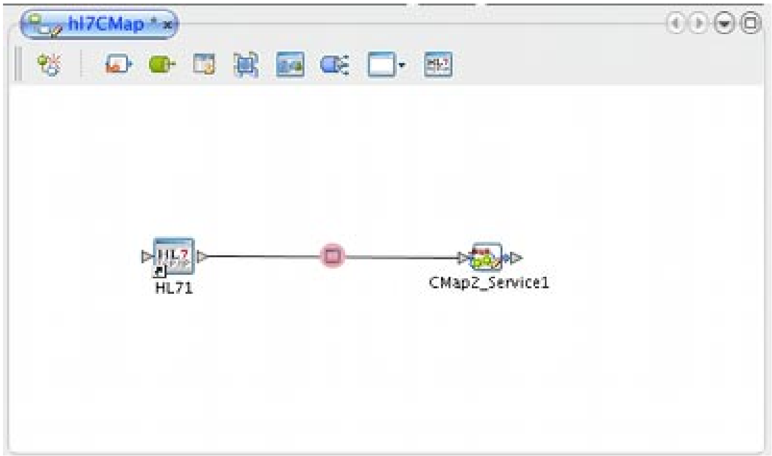


FIGURE 1 Adapter Location

6 **Modify the Adapter properties, as described in "Modifying the TCP/IP HL7 Adapter Properties in the Connectivity Map" on page 29.**

Modifying the TCP/IP HL7 Adapter Properties in the Connectivity Map

You can modify an Adapter's properties after it is created in the Connectivity Map. The properties you modify in the Connectivity Map apply only to the specific Adapter you are configuring. For information on modifying system-wide Adapter properties, see "Configuring Sun Adapter for TCP/IP HL7 Environment Properties" on page 91.

Tip – A description of each parameter is displayed in the Description box when that parameter is selected, providing an explanation of any required settings or options. Properties are also described in the following topics:

- "TCP/IP HL7 V2 Adapter Inbound Connectivity Map Properties" on page 30
- "TCP/IP HL7 V2 Adapter Outbound Connectivity Map Properties" on page 49
- "TCP/IP HL7 V3 Adapter Inbound Connectivity Map Properties" on page 64
- "TCP/IP HL7 V3 Adapter Outbound Connectivity Map Properties" on page 79

▼ Modifying Adapter Properties in the Connectivity Map

1 **From the Connectivity Map, double-click the Adapter icon located in the link between the TCP/IP HL7 External Application and the Service.**

The Adapter Properties Editor appears, and displays either inbound or outbound properties depending on the link to the Service.

2 **In the explorer panel on the left of the Properties Editor, expand the tree until you see the category you want to modify and then select that category.**

For example, to modify server port binding properties, expand TCPIP Inbound Settings and then select Server Port Binding.

3 **Modify a property by either selecting a new value from a drop-down list (if available) or by typing a new value in the property field.**

Tip – Click on the ellipsis button next to a property field to open a separate configuration dialog box. This is helpful for large values that cannot be fully displayed in the property field. Enter the property value in the dialog box and click OK. The value appears in the parameter's property field.

---

4 **(Optional) To record notes and information about the currently selected property, click inside the Comments box in the lower left of the editor and enter the text.**

This information is saved for future reference.

5 **When you are done configuring the properties, click OK.**

TCP/IP HL7 V2 Adapter Inbound Connectivity Map Properties

The TCP/IP HL7 V2 inbound adapter configuration properties are organized into the following sections on the Properties Editor accessed from the Connectivity Map:

- "General Inbound Settings – TCP/IP HL7 V2 Inbound Adapter" on page 31
- "TCPIP Inbound Settings — TCP/IP HL7 V2 Inbound Adapter" on page 32
- "TCPIP Inbound Settings - Server Port Binding — TCP/IP HL7 V2 Inbound Adapter" on page 35
- "TCPIP Inbound Settings - Client Connection Establishment — TCP/IP HL7 V2 Inbound Adapter" on page 35
- "TCPIP Inbound Settings - Inbound Connection Management — TCP/IP HL7 V2 Inbound Adapter" on page 35
- "TCPIP Inbound Schedules - Listener Schedule — TCP/IP HL7 V2 Inbound Adapter" on page 36
- "TCPIP Inbound Schedules - Service Schedule TCP/IP HL7 V2 Inbound Adapter" on page 38

General Inbound Settings – TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the TCP/IP HL7 V2 inbound adapter properties that appear on the General Inbound Settings page of the Properties Editor accessed from the Connectivity Map.

TABLE 2 Connectivity Map - General Inbound Settings

Name	Description
Max Data Size	A number that indicates the maximum amount of data that the programs can hold internally. The valid range is a numeric value from **1** to **2147483647** bytes (2GB), which is the maximum value of a Java integer.
Scope Of State	The scope of the state object, which is a Message Library node. Select one of the following options for this property: ■ **Resource Adapter Level** – The state has the same life cycle as the resource adapter. ■ **Connection Level** – The state has the same life cycle as the connection. ■ **OTD Level** – The state has the same life cycle as the Message Library object. This scope represents the life cycle of the state.
Dedicated Session Mode	An indicator of whether the server Dedicated Session Mode is enabled. When the server Dedicated Session Mode is enabled, the current client's request exclusively holds the server port to which it connects. The next client's request to the same port is blocked or rejected until the previous request concludes and releases the connection. Select **`true`** to enable the Dedicated Session Mode, or select **`false`** to disable the Dedicated Session Mode.

TCPIP Inbound Settings — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties on the TCPIP Inbound Settings page of the Properties Editor accessed from the Connectivity Map. These properties configure the Java socket and server socket options.

TABLE 3 Connectivity Map - TCPIP Inbound Settings

Name	Description
Connection Type	The way the adapter establishes the TCP/IP connection. Select one of the following options: ■ **Client** – The adapter connects to an external server (host/port) to establish the connection. The adapter is in active mode. ■ **Server** – The adapter waits and listens on a certain port for an incoming connection request from an external client. Once the request is received, the adapter accepts the request and establishes the connection. The adapter is in passive mode. Server is the default setting. Unless you specifically require Client mode, leave the default value.
ServerSO Timeout	The value (in milliseconds) of the `SO_TIMEOUT` parameter for `ServerSocket`. The timeout must be greater than zero (**0**). A timeout of zero is interpreted as an infinite timeout. This value is used for the `ServerSocket.accept()` method. When this option is set to a non-zero timeout, calling `accept()` for this `ServerSocket` will block for the configured length of time. If the timeout expires, a `java.net.SocketTimeoutException` (or `java.net.InterruptedIOException`) is thrown, but the `ServerSocket` remains valid. Enable this option prior to entering the blocking operation. This property is only used when the `Connection Type` property is set to Server.
Server Socket Factory Implementation Class Name	The name of the Java class that implements the server socket factory. This class is used to create the server socket. If you have provided your own server socket implementation, enter the name of the Java class that contains this implementation here. The factory implementation class must implement the `com.stc.connector.tcpip.model.factory.TCPIPSocketFactory` interface. A default interface, `com.stc.connector.tcpip.model.factory.TCPIPSocketFactoryImpl`, is provided.

TABLE 3 Connectivity Map - TCPIP Inbound Settings *(Continued)*

Name	Description
Keep Alive	An indicator of whether the client's SO_KEEPALIVE option is enabled or disabled. Select **`true`** to enable SO_KEEPALIVE; otherwise, select **`false`**. When the option is enabled for a TCP socket and no data has been exchanged across the socket in either direction for two hours, TCP automatically sends a KEEPALIVE probe to the peer (the actual value is implementation dependent). This probe is a TCP segment to which the peer must respond. One of three responses is expected: 1. The peer responds with the expected ACK. The application is not notified (since everything is OK). TCP will send another probe following another two hours of inactivity. 2. The peer responds with an RST, which tells the local TCP that the peer host has crashed and rebooted. The socket is closed. 3. There is no response from the peer. The socket is closed. The purpose of this option is to detect if the peer host has crashed. This is used for the accepted client Socket. **Note –** For some properties, the server socket itself does not have direct property settings associated with it. Instead, the properties map to the accepted client socket.
Receive Buffer Size	A number indicating the receive buffer size. This is the value of the SO_RCVBUF option for the current socket, which is the buffer size used by the operating system for input on this socket. It provides an estimate of the size of the underlying buffers used by the platform for incoming network I/O. When used in set mode, this is a suggestion for the kernel from the application regarding the size of buffers to use for the data to be received over the socket. When used in get mode, this must return the actual size of the buffer used by the platform when receiving data on this socket.
Send Buffer Size	A number indicating the send buffer size. This is the value of the SO_SNDBUF option for the current socket, which is the buffer size used by the operating system for output on this socket. It provides an estimate of the size of the underlying buffers used by the platform for outgoing network I/O. When used in set mode, this is a suggestion for the kernel from the application regarding the size of buffers to use for the data to be sent over the socket. When used in get mode, this must return the actual size of the buffer used by the platform when sending out data on this socket.

TABLE 3 Connectivity Map - TCPIP Inbound Settings *(Continued)*

Name	Description
SoLinger	An indicator of whether the adapter performs a “linger-on-close” timeout. This option disables or enables an immediate return from a call to the `close()` method for a TCP Socket. To enable the linger-on-close timeout, select **`true`**; otherwise, select **`false`**. If you enable this property, specify the maximum length of the timeout in the `SoLinger Timeout` property.
SoLinger Timeout	The server’s linger–on–close timeout in seconds. Use `SoLinger Timeout` when `SoLinger` is set to true (see the description for `SoLinger` above). You can specify an integer between -1 and 65535. The default is -1 seconds, which indicates that the `SoLinger` option is disabled. When `SoLinger` is set to **`true`**, the `SoLinger Timeout` value indicates the following: ■ **A non-zero integer** means that calling `close()` will block pending the transmission and acknowledgement of all data written to the peer. When all data is written, the socket is closed gracefully. Upon reaching the linger timeout value specified here, the socket is closed forcefully with a TCP RST. If the specified timeout value exceeds 65,535 it will be reduced to 65,535. ■ **A zero integer** means that a forceful close is performed immediately.
SoTimeout	The value of the SoTimeout in milliseconds. This is used for the accepted client socket. You can enter a value greater than or equal to zero (**0**). When set to zero (**0**), the timeout is infinite. With this option set to a non-zero value, calling the `read()` method on the input stream associated with this socket will block for only the configured length of time. If the timeout expires, a `java.io.InterruptedIOException` or `java.net.SocketTimeoutException` is thrown, but the socket remains valid. Enable this option prior to entering the blocking operation.
TcpNoDelay	An indicator of whether data packets that are smaller than the maximum transfer unit (MTU) size are sent out immediately over the network (this refers to Nagle’s algorithm). Select one of the following options: ■ **True** – Indicates that the server allows data packets that are smaller than the MTU size to be sent out immediately over the network. This can improve performance for higher-speed networks. ■ **False**– Indicates that the server does not allow data packets that are less than the MTU size be sent out immediately over the network. This is used for the accepted client socket.

TCPIP Inbound Settings - Server Port Binding — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties that appear on the Server Port Binding page of the Properties Editor accessed from the Connectivity Map. The properties define the server port binding retry options. This section is only used when the `Connection Type` under TCPIP Inbound Settings is set to Server.

TABLE 4 Connectivity Map - TCPIP Inbound Settings - Server Port Binding

Name	Description
Max Binding Retry	The maximum number of times the adapter attempts to bind to the specified TCP/IP port on the localhost. This value must be an integer.
Retry Binding Interval	The length of time (in milliseconds) the adapter waits between attempts to bind to the specified TCP/IP port on the localhost.

TCPIP Inbound Settings - Client Connection Establishment — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the property that appears on the Client Connection Establishment page of the Properties Editor accessed from the Connectivity Map. This property defines a wait time before connecting to the external system. This section is only used when the `Connection Type` under TCPIP Inbound Settings is set to Client.

TABLE 5 Connectivity Map - TCPIP Inbound Settings - Client Connection Establishment

Name	Description
Time to Wait Before Attempting Connection	The length of time (in milliseconds) that the adapter waits before attempting to connect to the external system.

TCPIP Inbound Settings - Inbound Connection Management — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties that appear on the Inbound Connection Management page of the Properties Editor accessed from the Connectivity Map. These properties manage the connection to inbound systems. For example, these properties include the connection pool and the life cycle of the accepted connection.

TABLE 6 Connectivity Map - TCPIP Inbound Settings - Inbound Connection Management

Name	Description
Max Connection Pool Size	The maximum number of concurrent connections allowed for the specific listener or monitor that is listening or monitoring the specified TCP/IP port. **0** (zero) indicates that there is no limit. This value indicates the capability or availability of this server's services. Each connection request from a client gains one concurrent connection. This value also indicates the maximum number of clients that can concurrently connect to this server's services and can be served by the specific listener or monitor at the same time.
Scope Of Connection	The scope of the accepted connection that is used by the adapter. Select one of the following options: ■ **Resource Adapter Level** – The resource adapter closes the connection upon request (by way of `ClosureCommandMessage`) so the connection may "keep alive" during multiple executions of the Collaboration. ■ **Collaboration Level** – The resource adapter closes the connection once the Collaboration has been executed so the connection has the same life cycle as the Collaboration.
Close Notification	A String indicating the trigger value that notifies the server to close the connection. When the server receives a notification with content that matches this parameter's value, the server safely closes the connection and cancels any corresponding schedules. The default value is **QUIT**.
Idle Timeout	The length of time (in milliseconds) for inactivity of the requestor (client). The adapter attempts to detect in/out activity from the client. If there is no client activity for a specified time period, then the connection is closed from the server side to release the resource. To disable idle timeout checking, specify **0** (zero) for this parameter.

TCPIP Inbound Schedules - Listener Schedule — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties that appear on the Listener Schedule page of the Properties Editor accessed from the Connectivity Map. These properties configure the scheduler used by the inbound TCP/IP server. The server waits for a new client connection establishment request. These parameters are used to configure the listener.

Two Java EE schedulers are available, both of which provide the functionality required by the inbound TCP/IP Server.

- **Timer Service** – Available for Java EE, this scheduler is configured using the At Fixed Rate, Delay, and Period properties.
- **Work Manager** – Available for Java EE (JCA 1.5 and above), this scheduler is configured using the Delay and Period properties.

TABLE 7 Connectivity Map - TCPIP Inbound Schedules - Listener Schedule

Name	Description
Scheduler	The scheduler type for this inbound communication. Select one of the following options: ■ **Timer Service** – The task is scheduled through the Java EE Timer Service. Timer Service is supported by Java EE. ■ **Work Manager** – The task is scheduled through the Java EE Work Manager. Work Manager is supported by Java EE (JCA 1.5 and above). If your container doesn't support JCA Work Manager, select **Timer Service**.
Schedule Type	This property, though visible from the Properties Editor, is disabled. The only available schedule type is Repeated, indicating that the task is scheduled for repeated execution at regular intervals defined by the Period property.
Delay	An integer indicating the length of time (in milliseconds) before the task is executed. This property applies to both the Timer Service and the Work Manager.
Period	An integer indicating the length of time (in milliseconds) between successive task executions. This property applies to both the Timer Service and the Work Manager.
At Fixed Rate	An indicator of whether a fixed-rate execution or fixed-delay execution is used. This property applies to the Timer Service configuration only. Select **true** to indicate fixed-rate; select **false** to indicate fixed-delay. ■ **Fixed-Rate** – Each execution is scheduled relative to the scheduled time of the initial execution. If an execution is delayed for any reason (such as garbage collection or other background activity), two or more executions occur in rapid succession to "catch up." In the long run, the frequency of execution is exactly the reciprocal of the specified period, assuming the system clock underlying Object.wait(long) is accurate. ■ **Fixed-Delay** – Each execution is scheduled relative to the actual time of the previous execution. If an execution is delayed for any reason (such as garbage collection or other background activity), subsequent executions are delayed as well. As a result, the frequency of execution is generally slightly lower than the reciprocal of the specified period, assuming the system clock underlying Object.wait(long) is accurate.

TCPIP Inbound Schedules - Service Schedule TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties that appear on the Service Schedule page of the Properties Editor accessed from the Connectivity Map. These properties configure the scheduler used by the TCP/IP server that executes the business tasks (Collaboration rules) over the existing connection. This scheduler affects the actual business rules you define.

You can use either of the following two Java EE schedulers, both of which provide the functionality required by the inbound TCP/IP server.

- **Timer Service** – Available for Java EE, this scheduler is configured using the `At Fixed Rate`, `Delay`, `Period`, and `Schedule Type`properties.
- **Work Manager** – Available for Java EE (JCA 1.5 and above), this scheduler is configured using the `Delay`, `Period`, and `Schedule Type`properties.

TABLE 8 Connectivity Map - TCPIP Inbound Schedules - Service Schedule

Name	Description
Scheduler	The scheduler type for this inbound communication. Select one of the following options: ■ **Timer Service** – The task is scheduled through the Java EE Timer Service. Timer Service is supported by Java EE. ■ **Work Manager** – The task is scheduled through the Java EE Work Manager. Work Manager is supported by Java EE (JCA 1.5 and above). If your container doesn't support JCA Work Manager, select Timer Service.
Schedule Type	An indicator of whether the task is scheduled to occur once or be repeated. This property applies to both the Timer Service and the Work Manager. Select one of the following options: ■ **OneTime** – The task is scheduled for one-time execution. ■ **Repeated** – The task is scheduled for repeated execution at regular intervals defined by `Period` property, described below.
Delay	An integer indicating the length of time (in milliseconds) before the task is executed. This property applies to both the Timer Service and the Work Manager.
Period	An integer indicating the length of time (in milliseconds) between successive task executions. This property applies to both the Timer Service and the Work Manager. This is used when the `Schedule Type` property is set to **`Repeated`**.

TABLE 8 Connectivity Map - TCPIP Inbound Schedules - Service Schedule *(Continued)*

Name	Description
At Fixed Rate	An indicator of whether a fixed-rate execution or fixed-delay execution is used. This property applies to the Timer Service configuration only, and is used when the `Schedule Type` property is set to **`Repeated`**. Select **`true`** to indicate fixed-rate; select **`false`** to indicate fixed-delay. ■ **Fixed-Rate** – Each execution is scheduled relative to the scheduled time of the initial execution. If an execution is delayed for any reason (such as garbage collection or other background activity), two or more executions occur in rapid succession to "catch up." In the long run, the frequency of execution is exactly the reciprocal of the specified period, assuming the system clock underlying `Object.wait(long)` is accurate. ■ **Fixed-Delay** – Each execution is scheduled relative to the actual time of the previous execution. If an execution is delayed for any reason (such as garbage collection or other background activity), subsequent executions are delayed as well. As a result, the frequency of execution is generally slightly lower than the reciprocal of the specified period, assuming the system clock underlying `Object.wait(long)` is accurate.

HL7 Acknowledgment — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties that appear on the HL7 Acknowledgment page of the Properties Editor accessed from the Connectivity Map. These properties define how the application acknowledgment events are handled.

TABLE 9 Connectivity Map - HL7 Acknowledgment

Name	Description
Acknowledgment Level	An indicator of whether the external application sends an Acknowledgement after successfully receiving a message, or after the message has been successfully committed to the application database. Select one of the following options: ■ **A** – Application acknowledgment. The acknowledgement is sent after the message is successfully and functionally processed by one receiving system. ■ **C** – Commit (accept) acknowledgment. The acknowledgement is sent when the message is successfully received.

TABLE 9 Connectivity Map - HL7 Acknowledgment *(Continued)*

Name	Description
eGate Sends App Acks	An indicator of whether the HL7 application acknowledgment sent to the external system is generated by the adapter or forwarded from the application server. Select one of the following options: - **true** – Indicates that the application server receives or creates the HL7 application acknowledgment and sends it to the adapter, which in turn forwards it to the external system. - **false** – Indicates that the adapter creates and sends the HL7 application acknowledgment directly to the external system. This property is used in the inbound Collaboration code.
Forward External Acks	An indicator of whether the HL7 application acknowledgment is forwarded to the application server. When an HL7 application acknowledgment is received, it is sometimes necessary to forward the contents of the HL7 application acknowledgment to the application server (as data). This property is used for inbound Collaboration code. Select `true` if the adapter forwards HL7 application acknowledgments from the external system to the application server for processing; otherwise select `false`.
Timeout For Delayed Ack	A number indicating the timeout value for delayed ACK in milliseconds. This property is used in the inbound Collaboration code.

Lower Layer Protocol — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties that appear on the Lower Layer Protocol page of the Properties Editor accessed from the Connectivity Map. These properties define the Lower Layer Protocol (LLP) configuration.

TABLE 10 Connectivity Map - Lower Layer Protocol

Name	Description
LLP Type	The lower layer protocol (LLP) type. Select one of the following options: - **MLLP** (Minimal Lower Layer Protocol) - **HLLP** (Hybrid Lower Layer Protocol) - **MLLP v2.0** (Minimal Lower Layer Protocol v2.0) For more information on the available envelope types, see "Lower Layer Protocol" on page 23.
Start Block Character	The first envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. Unless there is a conflict, the value should be ASCII VT (decimal 11).

TABLE 10 Connectivity Map - Lower Layer Protocol *(Continued)*

Name	Description
End Data Character	The second to the last envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. Unless there is a conflict, the value should be ASCII FS (decimal 28).
End Block Character	The last envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. To strictly comply with the HL7 Standard, this property must be set to a carriage return (decimal 13).
HLLP Checksum Enabled	An indicator of whether the HLLP Checksum is enabled or disabled. Select **`true`** to enable checksum values; otherwise select **`false`**.
Max Number of Retries	The maximum number of times the adapter tries to send a message upon receiving the MLLP v2.0 Negative Commit Acknowledgement from the peer before giving up. This property is used by the adapter in outbound mode. Enter any integer.

Sequence Number Protocol — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the property that appears on the Sequence Number Protocol page of the Properties Editor accessed from the Connectivity Map. This property enables or disables HL7 sequence numbering, which is used to help prevent duplication of data.

TABLE 11 Connectivity Map - Sequence Number Protocol

Name	Description
Sequence Number Enabled	An indicator of whether sequence numbering is enabled or disabled. Enabling sequence numbering helps prevent duplication of data. Select **`true`** to enable sequence numbering; otherwise select **`false`**.

HL7 MSH Segment — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties that appear on the HL7 MSH Segment page of the Properties Editor accessed from the Connectivity Map. These properties define the configuration of the MSH segment of the HL7 message. For more information about this segment, refer to the HL7 specification (`http://www.hl7.org`).

TABLE 12 Connectivity Map - HL7 MSH Segment

Name	Description
Field Separator	The character that separates the segment ID and the first real field. This value defines the character that is used as a separator for the rest of the message and is the first field in the HL7 MSH segment (MSH-01). The value is a decimal ASCII number, and the allowed range is 1 to 127. The default setting is 124, which is the pipe character (|).
Encoding Characters	Encoding characters in the following order: ■ Component separator ■ Repetition separator ■ Escape character ■ Subcomponent separator This is the second field in the HL7 MSH segment (MSH-02). The default is **`^~\&`** (ASCII 94, 126, 92, and 38) respectively.
Sending Application	A user-defined value for the sending application among other applications within the network enterprise. The network enterprise consists of the applications that participate in the exchange of HL7 messages within the enterprise. The default is **`Sun HL7 adapter`**. This is the third field in the HL7 MSH segment (MSH-03).
Sending Facility	A user-defined value that further identifies the sending application among multiple identical instances of the application running on behalf of different organizations. The default is **`Sun HL7 adapter`**. This is the fourth field in the HL7 MSH segment (MSH-04).
Receiving Application	A user-defined value for the receiving application among other applications within the network enterprise. The default value is **`Sun HL7 adapter`**. This is the fifth field in the HL7 MSH segment (MSH-05).
Receiving Facility	A user-defined value that further identifies the receiving application among multiple identical instances of the application running on behalf of different organizations. The default value is **`Sun HL7 adapter`**. This is the sixth field in the HL7 MSH segment (MSH-06).
Security	The implemented application level security features. This is the eighth field in the HL7 MSH segment (MSH-08).

TABLE 12 Connectivity Map - HL7 MSH Segment *(Continued)*

Name	Description
Processing ID	The subcomponent processing ID of the MSH-11 field. MSH-11 is used to indicate whether a message is processed as defined in the HL7 Application processing rules. Specify one of the following options: ■ **D** - The message is part of a debugging system. ■ **P** - The message is part of a production system. ■ **T** - The message is part of a training system. In some cases there may be an additional value, the *processing mode*, following the initial value. This value can be A (archive), R (restore from archive), or I (initial load).
Version ID	The HL7 version as displayed in HL7 Table 0104 - Version ID. This value is matched by the receiving system to its own version to ensure that messages are interpreted correctly. The default value is **`2.5`**. This is the 12th field in the HL7 MSH segment (MSH-12).
Country Code	A code that indicates the country of origin for the message (see HL7 Table 0399). Use the 3-character (alphabetic) form of ISO 3166. This value is used to specify default elements in a message, such as currency. The default value is **`USA`**. This is the 17th field in the HL7 MSH segment (MSH-17).
Character Set	The character set(s) used by the messages (see HL7 Table 0211). If the field is left blank, the character set is assumed to be the 7-bit ASCII set. The default value is **`8859/1`** (printable 7-bit ASCII character set). This is the 18th field in the HL7 MSH segment (MSH-18).
Principal Language of Message	The 2-character ISO 639 alphabetic code that specifies the principal language of the message. This is the 19th field in the HL7 MSH segment (MSH-19).
Alternate Character Set Handling Scheme	The value for the alternate character set handling scheme to be used when any alternative character sets are used and a special handling scheme is necessary (see HL7 Table 0356). Possible values are **`ISO 2022-1994`**, **`2.3`**, or **`<null>`** (blank). Leaving the field blank indicates that no character set switching will occur. This is the 20th field in the HL7 MSH segment (MSH-20).
Conformance Statement ID	A unique identifier that applies to a query's conformance statement. It can also be used as a Message Profile Identifier to assert constancy with a message profile (grammar, syntax, usage, and so on). This is the 21st field in the HL7 MSH segment (MSH-21).

TABLE 12 Connectivity Map - HL7 MSH Segment *(Continued)*

Name	Description
Validate MSH	An indicator of whether to validate the MSH segment of the data message (for inbound) and the MSH segment of the ACK (for outbound). Select **`true`** if you want the Collaboration to validate the MSH segment; otherwise select **`false`**. This parameter is used in Collaboration code. **Note –** This property does not affect structural validation of the entire HL7 message itself. Structural validation is always performed.

HL7 SFT Segment — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties that appear on the HL7 SFT Segment page of the Properties Editor accessed from the Connectivity Map. These properties define the configuration of the SFT segment of the HL7 message, which provides additional information about one or more software products used as sending applications. The primary purpose of this segment is for diagnostic use. There may be additional uses per site-specific agreements. For more information about this segment, refer to the HL7 specification (`http://www.hl7.org`).

Note – The SFT segment is available starting with HL7 version 2.5.

TABLE 13 Connectivity Map - HL7 SFT Segment

Name	Description
Enable	An indicator of whether the SFT optional segment is enabled in the ACK. Select **`true`** to enable the segment; otherwise select **`false`**. **Note –** If `Enable` is set to true, and the HL7 version is not configured as 2.5, the adapter will error upon startup.
Software Vendor Organization	The name of the company that publishes or distributes the sending software that created the transaction. This field identifies the vendor responsible for maintaining the application. The purpose of this field, along with the remaining fields in this segment, is to provide a more complete profile of the sending applications. This is the first field in the HL7 SFT segment (SFT-01).

TABLE 13 Connectivity Map - HL7 SFT Segment *(Continued)*

Name	Description
Software Certified Version or Release Number	The latest software version number or release number for the sending system, which helps provide a more complete profile of the application that is sending or receiving HL7 messages. Version numbers are important in identifying the specific release of an application. In some situations, the receiving application validates the software certified version or release number against a list of certified versions or releases of the particular software. This helps determine whether the sending application adheres to specific business rules required by the receiving application. Alternatively, the software may perform different processing, depending on the version of the sending software. This is the second field in the HL7 SFT segment (SFT-02).
Software Product Name	The name of the software product that submitted the transaction. The default value is **`Sun TCP/IP HL7 adapter Intelligent Adapter`**. This is the third field in the HL7 SFT segment (SFT-03).
Software Binary ID	The unique software binary ID. Software binary IDs are issued by a vendor for each unique software version instance. These IDs are used to differentiate between multiple versions of the same software. Identical primary IDs indicate that the software is identical at the binary level, but configuration settings may differ. This is the fourth field in the HL7 SFT segment (SFT-04).
Software Product Information	Any additional information about the sending application for more complete identification. This could include a description of the software application, configuration settings, modifications made to the software, and so on. This information is used for diagnostic purposes and provides greater flexibility in identifying the application software. This is the fifth field in the HL7 SFT segment (SFT-05).
Software Install Date	The date on which the submitting software was installed at the sending site. The software installation date on its own can often provide key information about the behavior of the application. This is the sixth field in the HL7 SFT segment (SFT-06).

Communication Control — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties that appears on the Communication Control page of the Properties Editor accessed from the Connectivity Map. These properties define how data is transferred (that is, sent and received) over the TCP/IP connection.

TABLE 14 Connectivity Map - Communication Control

Name	Description
Time To Wait For A Response	The amount of time (in milliseconds) that the adapter waits for a response from the external system before taking recourse action (see Action on No Response in HL7 Recourse Action). Any data from the external system is considered a response. This property corresponds to the initial read/receive operation timeout. Once a response is received, the subsequent read/receive operation uses the value specified for SoTimeout (see TCPIP Inbound Settings). A value of **0** (zero) indicates an infinite timeout.
Max Empty Read Retry	The maximum number of times the adapter attempts to read data from the external system after the read/receive operation returns nothing. This applies to the read or receive operation after a response starts to arrive. Empty Read means that a timeout occurs on the read/receive operation, which uses the SoTimeout parameter in the TCPIP Server Base Settings section as the timeout setting (see TCPIP Inbound Settings). The corresponding recourse action is specified by the Action on Max Failed Read Retry (see HL7 Recourse Action).
Max No Response	The maximum number of response timeouts the adapter allows while waiting for data from the external system before taking recourse action (see Action on Max No Response in HL7 Recourse Action). This property is used in the inbound Collaboration code. It is only used by outbound adapters and works in conjunction with the Resend option of the Action on No Response property. It configures the adapter to resend the last message for the specified maximum number of times before the subsequent recourse action is taken.
Max NAK Receive Retry	The maximum number of negative acknowledgments (NAKs) the adapter receives before taking recourse action (see Action on Max Nak Received in HL7 Recourse Action). This property is used for the inbound Collaboration code.
Max NAK Send Retry	The maximum number of negative acknowledgments (NAKs) the adapter sends before taking recourse action (see Action on Max Nak Sent in HL7 Recourse Action). This property is used in the inbound Collaboration code.
Max Canned NAK Send Retry	The maximum number of canned negative acknowledgments that the adapter sends before taking recourse action (see Action on Max Nak Sent in HL7 Recourse Action). A value of **0** (zero) indicates that the adapter will not attempt to create or send a canned NAK.
Enable Journaling	An indicator of whether message journaling is enabled. To enable message journaling, select **true**; otherwise select **false**. This property is used for inbound Collaboration code.

HL7 Recourse Action — TCP/IP HL7 V2 Inbound Adapter

The following table lists and describes the properties that appear on the HL7 Recourse Action page of the Properties Editor accessed from the Connectivity Map. These properties define the actions the adapter takes when operations occur outside the configured constraints.

TABLE 15 Connectivity Map - HL7 Recourse Action

Name	Description
Action on No Response	The action the adapter takes when no ACK is received from the external system in the allotted time. The amount of time is determined by the `Time To Wait For A Response` property (see Communication Control). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Resend** – The adapter attempts to resend the message to the external system. The Resend option is only allowed when sequence numbering is in effect. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for inbound Collaboration code.
Action on Max No Response	The action the adapter takes when it attempts to send a message to the external system the maximum allowed number of times and does not receive any response (HL7 Application Acknowledgement) from the external system. The maximum number times the adapter sends a message without receiving a response is determined by the `Max No Response` property (see Communication Control). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for inbound Collaboration code.
Action on Max Failed Read Retry	The action the adapter takes after it has reached the empty read limit set by the `Max Empty Read Retry` property. This property is used by inbound adapters only. Select one of the following recourse options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for inbound Collaboration code.

TABLE 15 Connectivity Map - HL7 Recourse Action *(Continued)*

Name	Description
Action on Nak Received	The action the adapter takes when it receives an HL7 Application NAK from the external system. Select one of the following options: ■ **Resend** – The adapter attempts to resend the message to the external system. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. ■ **Skip Message** – The adapter remains connected, but writes the message to an error queue. **Note –** Do not set both the `Action On NAK Received` and `Action On Max NAK Received` properties to Skip Message. This property is used for inbound Collaboration code.
Action on Max Nak Received	The action the adapter takes when the maximum number of HL7 Application NAKs have been received from the external system, as set by the `Max NAK Receive Retry` property (see Communication Control). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. ■ **Skip Message** – The adapter remains connected, but writes the message to an error queue. This property is used for inbound Collaboration code. **Note –** Do not set both the `Action On NAK Received` and `Action On Max NAK Received` properties to Skip Message.
Action on Max Nak Sent	The action the adapter takes when it has sent the maximum allowed number of NAKs to the external system, as set by the `Max NAK Send Retry` parameter (see Communication Control). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for inbound Collaboration code.

TCP/IP HL7 V2 Adapter Outbound Connectivity Map Properties

The TCP/IP HL7 V2 server outbound adapter configuration properties are organized into the following sections on the Properties Editor accessed from the Connectivity Map:

- "General Outbound Settings — TCP/IP HL7 V2 Outbound Adapter" on page 49
- "TCPIP Outbound Settings — TCP/IP HL7 V2 Outbound Adapter" on page 50
- "TCPIP Outbound Settings - Client Connection Establishment — TCP/IP HL7 V2 Outbound Adapter" on page 53
- "TCPIP Outbound Settings - Server Port Binding — TCP/IP HL7 V2 Outbound Adapter" on page 54
- "HL7 Acknowledgment — TCP/IP HL7 V2 Outbound Adapter" on page 55
- "Lower Layer Protocol — TCP/IP HL7 V2 Outbound Adapter" on page 55
- "Sequence Number Protocol — TCP/IP HL7 V2 Outbound Adapter" on page 56
- "HL7 MSH Segment — TCP/IP HL7 V2 Outbound Adapter" on page 57
- "HL7 SFT Segment — TCP/IP HL7 V2 Outbound Adapter" on page 59
- "Communication Control — TCP/IP HL7 V2 Outbound Adapter" on page 60
- "HL7 Recourse Action — TCP/IP HL7 V2 Outbound Adapter" on page 62

General Outbound Settings — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the TCP/IP HL7 V2 outbound adapter properties that appear on the General Outbound Settings page of the Properties Editor accessed from the Connectivity Map.

TABLE 16 Connectivity Map - General Outbound Settings

Name	Description
Max Data Size	A number that indicates the maximum amount of data that the programs can hold internally. The valid range is a numeric value from **1** to **2147483647** bytes (2GB), which is the maximum value of a Java integer.
Scope Of State	The scope of the state object, which is a Message Library node. Select one of the following options for this property: ■ **Resource Adapter Level** – The state has the same life cycle as the resource adapter. ■ **Connection Level** – The state has the same life cycle as the connection. ■ **OTD Level** – The state has the same life cycle as the Message Library object. This scope represents the life cycle of the state.

TCPIP Outbound Settings — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the properties on the TCPIP Outbound Settings page of the Properties Editor accessed from the Connectivity Map. These properties configure the Java socket and server socket options. For more information, see the Javadocs for Java SDK.

TABLE 17 Connectivity Map - TCPIP Outbound Settings

Name	Description
Connection Type	The way the adapter establishes the TCP/IP connection. Select one of the following options: ■ **Client** – The adapter connects to an external server (host/port) to establish the connection. The adapter is in active mode. ■ **Server** – The adapter waits and listens on a certain port for an incoming connection request from an external client. Once the request is received, the adapter accepts the request and establishes the connection. The adapter is in passive mode. Server is the default setting. Unless you specifically require Client mode, leave the default value.
ServerSo Timeout	The value (in milliseconds) of the `SO_TIMEOUT` parameter for `ServerSocket`. The timeout must be greater than zero (**0**). A timeout of zero is interpreted as an infinite timeout. This value is used for the `ServerSocket.accept()` method. When this option is set to a non-zero timeout, calling `accept()` for this `ServerSocket` will block for the configured length of time. If the timeout expires, a `java.net.SocketTimeoutException` (or `java.net.InterruptedIOException`) is thrown, but the `ServerSocket` remains valid. Enable this option prior to entering the blocking operation. This property is only used when the `Connection Type` property is set to Server.

TABLE 17 Connectivity Map - TCPIP Outbound Settings *(Continued)*

Name	Description
Keep Alive	An indicator of whether the client's SO_KEEPALIVE option is enabled or disabled. Select **true** to enable SO_KEEPALIVE; otherwise, select **false**. When the option is enabled for a TCP socket and no data has been exchanged across the socket in either direction for two hours, TCP automatically sends a KEEPALIVE probe to the peer (the actual value is implementation dependent). This probe is a TCP segment to which the peer must respond. One of three responses is expected: 1. The peer responds with the expected ACK. The application is not notified (since everything is OK). TCP will send another probe following another two hours of inactivity. 2. The peer responds with an RST, which tells the local TCP that the peer host has crashed and rebooted. The socket is closed. 3. There is no response from the peer. The socket is closed. The purpose of this option is to detect if the peer host has crashed. This is used for the accepted client Socket. **Note –** For some properties, the server socket itself does not have direct property settings associated with it. Instead, the properties map to the accepted client socket.
Receive Buffer Size	A number indicating the receive buffer size. This is the value of the SO_RCVBUF option for the current socket, which is the buffer size used by the operating system for input on this socket. It provides an estimate of the size of the underlying buffers used by the platform for incoming network I/O. When used in set mode, this is a suggestion for the kernel from the application regarding the size of buffers to use for the data to be received over the socket. When used in get mode, this must return the actual size of the buffer used by the platform when receiving data on this socket.
Send Buffer Size	A number indicating the send buffer size. This is the value of the SO_SNDBUF option for the current socket, which is the buffer size used by the operating system for output on this socket. It provides an estimate of the size of the underlying buffers used by the platform for outgoing network I/O. When used in set mode, this is a suggestion for the kernel from the application regarding the size of buffers to use for the data to be sent over the socket. When used in get mode, this must return the actual size of the buffer used by the platform when sending out data on this socket.

TABLE 17 Connectivity Map - TCPIP Outbound Settings *(Continued)*

Name	Description
SoLinger	An indicator of whether the adapter performs a "linger-on-close" timeout. This option disables or enables an immediate return from a call to the close() method for a TCP Socket. To enable the linger-on-close timeout, select **true**; otherwise, select **false**. If you enable this property, specify the maximum length of the timeout in the SoLinger Timeout property.
SoLinger Timeout	The server's linger–on–close timeout in seconds. Use SoLinger Timeout when SoLinger is set to true (see the description for SoLinger above). You can specify an integer between -1 and 65535. The default is -1 seconds, which indicates that the SoLinger option is disabled. When SoLinger is set to **true**, the SoLinger Timeout value indicates the following: ■ **A non-zero integer** means that calling close() will block pending the transmission and acknowledgement of all data written to the peer. When all data is written, the socket is closed gracefully. Upon reaching the linger timeout value specified here, the socket is closed forcefully with a TCP RST. If the specified timeout value exceeds 65,535 it will be reduced to 65,535. ■ **A zero integer** means that a forceful close is performed immediately.
SoTimeout	The value of the SoTimeout in milliseconds. You can enter a value greater than or equal to zero (**0**). When set to zero (**0**), the timeout is infinite. With this option set to a non-zero value, calling the read() method on the input stream associated with this socket will block for only the configured length of time. If the timeout expires, a java.io.InterruptedIOException or java.net.SocketTimeoutException is thrown, but the socket remains valid. Enable this option prior to entering the blocking operation.
TcpNoDelay	An indicator of whether data packets that are smaller than the maximum transfer unit (MTU) size are sent out immediately over the network (this refers to Nagle's algorithm). Select one of the following options: ■ **True** – Indicates that the server allows data packets that are smaller than the MTU size to be sent out immediately over the network. This can improve performance for higher-speed networks. ■ **False**– Indicates that the server does not allow data packets that are less than the MTU size be sent out immediately over the network. This is used for the accepted client socket.

TABLE 17 Connectivity Map - TCPIP Outbound Settings *(Continued)*

Name	Description
Socket Factory Implementation Class Name	The name of the Java class that implements the socket factory. This class is used to create the socket. If you have provided your own socket implementation, enter the name of the Java class that contains this implementation here. The factory implementation class must implement the `com.stc.connector.tcpip.model.factory.TCPIPSocketFactory` interface. The default value is `com.stc.connector.tcpip.model.factory.TCPIPSocketFactoryImpl`

TCPIP Outbound Settings - Client Connection Establishment — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the properties that appear on the Client Connection Establishment page of the Properties Editor accessed from the Connectivity Map. These properties configure how connections are established. This section is only used when the `Connection Type` under TCPIP Outbound Settings is set to Client.

TABLE 18 Connectivity Map - TCPIP Outbound Settings - Client Connection Establishment

Name	Description
Time To Wait Before Attempting Connection	The length of time (in milliseconds) the adapter waits before attempting to connect to the external system.
Always Create New Connection	An indicator of whether the adapter always attempts to create a new connection when a connection establishment request is received. Select on of the following options: ■ **true** – The adapter always attempts to create a new connection without attempting to match an existing connection. ■ **false** – The adapter attempts to match an existing connection (managed by the container).

TABLE 18 Connectivity Map - TCPIP Outbound Settings - Client Connection Establishment *(Continued)*

Name	Description
Auto Reconnect Upon Matching Failure	An indicator or whether to attempt to reconnect automatically when the adapter gets a matching connection from a container, even though this connection is not valid; for example, the external side of the connection is closed or reset due to the external application's logic. Select one of the following options: ■ **true** – The adapter discards the invalid matching connection and automatically attempts to reconnect using a new connection. ■ **false** – The adapter does not automatically attempt to reconnect using a new connection. Instead, the adapter defers the reconnect control to the user business rules. It is up to the business rules to detect this type of failure and act appropriately.
Max Connection Retry	The maximum number of times the adapter attempts to connect to a specific external TCP/IP destination (host/port) before giving up.
Retry Connection Interval	The length of time (in milliseconds) the adapter waits between attempts to connect to a specific external TCP/IP destination (host or port).

TCPIP Outbound Settings - Server Port Binding — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the properties that appear on the Server Port Binding page of the Properties Editor accessed from the Connectivity Map. The properties define the server port binding retry options. This section is only used when the `Connection Type` under TCPIP Outbound Settings is set to Server.

TABLE 19 Connectivity Map - TCPIP Outbound Settings - Server Port Binding

Name	Description
Max Binding Retry	The maximum number of times the adapter will attempt to bind to the specified TCP/IP port on the localhost before giving up.
Retry Binding Interval	The amount of time (in milliseconds) that the adapter waits between attempts to bind to the specified TCP/IP port on the localhost.

HL7 Acknowledgment — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the properties that appear on the HL7 Acknowledgment page of the Properties Editor accessed from the Connectivity Map. These properties define how the application acknowledgment events are handled.

TABLE 20 Connectivity Map - HL7 Acknowledgement

Name	Description
Acknowledgment Level	An indicator of whether the external application sends an Acknowledgement after successfully receiving a message, or after the message has been successfully committed to the application database. Select one of the following options: ■ **A** – Application acknowledgment. The acknowledgement is sent after the message is successfully and functionally processed by one receiving system. ■ **C** – Commit (accept) acknowledgment. The acknowledgement is sent when the message is successfully received.
eGate Sends App Acks	An indicator of whether the outbound Collaboration is in outbound delayed ACK mode; that is, the outbound adapter connects to an external system that communicates as a Delayed ACK receiver and sends two ACKs to the adapter. Select one of the following options: ■ **true** – Indicates that the adapter is expecting a Delayed ACK (two ACKS). ■ **false** – Indicates that the adapter does not expect a Delayed ACK.
Forward External Acks	An indicator of whether the HL7 application acknowledgment is forwarded to the application server. When an HL7 application acknowledgment is received, it is sometimes necessary to forward the contents of the HL7 application acknowledgment to the application server (as data). This property is used for outbound Collaboration code. Select **`true`** if the adapter forwards HL7 application acknowledgments from the external system to the application server for processing; otherwise select **`false`**.
Timeout For Delayed Ack	A number indicating the timeout value for delayed ACK in milliseconds. This property is used in the outbound Collaboration code.

Lower Layer Protocol — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the properties that appear on the Lower Layer Protocol page of the Properties Editor accessed from the Connectivity Map. These properties define the Lower Layer Protocol (LLP) configuration.

TABLE 21 Connectivity Map - Lower Layer Protocol

Name	Description
LLP Type	**MLLP** is the configured default value. The lower layer protocol (LLP) type. Select one of the following options: ■ **MLLP** (Minimal Lower Layer Protocol) ■ **HLLP** (Hybrid Lower Layer Protocol) ■ **MLLP v2.0** (Minimal Lower Layer Protocol v2.0) For more information on the available envelope types, see "Lower Layer Protocol" on page 23.
Start Block Character	The first envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. Unless there is a conflict, the value should be ASCII VT (decimal 11).
End Data Character	The second to the last envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. Unless there is a conflict, the value should be ASCII FS (decimal 28).
End Block Character	The last envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. To strictly comply with the HL7 Standard, this property must be set to a carriage return (decimal 13).
HLLP Checksum Enabled	An indicator of whether the HLLP Checksum is enabled or disabled. Select **`true`** to enable checksum values; otherwise select **`false`**.
Max Number of Retries	The maximum number of times the adapter tries to send a message upon receiving the MLLP v2.0 Negative Commit Acknowledgement from the peer before giving up. This property is used by the adapter in outbound mode. Enter any integer.

Sequence Number Protocol — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the property that appears on the Sequence Number Protocol page of the Properties Editor accessed from the Connectivity Map. This property enables or disables HL7 sequence numbering, which is used to help prevent duplication of data.

TABLE 22 Connectivity Map - Sequence Number Protocol

Name	Description
Sequence Number Enabled	An indicator of whether sequence numbering is enabled or disabled. Enabling sequence numbering helps prevent duplication of data. Select **`true`** to enable sequence numbering; otherwise select **`false`**.

HL7 MSH Segment — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the properties that appear on the HL7 MSH Segment page of the Properties Editor accessed from the Connectivity Map. These properties define the configuration of the MSH segment of the HL7 message. For more information about this segment, refer to the HL7 specification (`http://www.hl7.org`).

TABLE 23 Connectivity Map - HL7 MSH Segment

Name	Description
Field Separator	The character that separates the segment ID and the first real field. This value defines the character that is used as a separator for the rest of the message and is the first field in the HL7 MSH segment (MSH-01). The value is a decimal ASCII number, and the allowed range is 1 to 127. The default setting is 124, which is the pipe character (|).
Encoding Characters	Encoding characters in the following order: ■ Component separator ■ Repetition separator ■ Escape character ■ Subcomponent separator This is the second field in the HL7 MSH segment (MSH-02). The default is **`^~\&`** (ASCII 94, 126, 92, and 38) respectively.
Sending Application	A user-defined value for the sending application among other applications within the network enterprise. The network enterprise consists of the applications that participate in the exchange of HL7 messages within the enterprise. The default is **`Sun HL7 adapter`**. This is the third field in the HL7 MSH segment (MSH-03).
Sending Facility	A user-defined value that further identifies the sending application among multiple identical instances of the application running on behalf of different organizations. The default is **`Sun HL7 adapter`**. This is the fourth field in the HL7 MSH segment (MSH-04).
Receiving Application	A user-defined value for the receiving application among other applications within the network enterprise. The default value is **`Sun HL7 adapter`**. This is the fifth field in the HL7 MSH segment (MSH-05).

TABLE 23 Connectivity Map - HL7 MSH Segment *(Continued)*

Name	Description
Receiving Facility	A user-defined value that further identifies the receiving application among multiple identical instances of the application running on behalf of different organizations. The default value is **`Sun HL7 adapter`**. This is the sixth field in the HL7 MSH segment (MSH-06).
Security	The implemented application level security features. This is the eighth field in the HL7 MSH segment (MSH-08).
Processing ID	The sub-component processing ID of the MSH-11 field. MSH-11 is used to indicate whether a message is processed as defined in the HL7 Application processing rules. Specify one of the following options: ■ **D** - The message is part of a debugging system. ■ **P** - The message is part of a production system. ■ **T** - The message is part of a training system. In some cases there may be an additional value, the *processing mode*, following the initial value. This value can be A (archive), R (restore from archive), or I (initial load).
Version ID	The HL7 version as displayed in HL7 Table 0104 - Version ID. This value is matched by the receiving system to its own version to ensure that messages are interpreted correctly. The default value is **`2.5`**. This is the 12th field in the HL7 MSH segment (MSH-12).
Country Code	A code that indicates the country of origin for the message (see HL7 Table 0399). Use the 3-character (alphabetic) form of ISO 3166. This value is used to specify default elements in a message, such as currency. The default value is **`USA`**. This is the 17th field in the HL7 MSH segment (MSH-17).
Character Set	The character set(s) used by the messages (see HL7 Table 0211). If the field is left blank, the character set is assumed to be the 7-bit ASCII set. The default value is **`8859/1`** (printable 7-bit ASCII character set). This is the 18th field in the HL7 MSH segment (MSH-18).
Principal Language of Message	The 2-character ISO 639 alphabetic code that specifies the principal language of the message. This is the 19th field in the HL7 MSH segment (MSH-19).
Alternate Character Set Handling Scheme	The value for the alternate character set handling scheme to be used when any alternative character sets are used and a special handling scheme is necessary (see HL7 Table 0356). Possible values are **`ISO 2022-1994`**, **`2.3`**, or **`<null>`** (blank). Leaving the field blank indicates that no character set switching will occur. This is the 20th field in the HL7 MSH segment (MSH-20).

TABLE 23 Connectivity Map - HL7 MSH Segment *(Continued)*

Name	Description
Conformance Statement ID	A unique identifier that applies to a query's conformance statement. It can also be used as a Message Profile Identifier to assert constancy with a message profile (grammar, syntax, usage, and so on). This is the 21st field in the HL7 MSH segment (MSH-21).
Validate MSH	An indicator of whether to validate the MSH segment of the data message (for inbound) and the MSH segment of the ACK (for outbound). Select **`true`** if you want the Collaboration to validate the MSH segment; otherwise select **`false`**. This parameter is used in Collaboration code. **Note –** This property does not affect structural validation of the entire HL7 message itself. Structural validation is always performed.

HL7 SFT Segment — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the properties that appear on the HL7 SFT Segment page of the Properties Editor accessed from the Connectivity Map. These properties define the configuration of the SFT segment of the HL7 message, which provides additional information about one or more software products used as sending applications. The primary purpose of this segment is for diagnostic use. There may be additional uses per site-specific agreements. For more information about this segment, refer to the HL7 specification (`http://www.hl7.org`).

Note – The SFT segment is available starting with HL7 version 2.5.

TABLE 24 Connectivity Map - HL7 SFT Segment

Name	Description
Enable	An indicator of whether the SFT optional segment is enabled in the ACK. Select **`true`** to enable the segment; otherwise select **`false`**. **Note –** If `Enable` is set to true, and the HL7 version is not configured as 2.5, the adapter will error upon startup.
Software Vendor Organization	The name of the company that publishes or distributes the sending software that created the transaction. This field identifies the vendor responsible for maintaining the application. The purpose of this field, along with the remaining fields in this segment, is to provide a more complete profile of the sending applications. This is the first field in the HL7 SFT segment (SFT-01).

TABLE 24 Connectivity Map - HL7 SFT Segment *(Continued)*

Name	Description
Software Certified Version or Release Number	The latest software version number or release number for the sending system, which helps provide a more complete profile of the application that is sending or receiving HL7 messages. Version numbers are important in identifying the specific release of an application. In some situations, the receiving application validates the software certified version or release number against a list of certified versions or releases of the particular software. This helps determine whether the sending application adheres to specific business rules required by the receiving application. Alternatively, the software may perform different processing, depending on the version of the sending software. This is the second field in the HL7 SFT segment (SFT-02).
Software Product Name	The name of the software product that submitted the transaction. The default value is **`Sun TCP/IP HL7 adapter Intelligent Adapter`**. This is the third field in the HL7 SFT segment (SFT-03).
Software Binary ID	The unique software binary ID. Software binary IDs are issued by a vendor for each unique software version instance. These IDs are used to differentiate between multiple versions of the same software. Identical primary IDs indicate that the software is identical at the binary level, but configuration settings may differ. This is the fourth field in the HL7 SFT segment (SFT-04).
Software Product Information	Any additional information about the sending application for more complete identification. This could include a description of the software application, configuration settings, modifications made to the software, and so on. This information is used for diagnostic purposes and provides greater flexibility in identifying the application software. This is the fifth field in the HL7 SFT segment (SFT-05).
Software Install Date	The date on which the submitting software was installed at the sending site. The software installation date on its own can often provide key information about the behavior of the application. This is the sixth field in the HL7 SFT segment (SFT-06).

Communication Control — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the properties that appear on the Communication Control page of the Properties Editor accessed from the Connectivity Map. These properties define how data is transferred (that is, sent and received) over the TCP/IP connection.

TABLE 25 Connectivity Map - Communication Control

Name	Description
Time To Wait For A Response	The amount of time (in milliseconds) that the adapter waits for a response from the external system before taking recourse action (see `Action on No Response` in HL7 Recourse Action). Any data from the external system is considered a response. This property corresponds to the initial read/receive operation timeout. Once a response is received, the subsequent read/receive operation uses the value specified for `SoTimeout` (see Table 17). A value of **0** (zero) indicates an infinite timeout.
Max Empty Read Retry	The maximum number of times the adapter attempts to read data from the external system after the read/receive operation returns nothing. This applies to the read or receive operation after a response starts to arrive. `Empty Read` means that a timeout occurs on the read/receive operation, which uses the `SoTimeout` parameter in the TCPIP Outbound Settings section as the timeout setting (see TCPIP Outbound Settings). The corresponding recourse action is specified by the `Action on Max Failed Read Retry` (see HL7 Recourse Action).
Max No Response	The maximum number of response timeouts the adapter allows while waiting for data from the external system before taking recourse action (see `Action on Max No Response` in HL7 Recourse Action). This property is used in the Collaboration code. It is only used by outbound adapters and works in conjunction with the Resend option of the `Action on No Response` property.
Max NAK Receive Retry	The maximum number of negative acknowledgments (NAKs) the adapter receives before taking recourse action (see `Action on Max Nak Received` in HL7 Recourse Action). This property is used for the outbound Collaboration code.
Max NAK Send Retry	The maximum number of negative acknowledgments (NAKs) the adapter sends before taking recourse action (see `Action on Max Nak Sent` in HL7 Recourse Action).
Max Canned NAK Send Retry	The maximum number of canned negative acknowledgments that the adapter sends before taking recourse action (see `Action on Max Nak Sent` in HL7 Recourse Action). A value of **0** (zero) indicates that the adapter will not attempt to create or send a canned NAK.
Enable Journaling	An indicator of whether message journaling is enabled. To enable message journaling, select **`true`**; otherwise select **`false`**. This property is used for outbound Collaboration code.

HL7 Recourse Action — TCP/IP HL7 V2 Outbound Adapter

The following table lists and describes the properties that appear on the HL7 Recourse Action page of the Properties Editor accessed from the Connectivity Map. These properties define the actions the adapter takes when operations occur outside the configured constraints.

TABLE 26 Connectivity Map - HL7 Recourse Action

Name	Description
Action on No Response	The action the adapter takes when no ACK is received from the external system in the allotted time. The amount of time is determined by the `Time To Wait For A Response` property (see Communication Control). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Resend** – The adapter attempts to resend the message to the external system. The Resend option is only allowed when sequence numbering is in effect. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for outbound Collaboration code.
Action on Max No Response	The action the adapter takes when it attempts to send a message to the external system the maximum allowed number of times and does not receive any response (HL7 Application Acknowledgement) from the external system. The maximum number times the adapter sends a message without receiving a response is determined by the `Max No Response` property (see Communication Control). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for outbound Collaboration code.
Action on Max Failed Read Retry	The action the adapter takes after it has reached the empty read limit set by the `Max Empty Read Retry` property. Select one of the following recourse options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for outbound Collaboration code.

TABLE 26 Connectivity Map - HL7 Recourse Action *(Continued)*

Name	Description
Action on Nak Received	The action the adapter takes when it receives an HL7 Application NAK from the external system. Select one of the following options: - **Resend** – The adapter attempts to resend the message to the external system. - **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. - **Skip Message** – The adapter remains connected, but writes the message to an error queue. **Note** – Do not set both the `Action On NAK Received` and `Action On Max NAK Received` properties to Skip Message. This property is used for outbound Collaboration code.
Action on Max Nak Received	The action the adapter takes when the maximum number of HL7 Application NAKs have been received from the external system, as set by the `Max NAK Receive Retry` property (see Communication Control). Select one of the following options: - **Exit** – The adapter terminates its connection with the external system and shuts down. - **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. - **Skip Message** – The adapter remains connected, but writes the message to an error queue. This property is used for outbound Collaboration code. **Note** – Do not set both the `Action On NAK Received` and `Action On Max NAK Received` properties to Skip Message.
Action on Max Nak Sent	The action the adapter takes when it has sent the maximum allowed number of NAKs to the external system, as set by the `Max NAK Send Retry` parameter (see Communication Control). Select one of the following options: - **Exit** – The adapter terminates its connection with the external system and shuts down. - **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for inbound Collaboration code.

TCP/IP HL7 V3 Adapter Inbound Connectivity Map Properties

The TCP/IP HL7 V3 Server inbound adapter configuration properties are organized into the following sections on the Properties Editor accessed from the Connectivity Map:

General Inbound Settings — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the TCP/IP HL7 V3 inbound adapter properties that appear on the General Inbound Settings page of the Properties Editor accessed from the Connectivity Map.

TABLE 27 Connectivity Map - General Inbound Settings (V3)

Name	Description
Max Data Size	A number that indicates the maximum amount of data that the programs can hold internally. The valid range is a numeric value from **1** to **2147483647** bytes (2GB), which is the maximum value of a Java integer.

TABLE 27 Connectivity Map - General Inbound Settings (V3) *(Continued)*

Name	Description
Scope Of State	The scope of the state object, which is a Message Library node. Select one of the following options for this property: - **Resource Adapter Level** – The state has the same life cycle as the resource adapter. - **Connection Level** – The state has the same life cycle as the connection. - **OTD Level** – The state has the same life cycle as the Message Library object. This scope represents the life cycle of the state.
Dedicated Session Mode	An indicator of whether the server Dedicated Session Mode is enabled. When the server Dedicated Session Mode is enabled, the current client's request exclusively holds the server port to which it connects. The next client's request to the same port is blocked or rejected until the previous request concludes and releases the connection. Select `true` to enable the Dedicated Session Mode, or select `false` to disable the Dedicated Session Mode.

TCPIP Inbound Settings — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the properties on the TCPIP Inbound Settings page of the Properties Editor accessed from the Connectivity Map. These properties configure the Java socket and server socket options.

TABLE 28 Connectivity Map - TCPIP Inbound Settings (V3)

Name	Description
Connection Type	The way the adapter establishes the TCP/IP connection. Select one of the following options: - **Client** – The adapter connects to an external server (host/port) to establish the connection. The adapter is in active mode. - **Server** – The adapter waits and listens on a certain port for an incoming connection request from an external client. Once the request is received, the adapter accepts the request and establishes the connection. The adapter is in passive mode. Server is the default setting. Unless you specifically require Client mode, leave the default value.

TABLE 28 Connectivity Map - TCPIP Inbound Settings (V3) *(Continued)*

Name	Description
ServerSO Timeout	The value (in milliseconds) of the `SO_TIMEOUT` parameter for `ServerSocket`. The timeout must be greater than zero (**0**). A timeout of zero is interpreted as an infinite timeout. This value is used for the `ServerSocket.accept()` method. When this option is set to a non-zero timeout, calling `accept()` for this `ServerSocket` will block for the configured length of time. If the timeout expires, a `java.net.SocketTimeoutException` (or `java.net.InterruptedIOException`) is thrown, but the `ServerSocket` remains valid. Enable this option prior to entering the blocking operation. This property is only used when the `Connection Type` property is set to Server.
Server Socket Factory Implementation Class Name	The name of the Java class that implements the server socket factory. This class is used to create the server socket. If you have provided your own server socket implementation, enter the name of the Java class that contains this implementation here. The factory implementation class must implement the `com.stc.connector.tcpip.model.factory.TCPIPSocketFactory` interface. A default interface, `com.stc.connector.tcpip.model.factory.TCPIPSocketFactoryImpl`, is provided.
Keep Alive	An indicator of whether the client's SO_KEEPALIVE option is enabled or disabled. Select **`true`** to enable SO_KEEPALIVE; otherwise, select **`false`**. When the option is enabled for a TCP socket and no data has been exchanged across the socket in either direction for two hours, TCP automatically sends a KEEPALIVE probe to the peer (the actual value is implementation dependent). This probe is a TCP segment to which the peer must respond. One of three responses is expected: 1. The peer responds with the expected ACK. The application is not notified (since everything is OK). TCP will send another probe following another two hours of inactivity. 2. The peer responds with an RST, which tells the local TCP that the peer host has gone down and rebooted. The socket is closed. 3. There is no response from the peer. The socket is closed. The purpose of this option is to detect if the peer host has crashed. This is used for the accepted client Socket. **Note –** For some properties, the server socket itself does not have direct property settings associated with it. Instead, the properties map to the accepted client socket.

TABLE 28 Connectivity Map - TCPIP Inbound Settings (V3) *(Continued)*

Name	Description
Receive Buffer Size	A number indicating the receive buffer size. This is the value of the SO_RCVBUF option for the current socket, which is the buffer size used by the operating system for input on this socket. It provides an estimate of the size of the underlying buffers used by the platform for incoming network I/O. When used in set mode, this is a suggestion for the kernel from the application regarding the size of buffers to use for the data to be received over the socket. When used in get mode, this must return the actual size of the buffer used by the platform when receiving data on this socket.
Send Buffer Size	A number indicating the send buffer size. This is the value of the SO_SNDBUF option for the current socket, which is the buffer size used by the operating system for output on this socket. It provides an estimate of the size of the underlying buffers used by the platform for outgoing network I/O. When used in set mode, this is a suggestion for the kernel from the application regarding the size of buffers to use for the data to be sent over the socket. When used in get mode, this must return the actual size of the buffer used by the platform when sending out data on this socket.
SoLinger	An indicator of whether the adapter performs a "linger-on-close" timeout. This option disables or enables an immediate return from a call to the `close()` method for a TCP Socket. To enable the linger-on-close timeout, select **`true`**; otherwise, select **`false`**. If you enable this property, specify the maximum length of the timeout in the `SoLinger Timeout` property.
SoLinger Timeout	The server's linger–on–close timeout in seconds. Use `SoLinger Timeout` when `SoLinger` is set to true (see the description for `SoLinger` above). You can specify an integer between -1 and 65535. The default is -1 seconds, which indicates that the `SoLinger` option is disabled. When `SoLinger` is set to **`true`**, the `SoLinger Timeout` value indicates the following: ■ **A non-zero integer** means that calling `close()` will block pending the transmission and acknowledgement of all data written to the peer. When all data is written, the socket is closed gracefully. Upon reaching the linger timeout value specified here, the socket is closed forcefully with a TCP RST. If the specified timeout value exceeds 65,535 it will be reduced to 65,535. A value of –1 indicates the `SoLinger` property is disabled. ■ **A zero integer** means that a forceful close is performed immediately.

TABLE 28 Connectivity Map - TCPIP Inbound Settings (V3) *(Continued)*

Name	Description
SoTimeout	The value of the SoTimeout in milliseconds. This is used for the accepted client socket. You can enter a value greater than or equal to zero (**0**). When set to zero (**0**), the timeout is infinite. With this option set to a non-zero value, calling the `read()` method on the input stream associated with this socket will block for only the configured length of time. If the timeout expires, a `java.io.InterruptedIOException` or `java.net.SocketTimeoutException` is thrown, but the socket remains valid. Enable this option prior to entering the blocking operation.
TcpNoDelay	An indicator of whether data packets that are smaller than the maximum transfer unit (MTU) size are sent out immediately over the network (this refers to Nagle's algorithm). Select one of the following options: ■ **True** – Indicates that the server allows data packets that are smaller than the MTU size to be sent out immediately over the network. This can improve performance for higher-speed networks. ■ **False**– Indicates that the server does not allow data packets that are less than the MTU size be sent out immediately over the network. This is used for the accepted client socket.

TCPIP Inbound Settings - Server Port Binding — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the properties that appear on the Server Port Binding page of the Properties Editor accessed from the Connectivity Map. The properties define the server port binding retry options. This section is only used when the `Connection Type` under TCPIP Inbound Settings is set to Server.

TABLE 29 Connectivity Map - TCPIP Inbound Settings (V3) - Server Port Binding

Name	Description
Max Binding Retry	The maximum number of times the adapter attempts to bind to the specified TCP/IP port on the localhost. This value must be an integer.
Retry Binding Interval	The length of time (in milliseconds) the adapter waits between attempts to bind to the specified TCP/IP port on the localhost.

TCPIP Inbound Settings - Client Connection Establishment — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the property that appears on the Client Connection Establishment page of the Properties Editor accessed from the Connectivity Map. This property defines a wait time before connecting to the external system. This section is only used when the `Connection Type` under TCPIP Inbound Settings is set to Client.

TABLE 30 Connectivity Map - TCPIP Inbound Settings (V3) - Client Connection Establishment

Name	Description
Time to Wait Before Attempting Connection	The length of time (in milliseconds) that the adapter waits before attempting to connect to the external system.

TCPIP Inbound Settings - Inbound Connection Management — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the properties that appear on the Inbound Connection Management page of the Properties Editor accessed from the Connectivity Map. These properties manage the connection to inbound systems. For example, these properties include the connection pool and the life cycle of the accepted connection.

TABLE 31 Connectivity Map - TCPIP Inbound Settings (V3) - Inbound Connection Management

Name	Description
Max Connection Pool Size	The maximum number of concurrent connections allowed for the specific listener or monitor that is listening or monitoring the specified TCP/IP port. **0** (zero) indicates that there is no limit. This value indicates the capability or availability of this server's services. Each connection request from a client gains one concurrent connection. This value also indicates the maximum number of clients that can concurrently connect to this server's services and can be served by the specific listener or monitor at the same time.

TABLE 31 Connectivity Map - TCPIP Inbound Settings (V3) - Inbound Connection Management *(Continued)*

Name	Description
Scope Of Connection	The scope of the accepted connection that is used by the adapter. Select one of the following options: ■ **Resource Adapter Level** – The resource adapter closes the connection upon request (by way of `ClosureCommandMessage`) so the connection may "keep alive" during multiple executions of the Collaboration. ■ **Collaboration Level** – The resource adapter closes the connection once the Collaboration has been executed so the connection has the same life cycle as the Collaboration.
Close Notification	A String indicating the trigger value that notifies the server to close the connection. When the server receives a notification with content that matches this parameter's value, the server safely closes the connection and cancels any corresponding schedules. The default value is **QUIT**.
Idle Timeout	The length of time (in milliseconds) for inactivity of the requestor (client). The adapter attempts to detect in/out activity from the client. If there is no client activity for a specified time period, then the connection is closed from the server side to release the resource. To disable idle timeout checking, specify **0** (zero) for this parameter.

TCPIP Inbound Schedules - Listener Schedule — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the properties that appear on the Listener Schedule page of the Properties Editor accessed from the Connectivity Map. These properties configure the scheduler used by the inbound TCP/IP server. The server waits for a new client connection establishment request. These parameters are used to configure the listener.

Two Java EE schedulers are available, both of which provide the functionality required by the inbound TCP/IP Server.

- **Timer Service** – Available for Java EE, this scheduler is configured using the `At Fixed Rate`, `Delay`, and `Period` properties.
- **Work Manager** – Available for Java EE (JCA 1.5 and above), this scheduler is configured using the `Delay` and `Period` properties.

TABLE 32 Connectivity Map - TCPIP Inbound Schedules (V3) - Listener Schedule

Name	Description
Scheduler	The scheduler type for this inbound communication. Select one of the following options: ■ **Timer Service** – The task is scheduled through the Java EE Timer Service. Timer Service is supported by Java EE. ■ **Work Manager** – The task is scheduled through the Java EE Work Manager. Work Manager is supported by Java EE (JCA 1.5 and above). If your container doesn't support JCA Work Manager, select **`Timer Service`**.
Schedule Type	This property, though visible from the Properties Editor, is disabled. The only available schedule type is Repeated, indicating that the task is scheduled for repeated execution at regular intervals defined by the `Period` property.
Delay	An integer indicating the length of time (in milliseconds) before the task is executed. This property applies to both the Timer Service and the Work Manager.
Period	An integer indicating the length of time (in milliseconds) between successive task executions. This property applies to both the Timer Service and the Work Manager.
At Fixed Rate	An indicator of whether a fixed-rate execution or fixed-delay execution is used. This property applies to the Timer Service configuration only. Select **`true`** to indicate fixed-rate; select **`false`** to indicate fixed-delay. ■ **Fixed-Rate** – Each execution is scheduled relative to the scheduled time of the initial execution. If an execution is delayed for any reason (such as garbage collection or other background activity), two or more executions occur in rapid succession to "catch up." In the long run, the frequency of execution is exactly the reciprocal of the specified period, assuming the system clock underlying `Object.wait(long)` is accurate. ■ **Fixed-Delay** – Each execution is scheduled relative to the actual time of the previous execution. If an execution is delayed for any reason (such as garbage collection or other background activity), subsequent executions are delayed as well. As a result, the frequency of execution is generally slightly lower than the reciprocal of the specified period, assuming the system clock underlying `Object.wait(long)` is accurate.

TCPIP Inbound Schedules - Service Schedule — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the properties that appear on the Service Schedule page of the Properties Editor accessed from the Connectivity Map. These properties configure the scheduler used by the TCP/IP server that executes the business tasks (Collaboration rules) over the existing connection. This scheduler affects the actual business rules you define.

You can use either of the following two Java EE schedulers, both of which provide the functionality required by the inbound TCP/IP server.

- **Timer Service** – Available for Java EE, this scheduler is configured using the `At Fixed Rate`, `Delay`, `Period`, and `Schedule Type`properties.
- **Work Manager** – Available for Java EE (JCA 1.5 and above), this scheduler is configured using the `Delay`, `Period`, and `Schedule Type`properties.

TABLE 33 Connectivity Map - TCPIP Inbound Schedules (V3) - Service Schedule

Name	Description
Scheduler	The scheduler type for this inbound communication. Select one of the following options: ■ **Timer Service** – The task is scheduled through the Java EE Timer Service. Timer Service is supported by Java EE. ■ **Work Manager** – The task is scheduled through the Java EE Work Manager. Work Manager is supported by Java EE (JCA 1.5 and above). If your container doesn't support JCA Work Manager, select Timer Service.
Schedule Type	An indicator of whether the task is scheduled to occur once or be repeated. This property applies to both the Timer Service and the Work Manager. Select one of the following options: ■ **OneTime** – The task is scheduled for one-time execution. ■ **Repeated** – The task is scheduled for repeated execution at regular intervals defined by `Period` property, described below.
Delay	An integer indicating the length of time (in milliseconds) before the task is executed. This property applies to both the Timer Service and the Work Manager.
Period	An integer indicating the length of time (in milliseconds) between successive task executions. This property applies to both the Timer Service and the Work Manager. This is used when the `Schedule Type` property is set to **`Repeated`**.

TABLE 33 Connectivity Map - TCPIP Inbound Schedules (V3) - Service Schedule *(Continued)*

Name	Description
At Fixed Rate	An indicator of whether a fixed-rate execution or fixed-delay execution is used. This property applies to the Timer Service configuration only, and is used when the `Schedule Type` property is set to **`Repeated`**. Select **`true`** to indicate fixed-rate; select **`false`** to indicate fixed-delay. ■ **Fixed-Rate** – Each execution is scheduled relative to the scheduled time of the initial execution. If an execution is delayed for any reason (such as garbage collection or other background activity), two or more executions occur in rapid succession to "catch up." In the long run, the frequency of execution is exactly the reciprocal of the specified period, assuming the system clock underlying `Object.wait(long)` is accurate. ■ **Fixed-Delay** – Each execution is scheduled relative to the actual time of the previous execution. If an execution is delayed for any reason (such as garbage collection or other background activity), subsequent executions are delayed as well. As a result, the frequency of execution is generally slightly lower than the reciprocal of the specified period, assuming the system clock underlying `Object.wait(long)` is accurate.

HL7 Acknowledgment — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the property that appears on the HL7 Acknowledgment page of the Properties Editor accessed from the Connectivity Map. This property defines how the application acknowledgment events are handled.

TABLE 34 Connectivity Map - HL7 Acknowledgment (V3)

Name	Description	Required Value
Acknowledgment Type	The acknowledgment type provided by the Java Collaboration. Select one of the following types: ■ **Immediate** ■ **Deferred** ■ **Queued**	

Lower Layer Protocol — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the properties that appear on the Lower Layer Protocol page of the Properties Editor accessed from the Connectivity Map. These properties define the Lower Layer Protocol (LLP) configuration.

TABLE 35 Connectivity Map - Lower Layer Protocol (V3)

Name	Description
LLP Type	The lower layer protocol (LLP) type. The supported option is MLLP v2.0 (Minimal Lower Layer Protocol v2.0). For more information on MLLP v2.0, see "Lower Layer Protocol" on page 23.
Start Block Character	The first envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. Unless there is a conflict, the value should be ASCII VT (decimal 11).
End Data Character	The second to the last envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. Unless there is a conflict, the value should be ASCII FS (decimal 28).
End Block Character	The last envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. To strictly comply with the HL7 Standard, this property must be set to a carriage return (decimal 13).
Max Number of Retries	The maximum number of times the adapter tries to send a message upon receiving the MLLP v2.0 Negative Commit Acknowledgement from the peer before giving up. This property is used by the adapter in outbound mode. Enter any integer.

Sequence Number Protocol — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the property that appears on the Sequence Number Protocol page of the Properties Editor accessed from the Connectivity Map. This property enables or disables HL7 sequence numbering, which is used to help prevent duplication of data.

TABLE 36 Connectivity Map - Sequence Number Protocol (V3)

Name	Description
Sequence Number Enabled	An indicator of whether sequence numbering is enabled or disabled. Enabling sequence numbering helps prevent duplication of data. Select `true` to enable sequence numbering; otherwise select `false`.

HL7v3 Transmission Wrapper — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the properties that appear on the HL7 Transmission Wrapper page of the Properties Editor accessed from the Connectivity Map. The HL7 transmission wrapper includes information a sending application or message handling service needs to package and route the message to the specified receiving applications or message handling services. The transmission wrapper is a cluster of classes and identifies the sender and receiver of the message and the particular kind of message being communicated.

TABLE 37 Connectivity Map - HL7v3 Transmission Wrapper

Name	Description
Interaction ID	The identification of the unique information interchange. The attribute values are derived from the HL7 MDF interaction names; for example, **POLB_INI00100** and **COMT_IN300652**.
Processing Code	An indicator of the type of system the message is part of. Specify one of the following options: ■ **D** - The message is part of a debugging system. ■ **P** - The message is part of a production system. ■ **T** - The message is part of a training system.
Processing Mode Code	An indicator of the mode in which the message is processed. Specify one of the following options: ■ **T** – Current processing (online mode) ■ **A** – Archive mode ■ **I** – Initial mode ■ **R** – Restore from archive mode
Version Code	The HL7 version. This value is matched by the receiving system to its own version to ensure that messages are interpreted correctly. The default value is **v3.0**.
Validate Transmission Wrapper	An indicator of whether to validate the transmission wrapper of the data message (for inbound case) and transmission wrapper of the ACK response (for outbound case). Select **true** to validate messages; otherwise select **false**. This property is used in the Collaboration code.

Communication Control — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the properties that appears on the Communication Control page of the Properties Editor accessed from the Connectivity Map. These properties define how data is transferred (that is, sent and received) over the TCP/IP connection.

TABLE 38 Connectivity Map - Communication Control (V3)

Name	Description
Time To Wait For A Response	The amount of time (in milliseconds) that the adapter waits for a response from the external system before taking recourse action (see Action on No Response in HL7 Recourse Action). Any data from the external system is considered a response. This property corresponds to the initial read/receive operation timeout. Once a response is received, the subsequent read/receive operation uses the value specified for SoTimeout (see TCPIP Inbound Settings). A value of **0** (zero) indicates an infinite timeout.
Max Empty Read Retry	The maximum number of times the adapter attempts to read data from the external system after the read/receive operation returns nothing. This applies to the read or receive operation after a response starts to arrive. Empty Read means that a timeout occurs on the read/receive operation, which uses the SoTimeout parameter in the TCPIP Server Base Settings section as the timeout setting (see TCPIP Inbound Settings). The corresponding recourse action is specified by the Action on Max Failed Read Retry (see HL7 Recourse Action).
Max No Response	The maximum number of response timeouts the adapter allows while waiting for data from the external system before taking recourse action (see Action on Max No Response in HL7 Recourse Action). This property is used in the inbound Collaboration code.
Max NAK Receive Retry	The maximum number of negative acknowledgments (NAKs) the adapter receives before taking recourse action (see Action on Max Nak Received in HL7 Recourse Action). This property is used for the inbound Collaboration code.
Max NAK Send Retry	The maximum number of negative acknowledgments (NAKs) the adapter sends before taking recourse action (see Action on Max Nak Sent in HL7 Recourse Action). This property is used in the inbound Collaboration code.
Max Canned NAK Send Retry	The maximum number of canned negative acknowledgments that the adapter sends before taking recourse action (see Action on Max Nak Sent in HL7 Recourse Action). A value of **0** (zero) indicates that the adapter will not attempt to create or send a canned NAK.
Enable Journaling	An indicator of whether message journaling is enabled. To enable message journaling, select **true**; otherwise select **false**. This property is used for inbound Collaboration code.

HL7 Recourse Action — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the properties that appear on the HL7 Recourse Action page of the Properties Editor accessed from the Connectivity Map. These properties define the actions the adapter takes when operations occur outside the configured constraints.

TABLE 39 Connectivity Map - HL7 Recourse Action (V3)

Name	Description
Action on No Response	The action the adapter takes when no ACK is received from the external system in the allotted time. The amount of time is determined by the `Time To Wait For A Response` property (see Communication Control). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Resend** – The adapter attempts to resend the message to the external system. The Resend option is only allowed when sequence numbering is in effect. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for inbound Collaboration code.
Action on Max No Response	The action the adapter takes when it attempts to send a message to the external system the maximum allowed number of times and does not receive any response (HL7 Application Acknowledgement) from the external system. The maximum number times the adapter sends a message without receiving a response is determined by the `Max No Response` property (see Communication Control). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for inbound Collaboration code.
Action on Max Failed Read Retry	The action the adapter takes after it has reached the empty read limit set by the `Max Empty Read Retry` property. This property is used by inbound adapters only. Select one of the following recourse options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for inbound Collaboration code.

TABLE 39 Connectivity Map - HL7 Recourse Action (V3) *(Continued)*

Name	Description
Action on Nak Received	The action the adapter takes when it receives an HL7 Application NAK from the external system. Select one of the following options: ■ **Resend** – The adapter attempts to resend the message to the external system. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. ■ **Skip Message** – The adapter remains connected, but writes the message to an error queue. **Note –** Do not set both the `Action On NAK Received` and `Action On Max NAK Received` properties to Skip Message. This property is used for inbound Collaboration code.
Action on Max Nak Received	The action the adapter takes when the maximum number of HL7 Application NAKs have been received from the external system, as set by the `Max NAK Receive Retry` property (see Communication Control). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. ■ **Skip Message** – The adapter remains connected, but writes the message to an error queue. This property is used for inbound Collaboration code. **Note –** Do not set both the `Action On NAK Received` and `Action On Max NAK Received` properties to Skip Message.
Action on Max Nak Sent	The action the adapter takes when it has sent the maximum allowed number of NAKs to the external system, as set by the `Max NAK Send Retry` parameter (see Communication Control). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for inbound Collaboration code.

Schematron Validation — TCP/IP HL7 V3 Inbound Adapter

The following table lists and describes the properties that appears on the Schematron Validation page of the Properties Editor accessed from the Connectivity Map. The schematron uses the concept of finding tree patterns in a parsed document rather than grammar patterns.

TABLE 40 Connectivity Map - Communication Control (V3)

Name	Description
Enable Schematron Validation	An indicator of whether schematron validation is enabled. Select `true` to validate a document's tree patterns; otherwise select `false`. Click the ellipsis button next to this property to display a dialog box that allows you to enter an LDAP reference. Prefix the LDAP reference with `ldap://` or `ldaps://`.
Schematron Files	One or more files containing a predefined schema to validate an HL7 V3 document against. These files must have an .xml extension. Separate file names by commas. You can create these files or obtain files from organizations such as HL7.org.

TCP/IP HL7 V3 Adapter Outbound Connectivity Map Properties

The TCP/IP HL7 V3 server outbound adapter configuration properties are organized into the following sections on the Properties Editor accessed from the Connectivity Map:

- "General Outbound Settings — TCP/IP HL7 V3 Outbound Adapter" on page 80
- "TCPIP Outbound Settings — TCP/IP HL7 V3 Outbound Adapter" on page 80
- "TCPIP Outbound Settings - Client Connection Establishment — TCP/IP HL7 V3 Outbound Adapter" on page 83
- "TCPIP Outbound Settings - Server Port Binding — TCP/IP HL7 V3 Outbound Adapter" on page 84
- "HL7 Acknowledgment — TCP/IP HL7 V3 Outbound Adapter" on page 85
- "Lower Layer Protocol — TCP/IP HL7 V3 Outbound Adapter" on page 85
- "Sequence Number Protocol — TCP/IP HL7 V3 Outbound Adapter" on page 86
- "HL7v3 Transmission Wrapper — TCP/IP HL7 V3 Outbound Adapter" on page 86
- "Communication Control — TCP/IP HL7 V3 Outbound Adapter" on page 87
- "HL7 Recourse Action — TCP/IP HL7 V3 Outbound Adapter" on page 89

General Outbound Settings — TCP/IP HL7 V3 Outbound Adapter

The following table lists and describes the TCP/IP HL7 V3 outbound adapter properties that appear on the General Outbound Settings page of the Properties Editor accessed from the Connectivity Map.

TABLE 41 Connectivity Map - General Outbound Settings (V3)

Name	Description
Max Data Size	A number that indicates the maximum amount of data that the programs can hold internally. The valid range is a numeric value from **1** to **2147483647** bytes (2GB), which is the maximum value of a Java integer.
Scope Of State	The scope of the state object, which is a Message Library node. Select one of the following options for this property: ■ **Resource Adapter Level** – The state has the same life cycle as the resource adapter. ■ **Connection Level** – The state has the same life cycle as the connection. ■ **OTD Level** – The state has the same life cycle as the Message Library object. This scope represents the life cycle of the state.

TCPIP Outbound Settings — TCP/IP HL7 V3 Outbound Adapter

The following table lists and describes the properties on the TCPIP Outbound Settings page of the Properties Editor accessed from the Connectivity Map. These properties configure the Java socket and server socket options. For more information, see the Javadocs for Java SDK.

TABLE 42 Connectivity Map - TCPIP Outbound Settings (V3)

Name	Description
Connection Type	The way the adapter establishes the TCP/IP connection. Select one of the following options: - **Client** – The adapter connects to an external server (host/port) to establish the connection. The adapter is in active mode. - **Server** – The adapter waits and listens on a certain port for an incoming connection request from an external client. Once the request is received, the adapter accepts the request and establishes the connection. The adapter is in passive mode. Server is the default setting. Unless you specifically require Client mode, leave the default value.
ServerSoTimeout	The value (in milliseconds) of the `SO_TIMEOUT` parameter for `ServerSocket`. The timeout must be greater than zero (**0**). A timeout of zero is interpreted as an infinite timeout. This value is used for the `ServerSocket.accept()` method. When this option is set to a non-zero timeout, calling `accept()` for this `ServerSocket` will block for the configured length of time. If the timeout expires, a `java.net.SocketTimeoutException` (or `java.net.InterruptedIOException`) is thrown, but the `ServerSocket` remains valid. Enable this option prior to entering the blocking operation. This property is only used when the `Connection Type` property is set to Server.
Keep Alive	An indicator of whether the client's SO_KEEPALIVE option is enabled or disabled. Select **`true`** to enable SO_KEEPALIVE; otherwise, select **`false`**. When the option is enabled for a TCP socket and no data has been exchanged across the socket in either direction for two hours, TCP automatically sends a KEEPALIVE probe to the peer (the actual value is implementation dependent). This probe is a TCP segment to which the peer must respond. One of three responses is expected: 1. The peer responds with the expected ACK. The application is not notified (since everything is OK). TCP will send another probe following another two hours of inactivity. 2. The peer responds with an RST, which tells the local TCP that the peer host has crashed and rebooted. The socket is closed. 3. There is no response from the peer. The socket is closed. The purpose of this option is to detect if the peer host has crashed. This is used for the accepted client Socket. **Note –** For some properties, the server socket itself does not have direct property settings associated with it. Instead, the properties map to the accepted client socket.

TABLE 42 Connectivity Map - TCPIP Outbound Settings (V3) *(Continued)*

Name	Description
Receive Buffer Size	A number indicating the receive buffer size. This is the value of the SO_RCVBUF option for the current socket, which is the buffer size used by the operating system for input on this socket. It provides an estimate of the size of the underlying buffers used by the platform for incoming network I/O. When used in set mode, this is a suggestion for the kernel from the application regarding the size of buffers to use for the data to be received over the socket. When used in get mode, this must return the actual size of the buffer used by the platform when receiving data on this socket.
Send Buffer Size	A number indicating the send buffer size. This is the value of the SO_SNDBUF option for the current socket, which is the buffer size used by the operating system for output on this socket. It provides an estimate of the size of the underlying buffers used by the platform for outgoing network I/O. When used in set mode, this is a suggestion for the kernel from the application regarding the size of buffers to use for the data to be sent over the socket. When used in get mode, this must return the actual size of the buffer used by the platform when sending out data on this socket.
SoLinger	An indicator of whether the adapter performs a “linger-on-close” timeout. This option disables or enables an immediate return from a call to the `close()` method for a TCP Socket. To enable the linger-on-close timeout, select **`true`**; otherwise, select **`false`**. If you enable this property, specify the maximum length of the timeout in the `SoLinger Timeout` property.
SoLinger Timeout	The server’s linger–on–close timeout in seconds. Use `SoLinger Timeout` when `SoLinger` is set to true (see the description for `SoLinger` above). You can specify an integer between -1 and 65535. The default is -1 seconds, which indicates that the `SoLinger` option is disabled. When `SoLinger` is set to **`true`**, the `SoLinger Timeout` value indicates the following: ■ **A non-zero integer** means that calling `close()` will block pending the transmission and acknowledgement of all data written to the peer. When all data is written, the socket is closed gracefully. Upon reaching the linger timeout value specified here, the socket is closed forcefully with a TCP RST. If the specified timeout value exceeds 65,535 it will be reduced to 65,535. ■ **A zero integer** means that a forceful close is performed immediately.

TABLE 42 Connectivity Map - TCPIP Outbound Settings (V3) *(Continued)*

Name	Description
SoTimeout	The value of the SoTimeout in milliseconds. You can enter a value greater than or equal to zero (**0**). When set to zero (**0**), the timeout is infinite. With this option set to a non-zero value, calling the `read()` method on the input stream associated with this socket will block for only the configured length of time. If the timeout expires, a `java.io.InterruptedIOException` or `java.net.SocketTimeoutException` is thrown, but the socket remains valid. Enable this option prior to entering the blocking operation.
TcpNoDelay	An indicator of whether data packets that are smaller than the maximum transfer unit (MTU) size are sent out immediately over the network (this refers to Nagle's algorithm). Select one of the following options: ■ **True** – Indicates that the server allows data packets that are smaller than the MTU size to be sent out immediately over the network. This can improve performance for higher-speed networks. ■ **False**– Indicates that the server does not allow data packets that are less than the MTU size be sent out immediately over the network. This is used for the accepted client socket.
Socket Factory Implementation Class Name	The name of the Java class that implements the socket factory. This class is used to create the socket. If you have provided your own socket implementation, enter the name of the Java class that contains this implementation here. The factory implementation class must implement the `com.stc.connector.tcpip.model.factory.TCPIPSocketFactory` interface. The default value is `com.stc.connector.tcpip.model.factory.TCPIPSocketFactoryImpl`

TCPIP Outbound Settings - Client Connection Establishment — TCP/IP HL7 V3 Outbound Adapter

The following table lists and describes the properties that appear on the Client Connection Establishment page of the Properties Editor accessed from the Connectivity Map. These properties configure how connections are established. This section is only used when the `Connection Type` under TCPIP Outbound Settings is set to Client.

TABLE 43 Connectivity Map - TCPIP Outbound Settings (V3) - Client Connection Establishment

Name	Description
Time To Wait Before Attempting Connection	The length of time (in milliseconds) the adapter waits before attempting to connect to the external system.

TABLE 43 Connectivity Map - TCPIP Outbound Settings (V3) - Client Connection Establishment *(Continued)*

Name	Description
Always Create New Connection	An indicator of whether the adapter always attempts to create a new connection when a connection establishment request is received. Select on of the following options: ■ **true** – The adapter always attempts to create a new connection without attempting to match an existing connection. ■ **false** – The adapter attempts to match an existing connection (managed by the container).
Auto Reconnect Upon Matching Failure	An indicator or whether to attempt to reconnect automatically when the adapter gets a matching connection from a container, even though this connection is not valid; for example, the external side of the connection is closed or reset due to the external application's logic. Select one of the following options: ■ **true** – The adapter discards the invalid matching connection and automatically attempts to reconnect using a new connection. ■ **false** – The adapter does not automatically attempt to reconnect using a new connection. Instead, the adapter defers the reconnect control to the user business rules. It is up to the business rules to detect this type of failure and act appropriately.
Max Connection Retry	The maximum number of times the adapter attempts to connect to a specific external TCP/IP destination (host/port) before giving up.
Retry Connection Interval	The length of time (in milliseconds) the adapter waits between attempts to connect to a specific external TCP/IP destination (host or port).

TCPIP Outbound Settings - Server Port Binding — TCP/IP HL7 V3 Outbound Adapter

The following table lists and describes the properties that appear on the Server Port Binding page of the Properties Editor accessed from the Connectivity Map. The properties define the server port binding retry options. This section is only used when the `Connection Type` under TCPIP Outbound Settings is set to Server.

TABLE 44 Connectivity Map - TCPIP Outbound Settings (V3) - Server Port Binding

Name	Description
Max Binding Retry	The maximum number of times the adapter will attempt to bind to the specified TCP/IP port on the localhost before giving up.

TABLE 44 Connectivity Map - TCPIP Outbound Settings (V3) - Server Port Binding *(Continued)*

Name	Description
Retry Binding Interval	The amount of time (in milliseconds) that the adapter waits between attempts to bind to the specified TCP/IP port on the localhost.

HL7 Acknowledgment — TCP/IP HL7 V3 Outbound Adapter

The following table lists and describes the properties that appear on the HL7 Acknowledgment page of the Properties Editor accessed from the Connectivity Map. These properties define how the application acknowledgment events are handled.

TABLE 45 Connectivity Map - HL7 Acknowledgement (V3)

Name	Description
Acknowledgment Level	The acknowledgment type provided by the Java Collaboration. Select one of the following types: ■ **Immediate** ■ **Deferred** ■ **Queued**

Lower Layer Protocol — TCP/IP HL7 V3 Outbound Adapter

The following table lists and describes the properties that appear on the Lower Layer Protocol page of the Properties Editor accessed from the Connectivity Map. These properties define the Lower Layer Protocol (LLP) configuration.

TABLE 46 Connectivity Map - Lower Layer Protocol (V3)

Name	Description
LLP Type	The lower layer protocol (LLP) type. The supported option is MLLP v2.0 (Minimal Lower Layer Protocol v2.0). For more information on MLLP v2.0, see "Lower Layer Protocol" on page 23.
Start Block Character	The first envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. Unless there is a conflict, the value should be ASCII VT (decimal 11).

TABLE 46 Connectivity Map - Lower Layer Protocol (V3) *(Continued)*

Name	Description
End Data Character	The second to the last envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. Unless there is a conflict, the value should be ASCII FS (decimal 28).
End Block Character	The last envelope marker character in the HL7 envelope, as a decimal ASCII number. Enter a number from 1 to 127. To strictly comply with the HL7 Standard, this property must be set to a carriage return (decimal 13).
Max Number of Retries	The maximum number of times the adapter tries to send a message upon receiving the MLLP v2.0 Negative Commit Acknowledgement from the peer before giving up. This property is used by the adapter in outbound mode. Enter any integer.

Sequence Number Protocol — TCP/IP HL7 V3 Outbound Adapter

The following table lists and describes the property that appears on the Sequence Number Protocol page of the Properties Editor accessed from the Connectivity Map. This property enables or disables HL7 sequence numbering, which is used to help prevent duplication of data.

TABLE 47 Connectivity Map - Sequence Number Protocol (V3)

Name	Description
Sequence Number Enabled	An indicator of whether sequence numbering is enabled or disabled. Enabling sequence numbering helps prevent duplication of data. Select **`true`** to enable sequence numbering; otherwise select **`false`**.

HL7v3 Transmission Wrapper — TCP/IP HL7 V3 Outbound Adapter

The following table lists and describes the properties that appear on the HL7 Transmission Wrapper page of the Properties Editor accessed from the Connectivity Map. The HL7 transmission wrapper includes information a sending application or message handling service needs to package and route the message to the specified receiving applications or message handling services. The transmission wrapper is a cluster of classes and identifies the sender and receiver of the message and the particular kind of message being communicated.

TABLE 48 Connectivity Map - HL7v3 Transmission Wrapper

Name	Description
Interaction ID	The identification of the unique information interchange. The attribute values are derived from the HL7 MDF interaction names; for example, **POLB_INI00100** and **COMT_IN300652**.
Processing Code	An indicator of the type of system the message is part of. Specify one of the following options: ■ **D** - The message is part of a debugging system. ■ **P** - The message is part of a production system. ■ **T** - The message is part of a training system.
Processing Mode Code	An indicator of the mode in which the message is processed. Specify one of the following options: ■ **T** – Current processing (online mode) ■ **A** – Archive mode ■ **I** – Initial mode ■ **R** – Restore from archive mode
Version Code	The HL7 version. This value is matched by the receiving system to its own version to ensure that messages are interpreted correctly. The default value is **v3.0**.
Validate Transmission Wrapper	An indicator of whether to validate the transmission wrapper of the data message (for inbound case) and transmission wrapper of the ACK response (for outbound case). Select **true** to validate messages; otherwise select **false**. This property is used in the Collaboration code.

Communication Control — TCP/IP HL7 V3 Outbound Adapter

The following table lists and describes the properties that appear on the Communication Control page of the Properties Editor accessed from the Connectivity Map. These properties define how data is transferred (that is, sent and received) over the TCP/IP connection.

TABLE 49 Connectivity Map - Communication Control (V3)

Name	Description
Time To Wait For A Response	The amount of time (in milliseconds) that the adapter waits for a response from the external system before taking recourse action (see Action on No Response in HL7 Recourse Action). Any data from the external system is considered a response. This property corresponds to the initial read/receive operation timeout. Once a response is received, the subsequent read/receive operation uses the value specified for SoTimeout (see TCPIP Outbound Settings). A value of **0** (zero) indicates an infinite timeout.
Max Empty Read Retry	The maximum number of times the adapter attempts to read data from the external system after the read/receive operation returns nothing. This applies to the read or receive operation after a response starts to arrive. Empty Read means that a timeout occurs on the read/receive operation, which uses the SoTimeout property in the TCPIP Outbound Settings section as the timeout setting (see TCPIP Outbound Settings). The corresponding recourse action is specified by the Action on Max Failed Read Retry (see HL7 Recourse Action).
Max No Response	The maximum number of response timeouts the adapter allows while waiting for data from the external system before taking recourse action (see Action on Max No Response in HL7 Recourse Action). This property is used in the Collaboration code. It works in conjunction with the Resend option of the Action on No Response property (see HL7 Recourse Action table).
Max NAK Receive Retry	The maximum number of negative acknowledgments (NAKs) the adapter receives before taking recourse action (see Action on Max Nak Received in HL7 Recourse Action). This property is used for the outbound Collaboration code.
Max NAK Send Retry	The maximum number of negative acknowledgments (NAKs) the adapter sends before taking recourse action (see Action on Max Nak Sent in HL7 Recourse Action).
Max Canned NAK Send Retry	The maximum number of canned negative acknowledgments that the adapter sends before taking recourse action (see Action on Max Nak Sent in HL7 Recourse Action). A value of **0** (zero) indicates that the adapter will not attempt to create or send a canned NAK.
Enable Journaling	An indicator of whether message journaling is enabled. To enable message journaling, select **true**; otherwise select **false**. This property is used for outbound Collaboration code.

HL7 Recourse Action — TCP/IP HL7 V3 Outbound Adapter

The following table lists and describes the properties that appear on the HL7 Recourse Action page of the Properties Editor accessed from the Connectivity Map. These properties define the actions the adapter takes when operations occur outside the configured constraints.

TABLE 50 Connectivity Map - HL7 Recourse Action (V3)

Name	Description
Action on No Response	The action the adapter takes when no ACK is received from the external system in the allotted time. The amount of time is determined by the `Time To Wait For A Response` property (see Communication Control - TCP/IP). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Resend** – The adapter attempts to resend the message to the external system. The Resend option is only allowed when sequence numbering is in effect. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for outbound Collaboration code.
Action on Max No Response	The action the adapter takes when it attempts to send a message to the external system the maximum allowed number of times and does not receive any response (HL7 Application Acknowledgement) from the external system. The maximum number times the adapter sends a message without receiving a response is determined by the `Max No Response` property (see Communication Control - TCP/IP). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for outbound Collaboration code.
Action on Max Failed Read Retry	The action the adapter takes after it has reached the empty read limit set by the `Max Empty Read Retry` property. Select one of the following recourse options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for outbound Collaboration code.

TABLE 50 Connectivity Map - HL7 Recourse Action (V3) *(Continued)*

Name	Description
Action on Nak Received	The action the adapter takes when it receives an HL7 Application NAK from the external system. Select one of the following options: ■ **Resend** – The adapter attempts to resend the message to the external system. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. ■ **Skip Message** – The adapter remains connected, but writes the message to an error queue. **Note** – Do not set both the `Action On NAK Received` and `Action On Max NAK Received` properties to Skip Message. This property is used for outbound Collaboration code.
Action on Max Nak Received	The action the adapter takes when the maximum number of HL7 Application NAKs have been received from the external system, as set by the `Max NAK Receive Retry` property (see Communication Control - TCP/IP). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. ■ **Skip Message** – The adapter remains connected, but writes the message to an error queue. This property is used for outbound Collaboration code. **Note** – Do not set both the `Action On NAK Received` and `Action On Max NAK Received` properties to Skip Message.
Action on Max Nak Sent	The action the adapter takes when it has sent the maximum allowed number of NAKs to the external system, as set by the `Max NAK Send Retry` parameter (see Communication Control - TCP/IP). Select one of the following options: ■ **Exit** – The adapter terminates its connection with the external system and shuts down. ■ **Reset** – The adapter closes its connection with the external system and goes through the connection scenario. This property is used for inbound Collaboration code.

Configuring Sun Adapter for TCP/IP HL7 Environment Properties

The adapter environment configuration properties define how the adapter connects to and interacts with other Java CAPS components within the environment. The environment properties are accessed from the NetBeans IDE Services window. The following sections provide instructions on how to configure Java CAPS component environment properties and lists the environment properties for the various communications adapters.

Configuring TCP/IP HL7 Adapter Environment Properties

The adapter environment configuration properties contain parameters that define how the adapter connects to and interacts with other Java CAPS components within the environment. The environment properties are accessed from the NetBeans IDE Services window.

▼ To Configure the Environment Properties

1. **From the NetBeans Services window, expand the CAPS Environments node.**

2. **Expand the environment created for your project and locate the External System for your specific adapter.**

3. **Right-click the External System and select Properties.**

   The Environment Configuration Properties window appears.

4. **On the Properties Editor, click on any folder in the left panel to display the properties for that section.**

   For information about the folders and properties on the Properties Editor, see "TCP/IP HL7 Inbound Adapter Environment Properties" on page 92 and "TCP/IP HL7 Inbound Adapter Environment Properties" on page 94.

5. **Click on any property field to modify it.**

6. **Once you have finished modifying the properties, click OK to save your changes and close the editor.**

TCP/IP HL7 Inbound Adapter Environment Properties

The TCP/IP HL7 Adapter configuration parameters accessed from the Environment tree apply to both the inbound and outbound Adapters, and are the same for HL7 v2 and HL7 v3 Adapters.

The inbound TCP/IP HL7 Adapter's Environment properties are divided into the following sections:

- "HL7 Inbound Adapter - TCPIP Inbound Settings" on page 92
- "HL7 Inbound Adapter - MDB Pool Settings" on page 92
- "HL7 Inbound Adapter - Sequence Number Protocol" on page 93
- "HL7 Inbound Adapter - Database Settings" on page 94

HL7 Inbound Adapter - TCPIP Inbound Settings

The following table lists and describes the TCP/IP HL7 inbound adapter properties that appear on the TCPIP Inbound Settings page of the Properties Editor accessed from the Environment. These properties configure the Java socket and server socket options. For more information, see the Javadocs provided with Java SDK.

TABLE 51 Environment - HL7 Inbound Adapter - TCPIP Inbound Settings

Name	Description
Host	The host name or IP address used to establish a TCP/IP connection. This property is only used when the `Connection Type` is set to Client.
ServerPort	The port number of the TCP/IP destination. This is dependent on the value set for `Connection Type`. If `Connection Type` is set to Server, it indicates the port number on the local host; if `Connection Type` is set to Client, it indicates the port number of the external host. Enter an integer between 0 and 65535.
Backlog	An integer indicating the maximum length of the queue for incoming connection requests when creating the server socket. When a connection indication arrives and the queue is full, the connection is refused. This parameter is only used when `Connection Type` is set to Server.

HL7 Inbound Adapter - MDB Pool Settings

The following table lists and describes the properties that appear on the MDB Pool Setting page of the Properties Editor accessed from the Environment. These properties are specific to the message–driven bean (MDB) pool of the GlassFish Server or Sun Runtime Server. These properties settings are packaged into `sun-ejb-jar.xml`.

TABLE 52 Environment - HL7 Inbound Adapter - TCPIP Inbound Settings

Name	Description
Steady Pool Size	An integer indicating the minimum number of message–driven beans to maintain. When the value is set to a number greater than 0 (zero), the container pre-populates the MDB pool with the specified number and tries to ensure that there are always this number of beans in the free pool. This ensures that there are enough MDB beans in the ready-to-serve state to process user requests. This property does not necessarily guarantee that no more than steady-pool-size instances exist at a given time. It only governs the number of instances that are pooled over a long period of time. For example, if an idle stateless session container has a fully-populated pool with a steady-pool-size of 10 and 20 concurrent requests arrive for the MDB component, the container creates 10 additional instances to satisfy the burst of requests. The advantage of this is that it prevents the container from blocking any of the incoming requests. However, if the activity is reduced to 10 or fewer concurrent requests, the additional 10 instances are discarded.
Max Pool Size	An integer indicating the maximum number of message–driven beans in the pool. A value of **0** (zero) indicates that the pool is unbounded.
Pool Idle Timeout in Seconds	The maximum amount of time (in seconds) that an MDB instance can remain idle in the pool. When an MDB has exceeded the configured timeout, a timer thread removes the unused MDB. This property defines the interval at which this thread runs. A value greater than 0 indicates that the container removes or destroys any MDB instance that is idle at this specified duration. A value of 0 (zero) specifies that idle MDB instances can remain in the pool indefinitely.

HL7 Inbound Adapter - Sequence Number Protocol

The following table lists and describes the property that appears on the Sequence Number Protocol page of the Properties Editor accessed from the Environment. Sequence numbering helps prevent duplication of data.

TABLE 53 Environment - HL7 Inbound Adapter - Sequence Number Protocol

Name	Description
Sequence Number File Location	The location of the sequence number file (a local directory). This is required when the Sequence Number Protocol is enabled. The sequence number folder is a nonvolatile directory that stores the sequence number files used to persist the HL7 sequence number. The unique base file name is automatically generated according to project or Collaboration information. For the inbound Adapter the file names are created as follows: *ProjectName* `+` *DeploymentName* `+` *ExternalApplicationName* `+` *CollaborationName* `+ .seqno` For example, `prjHL7Inbound_dpIn_eaHL7Inbound_ jcdHL7inbound1.seqno` The default setting is `/temp/hl7inbound/seq`.

HL7 Inbound Adapter - Database Settings

The following table lists and describes the property that appears on the Database Settings page of the Properties Editor accessed from the Environment. These properties are used for storing HL7 messages as part of MLLP v2 support.

TABLE 54 Environment - HL7 Inbound Adapter - Database Settings

Name	Description
JNDI Name of the JDBC Datasource	The JNDI name of the JDBC data source as defined in the application server. This data source is used by MLLP v2 protocol to persist the HL7 messages before sending a commit acknowledgment. For more information, see "MLLP V2 and the Sample Projects" on page 140.

TCP/IP HL7 Inbound Adapter Environment Properties

The TCP/IP HL7 Adapter configuration parameters accessed from the Environment tree apply to both the inbound and outbound Adapters, and are the same for HL7 v2 and HL7 v3 Adapters.

The onbound TCP/IP HL7 Adapter's Environment properties are divided into the following sections:

- "HL7 Outbound Adapter - TCPIP Outbound Settings" on page 95
- "HL7 Outbound Adapter - Connection Pool Settings" on page 95
- "HL7 Outbound Adapter - Sequence Number Protocol" on page 96

HL7 Outbound Adapter - TCPIP Outbound Settings

The following table lists and describes the properties that appear on the MDB Pool Setting page of the Properties Editor accessed from the Environment. These properties configure the Java socket and server socket options. For more information, see the Javadocs provided with Java SDK.

TABLE 55 Environment - HL7 Outbound Adapter - TCPIP Outbound Settings

Name	Description
Host	The host name or IP address used to establish a TCP/IP connection. This property is only used when the Connection Type is set to Client. The default value is **`localhost`**.
ServerPort	An integer between 0 and 65535, indicating the port number of the TCP/IP destination. This is dependent on the value set for `Connection Type`. If `Connection Type` is set to Server, it indicates the port number on the local host; if `Connection Type` is set to Client, it indicates the port number of the external host.
Backlog	An integer indicating the maximum length of the queue for incoming connection requests when creating the server socket. When a connection indication arrives and the queue is full, the connection is refused. This parameter is only used when `Connection Type` is set to Server.
Connection Mode	An indicator of how the Adapter connects to the external system. Select one of the following options: ■ **Automatic** – The Adapter connects to the external system during initialization based on the connection settings (host and port number) defined in the environment. This enables dynamic connections ■ **Manual** – The connection settings are provided in the Java Collaboration through method calls, so the connection is done manually.

HL7 Outbound Adapter - Connection Pool Settings

The following table lists and describes the properties that appear on the Connection Pool Setting page of the Properties Editor accessed from the Environment. These properties are specific to the resource adapter pool of the GlassFish Server or Sun Runtime Server. These properties settings are packaged into `sun-ra.xml`.

TABLE 56 Environment - HL7 Outbound Adapter - Connection Pool Settings

Name	Description
Steady Pool Size	An integer indicating the minimum number of resource adapter connections to be maintained. When the value is set to a number greater than 0 (zero), the container pre-populates the RA connection pool with the specified number and tries to ensure that there are always this many connections in the free pool. This ensures that there are enough connections in the ready-to-serve state to process user requests. This property does not necessarily guarantee that no more than steady-pool-size instances exist at a given time. It only governs the number of instances that are pooled over a long period of time. For example, if an idle stateless session container has a fully-populated pool with a steady-pool-size of 10 and 20 concurrent requests arrive for the resource adapter connection component, the container creates 10 additional instances to satisfy the burst of requests. The advantage of this is that it prevents the container from blocking any of the incoming requests. However, if the activity is reduced to 10 or fewer concurrent requests, the additional 10 instances are discarded.
Max Pool Size	An integer indicating the maximum number of resource adapter connections in the pool. A value of **0** (zero) indicates that the pool is unbounded.
Pool Idle Timeout in Seconds	The maximum amount of time (in seconds) that a resource adapter connection instance can remain idle in the pool. When a connection has exceeded the configured timeout, a timer thread removes the unused connection. This property defines the interval at which this thread runs. A value greater than 0 (zero) indicates that the container removes or destroys any resource adapter connection instance that is idle at this specified duration. A value of 0 (zero) specifies that idle connections can remain in the pool indefinitely.

HL7 Outbound Adapter - Sequence Number Protocol

The following table lists and describes the property that appears on the Sequence Number Protocol page of the Properties Editor accessed from the Environment. Sequence numbering helps prevent duplication of data.

TABLE 57 Environment - HL7 Outbound Adapter - Sequence Number Protocol

Name	Description
Sequence Number File Location	The location of the sequence number file (a local directory). This is required when the Sequence Number Protocol is enabled. The sequence number folder is a nonvolatile directory that stores the sequence number files used to persist the HL7 sequence number. The unique base file name is automatically generated according to project or Collaboration information. For the outbound Adapter the file names are created as follows: *ProjectName* + *DeploymentName* + *CollaborationName* + *ExternalApplicationName* + .seqno For example, `prjHL7Outbound_dpOut_jcolHL7Outbound_eaHL7Outbound.seqno` The default setting is `/temp/hl7outbound/seq`.

Using the TCP/IP HL7 Predefined Templates

The TCP/IP HL7 Adapter provides several sample Projects to help you customize a solution using either HL7 V2.x or HL7 V3 messaging. In addition to the HL7 Adapter, these Projects include several Collaborations that define processing logic in addition to the rules configured for the Adapter. You can use these Collaborations as a template for the Projects you create for your own HL7 system.

The following topics provide instructions for copying and customizing HL7 Collaborations:

- "Prerequisites for the HL7 V3 Sample Projects" on page 97
- "Creating a Copy of an HL7 Sample Project" on page 98
- "Customizing Predefined Collaborations for HL7" on page 102
- "Creating Copies of HL7 Collaborations" on page 102
- "Adding an HL7 Message Library to an Existing Collaboration" on page 104

The following topics provide information about how the Collaborations are defined and the functional flow of the Collaborations:

- "About TCP/IP HL7 V2 Collaborations" on page 107
- "About TCP/IP HL7 V3 Collaborations" on page 120

Prerequisites for the HL7 V3 Sample Projects

The predefined Collaborations are designed to be extended and modified, however, for HL7 V3 compliant systems this is not necessary. If you need to modify an HL7 V3 Collaboration, it is strongly suggested that you use these template Collaborations as the basis for any new

Collaborations. Therefore, it is important to maintain the original predefined jcdHL7V3Inbound and jcdHL7V3Outbound Collaborations in their initial form for future use.

Before working with the HL7 V3 sample Projects, be sure that the required HL7 V3 SAR files are uploaded to the repository. The required files are listed below:

1. HL7eWay.sar
2. HL7OTDLibrary.sar
3. HL7V32006ACCTBilling.sar
4. HL7V32006ClaimsAndReimb.sar
5. HL7V32006ClinicalGenomics.sar
6. HL7V32006MedicalRecords.sar
7. HL7V32006MsgContActInfra.sar
8. HL7V32006PatientAdmin.sar
9. HL7V32006PersonnelManagement.sar
10. HL7V32006PublicHealthRepot.sar
11. HL7V32006QueryInfra.sar
12. HL7V32006RegulateStudies.sar
13. HL7V32006Scheduling.sar
14. HL7V32006SharedMessages.sar
15. HL7V32006TransInfra.sar

Creating a Copy of an HL7 Sample Project

It is recommended that you retain the sample Projects as they are and create copies of the Projects to use as a basis for your new customized Projects. To create a copy of a Project, you first export the original Project to a file, change the name of the original Project, and import the exported Project back into the Repository.

▼ To Create a Copy of a Project

1. **Start the CAPS Repository.**

2. **Start the NetBeans IDE.**

3. **On the NetBeans toolbar, click Tools, point to CAPS Repository, and then select Connect.**

   The Connect to CAPS Repository dialog box appears.

4. **Enter the connection information and click Connect.**

Note – When connected the icon in the status bar turns from red to green.

---

5 **Export the Project you want to copy:**

a. **On the Projects window, right-click the Project to copy, click Export, and then click Project.**

The Export Manager dialog box appears.

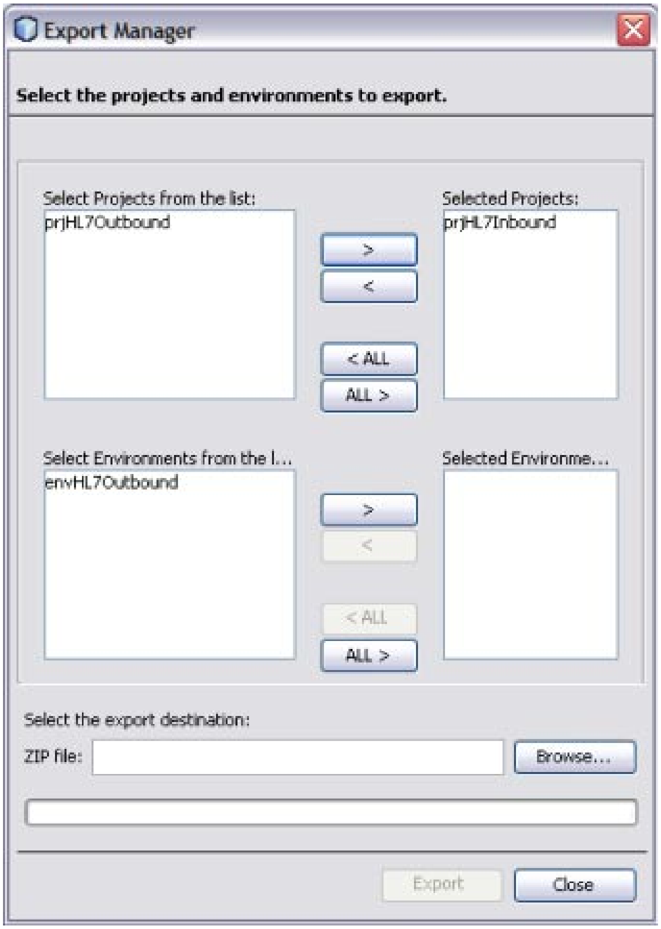


FIGURE 2 Export Manager

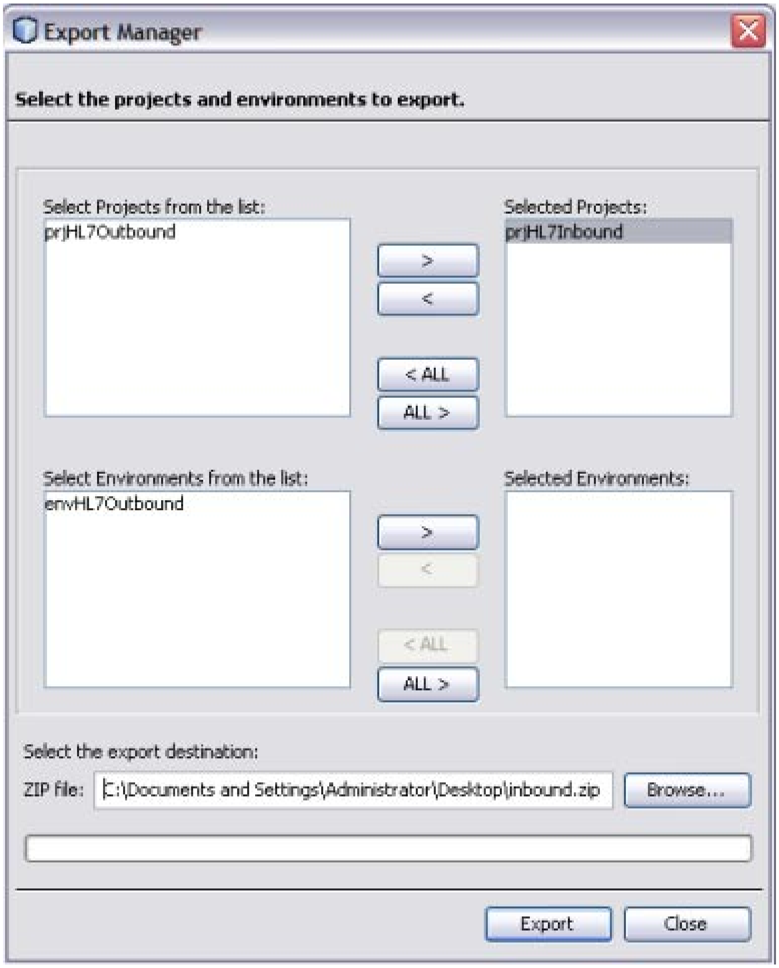
b. **Select the Projects and Environments to export.**

The selected Project appears in the Selected Projects: pane of the Export Manager.

c. **Select the Project or the Environment from the left pane and click the right-arrow button to add the Project to the Selected Projects pane.**

Tip – Click the left-arrow button , if you want to move the selected projects from the right pane to the left pane.

d. **Click Browse and select an appropriate directory to save the exported Project.**



Export Manager

Select the projects and environments to export.

Select Projects from the list: prjHL7Outbound

Selected Projects: prjHL7Inbound

Select Environments from the list: envHL7Outbound

Selected Environments:

Select the export destination:

ZIP file: C:\Documents and Settings\Administrator\Desktop\inbound.zip

FIGURE 3 Export Manager — Browse Option

e. **Click Export.**

The Project is compressed and saved to the specified directory and file name.

6 **Rename the original Project in the NetBeans IDE.**

Note – The NetBeans IDE will not import identically named Projects to the same root, so you must perform this step before importing the Project back in.

---

7 **On the NetBeans IDE, click Save All.**

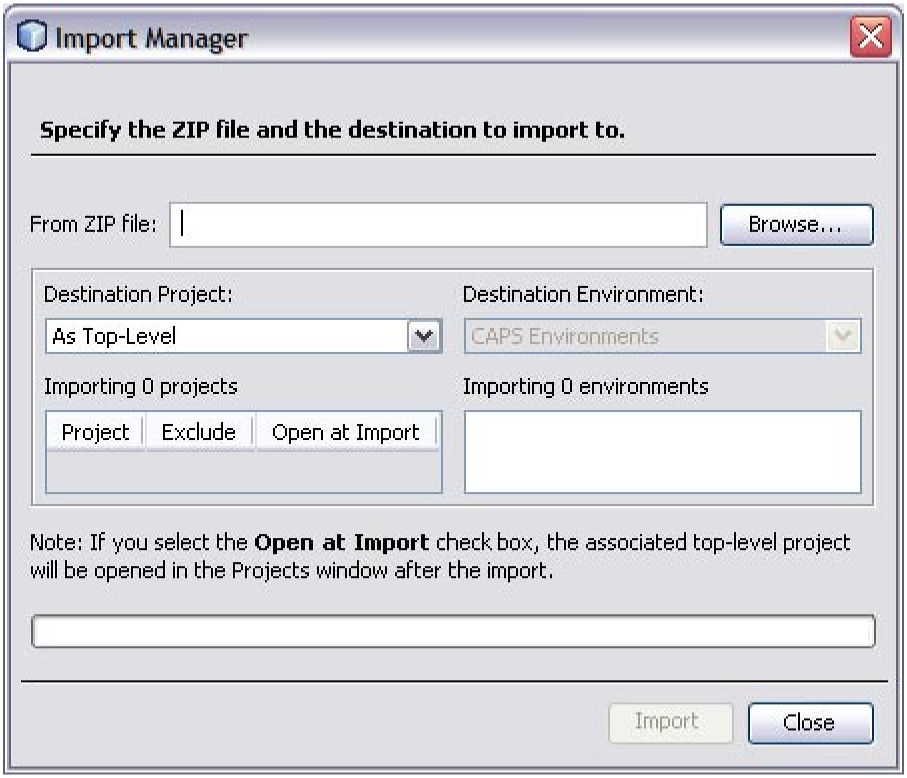
8 **Import the Project back in to the Repository.**

a. **On the NetBeans menu bar, select Tools, point to CAPS Repository, and then select Import Project from the drop-down menu.**

A confirmation dialog box appears asking if you need to save any changes.

b. **Click Yes to proceed with importing a project.**

The Import Manager appears.



Import Manager

Specify the ZIP file and the destination to import to.

From ZIP file:

Destination Project: As Top-Level

Destination Environment: CAPS Environments

Importing 0 projects

Project	Exclude	Open at Import

Importing 0 environments

Note: If you select the **Open at Import** check box, the associated top-level project will be opened in the Projects window after the import.

c. **Click Browse, navigate to the location you specified for the export, and select the Project you just exported.**

d. **Click Import.**

It may take a few seconds to import the project. When the project is imported, the Import Status dialog box appears.

e. **Click OK on the dialog box.**

The CAPS Repository is refreshed, and you now have two copies of the same Project in the Repository.

Customizing Predefined Collaborations for HL7

The predefined Collaborations are designed to be extended and modified, however, for HL7 compliant systems this is not necessary. If you need to modify an HL7 Collaboration, it is strongly suggested that you use the template Collaborations as the basis for any new Collaborations. You should always keep a copy of the sample Projects in their original form for this use.

The following steps describe creating copies of the HL7 Collaborations and adding a new Message Library to an existing Collaboration:

- "Creating Copies of HL7 Collaborations" on page 102
- "Adding an HL7 Message Library to an Existing Collaboration" on page 104

Creating Copies of HL7 Collaborations

Production Projects contain several Collaborations. To create multiple copies of the HL7 Collaborations for your Project, copy and paste the original Collaborations into your project.

▼ To Create Copies of HL7 Collaborations

1 **On the Projects window of the NetBeans IDE, expand the Project containing the Collaboration you want to copy until you see the Collaboration.**

2 **Right-click the Collaboration you want to copy, and select Copy.**

3 **Right-click the Project to which you want to copy the Collaboration, and select Paste.**

Note – A copy of the Collaboration appears in your Project. The Collaboration can be copied within the Project or to another Project. If you copy it to the same Project, a number is appended to the Collaboration name.

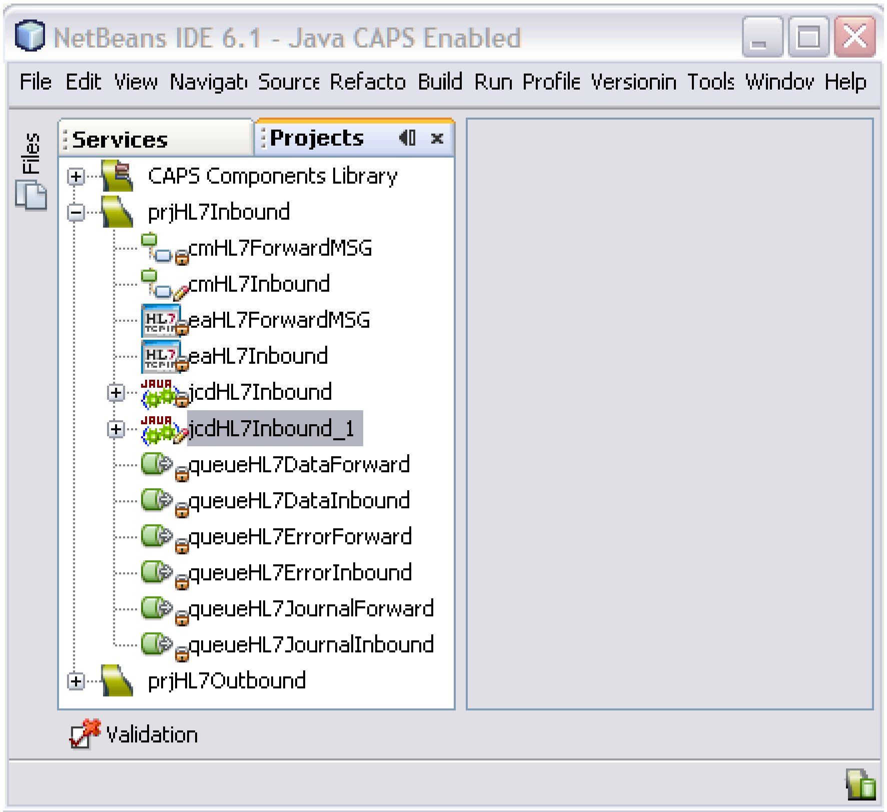


FIGURE 4 Collaboration — Copied to Project Tree

4 **Repeat this step for additional copies of the Collaboration.**

5 **To add each new Collaboration to a Project's Connectivity Map, drag and drop the Collaboration from the Projects Explorer tree to the Connectivity Map Editor.**

The new Collaborations can now be associated with the Project's components.

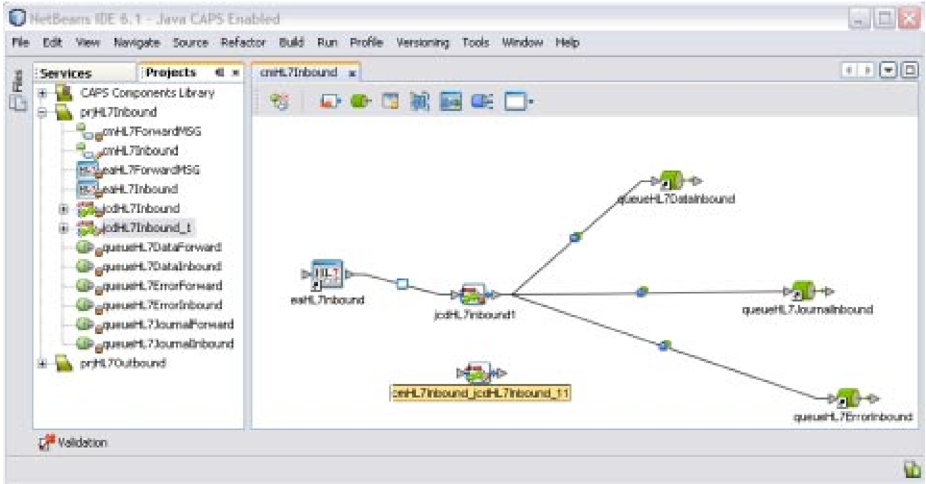


FIGURE 5 Connectivity Map — Newly Added Collaborations

Adding an HL7 Message Library to an Existing Collaboration

In some cases, a specific HL7 message or messages might need to be added to the Collaboration. Perform the following to add a HL7 Message Library to an existing Collaboration.

Note – Make sure you are using the correct version of the Message Library for the Collaboration you are working with.

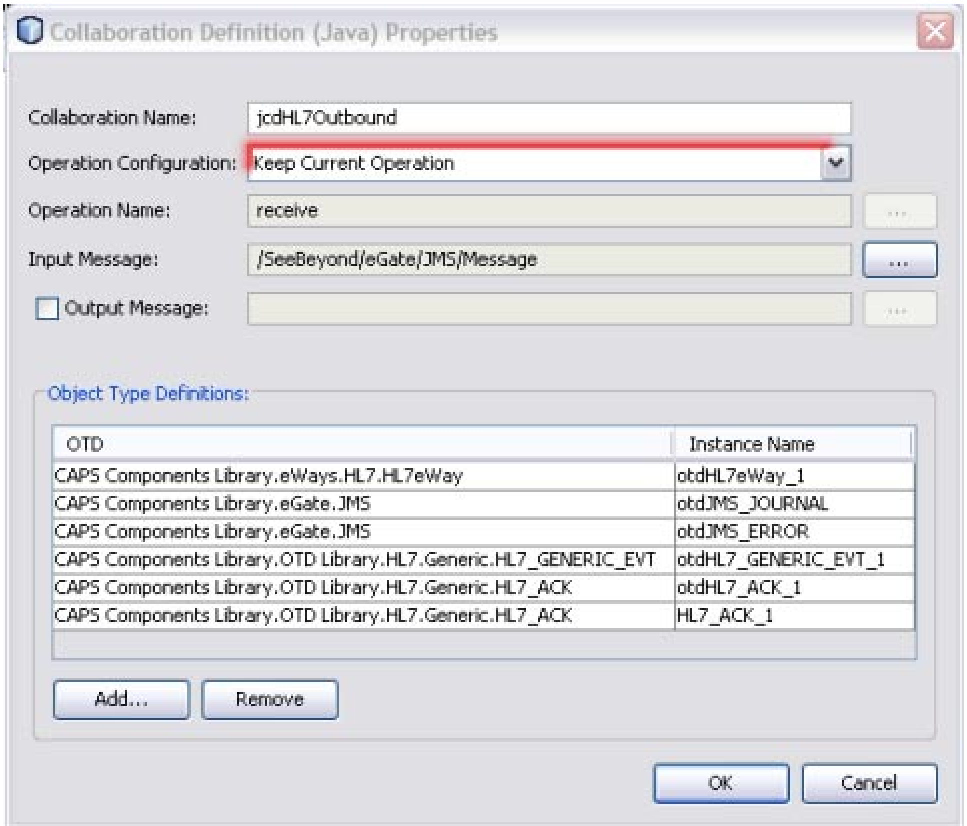
▼ To Add HL7 Message Library to an Existing Collaboration

1. **On the NetBeans Project window, right-click the Collaboration to which you want to add the HL7 Message Library.**

2. **Select Properties from the drop-down menu.**

   The Collaboration Definition (Java) Properties dialog box appears.

3. **Select Keep Current Operation as the value for the Operation Configuration field.**



Collaboration Definition (Java) Properties

Collaboration Name: jcdHL7Outbound

Operation Configuration: Keep Current Operation

Operation Name: receive

Input Message: /SeeBeyond/eGate/JMS/Message

□ Output Message:

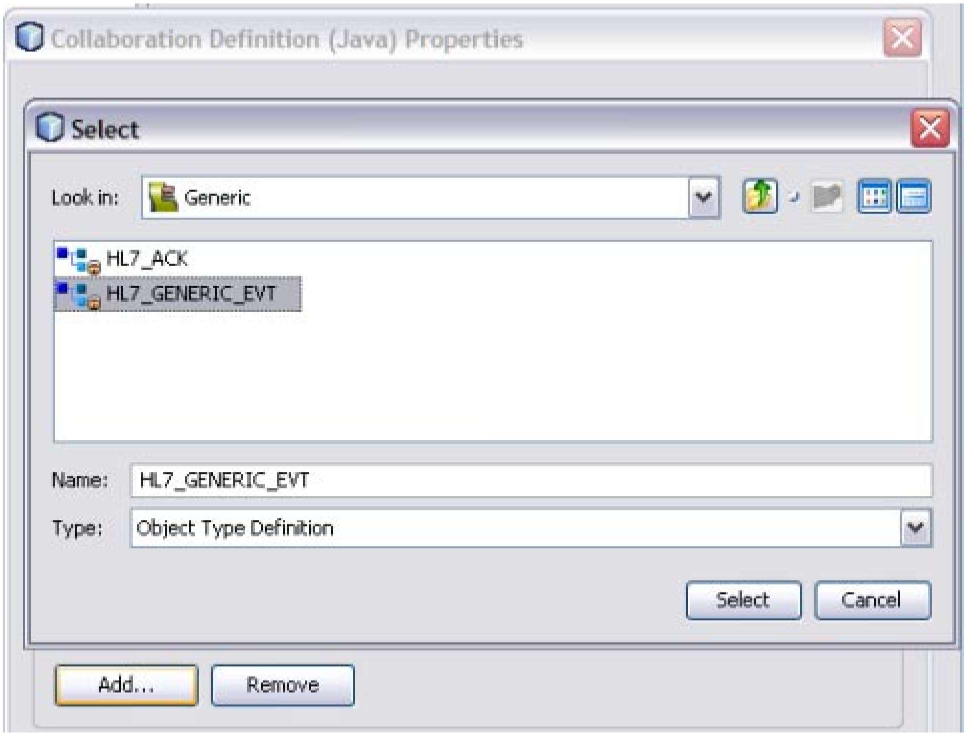
Object Type Definitions:

OTD	Instance Name
CAPS Components Library.eWays.HL7.HL7eWay	otdHL7eWay_1
CAPS Components Library.eGate.JMS	otdJMS_JOURNAL
CAPS Components Library.eGate.JMS	otdJMS_ERROR
CAPS Components Library.OTD Library.HL7.Generic.HL7_GENERIC_EVT	otdHL7_GENERIC_EVT_1
CAPS Components Library.OTD Library.HL7.Generic.HL7_ACK	otdHL7_ACK_1
CAPS Components Library.OTD Library.HL7.Generic.HL7_ACK	HL7_ACK_1

FIGURE 6 Collaboration Definition (Java) Properties

4 **Click Add to select the object type definition.**

A dialog box appears where you can select the Message Library to add.



Name: HL7_GENERIC_EVT

Type: Object Type Definition

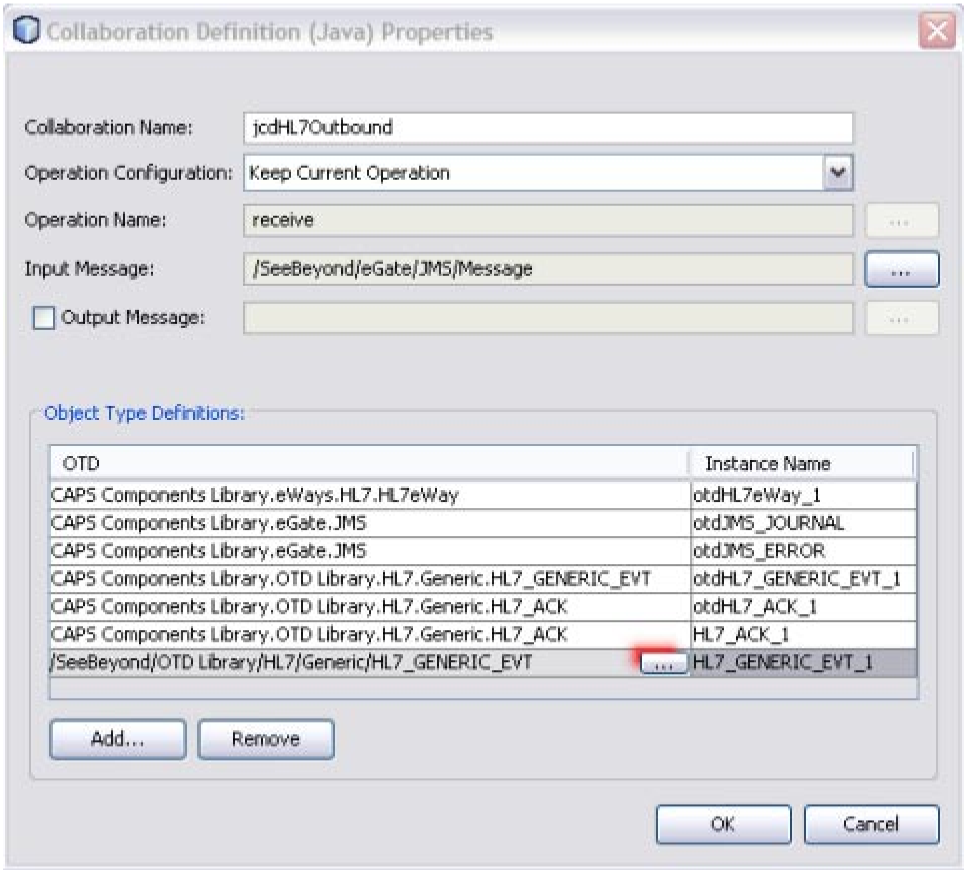
FIGURE 7 Select Object Type Definition

5 **In the Look In field, navigate to the location of the Message Library.**

The HL7 Message Libraries are located in CAPS Components Library > Message Library > HL7.

6 **Select the Message Library and then click Select.**

The Message Library is added to the list.



Collaboration Name: jcdHL7Outbound

Operation Configuration: Keep Current Operation

Operation Name: receive

Input Message: /SeeBeyond/eGate/JMS/Message

□ Output Message:

Object Type Definitions:

OTD	Instance Name
CAPS Components Library.eWays.HL7.HL7eWay	otdHL7eWay_1
CAPS Components Library.eGate.JMS	otdJMS_JOURNAL
CAPS Components Library.eGate.JMS	otdJMS_ERROR
CAPS Components Library.OTD Library.HL7.Generic.HL7_GENERIC_EVT	otdHL7_GENERIC_EVT_1
CAPS Components Library.OTD Library.HL7.Generic.HL7_ACK	otdHL7_ACK_1
CAPS Components Library.OTD Library.HL7.Generic.HL7_ACK	HL7_ACK_1
/SeeBeyond/OTD Library/HL7/Generic/HL7_GENERIC_EVT	HL7_GENERIC_EVT_1

FIGURE 8 Message Library Added to the List

Tip – You can also modify an existing HL7 V2 Message Library by clicking the Message Library to change, clicking the ellipsis that appears, and selecting the appropriate Message Library.

7 **On the Collaboration Definition Properties dialog box, click OK.**

About TCP/IP HL7 V2 Collaborations

This section provides an overview and description of the structure and functionality of the inbound and outbound Collaborations provided as part of the TCP/IP HL7 Adapter for V2.x.

The following topics provide information about the sample V2 Collaborations and provide instructions for copying and customizing them:

- "TCP/IP HL7 V2 Adapter Projects Overview" on page 108
- "TCP/IP HL7 V2 Adapter Collaborations" on page 108
- "Inbound HL7 V2 Collaboration Overview" on page 109
- "Outbound HL7 V2 Collaboration Overview" on page 114

TCP/IP HL7 V2 Adapter Projects Overview

Several sample Projects are provided with the TCP/IP HL7 Adapter to illustrate different scenarios for connecting with external HL7 systems. You can download the sample Projects from the Downloads page of the Java CAPS Suite Installer.

While this section focuses on the standard inbound and outbound HL7 scenarios, you can choose from any of the following HL7 V2 samples:

- HL7 Adapter Inbound Collaboration Projects
- HL7 Adapter Outbound Collaboration Projects
- HL7 Adapter Inbound Collaboration Projects With XML Support Over MLLPV2
- HL7 Adapter Outbound Collaboration Projects With XML Support Over MLLPV2
- HL7 Adapter Inbound Collaboration Projects With XML Support Over MLLPV1
- HL7 Adapter Outbound Collaboration Projects With XML Support Over MLLPV1

TCP/IP HL7 V2 Adapter Collaborations

This section describes the inbound Collaboration (jcdHL7Inbound) and the outbound Collaboration (jcdHL7Outbound), provided within the standard sample Projects for inbound and outbound HL7 V2 messaging. These Collaborations are designed to work as is for HL7 V2 compliant interfaces, and can be configured for your specific needs using only the property configuration files. If an interface requires special functionality, the Collaboration's Java code is easily accessible for modification, much of which can be created graphically using the Collaboration Editor's Business Rules Designer.

The Collaborations contain a number of Message Libraries that extend functionality for HL7 V2 message handling, logging, error messaging, journaling, and sequence numbering. These include both generic HL7 Message Libraries for HL7 ACK/NAK generation or verification, and the Resource Adapter that communicates to the external system and offers services to the application server. The Collaborations control messaging protocol and all business logic.

The Collaborations are designed to target one unit of work at a time, meaning the resolution of one message at a time. The basic structure of both Collaborations is a state machine implemented in a Java switch statement. The state machine keeps track of the messaging protocol so when a Collaboration is invoked, it can retrieve the state of the connection just handed to it by the RA, and then execute the proper actions based on the state machine.

At the end of each action, the state is set for the next execution of the Collaboration. There are three main states:

- **To Establish:** A new or reset connection needs to have an HL7 session established. If sequence numbering is used, the sequence numbers need to be negotiated.
- **Messaging:** This is where the exchange of messages and ACKs takes place.
- **Shutdown:** This is where any cleanup can happen before the connection is closed, or to close the connection.

Additional Collaborations can be added to a Project to increase message flow.

Note – The TCP/IP HL7 V2 inbound Collaboration publishes received data as a Byte message in JMS using the `sendBytes()` method. However, the HL7 V2 outbound Collaboration expects a Text message from JMS. The adapter is not designed for the HL7 V2 outbound Collaboration to subscribe to a JMS data queue created by the HL7 V2 inbound Collaboration directly. HL7 V2 inbound and outbound Collaborations are designed to communicate through an HL7 V2 TCP/IP adapter connection.

Inbound HL7 V2 Collaboration Overview

The inbound HL7 V2 Collaboration, jcdHL7inbound, contains Message Libraries for the HL7 Resource Adapter, JMS Data, HL7 ACK, JMS Journal, and JMS Error, as well as the Generic HL7 Event. The Collaboration works with its own internal code and the properties configuration files.

This following topics describe the business logic defined in the jcdHL7Inbound Collaboration:

- “Inbound HL7 V2 Collaboration - Part 1” on page 109
- “Inbound HL7 V2 Collaboration - Part 2” on page 112
- “Inbound HL7 V2 Collaboration - Part 3” on page 114

Inbound HL7 V2 Collaboration - Part 1

The inbound Collaboration is triggered by an HL7 V2 message received from an external system or an outbound HL7 V2 client. The Collaboration calls the HL7 V2 user Collaboration rule by executing `receive()`.

The receive method is the entry point to the HL7 V2 user Collaboration, with the following signature:

```
public void
receive(com.stc.connector.appconn.tcpip.hl7.HL7ServerApplication input,
com.stc.connectors.jms.JMS otdJMS_DATA,
com.stc.SeeBeyond.OTD_Library.HL7.Generic.HL7_GENERIC_EVT.GENERIC_EVT
otdHL7_GENERIC_EVT_1,
com.stc.SeeBeyond.OTD_Library.HL7.Generic.HL7_ACK.AC K
otdHL7_ACK_1,com.stc.connectors.jms.JMS otdJMS_JOURNAL,
com.stc.connectors.jms.JMS otdJMS_ERROR) throws Throwable
```

Note – The above text has been wrapped to fit the page.

Once the message is received, the Collaboration determines whether the message needs to be validated. The HL7 V2 message is then validated making sure that the message structure is correct. Various fields in the MSH segment of the message are also validated, such as Version ID and Sending Facility. If these fields do not match the configuration, a NAK is returned.

If sequence numbering is enabled the Collaboration checks to see if the messages sequence number is valid. If the sequence number is not valid, the adapter sends a NAK.

The validated HL7 V2 message moves on to `processInitialHandshake()` and the sequence numbers are synchronized. The sequence number within the message is checked against the expected sequence number. If the numbers match, the Collaboration sends an ACK, if not it sends a NAK. The ACK or NAK includes information from various fields of the incoming MSH segment. The ACKs level of acknowledgement is set to A (acknowledgement is sent after the message is successfully processed) or C (acknowledgement is sent when the message is successfully received).

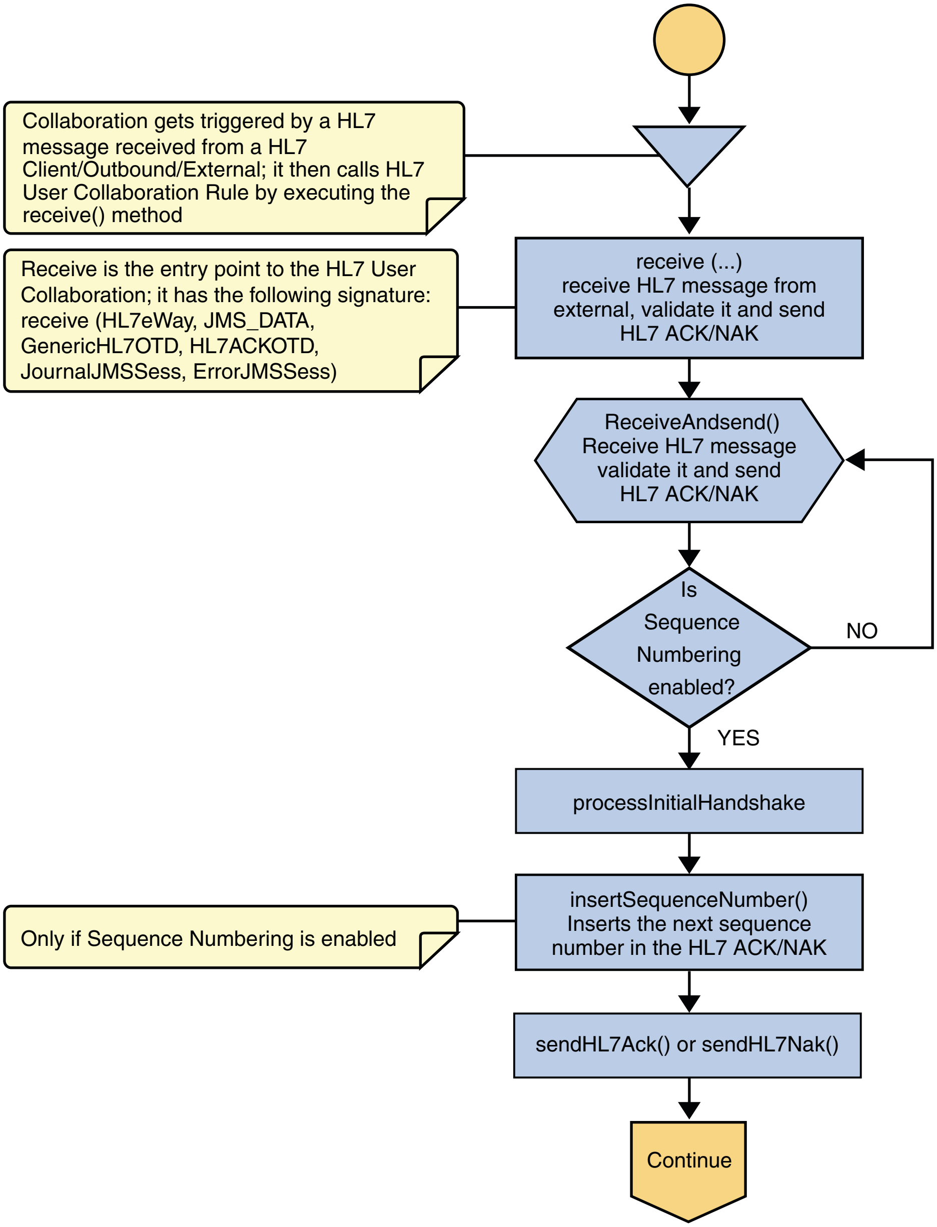

Collaboration gets triggered by a HL7 message received from a HL7 Client/Outbound/External; it then calls HL7 User Collaboration Rule by executing the receive() method
Receive is the entry point to the HL7 User Collaboration; it has the following signature: receive (HL7eWay, JMS_DATA, GenericHL7OTD, HL7ACKOTD, JournalJMSSess, ErrorJMSSess)
receive (...) receive HL7 message from external, validate it and send HL7 ACK/NAK
ReceiveAndsend() Receive HL7 message validate it and send HL7 ACK/NAK
Is Sequence Numbering enabled?
NO
YES
processInitialHandshake
Only if Sequence Numbering is enabled
insertSequenceNumber() Inserts the next sequence number in the HL7 ACK/NAK
sendHL7Ack() or sendHL7Nak()
Continue

FIGURE 9 Inbound HL7 V2 Collaboration - Part 1

Inbound HL7 V2 Collaboration - Part 2

The Collaboration receives the HL7 V2 message from the external using `receiveHL7message()`. If an exception occurs due to incomplete data and the adapter fails to read the data within the configured number of retries, the associated recourse action is taken. If the exception is due to no response, the associated recourse action is taken.

If no exception occurs, `validateHL7Message()` is called, which validates the message to determine whether to ACK or NAK the message. Other helper methods are also called to validate the HL7 V2 message.

If the HL7 V2 message does not pass validation, the Collaboration calls `makeNak()` and `sendHL7Nak()` to create and send the NAK to the external system. The HL7 V2 message, with the NAK, is archived to the error queue. If the number of consecutive NAKs sent surpasses the maximum number of retries, the associated recourse action is taken.

If the HL7 V2 message passes validation, the Collaboration calls `makeAck()` and `sendHL7Ack()` to create and send the ACK to the external.

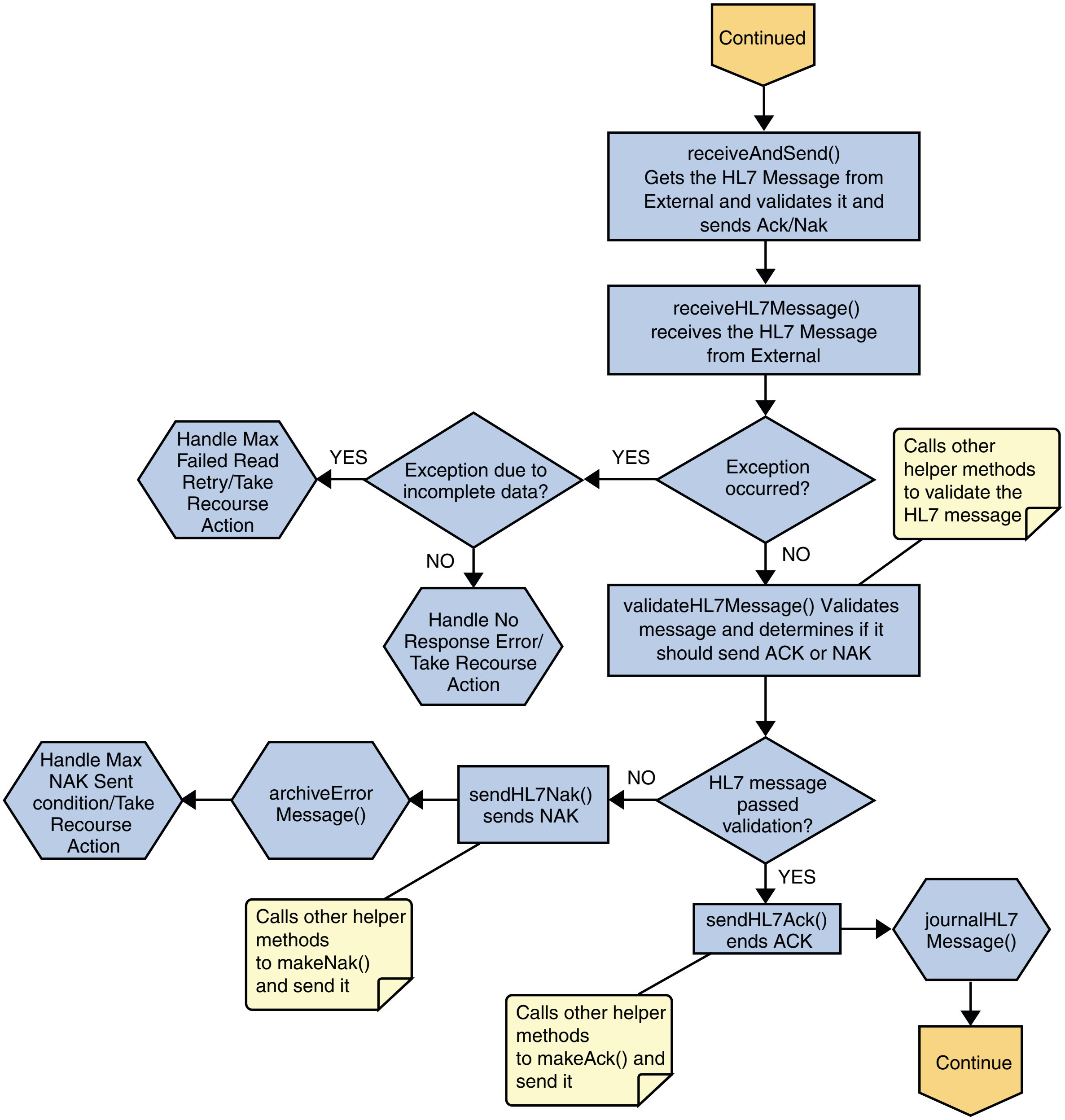

Continued
receiveAndSend() Gets the HL7 Message from External and validates it and sends Ack/Nak
receiveHL7Message() receives the HL7 Message from External
Exception occurred?
YES
Exception due to incomplete data?
YES
Handle Max Failed Read Retry/Take Recourse Action
NO
Handle No Response Error/ Take Recourse Action
NO
Calls other helper methods to validate the HL7 message
validateHL7Message() Validates message and determines if it should send ACK or NAK
HL7 message passed validation?
NO
sendHL7Nak() sends NAK
archiveError Message()
Handle Max NAK Sent condition/Take Recourse Action
Calls other helper methods to makeNak() and send it
YES
sendHL7Ack() ends ACK
journalHL7 Message()
Calls other helper methods to makeAck() and send it
Continue

FIGURE 10 Inbound HL7 V2 Collaboration - Part 2

Inbound HL7 V2 Collaboration - Part 3

After the ACK is sent, the HL7 V2 message and the ACK are journaled to the JMS queue Journal destination. If the message fails to journal, the associated recourse action is taken.

If sequence numbering is enabled, the `processAckNakSequenceNumbering` method calculates the next sequence number and stores the number in the sequence number file by calling the `updateSequenceNumberFile` method to persist the next sequence number.

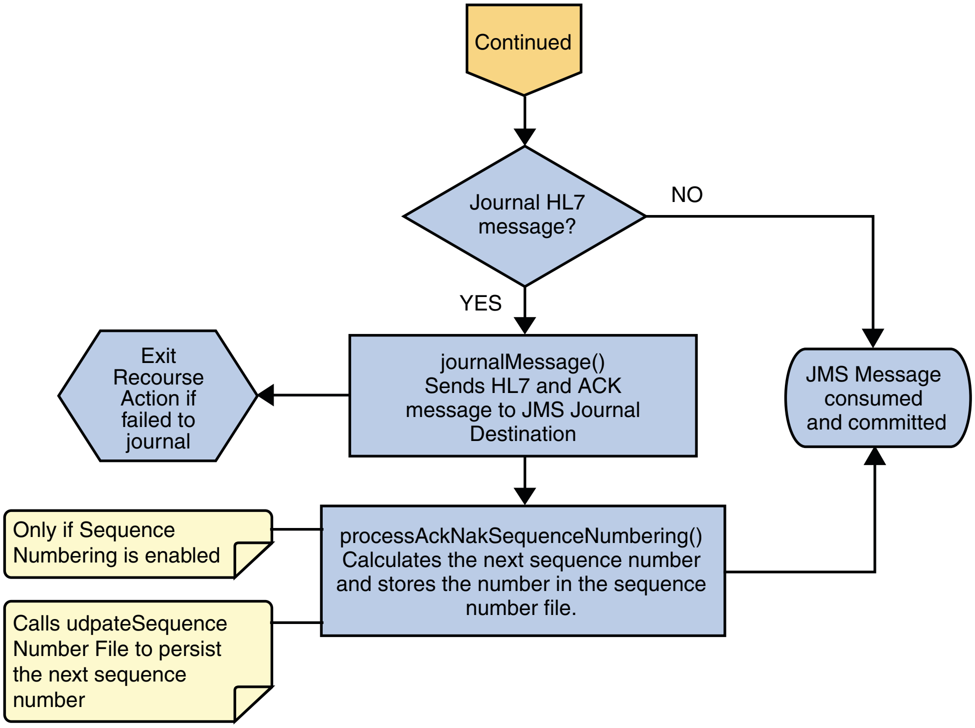

Continued
Journal HL7 message?
NO
YES
Exit Recourse Action if failed to journal
journalMessage() Sends HL7 and ACK message to JMS Journal Destination
JMS Message consumed and committed
Only if Sequence Numbering is enabled
processAckNakSequenceNumbering() Calculates the next sequence number and stores the number in the sequence number file.
Calls udpateSequence Number File to persist the next sequence number

FIGURE 11 Inbound HL7 V2 Collaboration - Part 3

Outbound HL7 V2 Collaboration Overview

The outbound HL7 V2 Collaboration, jcdHL7Outbound, contains Message Libraries for the HL7 Resource Adapter, JMS Data, HL7 ACK, JMS Journal, and JMS Error, as well as the Generic HL7 Event. The Collaboration works with its own internal code and the properties configuration files. The outbound Collaboration assumes that it is reading valid HL7 V2 messages, so the data flow that feeds this Collaboration must ensure this.

The following topics describe the business logic defined in the jcdHL7Outbound Collaboration:

- "Outbound HL7 V2 Collaboration - Part 1" on page 115
- "Outbound HL7 V2 Collaboration - Part 2" on page 117
- "Outbound HL7 V2 Collaboration - Part 3" on page 119
- "HL7 V2 Outbound Test Collaboration" on page 119

Outbound HL7 V2 Collaboration - Part 1

The Collaboration is triggered by a JMS HL7 V2 message. The Collaboration then calls the HL7 User Collaboration Rule by executing the receive method. Receive is the entry point to the HL7 User Collaboration, with the following signature,

```
receive (input, otdHL7eWay_1, otdJMS_JOURNAL, otdJMS_ERROR, otdHL7_ACK_1,
otdHL7_GENERIC_EVT_1)
```

The incoming HL7 V2 message is then validated, making sure that the message structure is correct. Various fields of the message are also validated, such as Sending Facility, Version ID, and MSH.

If the message does not pass validation, an error occurs and the associated recourse action is applied. If the HL7 V2 message passes validation, the message moves on to processInitialHandshake() to receive a sequence number (if sequences numbering is enabled). The Collaboration takes the sequence number from the sequence numbering file and determines the next number to use. This number is then inserted into the HL7 V2 message.

Next, the message moves on to processMessage(), which calls the helper method, sendAndReceive(). The sendAndReceive method sends the HL7 message, waits for an HL7 ACK message, and processes the ACK or NAK. The validation also checks the message structure to see if the message is unmarshaled. If a valid ACK is not received, it continues to send the HL7 V2 message up to the configured number of retries, at which time an error occurs and the associated recourse action is taken. If a valid ACK is received, the message moves on to insertSequenceNumber().

If sequence numbering is enabled, the insertSequenceNumber method inserts the sequence number and calls sendHL7Message(). The sendHL7Message method sends the HL7 V2 message to the external using the HL7 adapter Message Library.

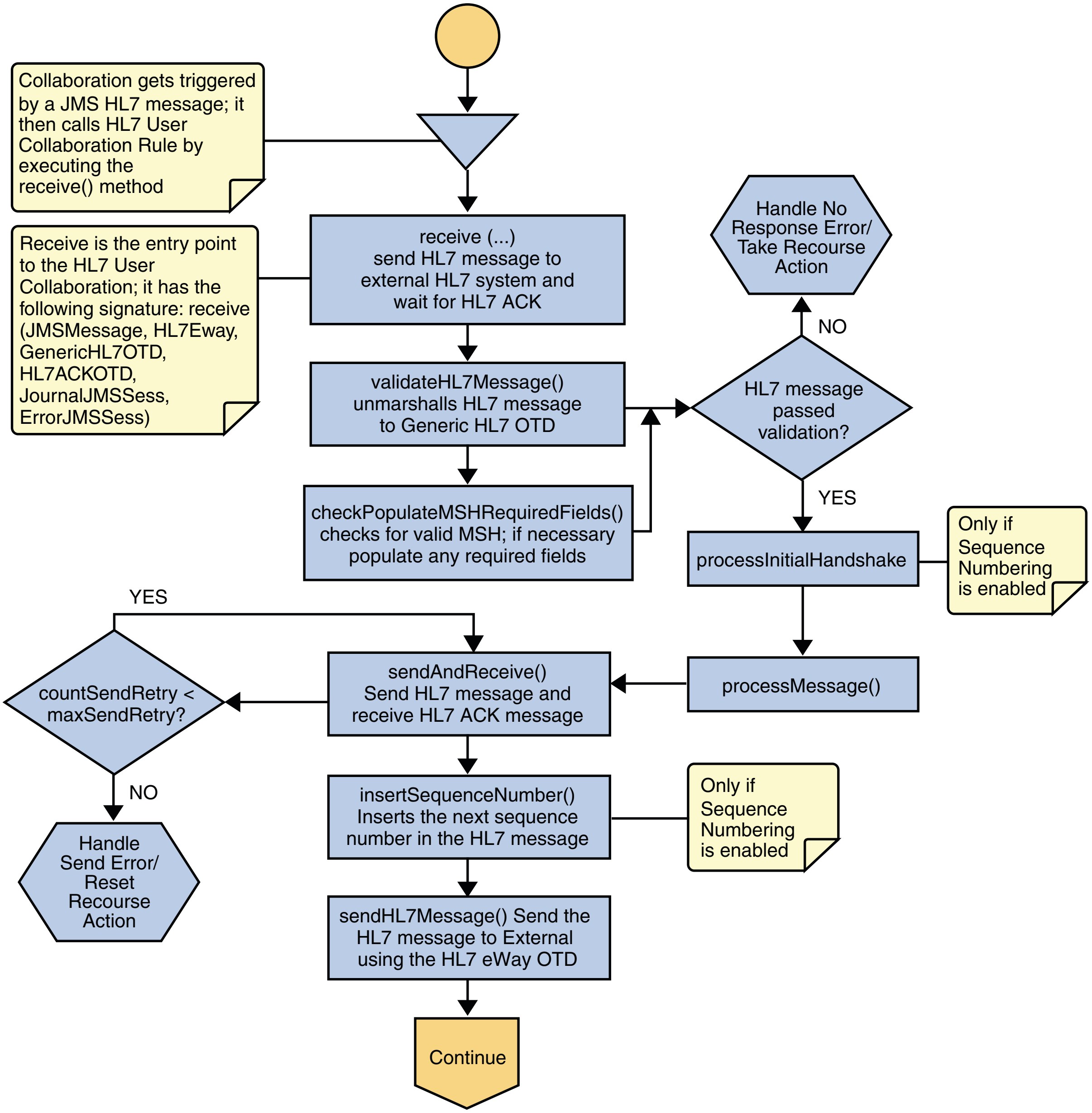

Collaboration gets triggered by a JMS HL7 message; it then calls HL7 User Collaboration Rule by executing the receive() method
Receive is the entry point to the HL7 User Collaboration; it has the following signature: receive (JMSMessage, HL7Eway, GenericHL7OTD, HL7ACKOTD, JournalJMSSess, ErrorJMSSess)
receive (...) send HL7 message to external HL7 system and wait for HL7 ACK
validateHL7Message() unmarshalls HL7 message to Generic HL7 OTD
checkPopulateMSHRequiredFields() checks for valid MSH; if necessary populate any required fields
Handle No Response Error/ Take Recourse Action
NO
HL7 message passed validation?
YES
processInitialHandshake
Only if Sequence Numbering is enabled
processMessage()
YES
countSendRetry < maxSendRetry?
NO
Handle Send Error/ Reset Recourse Action
sendAndReceive() Send HL7 message and receive HL7 ACK message
insertSequenceNumber() Inserts the next sequence number in the HL7 message
Only if Sequence Numbering is enabled
sendHL7Message() Send the HL7 message to External using the HL7 eWay OTD
Continue

FIGURE 12 Outbound HL7 V2 Collaboration - Part 1

Outbound HL7 V2 Collaboration - Part 2

The Collaboration receives the HL7 ACK or NAK from the external system using `receiveHL7AckNak()`. If an exception occurs due to incomplete data and the adapter fails to read the data within the configured number of retries, the associated recourse action is taken. If the exception is due to no response, the associated recourse action is taken. If no exception occurs, the ACK or NAK message moves on to `isAckMessage()`, which validates the message to determine whether the message is an ACK or a NAK.

Next the `validateAckNak` method unmarshals the message to the ACK Message Library and validates the MSH/MSA data. It also calls other helper methods such as `checkSendingReceivingApplication()` and `checkSendingReceivingFacility()` to validate the ACK message.

If the message does not pass validation it is handled as a NAK, the associated recourse action is taken, and the message is archived in the error queue.

If the message is a NAK, the associated recourse action is taken, and the message is archived in the error queue, along with the NAK message, as a JMS property.

If the message is an ACK and passes validation, the message is sent on to the `journalMessage` method

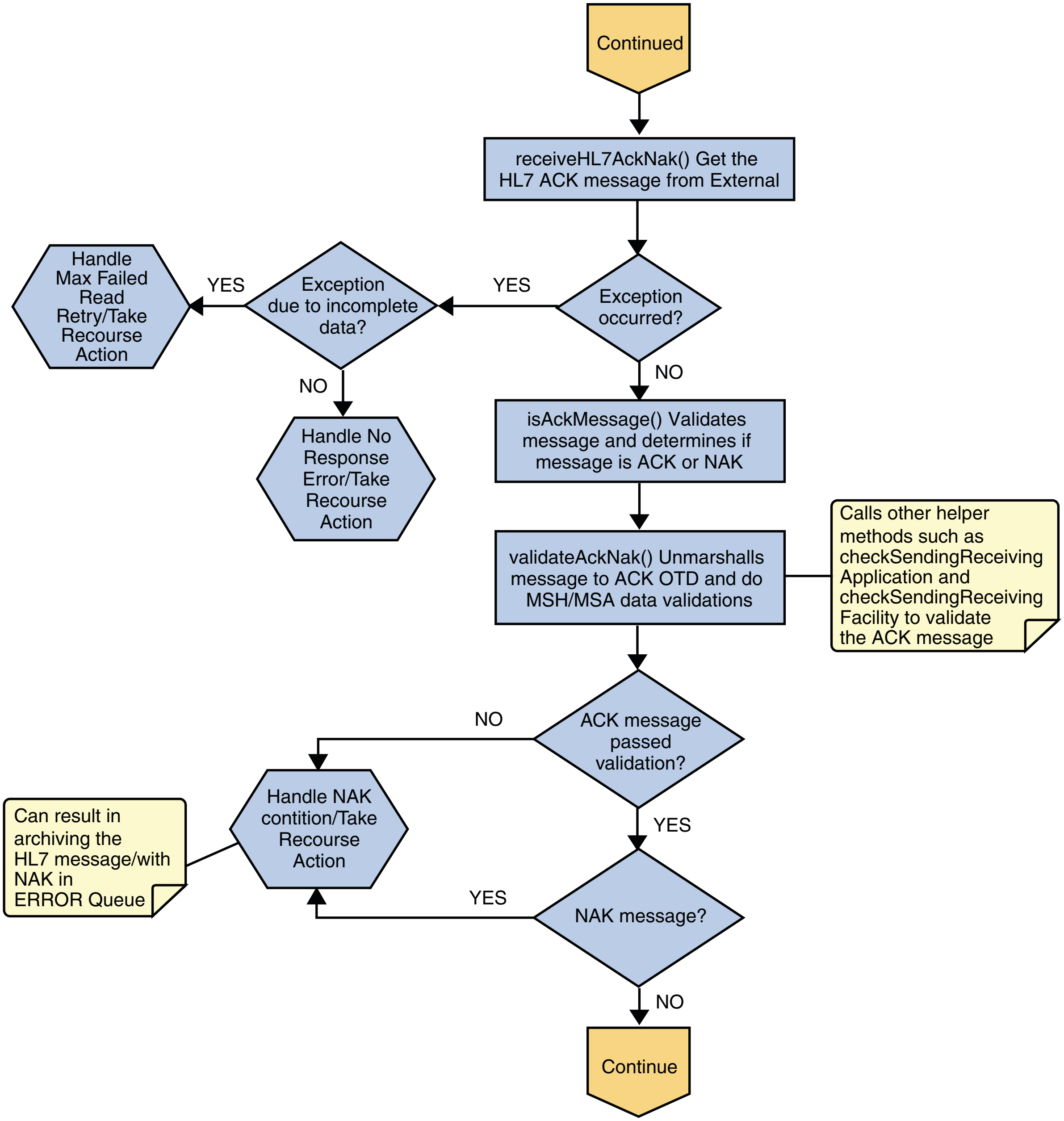

Continued
receiveHL7AckNak() Get the HL7 ACK message from External
Exception occurred?
YES
Exception due to incomplete data?
YES
Handle Max Failed Read Retry/Take Recourse Action
NO
Handle No Response Error/Take Recourse Action
NO
isAckMessage() Validates message and determines if message is ACK or NAK
validateAckNak() Unmarshalls message to ACK OTD and do MSH/MSA data validations
Calls other helper methods such as checkSendingReceiving Application and checkSendingReceiving Facility to validate the ACK message
ACK message passed validation?
NO
Handle NAK contition/Take Recourse Action
Can result in archiving the HL7 message/with NAK in ERROR Queue
YES
NAK message?
YES
NO
Continue

FIGURE 13 Outbound HL7 V2 Collaboration - Part 2

Outbound HL7 V2 Collaboration - Part 3

If the ACK message passes validation, the HL7 V2 message and ACK message are sent to the JMS journal destination. If the message fails to journal, the associated recourse action is taken.

If sequence numbering is enabled, the `processAckNakSequenceNumbering` method calculates the next sequence number and stores the number in the sequence number file, calling `updateSequenceNumberFile` to persist the next sequence number.

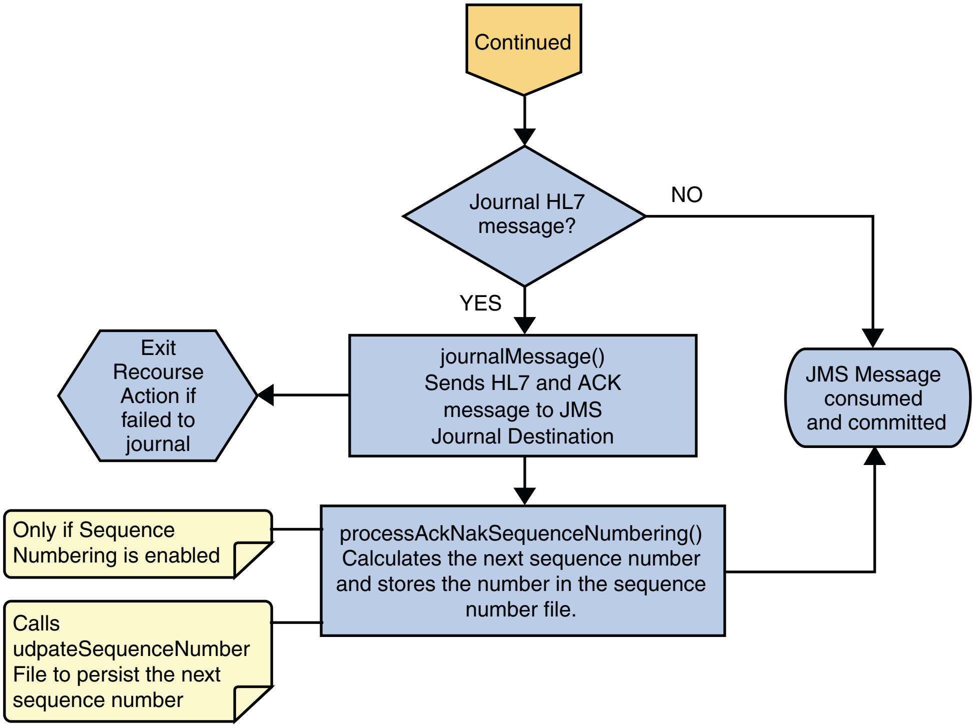

Continued
Journal HL7 message?
NO
YES
journalMessage() Sends HL7 and ACK message to JMS Journal Destination
Exit Recourse Action if failed to journal
JMS Message consumed and committed
Only if Sequence Numbering is enabled
processAckNakSequenceNumbering() Calculates the next sequence number and stores the number in the sequence number file.
Calls udpateSequenceNumber File to persist the next sequence number

FIGURE 14 Outbound HL7 V2 Collaboration - Part 3

HL7 V2 Outbound Test Collaboration

In addition to the inbound and outbound HL7 V2 Collaborations, an outbound test Collaboration, jcdHL7OutboundTestDriver, is provided to test the HL7 V2 outbound and HL7 V2 outbound delayed ACK samples.

The jcdHL7OutboundTestDriver Collaboration simply picks up HL7 V2 messages from the File Adapter and sends the message to the JMS queue. This is used by the sample Projects to test the number of HL7 V2 messages processed per minute.

About TCP/IP HL7 V3 Collaborations

The following topics provide a brief overview of HL7 V3, and describe the HL7 V3 Collaborations and provide instructions on copying and customizing the Collaborations:

For more information about HL7 V3, refer to the HL7 V3 Guide.

About HL7 V3

HL7 V3, like V2.x, is a standard for exchanging messages among healthcare information systems. However, HL7 V3 is not compatible with V2. V3 defines a more fully-specified data model with less flexibility for customization. This presents a tighter standard that is easier to use, and provides a higher level of formal modeling, complexity, and internal consistency.

HL7 V3 is based on information models, which define the specifications of sets of data that are specific to one area of interest. There are different types of information models, including:

- **Reference Information Model (RIM)** – Defines the information content for a domain.
- **Domain Information Model (D-MIM)** – Defines a set of class clones, attributes, and relationships, and is a subset of the RIM. Domains include the following:

  Domains under Administrative Management:

  - Accounting and Billing
  - Claims & Reimbursements
  - Patient Administration
  - Personnel Management
  - Scheduling

  Domains under Health and Clinical Management:

  - Clinical Document Architecture
  - Medical Records
  - Public Health Reporting
  - Clinical Genomics
  - Specimen Domain
  - Regulated Studies
- **Refined Message Information Model (R-MIM)** – Defines message-specific content, and is a subset of the D-MIM).

The V3 standard also includes interactions, such as Create Patient Billing Account.

There are three modes of Acknowledgement process in HL7 V3:

- Immediate Mode
- Deferred Mode
- Queued Mode

TCP/IP HL7 V3 Adapter Projects Overview

Several HL7 V3 sample Projects are provided with the TCP/IP HL7 Adapter to illustrate different scenarios for connecting with external HL7 systems. You can download the sample Projects from the Downloads page of the Java CAPS Suite Installer.

While this section focuses on the standard inbound and outbound HL7 scenarios, you can choose from any of the following samples:

- HL7 Adapter Inbound Collaboration Project for HL7V3 PRPA_IN403001UV01 Interaction over MLLPV2
- HL7 Adapter Outbound Collaboration Project for HL7V3 PRPA_PRPA_IN403001UV01 Interaction over MLLPV2
- HL7 Adapter Inbound Collaboration Project for HL7V3 PRPA_IN403001UV01 With Schematron Validation Enabled
- HL7 Adapter Collaboration Project for HL7 V2 to V3 Conversion and ebXML Wrapper on HL7V3 Message
- HL7 Adapter Inbound Collaboration Project That Invokes Sub Collaborations for HL7V3

TCP/IP HL7 V3 Adapter Collaborations

This section describes two inbound Collaboration (jcdHL7V3Inbound and jcdHL7V3DefrdInbound) and an outbound Collaboration (jcdHL7v3Outbound), provided within the standard sample Projects for inbound and outbound HL7 V3 messaging. These Collaborations are designed to work as is for HL7 V3 compliant interfaces, and can be configured for your specific needs using the property configuration files. If an interface requires special functionality, the Collaboration's Java code is easily accessible for modification, much of which can be created graphically using the Collaboration Editor's Business Rules Designer.

The Collaborations contain a number of Message Libraries that extend functionality for HL7 V3 message handling, logging, error messaging, journaling, and sequence numbering. These include both HL7 Message Libraries for the V3 domains, and the Resource Adapter that communicates to the external system and offers services to the application server. The Collaborations control messaging protocol and all business logic.

There are four inbound Collaborations in the prjHL7V3Inbound Project:

1. Inbound HL7 V3 Immediate Collaboration
2. Inbound HL7 V3 Deferred Collaboration
3. Inbound HL7 V3 Queue Manager
4. Inbound HL7 V3 Message Publisher

Inbound HL7 V3 Immediate Collaboration Overview

The Inbound HL7 V3 Collaboration, jcdHL7V3Inbound, contains Message Libraries for the HL7 V3 Resource Adapter, JMS Data, JMS Journal, and JMS Error, as well as the HL7 V3 Patient Administration Domain Interaction Event (PRPA_IN403001UV01) and the corresponding HL7 V3 Acknowledgements (MCCI_IN000004UV01). The Collaboration works with its own internal code and the properties configuration files.

Note – Immediate mode contains only one type of Acknowledgement Message Library, MCCI_IN000004UV01.

The following topics describe the business logic defined in the jcdHL7V3Inbound Collaboration:

- “HL7 V3 Standard Inbound Message Mode Data Flow (For Immediate Mode of ACK Process) — Part 1” on page 122
- “HL7 V3 Standard Inbound Message Mode Data Flow (For Immediate Mode of ACK Process) — Part 2” on page 125

HL7 V3 Standard Inbound Message Mode Data Flow (For Immediate Mode of ACK Process) — Part 1

An HL7 V3 message triggers an inbound Collaboration received from an external system or an outbound HL7 V3 client. The Collaboration calls the HL7 V3 user Collaboration rule by executing `receive()`.

The receive method is the entry point to the HL7 V3 User Collaboration, with the following signature:

```
public void
receive(com.stc.connector.appconn.tcpip.hl7.HL7ServerApplication input,
com.stc.connectors.jms.JMS otdJMS_DATA,
xsd.hl7v3.PRPA_IN403001UV01.PRPA_IN403001UV01_  otd_PRPA_IN403001UV01_1
xsd.hl7v3.MCCI_IN000004UV01.MCCI_IN000004UV01_  otd_MCCI_IN000004UV01
otdHL7_ACK_1,com.stc.connectors.jms.JMS otdJMS_JOURNAL,
com.stc.connectors.jms.JMS otdJMS_ERROR) throws Throwable.
```

Note – The above text has been wrapped to fit the page.

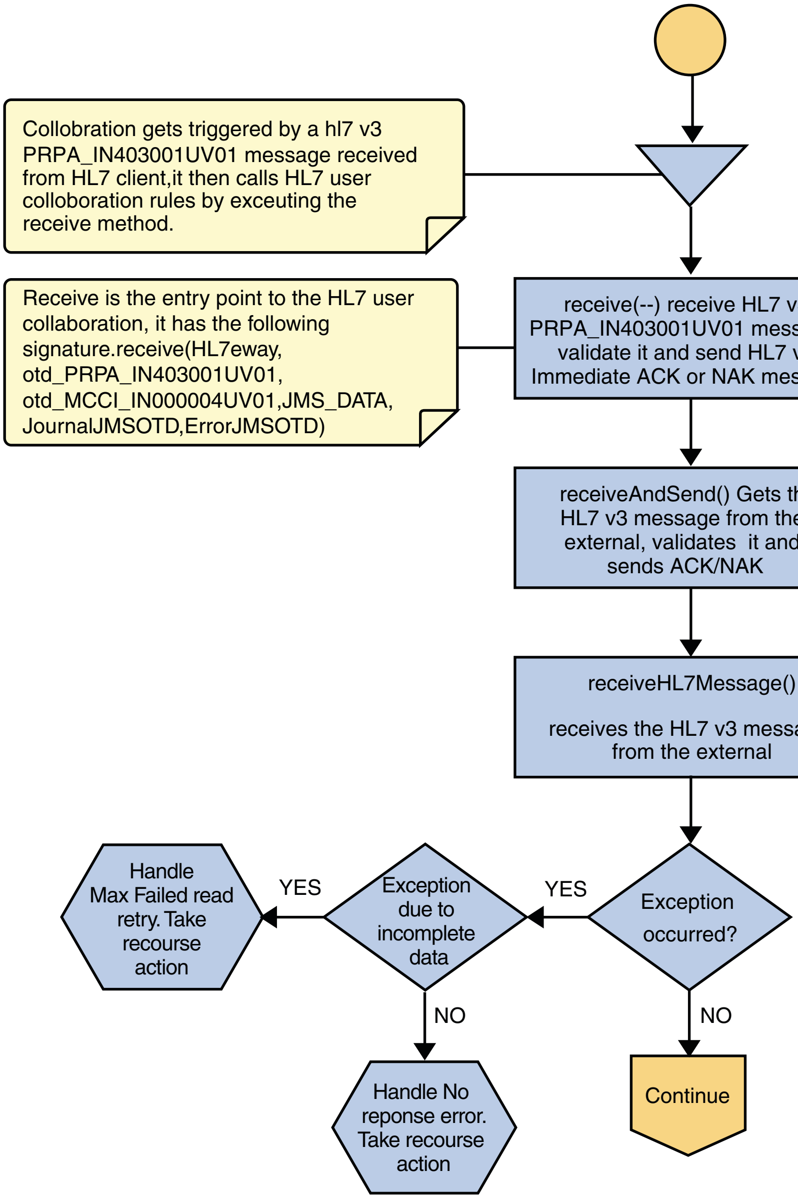

Collobration gets triggered by a hl7 v3 PRPA_IN403001UV01 message received from HL7 client,it then calls HL7 user colloboration rules by exceuting the receive method.
Receive is the entry point to the HL7 user collaboration, it has the following signature.receive(HL7eway, otd_PRPA_IN403001UV01, otd_MCCI_IN000004UV01,JMS_DATA, JournalJMSOTD,ErrorJMSOTD)
receive(--) receive HL7 v3 PRPA_IN403001UV01 message, validate it and send HL7 v3 Immediate ACK or NAK message
receiveAndSend() Gets the HL7 v3 message from the external, validates it and sends ACK/NAK
receiveHL7Message()
receives the HL7 v3 message from the external
Exception occurred?
YES
NO
Exception due to incomplete data
YES
NO
Handle Max Failed read retry. Take recourse action
Handle No reponse error. Take recourse action
Continue

FIGURE 15 Immediate Mode of ACK Process — Part 1

Once the message is received, the Collaboration determines whether the message needs to be validated. The HL7 V3 message is then validated, making sure that the message structure is correct. Various fields in the Transmission Wrapper of the message are also validated, such as

Version Code, Processing Code, Processing Mode Code, and Interaction ID. If these fields do not match the configuration, a NAK is returned.

The Collaboration receives the HL7 V3 message from the external using `receiveHL7Message()`. If an exception occurs due to incomplete data, and the adapter fails to read the data within the configured number of retries, the associated recourse action is taken. If the exception is due to no response, the associated recourse action is taken.

HL7 V3 Standard Inbound Message Mode Data Flow (For Immediate Mode of ACK Process) — Part 2

If no exception occurs, `validateHL7Message()` is called, which validates the message to determine whether to ACK or NAK the message. Other helper methods are also called to validate the HL7 V3 message.

If the HL7 V3 message does not pass validation, the Collaboration calls `makeNak()` and `sendHL7Nak()` to create and send the NAK to the external. The HL7 V3 message, with the NAK, is archived to the error queue. If the number of consecutive NAKs sent surpasses the maximum number of retries, the associated recourse action is taken.

If the HL7 V3 message passes validation, the Collaboration calls `makeAck()` and `sendHL7Ack()` to create and send the ACK to the external.

After the ACK is sent, the HL7 V3 message and the ACK are journaled to the JMS queue journal destination. If the message fails to journal the associated recourse action is taken.

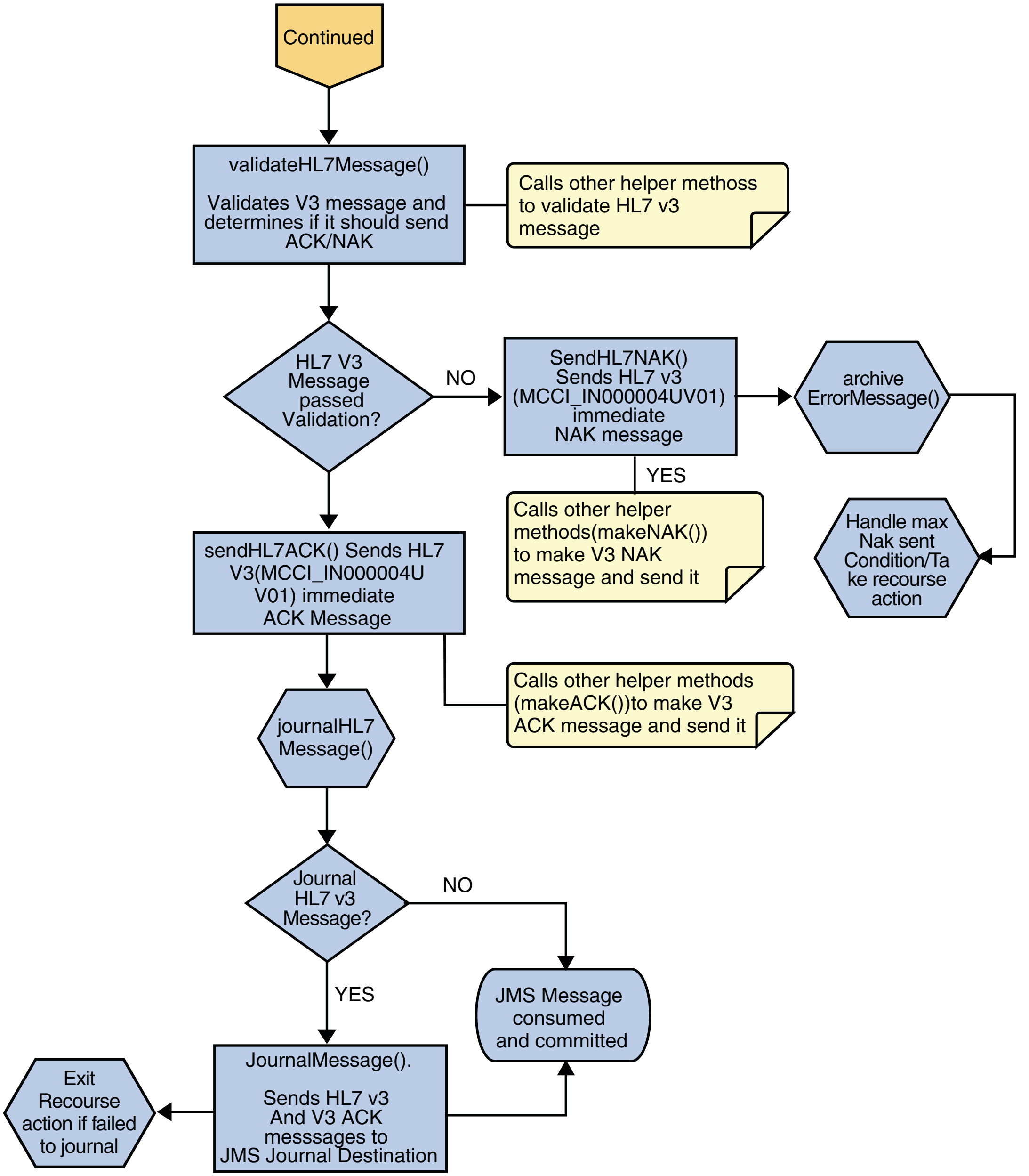

Continued
validateHL7Message()
Validates V3 message and determines if it should send ACK/NAK
Calls other helper methoss to validate HL7 v3 message
HL7 V3 Message passed Validation?
NO
SendHL7NAK() Sends HL7 v3 (MCCI_IN000004UV01) immediate NAK message
archive ErrorMessage()
YES
Calls other helper methods(makeNAK()) to make V3 NAK message and send it
Handle max Nak sent Condition/Ta ke recourse action
sendHL7ACK() Sends HL7 V3(MCCI_IN000004U V01) immediate ACK Message
Calls other helper methods (makeACK())to make V3 ACK message and send it
journalHL7 Message()
Journal HL7 v3 Message?
NO
YES
JMS Message consumed and committed
Exit Recourse action if failed to journal
JournalMessage().
Sends HL7 v3 And V3 ACK messsages to JMS Journal Destination

FIGURE 16 Immediate Mode of ACK Process — Part 2

Inbound HL7 V3 Deferred Collaboration Overview

The Inbound HL7 V3 Collaboration, jcdHL7V3DeferredInbound, contains Message Libraries for the HL7 V3 Resource Adapter, JMS Data, JMS Journal, JMS Error, and JMS APP ACK as well as the HL7 V3 Patient Administration Domain Interaction Event (PRPA_IN403001UV01) and the corresponding HL7 V3 Commit Acknowledgement (MCCI_IN000006UV01) and Application Acknowledgement (MCCI_IN000007UV01). The Collaboration works with its own internal code and the Properties Configuration files.

Note – Deferred mode contains the following two types of Acknowledgement Message Libraries:

- Commit Acknowledgement (MCCI_IN000006UV01)
- Application Acknowledgement (MCCI_IN000007UV01)

The above JCD (jcdHL7V3DeferredInbound) will also work for both immediate mode and deferred mode.

The following topics describe the business logic defined in the jcdHL7V3DeferredInbound Collaboration:

- "HL7 V3 Standard Inbound Message Mode Data Flow (For Deferred Mode of ACK Process) — Part 1" on page 127
- "HL7 V3 Standard Inbound Message Mode Data Flow (For Deferred Mode of ACK Process) — Part 2" on page 130
- "HL7 V3 Standard Inbound Message Mode Data Flow (For Deferred Mode of ACK Process) — Part 3" on page 132

HL7 V3 Standard Inbound Message Mode Data Flow (For Deferred Mode of ACK Process) — Part 1

A HL7 V3 message triggers an inbound Collaboration received from an external system or an outbound HL7 V3 Client. The Collaboration calls the HL7 V3 user Collaboration rule by executing `receive()`.

The receive method is the entry point to the HL7 V3 User Collaboration, with the following signature:

```
public void
receive(com.stc.connector.appconn.tcpip.hl7.HL7ServerApplication input,
com.stc.connectors.jms.JMS otdJMS_DATA,
xsd.hl7v3.PRPA_IN403001UV01.PRPA_IN403001UV01_  otd_PRPA_IN403001UV01_1
xsd.hl7v3.MCCI_IN000004UV01.MCCI_IN000004UV01_  otd_MCCI_IN000004UV01
xsd.hl7v3.MCCI_IN000006UV01.MCCI_IN000006UV01_  otd_MCCI_IN000006UV01
xsd.hl7v3.MCCI_IN000007UV01.MCCI_IN000007UV01_  otd_MCCI_IN000007UV01
com.stc.connectors.jms.JMS otdJMS_APPACK
otdHL7_ACK_1,com.stc.connectors.jms.JMS otdJMS_JOURNAL,
com.stc.connectors.jms.JMS otdJMS_ERROR) throws Throwable.
```

Note – The above text is wrapped to fit the page.

Once the message is received, the Collaboration determines whether the message needs to be validated. The HL7 V3 message is then validated, making sure that the message structure is correct. Various fields in the Transmission Wrapper of the message are also validated, such as Version Code, Processing Code, Processing Mode Code, and Interaction ID. If these fields do not match the configuration, a NAK is returned.

If sequence numbering is enabled the Collaboration checks to see if the messages sequence number is valid. If the sequence number is not valid, the adapter sends a NAK. The validated HL7 V3 message moves on to `processInitialHandshake()`.

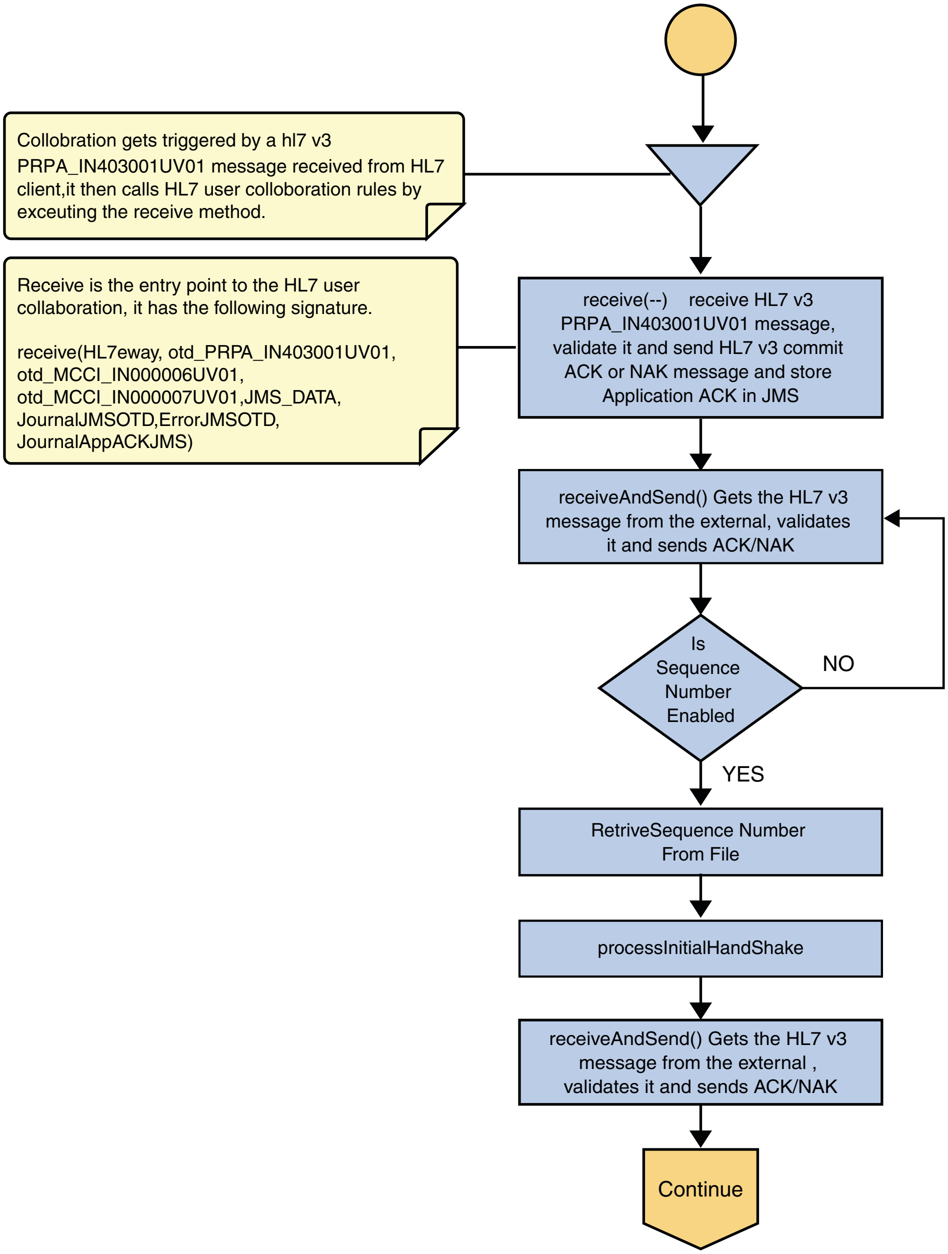

Collobration gets triggered by a hl7 v3 PRPA_IN403001UV01 message received from HL7 client,it then calls HL7 user colloboration rules by exceuting the receive method.
Receive is the entry point to the HL7 user collaboration, it has the following signature.
receive(HL7eway, otd_PRPA_IN403001UV01, otd_MCCI_IN000006UV01, otd_MCCI_IN000007UV01,JMS_DATA, JournalJMSOTD,ErrorJMSOTD, JournalAppACKJMS)
receive(--) receive HL7 v3 PRPA_IN403001UV01 message, validate it and send HL7 v3 commit ACK or NAK message and store Application ACK in JMS
receiveAndSend() Gets the HL7 v3 message from the external, validates it and sends ACK/NAK
Is Sequence Number Enabled
NO
YES
RetriveSequence Number From File
processInitialHandShake
receiveAndSend() Gets the HL7 v3 message from the external , validates it and sends ACK/NAK
Continue

FIGURE 17 Deferred Mode of ACK Process — Part 1

HL7 V3 Standard Inbound Message Mode Data Flow (For Deferred Mode of ACK Process) — Part 2

The Collaboration receives the HL7 V3 message from the external using `receiveHL7message()`. If an exception occurs due to incomplete data, and the adapter fails to read the data within the configured number of retries, the associated recourse action is taken. If the exception is due to no response, the associated recourse action is taken.

If no exception occurs, `validateHL7Message()` is called, which validates the message to determine whether to ACK or NAK the message. Other helper methods are also called to validate the HL7 V3 message.

If the HL7 V3 message passes validation, the Collaboration calls `makeCommitAck()` and `sendHL7CommitAck()` to create and send the Commit ACK (MCCI_IN000006UV01) to the external. It then calls `journalHL7AppAck()` to create Application ACK (MCCI_IN000007UV01) and store it into the JMS.

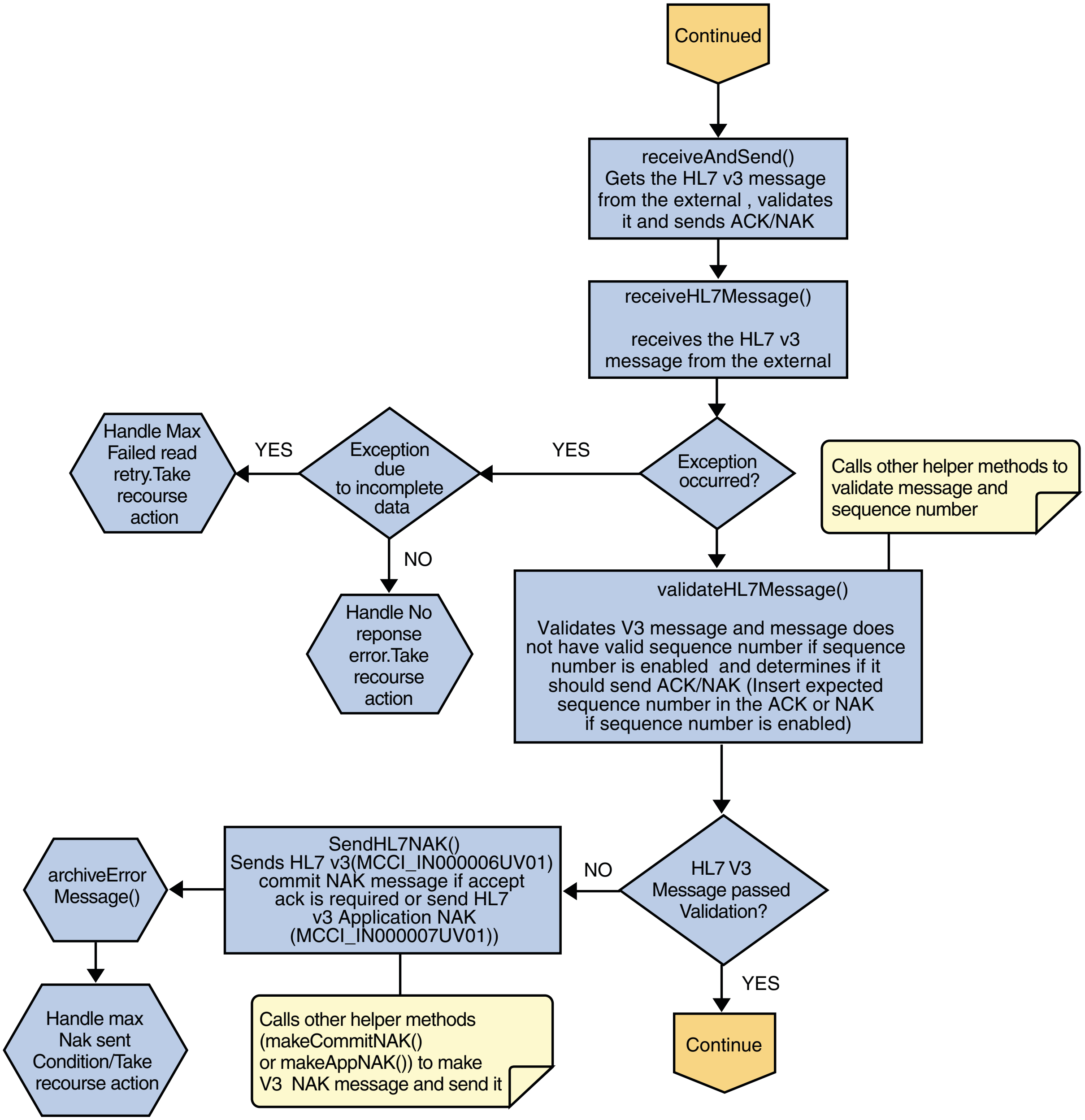


FIGURE 18 Deferred Mode of ACK Process — Part 2

HL7 V3 Standard Inbound Message Mode Data Flow (For Deferred Mode of ACK Process) — Part 3

After the Commit ACK is sent and the Application ACK is stored in JMS, the HL7 V3 message and the ACKs are journaled to the JMS queue journal destination. If the message fails to journal the associated recourse action is taken.

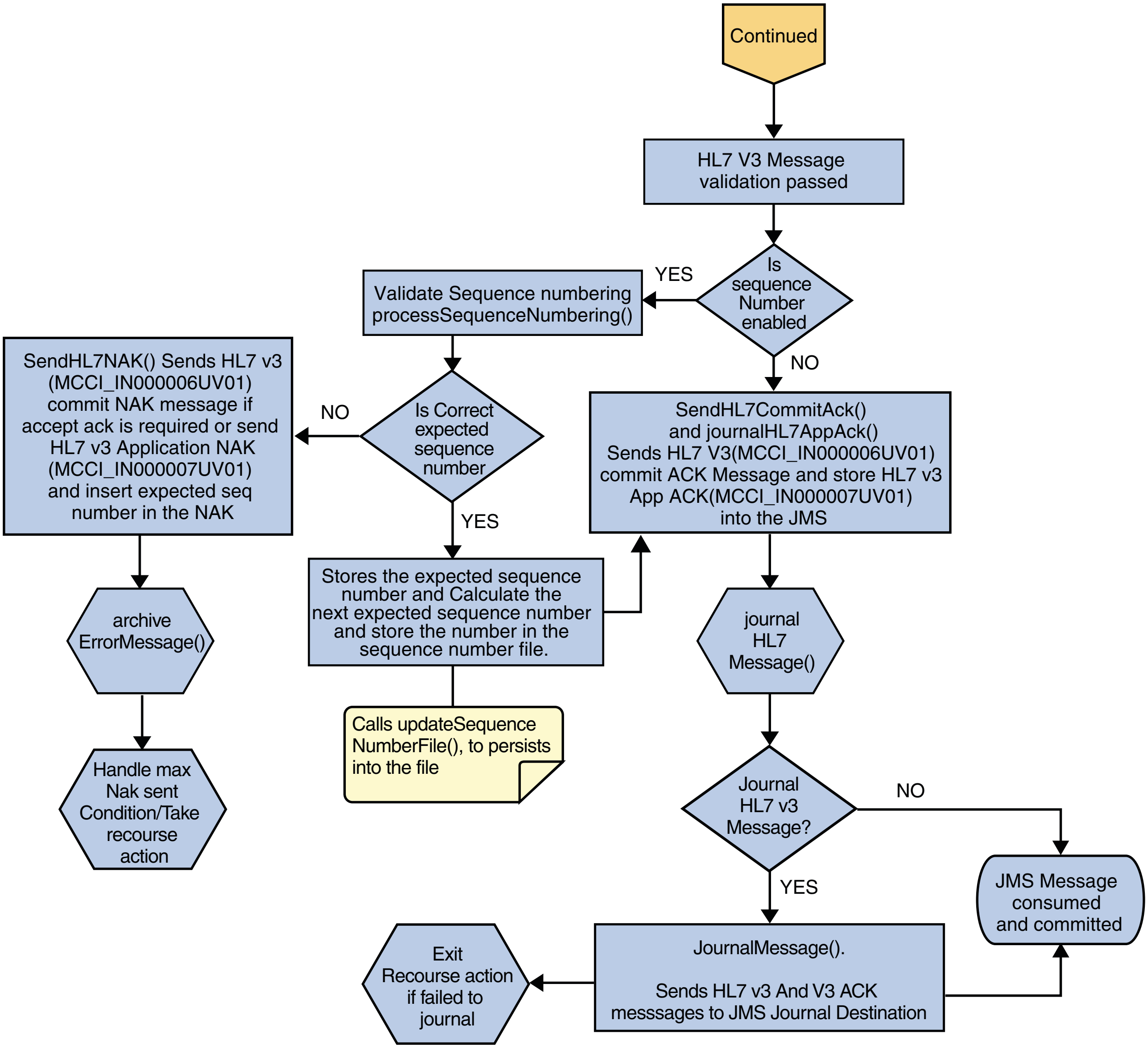

Continued
HL7 V3 Message validation passed
Is sequence Number enabled
YES
NO
Validate Sequence numbering processSequenceNumbering()
Is Correct expected sequence number
NO
YES
SendHL7NAK() Sends HL7 v3 (MCCI_IN000006UV01) commit NAK message if accept ack is required or send HL7 v3 Application NAK (MCCI_IN000007UV01) and insert expected seq number in the NAK
archive ErrorMessage()
Handle max Nak sent Condition/Take recourse action
Stores the expected sequence number and Calculate the next expected sequence number and store the number in the sequence number file.
Calls updateSequence NumberFile(), to persists into the file
SendHL7CommitAck() and journalHL7AppAck() Sends HL7 V3(MCCI_IN000006UV01) commit ACK Message and store HL7 v3 App ACK(MCCI_IN000007UV01) into the JMS
journal HL7 Message()
Journal HL7 v3 Message?
NO
YES
JMS Message consumed and committed
JournalMessage().
Sends HL7 v3 And V3 ACK messsages to JMS Journal Destination
Exit Recourse action if failed to journal

FIGURE 19 Deferred Mode of ACK Process — Part 3

Outbound HL7 V3 Collaboration Overview

The Outbound HL7 V3 Collaboration, jcdHL7V3Outbound, contains Message Libraries for the HL7 V3 Resource Adapter, JMS Data, JMS Journal, and JMS Error, as well as the HL7 V3 Patient Administration Domain Interaction Event (PRPA_IN403001UV01) and the corresponding HL7 V3 Acknowledgements. MCCI_IN000004UV01 for Immediate Mode and MCCI_IN000006UV01 for Deferred Mode. The Collaboration works with its own internal code and the Properties Configuration files. The outbound Collaboration assumes that it is reading valid HL7 V3 messages, so the data flow that feeds this Collaboration must ensure this.

The Outbound Collaboration is for both immediate and deferred mode.

The following topics describe the processing logic defined in the jcdHL7V3Outbound Collaboration:

- "HL7 V3 Standard Outbound Message Mode Data Flow (For Immediate and Deferred Mode of ACK Process) — Part 1" on page 134
- "HL7 V3 Standard Outbound Message Mode Data Flow (For Immediate and Deferred Mode of ACK Process) — Part 2" on page 136
- "HL7 V3 Standard Outbound Message Mode Data Flow (For Immediate and Deferred Mode of ACK Process) — Part 3" on page 138
- "HL7 V3 Outbound Test Collaboration" on page 139

HL7 V3 Standard Outbound Message Mode Data Flow (For Immediate and Deferred Mode of ACK Process) — Part 1

The Collaboration is triggered by a JMS HL7 V3 message. The Collaboration then calls the HL7 V3 User Collaboration Rule by executing the receive method. Receive is the entry point to the HL7 V3 User Collaboration, with the following signature:

```
receive (input, otdHL7eWay_1, otdJMS_JOURNAL, otdJMS_ERROR, otd_MCCI_IN000004UV01_1,
otd_MCCI_IN000006UV01_1, otd_PRPA_IN403001UV01_1)
```

The incoming HL7 V3 message is then validated, making sure that the message structure is correct. Various fields in the Transmission Wrapper of the message are also validated, such as Version Code, Processing Code, Processing Mode Code, and Interaction ID.

If the message does not pass validation, an error occurs and the associated recourse action is applied. If the HL7 V3 message passes validation, the message moves on to `processInitialHandshake()` to receive a sequence number (if sequences numbering is enabled only for Deferred Mode). The Collaboration takes the sequence number from the sequencing numbering file and determines the next number to use. This number is then inserted into the HL7 V3 message.

Next, the message moves on to `processMessage()`, which calls the helper method, `sendAndReceive()`. The `sendAndReceive` method sends the HL7 V3 message, waits for an HL7 V3 ACK message, and processes the ACK or NAK. The validation also checks the message structure to see if the message is unmarshaled. If a valid ACK is not received, it continues to send the HL7 V3 message up to the configured number of retries, at which time an error occurs and the associated recourse action is taken.

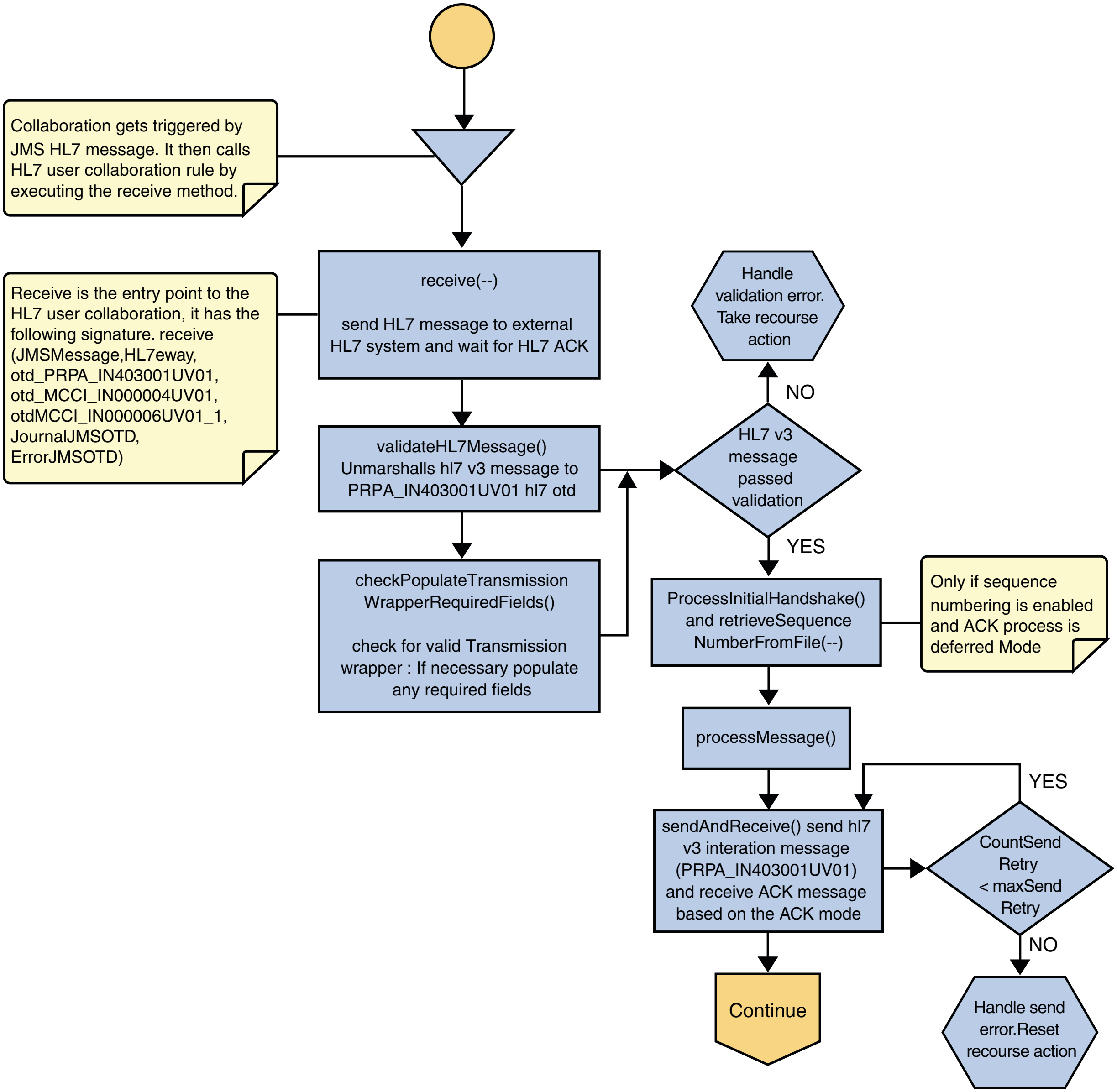

Collaboration gets triggered by JMS HL7 message. It then calls HL7 user collaboration rule by executing the receive method.
Receive is the entry point to the HL7 user collaboration, it has the following signature. receive (JMSMessage,HL7eway, otd_PRPA_IN403001UV01, otd_MCCI_IN000004UV01, otdMCCI_IN000006UV01_1, JournalJMSOTD, ErrorJMSOTD)
receive(--)
send HL7 message to external HL7 system and wait for HL7 ACK
Handle validation error. Take recourse action
NO
validateHL7Message() Unmarshalls hl7 v3 message to PRPA_IN403001UV01 hl7 otd
HL7 v3 message passed validation
YES
checkPopulateTransmission WrapperRequiredFields()
check for valid Transmission wrapper : If necessary populate any required fields
ProcessInitialHandshake() and retrieveSequence NumberFromFile(--)
Only if sequence numbering is enabled and ACK process is deferred Mode
processMessage()
YES
sendAndReceive() send hl7 v3 interation message (PRPA_IN403001UV01) and receive ACK message based on the ACK mode
CountSend Retry < maxSend Retry
NO
Continue
Handle send error.Reset recourse action

FIGURE 20 Immediate and Deferred Mode of ACK Process — Part 1

HL7 V3 Standard Outbound Message Mode Data Flow (For Immediate and Deferred Mode of ACK Process) — Part 2

In `processMessage()`, the message moves on to `insertSequenceNumber()`. If sequence numbering is enabled and is in Deferred Mode, the `insertSequenceNumber` method inserts the sequence number and call `sendHL7Message()`. The `sendHL7Message` method sends the HL7 V3 message to the external using the HL7 V3 adapter Message Library

The Collaboration receives the HL7 V3 ACK or NAK from the external using `receiveHL7AckNak()`. If an exception occurs due to incomplete data, and the adapter fails to read the data within the configured number of retries, the associated recourse action is taken. If the exception is due to no response, the associated recourse action is taken. If no exception occurs, the ACK or NAK message moves on to `isAckMessage()`, which validates the message to determine whether the message is an ACK or a NAK.

Next, the `validateAckNak` method unmarshals the message to the ACK Message Library in MCCI_IN000004UV01 Message Library for Immediate Mode and MCCI_IN000006UV01 Message Library for Deferred Mode.

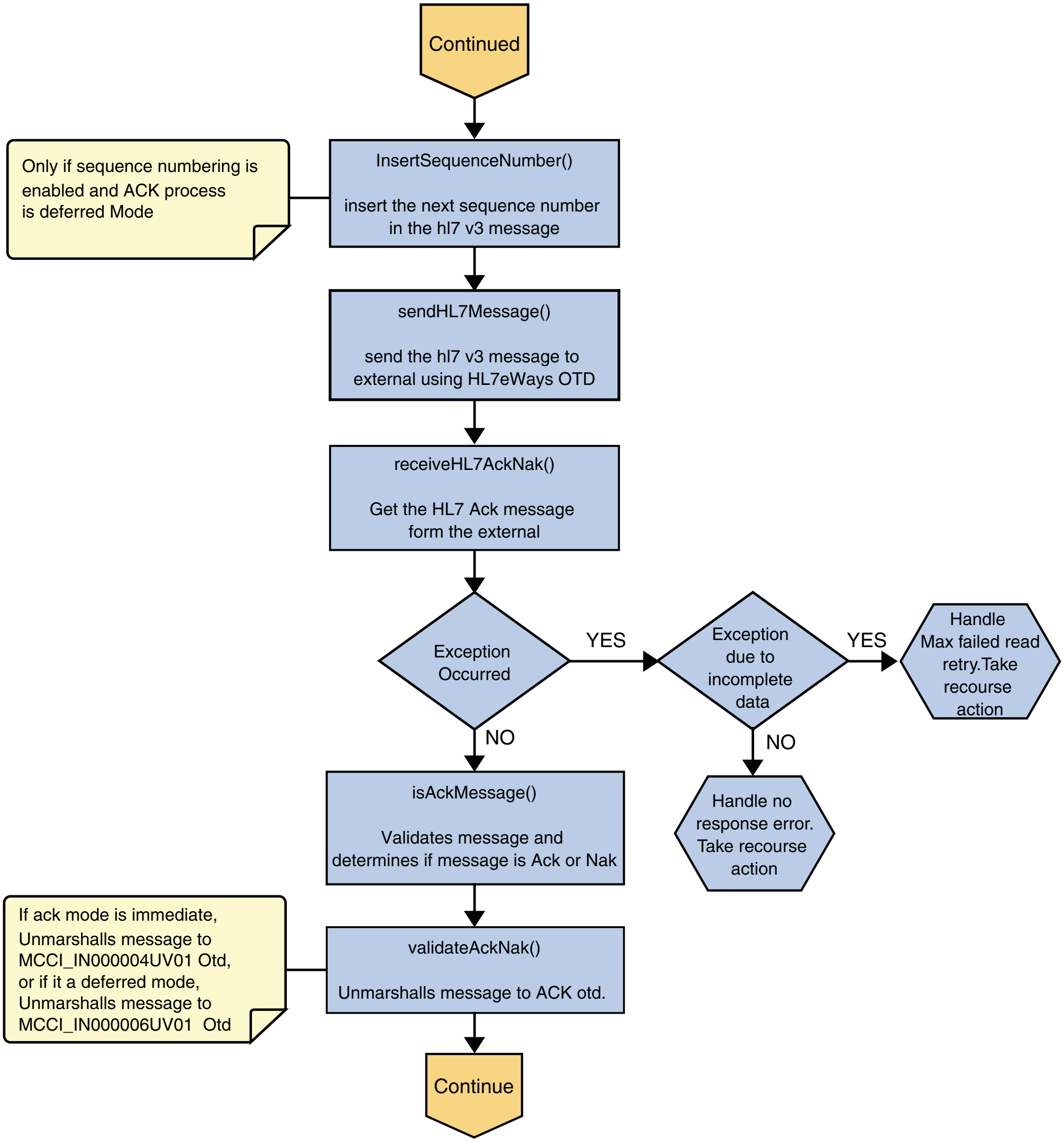

Continued
Only if sequence numbering is enabled and ACK process is deferred Mode
InsertSequenceNumber()
insert the next sequence number in the hl7 v3 message
sendHL7Message()
send the hl7 v3 message to external using HL7eWays OTD
receiveHL7AckNak()
Get the HL7 Ack message form the external
Exception Occurred
YES
Exception due to incomplete data
YES
Handle Max failed read retry.Take recourse action
NO
NO
Handle no response error. Take recourse action
isAckMessage()
Validates message and determines if message is Ack or Nak
If ack mode is immediate, Unmarshalls message to MCCI_IN000004UV01 Otd, or if it a deferred mode, Unmarshalls message to MCCI_IN000006UV01 Otd
validateAckNak()
Unmarshalls message to ACK otd.
Continue

FIGURE 21 Immediate and Deferred Mode of ACK Process — Part 2

HL7 V3 Standard Outbound Message Mode Data Flow (For Immediate and Deferred Mode of ACK Process) — Part 3

If the message does not pass validation it is handled as a NAK, the associated recourse action is taken, and the message is archived in the error queue.

If the message is a NAK, the associated recourse action is taken, and the message is archived in the error queue, along with the NAK message, as a JMS property

If the message is an ACK and passes validation, the message is sent on to the `journalMessage` method.

If the ACK message validates, the HL7 V3 message and ACK message are sent to the JMS journal Destination. If the message fails to journal, the associated recourse action is taken.

If sequencing numbering is enabled for Deferred Mode, the `processAckNakSequenceNumbering` method calculates the next sequence number and stores the number in the sequence number file, calling `updateSequenceNumberFile` to persist the next sequence number.

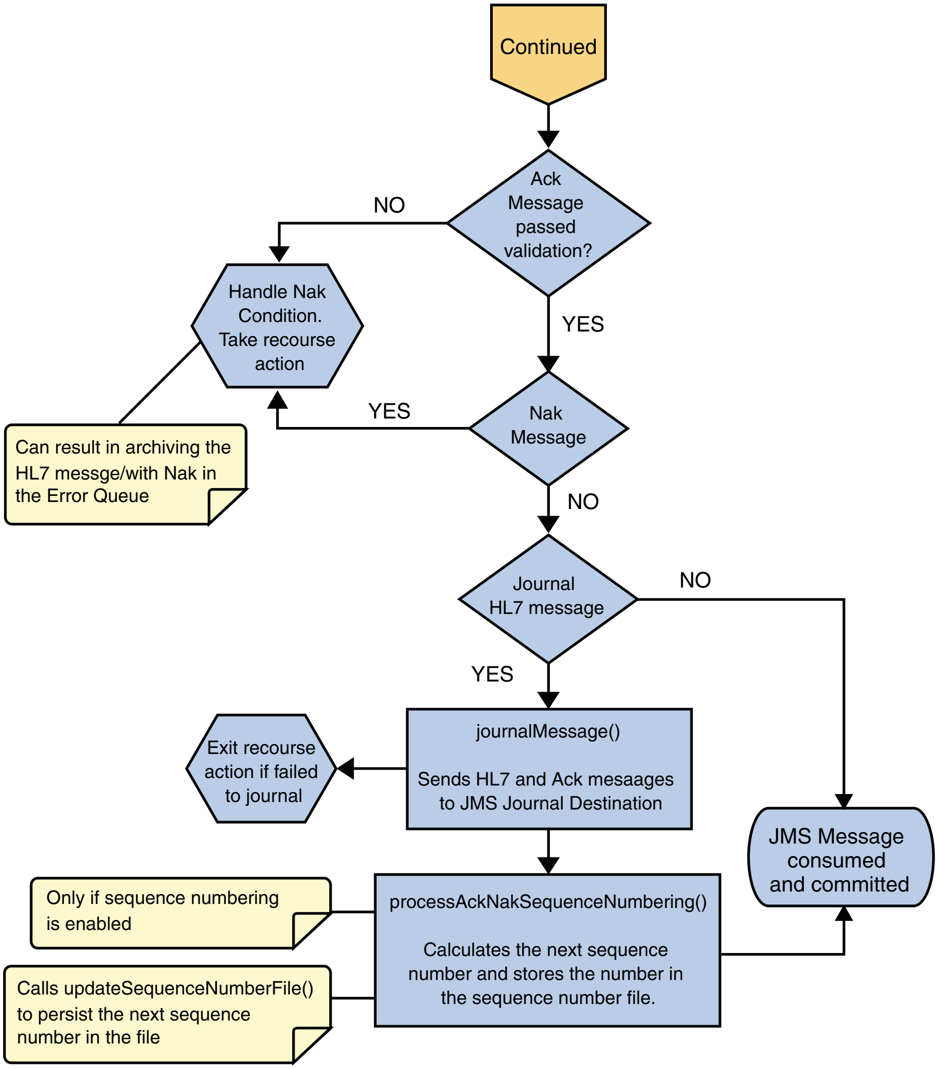

Continued
Ack Message passed validation?
NO
Handle Nak Condition. Take recourse action
YES
Nak Message
YES
Can result in archiving the HL7 messge/with Nak in the Error Queue
NO
Journal HL7 message
NO
YES
journalMessage()
Sends HL7 and Ack mesaages to JMS Journal Destination
Exit recourse action if failed to journal
JMS Message consumed and committed
processAckNakSequenceNumbering()
Calculates the next sequence number and stores the number in the sequence number file.
Only if sequence numbering is enabled
Calls updateSequenceNumberFile() to persist the next sequence number in the file

FIGURE 22 Immediate and Deferred Mode of ACK Process — Part 3

HL7 V3 Outbound Test Collaboration

For information, see "HL7 V2 Outbound Test Collaboration" on page 119.

MLLP V2 and the Sample Projects

Minimal Lower Layer Protocol Release 2 (MLLP V2.0) is a message transport protocol, and is required for all message transport protocols used to transport HL7 Version 3 content. This protocol wraps the HL7 message to indicate the beginning and end of each message.

The following topics provide instruction on how to set up the MLLP database, and describe MLLP V2.0 and the MLLP V2.0 sample Project provided with the TCP/IP HL7 Adapter:

- "Creating and Configuring the MLLP V2.0 Database" on page 140
- "MLLP V2 Content Exchange Model" on page 144
- "Standard Inbound HL7 V2 Collaboration Overview over MLLPV2" on page 146

Creating and Configuring the MLLP V2.0 Database

Using the HL7 Adapter with MLLP V2.0 transport scenarios requires an MLLP database. This topic describes the steps to connect to Sun's Java Database (based on the Apache Derby database) and run the MLLP V2.0 database scripts.

The TCP/IP HL7 Adapter supports the following database platforms for the MLLP database:

- DB2
- Java DB (Derby)
- Oracle
- Microsoft SQL Server
- Sybase

Perform the following steps in the order given to set up the MLLP database:

- "To Create and Connect to the Database" on page 140
- "To Run the MLLP V2.0 Database Scripts" on page 142
- "To Create the Connection Pool and JDBC Resource" on page 144
- "Associating the MLLP Database With the Adapter" on page 144

▼ To Create and Connect to the Database

1. **Create a database instance using your database platform tools. If you are using a Java DB database, perform the following steps to create the database:**

   a. **On the NetBeans Services window, expand Databases.**

   b. **Right-click Java DB, and then click Create Database.**

      The Create Java DB Database dialog box appears.

   c. **Enter a name for the database, along with a login user name and password.**

d. **Click OK.**

The database appears in the list of databases on the Services window.

2 **Make sure your database instance is running.**

3 **For Oracle, DB2, SQL Server, and Sybase databases, do the following:**

a. **Copy the database driver file to** *JavaCAPS_Home*`/appserver/lib`**.**

b. **Under Databases on the NetBeans Services window, right-click Drivers and then select New Driver.**

The New JDBC Driver dialog box appears.

c. **Click Add, and then brows to and select the database driver file.**

d. **Select or look for the driver class to use.**

e. **Accept the default name or enter a name for the driver.**

f. **Click OK.**

g. **In the Drivers list on the Services window, right-click the driver you just added and click Connect Using.**

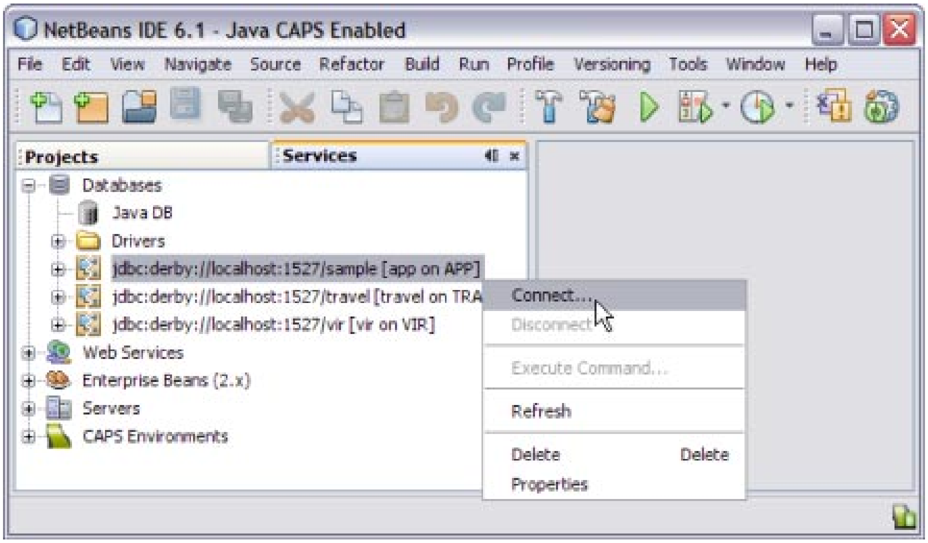
The New Database Connection dialog box appears.

h. **Enter the connection information for the database instance you created for MLLP and then click OK.**

The new database connection appears in the Databases list.

4 **On the NetBeans Services window, right-click the MLLP database, and click Connect.**

NetBeans connects to the database and the icon changes for that database. The following figure illustrates a Java DB connection.

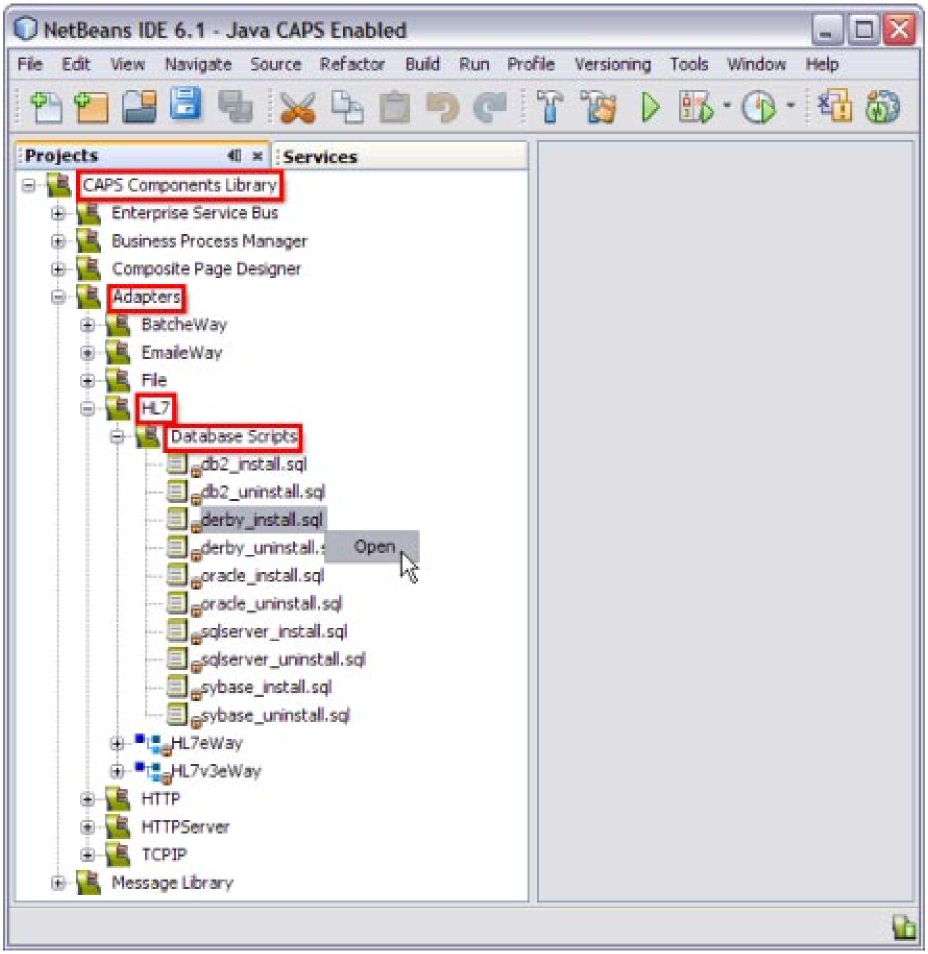

NetBeans IDE 6.1 - Java CAPS Enabled
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Projects
Services
Databases
Java DB
Drivers
jdbc:derby://localhost:1527/sample [app on APP]
jdbc:derby://localhost:1527/travel [travel on TRA
jdbc:derby://localhost:1527/vir [vir on VIR]
Web Services
Enterprise Beans (2.x)
Servers
CAPS Environments
Connect...
Disconnect
Execute Command...
Refresh
Delete Delete
Properties

Note – A broken icon represents that the database is not connected.

▼ To Run the MLLP V2.0 Database Scripts

You can either run the MLLP database scripts from NetBeans or from a SQL editor. The following steps take you through running the scripts from NetBeans.

1. **On the NetBeans IDE, click the Projects tab.**
2. **Expand CAPS Components Library > Adapters > HL7 > Database Scripts.**
3. **Right-click** *db_platform*`_install.sql`**; for example,** `derby_install.sql`**.**
4. **Click Open.**



This opens the SQL Editor window along with the SQL Statement.

```
CREATE TABLE HL7MESSAGELOG (MESSAGEID VARCHAR(250), APPLICATIONID VARCHAR(250) ,
REQUESTMESSAGE CLOB, RESPONSEMESSAGE CLOB, STATUS SMALLINT, CREATEDTIME TIMESTAMP,
LASTUPDATEDTIME TIMESTAMP)
```

5 **In the SQL Editor toolbar, click in the Connection field and select the database connection from the drop-down list.**

6 **In the toolbar, click Run SQL.**

A message appears in the Output panel indicating whether the script was run successfully.

7 **To view the new tables, return to the Services window, right-click the database connection, and then click Refresh. Expand the database connection and then Tables to view the database.**

▼ To Create the Connection Pool and JDBC Resource

In order to connect to the database, you need to define a JDBC connection pool and JDBC resource using the GlassFish Admin Console or the command-line interface.

1 **Start the GlassFish server.**

2 **To create the connection pool and JDBC resource, follow the instructions under "Creating a JDBC Connection Pool" in *Sun GlassFish Enterprise Server 2.1 Administration Guide*.**

Use the information from the database connection you created earlier in "To Create and Connect to the Database" on page 140.

▼ Associating the MLLP Database With the Adapter

Once you create the MLLP database, you can connect to it from the inbound TCP/IP HL7 Adapter. You do this in the HL7 External Systems properties located in the CAPS Environment you create for the HL7 Project. This procedure assumes you have already created the Environment and that it includes an HL7 External System.

1 **On the NetBeans Services window, expand CAPS Environments, and then expand the HL7 Environment you created for your Project.**

2 **Right-click the HL7 External System, and select Properties.**

The Properties Editor appears.

3 **In the left pane, expand HL7 Inbound eWay.**

4 **Select Database Settings.**

5 **In the `JNDI Name of the JDBC Datasource` property, enter the JNDI name of the JDBC Resource you created under "To Create the Connection Pool and JDBC Resource" on page 144.**

6 **Click OK.**

MLLP V2 Content Exchange Model

MLLP Release 2 is a reliable message transport protocol. It guarantees in order delivery and at least once delivery of HL7 content. HL7 content is framed in a block and sent to the destination system. The destination system acknowledges the receipt of the message by returning a commit acknowledgement message. The MLLP V2 acknowledgement protocol is synchronous: the source system will not send new HL7 content until an acknowledgement for the previous HL7 content has been received.

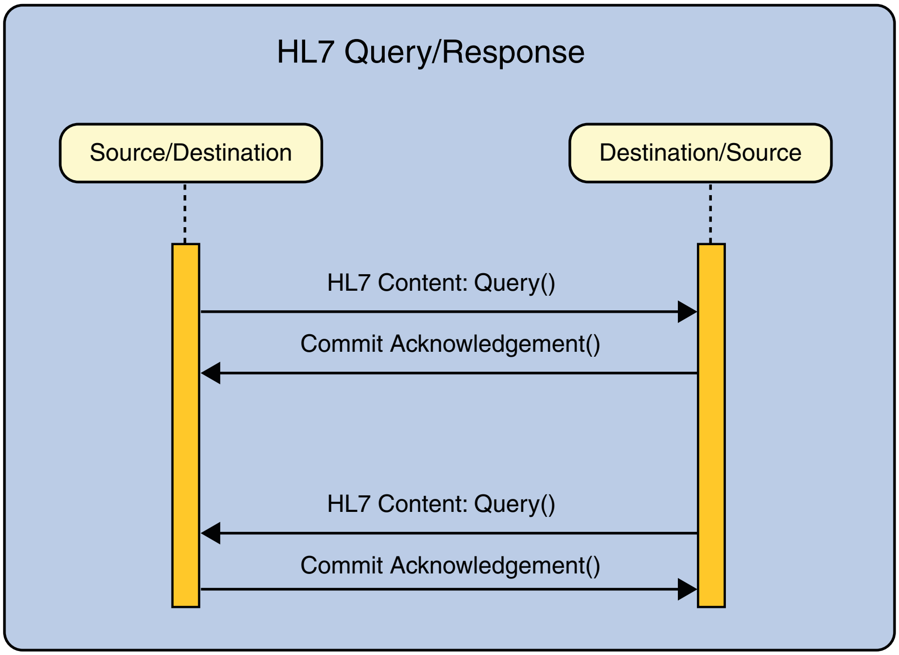


FIGURE 23 Interaction Diagram

If the HL7 content (a query in the example below) triggers the sending of HL7 content (a response) by the destination system, then this HL7 content is framed in a block and sent. MLLP has no knowledge of the HL7 content, nor does it base any part of its behavior on HL7 content. With the TCP/IP HL7 Adapter, a database is used to persist the message.

In inbound mode, once the message is received by the HL7 Adapter, it is persisted into the database and a commit acknowledgement is sent to the sender before the HL7 message is passed to the Collaboration level. If the persistence fails, a negative acknowledgement is sent to the sender and the Collaboration receives null as its message. Inside the Collaboration, the message received is null and the Collaboration returns without proceeding.

If the message received is a duplicate message and if an ACK/NAK is already available in the persistence database, the same is retrieved from the database and sent to the sender. The Collaboration receives null as the received message.

In outbound mode, the message is sent to the sender and waits for the commit acknowledgement or negative acknowledgement from the receiver. If the adapter receives a negative acknowledgement, the message is sent again until the configured maximum number of retries is made.

Standard Inbound HL7 V2 Collaboration Overview over MLLPV2

In the MLLP V2 sample Projects, the inbound HL7 V2 Collaboration, jcdHL7Inbound, contains Message Libraries (previously called OTDs) for the HL7 Resource Adapter, JMS Data, HL7 ACK, JMS Journal, and JMS Error, as well as the Generic HL7 Event. The Collaboration works with its own internal code and the properties configuration files.

The following topics describe the business logic defined in the jcdHL7Inbound Collaboration for MLLP V2:

- "HL7 V2 Standard Inbound Message Mode Data Flow over MLLPV2 — Part 1" on page 146
- "HL7 V2 Standard Inbound Message Mode Data Flow over MLLPV2 — Part 2" on page 148
- "HL7 V2 Standard Inbound Message Mode Data Flow over MLLPV2 — Part 3" on page 149

HL7 V2 Standard Inbound Message Mode Data Flow over MLLPV2 — Part 1

Once the message is received, the Collaboration determines whether the message needs to be validated. The HL7 V2 message is then validated making sure that the message structure is correct. Various fields in the MSH segment of the message are also validated, such as Version ID and Sending Facility. If these fields do not match the configuration, a NAK is returned.

If sequence numbering is enabled the Collaboration checks to see if the messages sequence number is valid. If the sequence number is not valid, the adapter sends a NAK.

The validated HL7 V2 message moves on to `processInitialHandshake()` and the sequence numbers are synchronized. The sequence number within the message is checked against the expected sequence number. If the numbers match, the Collaboration sends an ACK, if not it sends a NAK. The ACK or NAK includes information from various fields of the incoming MSA segment. The ACKs level of acknowledgement is set to A (acknowledgement is sent when the message is successfully received) or C (acknowledgement is sent after the message is successfully processed).

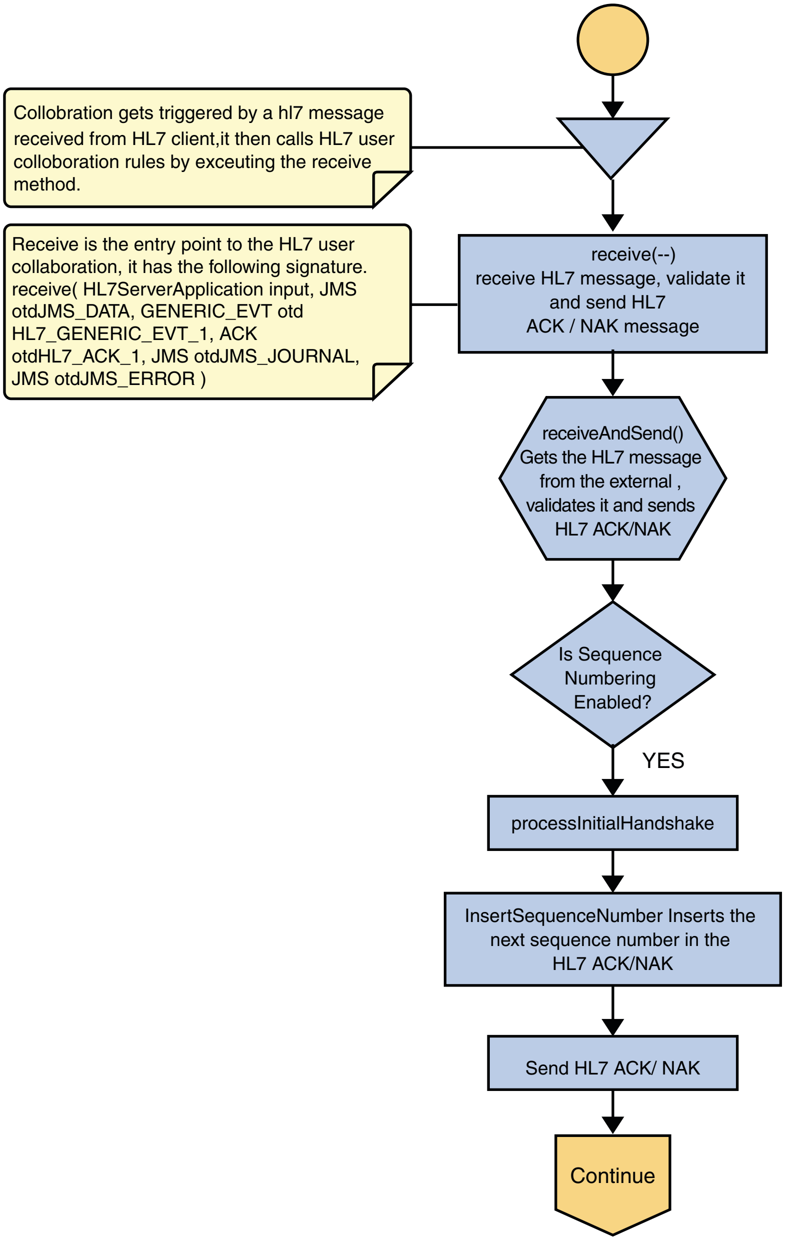

Collobration gets triggered by a hl7 message received from HL7 client,it then calls HL7 user colloboration rules by exceuting the receive method.
Receive is the entry point to the HL7 user collaboration, it has the following signature. receive(HL7ServerApplication input, JMS otdJMS_DATA, GENERIC_EVT otd HL7_GENERIC_EVT_1, ACK otdHL7_ACK_1, JMS otdJMS_JOURNAL, JMS otdJMS_ERROR)
receive(--) receive HL7 message, validate it and send HL7 ACK / NAK message
receiveAndSend() Gets the HL7 message from the external , validates it and sends HL7 ACK/NAK
Is Sequence Numbering Enabled?
YES
processInitialHandshake
InsertSequenceNumber Inserts the next sequence number in the HL7 ACK/NAK
Send HL7 ACK/ NAK
Continue

FIGURE 24 HL7 V2 Standard Inbound Message Mode Data Flow over MLLPV2 — Part 1

HL7 V2 Standard Inbound Message Mode Data Flow over MLLPV2 — Part 2

The Collaboration receives the HL7 V2 message from the external using `receiveHL7message()`. If an exception occurs due to incomplete data, and the adapter fails to read the data within the configured number of retries, the associated recourse action is taken. If the exception is due to no response, the associated recourse action is taken.

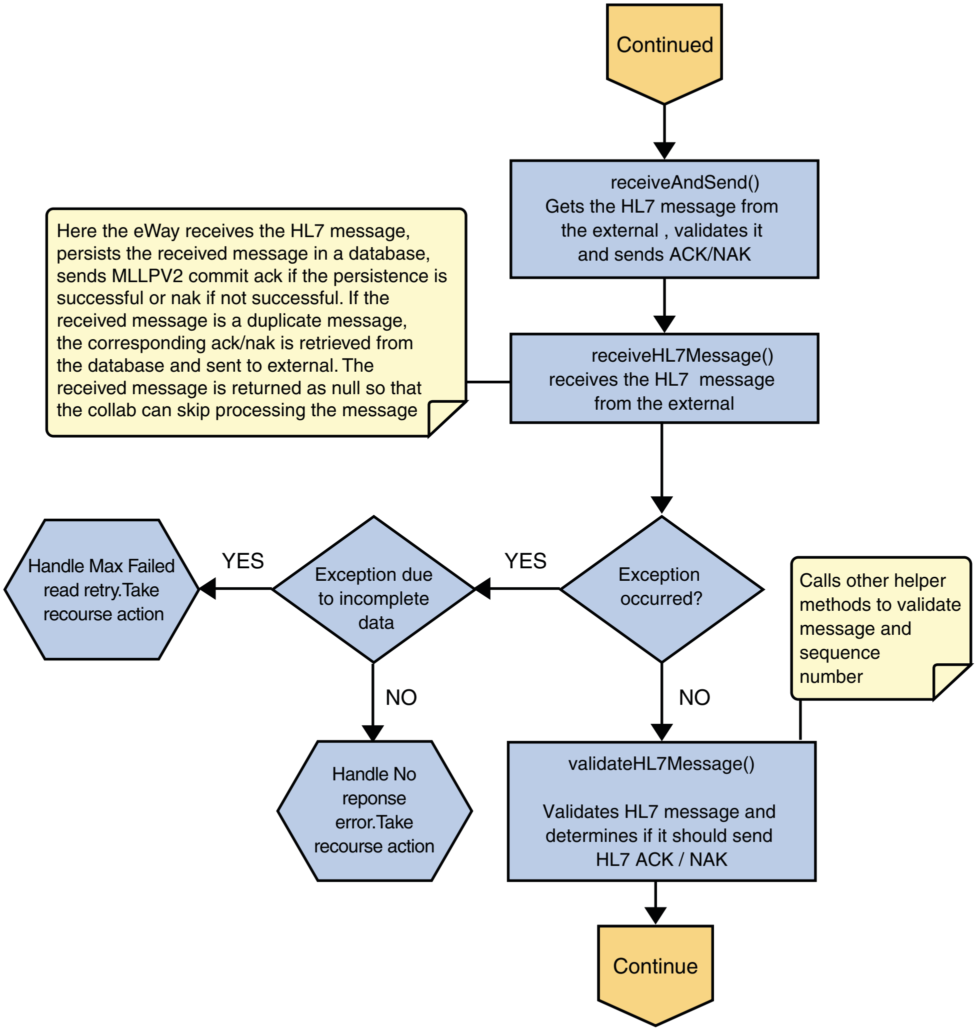

Continued
receiveAndSend()
Gets the HL7 message from the external , validates it and sends ACK/NAK
Here the eWay receives the HL7 message, persists the received message in a database, sends MLLPV2 commit ack if the persistence is successful or nak if not successful. If the received message is a duplicate message, the corresponding ack/nak is retrieved from the database and sent to external. The received message is returned as null so that the collab can skip processing the message
receiveHL7Message()
receives the HL7 message from the external
Handle Max Failed read retry.Take recourse action
YES
Exception due to incomplete data
YES
Exception occurred?
Calls other helper methods to validate message and sequence number
NO
NO
Handle No reponse error.Take recourse action
validateHL7Message()
Validates HL7 message and determines if it should send HL7 ACK / NAK
Continue

FIGURE 25 HL7 V2 Standard Inbound Message Mode Data Flow over MLLPV2 — Part 2

HL7 V2 Standard Inbound Message Mode Data Flow over MLLPV2 — Part 3

If no exception occurs, `validateHL7Message()` is called, which validates the message to determine whether to ACK or NAK the message. Other helper methods are also called to validate the HL7 message.

If the HL7 message does not pass validation, the Collaboration calls `makeNak()` and `sendHL7Nak()` to create and send the NAK to the external system. The HL7 message, with the NAK, is archived to the error queue. If the number of consecutive NAKs sent surpasses the maximum number of retries, the associated recourse action is taken.

If the HL7 message passes validation, the Collaboration calls `makeAck()` and `sendHL7Ack()` to create and send the ACK to the external system. The HL7 ACK/NAK messages are stored in the persistent database. This ACK/NAK is stored against the inbound HL7 message received.

After the ACK is sent, the HL7 message and the ACK are journaled to the JMS queue journal destination. If the message fails to journal the associated recourse action is taken.

If sequencing numbering is enabled, the `processAckNakSequenceNumbering` method calculates the next sequence number and stores the number in the sequence number file by calling the `updateSequenceNumberFile` method to persist the next sequence number.

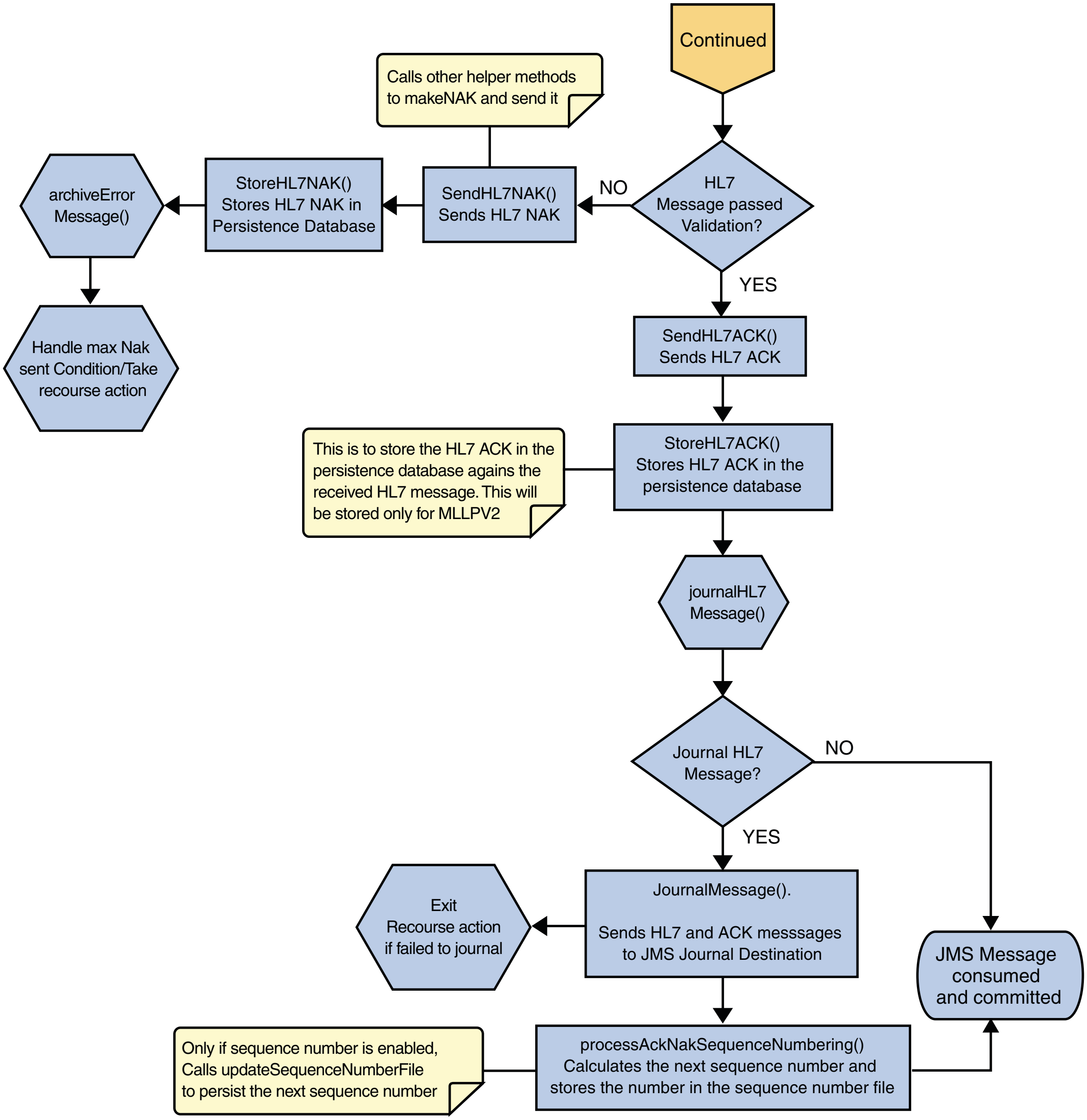

Continued
Calls other helper methods to makeNAK and send it
HL7 Message passed Validation?
NO
SendHL7NAK() Sends HL7 NAK
StoreHL7NAK() Stores HL7 NAK in Persistence Database
archiveError Message()
Handle max Nak sent Condition/Take recourse action
YES
SendHL7ACK() Sends HL7 ACK
This is to store the HL7 ACK in the persistence database agains the received HL7 message. This will be stored only for MLLPV2
StoreHL7ACK() Stores HL7 ACK in the persistence database
journalHL7 Message()
Journal HL7 Message?
NO
YES
JournalMessage().
Sends HL7 and ACK messsages to JMS Journal Destination
Exit Recourse action if failed to journal
JMS Message consumed and committed
Only if sequence number is enabled, Calls updateSequenceNumberFile to persist the next sequence number
processAckNakSequenceNumbering() Calculates the next sequence number and stores the number in the sequence number file

FIGURE 26 HL7 V2 Standard Inbound Message Mode Data Flow over MLLPV2 — Part 3