SeeBeyond™ eBusiness Integration Suite

TCP/IP HL7 e*Way Intelligent Adapter User's Guide

Release 4.5.3

SᴇᴇBᴇʏᴏɴᴅ™

Contents

Contents

Chapter 4

Implementation 36

Introduction

1.1 Overview

The TCP/IP HL7 e*Way Intelligent Adapter enables the e*Gate Integrator system to exchange data with an external TCP/IP backend, using the HL7 data protocol. This document describes how to install and configure the TCP/IP HL7 e*Way.

Note: *This e*Way is based on the TCP/IP HL7 Super Communications Client, the existence of which predates the use of the name "e*Gate." The term "DataGate" has been retained in some parameter names to ensure backwards-compatibility, even though the functionality now pertains to e*Gate.*

1.1.1. Intended Reader

The reader of this guide is presumed to be a developer or system administrator with responsibility for maintaining the e*Gate system; to have expert-level knowledge of Windows NT and UNIX operations and administration; to be familiar with TCP/IP and HL7 protocols; and to be thoroughly familiar with Windows-style GUI operations.

1.1.2. Components

The TCP/IP HL7 e*Way encompasses the following:

- stcdgwtcpiphl7.exe, the executable component
- Configuration files, which the e*Way Editor uses to define configuration parameters

A complete list of installed files appears in **Table 1 on page 11**.

1.1.3. Features

The TCP/IP HL7 e*Way:

- Handles both HL7 HLLP and MLLP protocols
- Has a wide variety of Recourse action configurations
- Has full support of the HL7 sequence numbering protocol
- Has full support for HL7 acknowledgment generation

and may be configured:

- For Inbound or Outbound communication

- For acknowledgment to arrive from another port or e*Gate IQ

- As a TCP/IP Server or Client

- As a remote client

- To Support multiple Outbound Collaborations

1.2 Supported Operating Systems

The TCP/IP HL7 e*Way is available on the following operating systems:

- Windows 2000, Windows 2000 SP1, Windows 2000 SP2, and Windows 2000 SP3

- Windows NT 4.0 SP6a

- Solaris 2.6, 7, and 8

- AIX 4.3.3

- HP-UX 11.0 and HP-UX 11i

- Compaq Tru64 V4.0F, V5.0A, and V5.1A

- Linux RedHat 6.2

1.3 System Requirements

To use the TCP/IP HL7 e*Way, you need the following:

- An e*Gate Participating Host, version 4.5.1 or later.

- A TCP/IP network connection.

- Additional disk space for e*Way executable, configuration, library, and script files. The disk space is required on both the Participating and the Registry Host. Additional disk space is required to process and queue the data that this e*Way processes; the amount necessary varies based on the type and size of the data being processed, and any external applications performing the processing.

Installation

2.1 Windows NT and Windows 2000

2.1.1. Pre-installation

1 Exit all Windows programs before running the setup program, including any anti-virus applications.

2 You must have Administrator privileges to install this e*Way.

2.1.2. Installation Procedure

To install the TCP/IP HL7 e*Way on a Windows NT or Windows 2000 system

1 Log in as Administrator on the workstation on which you want to install the e*Way.

2 Insert the e*Way installation CD-ROM into the CD-ROM drive.

3 If the CD-ROM drive's "Autorun" feature is enabled, the setup application should launch automatically; skip ahead to step 4. Otherwise, use the Windows NT Explorer or the Control Panel's **Add/Remove Applications** feature to launch the file **setup.exe** on the CD-ROM drive.

4 The InstallShield setup application will launch. Follow the on-screen instructions to install the e*Way.

Note: *Be sure to install the e*Way files in the suggested "client" installation directory.*
The installation utility detects and suggests the appropriate installation directory.
Unless you are directed to do so by SeeBeyond support personnel, do not change the suggested "installation directory" setting.

5 After the installation is complete, exit the install utility and launch the Enterprise Manager.

6 In the Component editor, create a new e*Way.

7 Display the new e*Way's properties.

8 On the General tab, under **Executable File**, click **Find**.

9 Select the file **stcdgwtcpiphl7.exe**.

10 Click **OK** to close the properties dialog box, or continue to configure the e*Way. Configuration parameters are discussed in **Chapter 3**.

*Note: Once you have installed and configured this e*Way, you must incorporate it into a schema by defining and associating the appropriate Collaborations, Collaboration Rules, IQs, and Event Types before this e*Way can perform its intended functions. For more information about any of these procedures, please see the online Help system.*

*For more information about configuring e*Ways or how to use the e*Way Editors, see the **e*Gate Integrator User's Guide**.*

2.2 UNIX

2.2.1. Pre-installation

1 Do not install the e*Way as root. Log in under the user name of the user who will own the e*Way files. Be sure that this user has sufficient privilege to create files in the e*Gate directory tree.

2.2.2. Installation Procedure

To install the TCP/IP HL7 e*Way on a UNIX system

1 Log in on the workstation containing the CD-ROM drive, and insert the CD-ROM into the drive.

2 If necessary, mount the CD-ROM drive.

3 At the shell prompt, type

cd /cdrom

4 Start the installation script by typing:

setup.sh

5 A menu of options will appear. Select the "install e*Way" option. Then, follow any additional on-screen directions.

*Note: Be sure to install the e*Way files in the suggested "client" installation directory. The installation utility detects and suggests the appropriate installation directory. **Unless you are directed to do so by SeeBeyond support personnel, do not change the suggested "installation directory" setting.***

6 After installation is complete, exit the installation utility and launch the Enterprise Manager.

7 In the Component editor, create a new e*Way.

8 Display the new e*Way's properties.

9   On the General tab, under **Executable File**, click **Find**.

10   Select the file **stcdgwtcpiphl7.exe**.

11   Click **OK** to close the properties dialog box, or continue to configure the e*Way. Configuration parameters are discussed in **Chapter 3**.

Note: *Once you have installed and configured this e*Way, you must incorporate it into a schema by defining and associating the appropriate Collaborations, Collaboration Rules, IQs, and Event Types before this e*Way can perform its intended functions. For more information about any of these procedures, please see the online Help system.*

*For more information about configuring e*Ways or how to use the e*Way Editors, see the* ***e*Gate Integrator User's Guide.***

2.3   Files/Directories Created by the Installation

The TCP/IP HL7 e*Way installation process will install the following files within the e*Gate "client" directory tree:

Table 1   Files created by the installation

e*Gate "Client" Directory	Files
\bin\	stcdgwtcpiphl7.exe
\configs\stcdgwtcpiphl7\	stcdgwtcpiphl7.def

Configuration

3.1 Configurable Properties

3.1.1. Connection Type

You can configure the TCP/IP HL7 e*Way as either a TCP/IP client or a TCP/IP server. Some additional configuration parameters are dependent upon these settings.

Connected as a TCP/IP Client

As a TCP/IP client, the e*Way obtains the target hostname and TCP/IP port number from its configuration file, and attempts to establish connection with the specified host. If the connection attempt fails, the e*Way continues to try to connect until the connection is made. The e*Way attempts to connect immediately at start-up; however, it waits for a configurable number of seconds between connection resets before attempting further connections.

Connected as a TCP/IP Server

As a TCP/IP server, the e*Way obtains the TCP/IP port number from its configuration file—the hostname specification in the configuration file is not used. The e*Way becomes a TCP/IP server on the local machine, at the pre-configured TCP/IP port number.

The e*Way attempts to become a server immediately at start-up; however, it waits for a configurable number of seconds between connection resets before making any further attempt to become a server. If the initial attempt to become a server fails, it continues to try until it is successful.

Once it becomes a server, the e*Way listens on its TCP/IP port for any incoming connections. When the external system tries to connect, the e*Way accepts the connection and the link is established.

The e*Way continues to listen for any other incoming connections and tracks the time of receipt of the last transaction from the external system. If a second connection is attempted, the e*Way checks the elapsed time to determine whether or not the last transaction received from the external was larger than its configurable **Time to Wait for a Response** parameter. If so, it abandons the old connection and accepts the new one. If the time elapsed since the last transaction from the external is still within the response timeout period, however, the new incoming connection is dropped.

3.1.2. Direction

You can configure the TCP/IP HL7 e*Way as Inbound to e*Gate (receiving data from the external system) or as Outbound from e*Gate (sending data to the external system). Some additional configuration parameters are dependent upon these settings.

Inbound to e*Gate

HL7 data received from the external system is processed and sent to e*Gate. The e*Way waits indefinitely for an e*Gate Acknowledgement from e*Gate. As determined by the e*Way's configuration in the **Acknowledgment Generation** section, one of two things then happen:

- e*Gate sends an e*Gate Acknowledgement to the e*Way. The e*Way then generates an HL7 Application Acknowledgment and sends it to the external system.

- e*Gate sends an HL7 Application Acknowledgment to the e*Way, which formats and sends it to the external system.

Outbound from e*Gate

HL7 Data coming from e*Gate is verified to ensure it contains HL7 Data only. It then is enveloped into its configured format and sent to the external system.

The e*Way waits for a configurable number of seconds (response timeout) for an incoming HL7 Application Acknowledgment from the external system, as configured in the **Time to Wait for a Response** and **Max No Response** parameters.

If the incoming response is an HL7 Application Acknowledgment, the e*Way may do either of the following, as dictated by its configuration in the **Acknowledgment Generation** section:

- Forward the external system's HL7 response to e*Gate

- Send an e*Gate Acknowledgement to e*Gate

Acknowledgements

Usually, an HL7 Application Acknowledgment is generated by the e*Way and sent directly to the external application. However, sometimes it is necessary to create the HL7 Application Acknowledgment within e*Gate. It also is possible that the HL7 Application Acknowledgment may be sent to e*Gate from yet another source and e*Gate simply forwards it to the Inbound e*Way.

Do not confuse an HL7 Application Acknowledgment being sent by e*Gate, with an e*Gate Acknowledgment. The HL7 Application Acknowledgment looks like Data to both e*Gate and the e*Way. The e*Gate Acknowledgment is a lower level communication acknowledgment used **only** between e*Gate and its connected e*Ways.

3.1.3. Envelope Type

The e*Way uses the envelope to know when to start and stop reading the data. This section describes the two envelope types used in the HL7 protocol.

The TCP/IP HL7 e*Way supports the following two envelope types:

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

Note: *According to the HL7 Standard, the End of Block (EOB) Character for the HLLP and MLLP Envelopes must be set to a Carriage Return (CR). Although the End of Block Character is a configurable parameter, it is strongly recommended that this parameter be set to a Carriage Return for the e*Way to be fully compliant with the HL7 Standard.*

Both envelope types use the following configuration parameters:

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

MLLP

The **MLLP** envelope structure is displayed in Figure 1.

Figure 1 MLLP Envelope Structure
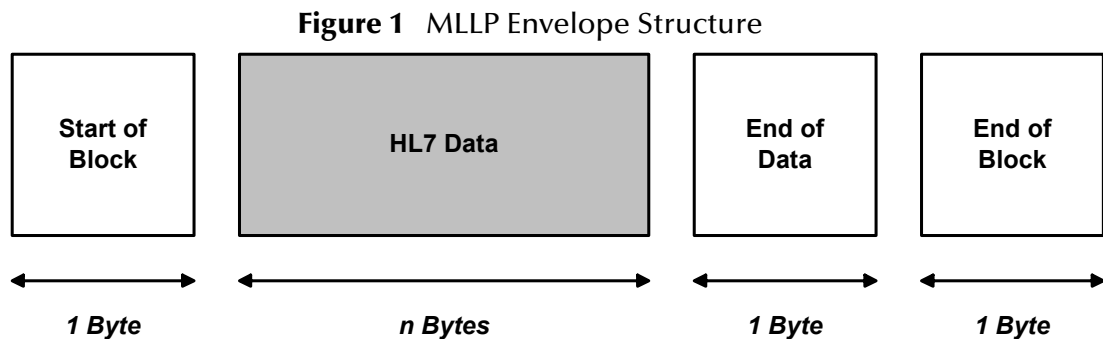
This envelope consists of a **Start of Block** component, a **Data** component, an **End of Data** component, and an **End of Block** component. The size of the **HL7 Data** field is determined by the length of the data (number of bytes between start and end), with a maximum size of 99999 Bytes.

For more detail, refer to Appendix C (Lower Layer Protocols) of the *HL7 Implementation Support Guide* for the HL7 version you are using.

HLLP

The **HLLP** envelope structure is displayed in Figure 2.

Figure 2   HLLP Envelope Structure
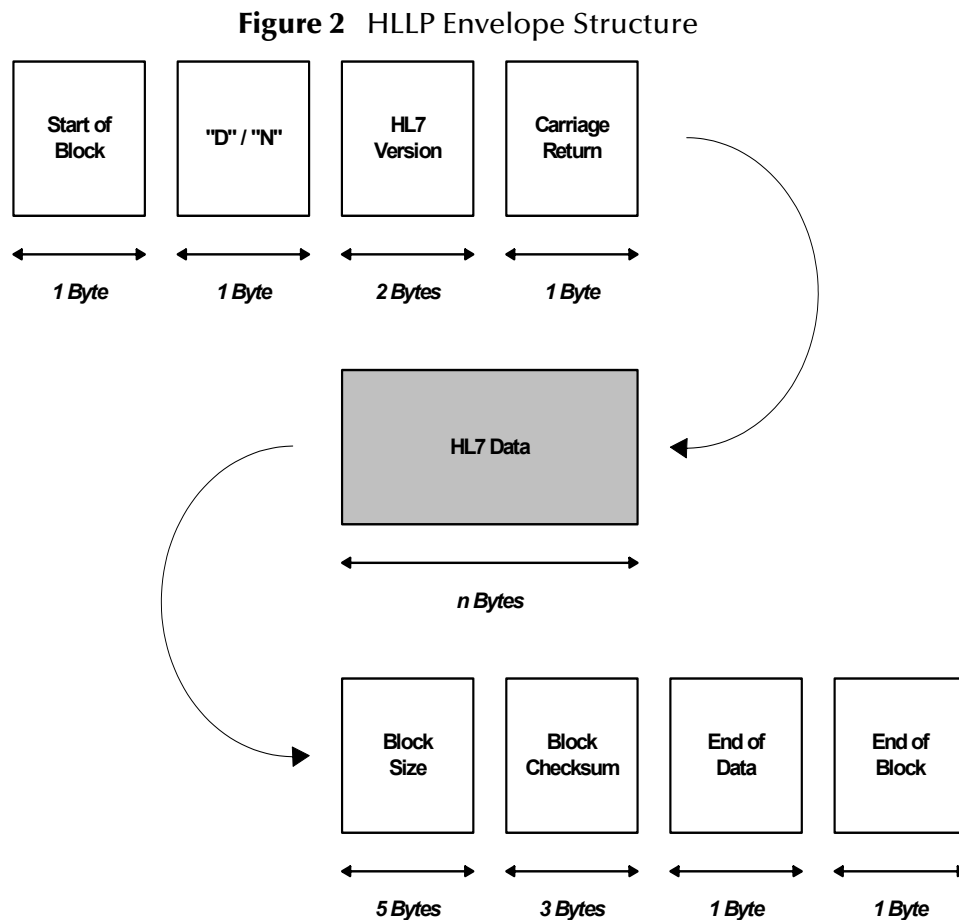
This envelope consists of a **Start of Block** component, a **'D'** (Data) or **'N'** (NAK) indicator, an **HL7 Version** component, a **Carriage Return**, a **Data** component, a **Block Size** component, a **Block Checksum** component, an **End of Data** component, and an **End of Block** component. 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.

For more detail, refer to Appendix C (Lower Layer Protocols) of the *HL7 Implementation Support Guide* for the HL7 version you are using.

3.2   Configuration Parameters

e*Way configuration parameters are set using the e*Way Editor. For a description of how a particular parameter is used in e*Gate, click the 📖 button in the Toolbar for that parameter.

To Change e*Way Configuration Parameters

1 In the Enterprise Manager's Component editor, select the e*Way you want to configure and display its properties.

2 Under **Configuration File**, click **New** to create a new file, **Find** to select an existing configuration file, or **Edit** to edit the currently selected file.

3 In the **Additional Command Line Arguments** box, type any additional command line arguments that the e*Way may require, taking care to insert them *at the end* of the existing command-line string. Be careful not to change any of the default arguments unless you have a specific need to do so.

For more information about how to use the e*Way Editor, see the e*Way Editor's online Help system or the *e*Gate Integrator Intelligent Queue Services Reference Guide*.

Note: *This e*Way is ported from the TCP/IP HL7 Super Communications Client, the existence of which predates the use of the name "e*Gate." The term "DataGate" has been retained in some parameter names to ensure compatibility, even though the functionality now pertains to e*Gate.*

The e*Way's configuration parameters are organized into the following sections:

- **General Settings** on page 16
- **Alert Notification** on page 19
- **Remote Client Setup** on page 19
- **Internal Buffer Configuration** on page 20
- **TCP/IP Setup** on page 22
- **Communication Setup** on page 23
- **HL7 Envelope Setup** on page 26
- **HL7 Application Protocol Setup** on page 28
- **Acknowledgment Generation** on page 32
- **Recourse Action** on page 33

3.2.1. General Settings

The **General Settings** section controls the direction of data to or from the e*Way.

Note: *Many of the parameters for the e*Way are specific to the direction the data is travelling, that is whether the e*Way is Inbound or Outbound to e*Gate.*

Data Direction

Description

Sets the direction of data flow (inbound to or outbound from e*Gate).

Required Values

Inbound or **Outbound**.

- ▪ **Inbound**: When configured as Inbound, this e*Way receives HL7 messages from the external system and forwards them to an e*Gate IQ. Depending upon the **Acknowledgment Generation** configuration, the e*Way either generates an HL7 Application Acknowledgment or expects one to arrive from the e*Gate IQ.

- ▪ **Outbound**: When configured as Outbound, the e*Way sends HL7 messages to the external system and expects a proper HL7 Application Acknowledgment. Depending on the **Acknowledgment Generation** configuration, the e*Way may forward the external's HL7 Application Acknowledgment to an e*Gate IQ.

Debug Flags

Note: *These Debug flags apply to older versions of e*Gate and DataGate. They have no effect on the current version of e*Gate and should therefor be disregarded.*

The purpose of this parameter is to override the value of the debug flags as supplied to the communication client via the -c option. If no options are selected, then the value, as set by the **-c** option takes precedence.

Table 2   Debug Flag Options

Value	Description
CHILD	General e*Way debug message
DBGLD	Configuration file loading information (not used—has no effect)
IDDBG	e*Way Identification debug message
IFCER	Interface ERROR debug message
IFCIN	Interface INFO debug message
IFCWA	Interface WARNING debug message
PROTO	Protocol-related debug message
QUEUE	Queue-related debug message (not used—has no effect)
RDDTA	General Read operation debug message
RDMSG	Message Read operation debug message
ROUTE	Routing specific debug message (not used—has no effect)
RTRSR	Pending Read debug message (not used—has no effect)
RWDTA	General Read/Write operation debug message
SLECT:	Select() call debug messages (not used—has no effect)
TMOUT	Read/Write Timeout debug messages (not used—has no effect)
VBOSE	Very verbose Read operation debug message (character by character—should be used sparingly)
VERIF	Message verification debug message (not used -- has no effect)

Table 2   Debug Flag Options

Value	Description
WTDTA	General Write operation debug message (with respect to system level calls)—also used to log Outgoing messages to the external system
WTMSG	Message Write operation debug message (with respect to Message/Data sending operations)

Interface Name

The value of this parameter is used to communicate a more verbose name to the **status** commands that originate from **stccmd.exe** or the e*Gate Monitor.

Example:

    status lab2xray Monitor Screen:

Interface name:	Labratory_To_XRAY
^^^^^^^^^^^^^^	
Transaction to be processed:	0
Transaction to be sent:	
External Connection Status:	connected
External Retries:	0
Interface Status:	up
Last Transaction Sent:	
Last Transaction Received:	Fri Jan 10 21:41:01 1997
Type of Interface:	dynamic
Sequence Number:	10330

Internal Timeout

Description

Internal Timeout is a configurable variable used to fine tune the select timeout in terms of microseconds inside the wrapper code. The higher the timeout value, the less CPU time committed to the process.

Required Values

An integer ranging from 0 to 2147483647. The configured default is 10000.

3.2.2. Alert Notification

New Control Broker Host Name

Description

This parameter is used for Alert Notification purposes. Alert notifications are done via New Control Broker (newcb). This parameter specifies the name of the machine on which newcb is running.

Required Values

A valid host name. (No default value.)

New Control Broker Port Number

Description

This parameter is used for Alert Notification purposes. Alert notifications are done via New Control Broker (newcb). This parameter specifies the TCP/IP port number that newcb is listening for incoming connections.

Required Values

A valid port number between 2000 and 32767. (No default value.)

Client ID

Description

This parameter is used for Alert Notification purposes. Alert notifications are done via New Control Broker (newcb). All e*Ways must register with newcb before they can send out any alert notifications. This parameter will be used as a unique identifier for registering with newcb.

Required Values

A valid Client ID number between 0 and 2147483647. (No default value.)

3.2.3. Remote Client Setup

This e*Way can be set up to run either Local to e*Gate or on a Remote machine. If the e*Way is being set up as a Remote client, then the machine on which this e*Way will reside must be capable of TCP/IP communication with the outside world (specifically with the host on which e*Gate will be running).

The parameters in this section help you set this communication as a remote one which will connect and communicate with e*Gate on a different machine. Note that the term "DataGate" appears in some parameter names, even though the functionality now pertains to e*Gate.

*Note: To set up this e*Way remotely, you need to follow the procedures in the e*Gate Integrator Installation Guide.*

Communication Client Location

Description

Whether the e*Way is to be Local to e*Gate or run as a Remote client.

Required Values

Local or **Remote**. The default value is **Local**.

e*Gate Server Host Name

Description

This parameter should be set if you are planning to set up this e*Way as a REMOTE CLIENT. Otherwise, this parameter is ignored.

The value of this parameter is the name of the host on which e*Gate will be running. You may also provide the IP address of the e*Gate registry host. When starting this e*Way remotely, it will use either the value of this parameter or the value as passed by the **-rh** option to indicate which host e*Gate is running on. The **-rh** option takes precedence.

Required Values

A valid host name. (No default value.)

e*Gate Server Port Number

Description

This parameter should be set if you are planning to set up this e*Way as a REMOTE CLIENT. Otherwise, this parameter is ignored.

The value of this parameter is the port number to which e*Gate listens for connections from e*Ways. When starting this e*Way remotely, e*Gate will use either the value of this parameter or the value as passed by the **-p** option to indicate which port e*Gate is listening for connections. The **-p** option takes precedence.

This parameter must be used in conjunction with **DataGate Server Host Name**.

Required Values

A valid port number between 2000 and 32767. (No default value.)

3.2.4. Internal Buffer Configuration

Internal buffers are used to store messages that are received or are being prepared to be sent. At runtime, these buffers store information in RAM. The **Internal Buffer Configuration** parameters allow you to optimize the memory consumption by the e*Way on your e*Gate Integrator system.

Initial Size

Description

Specifies the initial size of the internal buffer.

Required Values

An integer. You must enter an initial buffer size. The default is 4096 bytes. The allowed range is 1024 to 8192 bytes.

Additional Information

To enter the initial buffer size in kilobytes or megabytes, select the desired unit from the **Units** drop-down list. The current **Range** must be able to be converted to the specified unit.

If larger messages are to be sent (or were received), this internal buffer is expanded until the **Absolute Max Size** is reached. Make sure the value of this parameter is less than the value set for **Absolute Max Size**.

If the **Initial Size** is configured to be larger than the **Absolute Max Size**, then at load time the e*Way will make a correction by setting the **Absolute Max Size** to the **Initial Size** value plus 8192 bytes—provided that the resulting value does not exceed 2.15 GB.

Absolute Max Size

Description

This maximum buffer size is the largest amount of memory the e*Way will allocate for sending and receiving messages. If the size of internal memory allocated exceeds this value, the e*Way will RESET its external connection and ignore all data received.

The purpose of this parameter is to control memory consumption. It helps prevent any misbehavior by the external system from exhausting memory in the e*Gate system by ignoring overly large Events.

Required Values

An integer between 8192 and 131072 (Bytes). The default value is 65536 Bytes.

Additional Information

To enter the maximum buffer size in kilobytes or megabytes, select the desired unit from the **Units** drop-down list. The current **Range** must be able to be converted to the specified unit.

At startup, the internal buffer is set to the value of the **Initial Size** parameter. The size of the internal buffer is increased automatically in increments of 4096 bytes to accommodate incoming data. Once the **Absolute Max Size** is reached, however, the e*Way continues reading—but ignoring—the data until all of the data is read. The e*Way then generates a NAK (Canned or regular) and sends it to the external system.

If the **Initial Size** is configured to be larger than the **Absolute Max Size**, then at load time the e*Way will make a correction by setting the **Absolute Max Size** to the **Initial Size** value plus 8192 bytes—provided that the resulting value does not exceed 2.15 GB.

3.2.5. TCP/IP Setup

The **TCP/IP Setup** section contains parameters that set up the communication channel between the external system and the e*Way.

Connection Type

Description

TCP/IP clients request a connection from a TCP/IP server. TCP/IP servers listen for requests originating from TCP/IP clients. This parameter identifies the e*Way as either a client or server.

Required Values

Client or **Server**.

- **Client**: The e*Way will actively make a connection with the external system.

- **Server**: The e*Way waits for an incoming connection from the external system.

Additional Information

Generally, Inbound e*Ways are set up as servers and Outbound e*Ways are set up as clients. However, this may differ depending on your configuration and the nature of the external system to which the e*Way is connecting.

External Application Host Name

Description

Specifies the name of the system on which the external system is running.

Required Values

One of the following options:

- localhost

- 127.0.0.1

- local

The default value is **localhost**.

Additional Information

This parameter is required whether you are configuring this e*Way as a Server or as a Client. You may obtain the host name from the vendor or system administrator of the External System with which this e*Way is to communicate.

External Application Port Number

Description

Specifies the port number through which the e*Way will communicate with the external system.

Required Values

An integer between 2000 and 32767. The default value is 3000.

Additional Information

If the e*Way is set up as a server, it listens to this port for a connection (incoming from the external system). In this case, the external system is the client attempting to connect to the e*Way over this port.

If the e*Way is set up as a client, it attempts to connect to the external system through this port. In this case, the external is the server listening for connections through this port.

3.2.6. Communication Setup

The **Communication Setup** section controls timeouts and retry actions for the e*Way's connection to the external system.

Time to Wait Before Attempting Connection

Description

If the e*Way is set up as a client, this parameter specifies the length of time the e*Way waits before attempting to connect to the external system. If the e*Way is set up as a server, it specifies the length of time the e*Way waits before allowing a connection from the external system.

Required Values

An integer representing the number of seconds; the default is 10 seconds. The allowable range is 1 second to 120 seconds.

Additional Information

To enter the value in minutes or hours, select the desired unit from the **Units** drop-down list. The current **Range** must be able to be converted to the specified unit.

Time to Wait for a Response

Description

When the e*Way is set up as Outbound, specifies the amount of time the e*Way waits for an acknowledgment to arrive from the external system before taking action. If the e*Way is set up as a server, determines whether or not to allow a new connection coming in when another connection to the external system already exists. If the existing connection has been idle for more than the time specified by this parameter, the new connection forces the existing connection to be abandoned.

Required Values

An integer representing the number of seconds; the default is 30 seconds. The allowable range is 1 second to 120 seconds.

Additional Information

This parameter is used in two (2) ways:

1   If you have set up this e*Way as a sending (or Outbound) Client, then it will use the value of this parameter to wait for a response before taking action. See **"Recourse Action" on page 33**.

2   If you have setup this e*Way as a TCP/IP Server, then the value of this parameter is used to allow or reject a new incoming connection being made whether or not another connection has already been established. If the external connection has been IDLE for more than the amount of time as specified by this parameter, then a new connection will force the current IDLE connection to be abandoned.

The value of this parameter is ignored if you have setup this e*Way as a receiving (or Inbound) Client.

To enter the value in minutes or seconds, select the desired unit from the **Units** drop-down list. The current **Range** must be able to be converted to the specified unit.

This parameter works in conjunction with the **Max No Response** parameter, described below.

Max No Response

Description

Specifies the maximum number of response timeouts the e*Way allows while waiting for an Acknowledgment from the external system before taking further action.

Required Values

An integer representing the number of timeouts; the default is 30. The allowable range is 1 to 120 timeouts.

Additional Information

This parameter is used by Outbound e*Ways only and works in conjunction with the **Resend** option of the **Recourse Action** parameter **Action on No Response**. It configures the e*Way to resend the last Event for the specified maximum number of times before the subsequent recourse action is taken.

When an Outbound e*Way receives more than the maximum no response timeouts, the recourse action specified by the **Action on Max No Response** parameter is taken.

Time to Wait for Read to Complete

Description

Whether you have set this e*Way as Inbound or Outbound, it is possible that one of the following could be slow in getting the Data or Acknowledgment to the e*Way:

- The sender (i.e. external application or host)

- The network

- The communication line

As a result, the read/receive operation may appear to the e*Way to be slow. This parameter allows you to adjust how long a read/receive operation will take to complete—thereby avoiding an immediate timeout, causing the read or receive operation to fail.

Specifies the amount of time allowed for a read or receive operation to be completed.

Required Values

An integer representing the number of milliseconds; the default is 500 milliseconds. The allowable range is 100 milliseconds to 1000 milliseconds.

Additional Information

To enter the value in seconds or minutes, select the desired unit from the **Units** drop-down list. The current **Range** must be able to be converted to the specified unit.

Max Empty Read Retry

Description

Specifies the maximum number of times the e*Way attempts to read data from the external system when the read or receive operation returns nothing.

Required Values

An integer representing the number of retrys; the default is 5. The allowable range is 1 to 100 times.

Max NAK Receive Retry

Description

Specifies the maximum number of negative acknowledgments the e*Way receives before taking recourse action.

Required Values

An integer representing the number of NAKs; the default is 30. The allowable range is 1 to 120.

Additional Information

This parameter is used by Outbound e*Ways only.

This parameter works in conjunction with the recourse action parameters **Action on Nak Received** and **Action on Max Nak Received**.

Max NAK Send Retry

Description

Specifies the maximum number of acknowledgments the e*Way sends before taking recourse action.

Required Values

An integer representing the number of NAKs; the default is 30. The allowable range is 1 to 120.

Additional Information

This parameter is used by Inbound e*Ways only.

This parameter works in conjunction with the recourse action parameter **Action on Max Nak Sent**.

Log Incoming Raw Message

Description

Determines whether or not to send the raw data coming from the external system to a log file.

Required Values

Yes or **No**.

Journal File Name

Description

Specifies the name of the journal file, which stores Events that are not picked up from the queue due to "no interest," "bad data," or a "down system."

Required Values

A valid filename, optionally including an absolute path (for example, **c:\temp\filename.txt**). If an absolute path is not specified, the file is stored in the e*Gate "SystemData" directory. See the *e*Gate Integrator System Administration and Operations Guide* for more information about file locations.

Additional Information

Journalling is enabled if this parameter contains a filename. Leaving this parameter blank disables journalling for the e*Way.

3.2.7. HL7 Envelope Setup

The **HL7 Envelope Setup** section specifies the envelope type and configuration for the e*Way.

HL7 Envelope Type

Description

Specifies the particular HL7 protocol to be used by the e*Way.

Required Values

Either of the following options:

- **HLLP -- Hybrid Lower Level Protocol**

- **MLLP -- Minimal Lower Level Protocol**

Start Block Character

Description

The first envelope marker character in the HL7 envelope.

Required Values

An ASCII character expressed in ASCII, decimal (DEC), octal (OCT), or hexadecimal (HEX). The ASCII form is provided to allow characters such as:

- NewLine (\n)
- CarriageReturn (\r)
- Tabs (\t)
- Bells (\b)
- FormFeeds (\f)

The default value is **b** (HEX). The allowed range is 1 to 31 (DEC), 1 to 37 (OCT), or 1 to 1f (HEX).

End Data Character

Description

The second to the last envelope marker character in the HL7 envelope.

Required Values

An ASCII character expressed in ASCII, decimal (DEC), octal (OCT), or hexadecimal (HEX). The ASCII form is provided to allow characters such as:

- NewLine (\n)
- CarriageReturn (\r)
- Tabs (\t)
- Bells (\b)
- FormFeeds (\f)

The default value is **1c** (HEX). The allowed range is 1 to 31 (DEC), 1 to 37 (OCT), or 1 to 1f (HEX).

End Block Character

Description

The last envelope marker character in the HL7 envelope.

Required Values

An ASCII character expressed in ASCII, decimal (DEC), octal (OCT), or hexadecimal (HEX). The ASCII form is provided to allow characters such as:

- NewLine (\n)

- CarriageReturn (\r)

- Tabs (\t)

- Bells (\b)

- FormFeeds (\f)

The default value is **\r** (ASCII). The allowed range is 1 to 31 (DEC), 1 to 37 (OCT), or 1 to 1f (HEX). To be strictly compliant with the HL7 Standard, this parameter **must** be set to a Carriage Return.

3.2.8. HL7 Application Protocol Setup

The **HL7 Application Protocol Setup** section specifies the HL7 Protocol parameters for the e*Way.

Version

Description

Specifies the particular HL7 version to be used by the e*Way.

Required Values

One of the following options:

- **2.1**

- **2.2**

- 2.3

The default value is **2.1**.

Segment Delimiter

Description

The delimiter used to mark the end of each HL7 segment, when the e*Way is used in Inbound mode.

Required Values

An ASCII character expressed in ASCII, decimal (DEC), octal (OCT), or hexadecimal (HEX). The ASCII form is provided to allow characters such as:

- NewLine (\n)

- CarriageReturn (\r)

- Tabs (\t)

- Bells (\b)

- FormFeeds (\f)

The allowed range is **1** to **127** (DEC), **1** to **165** (OCT), or **1** to **7f** (HEX).

The default value is **\r** (ASCII).

Maximum Fields in MSH Segment

Description

The maximum number of fields in the MSH segment. The value must correspond with the selected version of the HL7 standard. The value of this parameter is used to construct the HL7 Data or Canned Acknowledgment envelopes, when the e*Way is used in Inbound mode.

Required Values

An integer between **14** and **19**; the default value is **15**.

Location of the Control ID Field

Description

The field position of the control ID within the MSH segment.

Required Values

An integer between **10** and **19**; the default value is **10**.

Additional Information

The location of the (HL7) **Control ID** field in the MSH segment usually is 10. Do not set this parameter to any other value unless you are very certain that the sending or receiving external system requires the control ID field to be other than 10.

Location of the Sequence Number Field

Description

The field position of the sequence number within the HL7 segment.

Required Values

An integer between **13** and **19**; the default value is **13**.

Additional Information

The location of the (HL7) **Sequence Number** field in the MSH segment usually is 13. Do not set this parameter to any other value unless you are very certain that the sending or receiving external system requires the control ID field to be other than 13.

Use HL7 Sequence Numbering

Description

This e*Way is designed to process, generate, and verify the HL7 sequence numbering scheme. It is highly recommended to have both sides of the Collaboration use the HL7 sequence numbering scheme to prevent duplication of data.

Required Values

Yes or **No**. The default value is **No**.

Sequence Number File Name

Description

If sequence numbering is active; that is, you have set **Use HL7 Sequence Numbering** to **Yes**, then the most recent sequence number will be stored in a non-volatile area referred to by the value of this parameter. If the specified file does not exist it will be created, and the e*Way will re-synchronize (negotiate) a sequence number with the external application.

Required Values

The full path name of the sequence number file. The default value is **data/ScTcpHL7In.seq**.

Additional Information

If an absolute path is not provided, the file is stored in the e*Gate /data/ directory. See the *e*Gate Integrator System Administration and Operations Guide* for more information about file locations.

e*Gate Application Name for MSH

Description

The e*Way uses the value of this parameter when generating a Canned HL7 NAK. This parameter is used only if you have configured this e*Way as Inbound. The value you provide for this parameter helps the external system identify the source of an HL7 acknowledgment sent by this e*Way.

Required Values

Any name up to 20 characters.

Additional Information

The name of the HL7 Sending Application (in this case e*Gate) is the third field in the HL7 MSH segment. Refer to your HL7 manual and your site's operations manual in order to set the value of this parameter properly.

e*Gate Facility Name for MSH

Description

The e*Way uses the value of this parameter when generating a Canned HL7 NAK. This parameter is used only if you have configured this e*Way as Inbound. The value you provide for this parameter helps the external system identify the source of an HL7 acknowledgment sent by this e*Way.

Required Values

Any name up to 15 characters.

Additional Information

The name of the HL7 Sending Facility (in this case where e*Gate resides) is the fourth field in the HL7 MSH segment. Refer to your HL7 manual and your site's operations manual in order to set the value of this parameter properly.

External Application Name for MSH

Description

The e*Way uses the value of this parameter when generating a Canned HL7 NAK. This parameter is used only if you have configured this e*Way as Inbound. It specifies the name of the application receiving the NAK from the e*Way.

Required Values

Any name up to 30 characters.

Additional Information

The name of the external application is the fifth field in the HL7 MSH segment. Refer to your HL7 manual and your site's operations manual in order to set the value of this parameter properly.

External Facility Name for MSH

Description

The e*Way uses the value of this parameter when generating a Canned HL7 NAK. This parameter is used only if you have configured this e*Way as Inbound. It specifies the facility name for the application receiving the NAK from the e*Way.

Required Values

Any name up to 30 characters.

Additional Information

The facility name for the external application is the sixth field in the HL7 MSH segment. Refer to your HL7 manual and your site's operations manual in order to set the value of this parameter properly.

HL7 Acknowledgement Level

Description

This parameter specifies whether the external application is configured to send an HL7 Application Acknowledgement, either after successfully receiving the Event or after the Event has been successfully Committed to the application database.

Required Values

Either Application Level or Commit Level. The default value is Application Level.

Use HL7 Processing ID

Description

This parameter specifies whether or not to use the HL7 Processing ID.

Required Values

Yes or **No**. The default value is **No**.

3.2.9. Acknowledgment Generation

The **Acknowledgment Generation** section controls how the application acknowledgment Events are handled.

e*Gate Server Sends Application Acknowledgment

Description

Specifies whether the **HL7 Application Acknowledgment** sent to the external system is generated by the e*Way or forwarded from e*Gate. The value of this parameter has **no** effect on **e*Gate Acknowledgments**. See **"Acknowledgements" on page 13**.

Required Values

Yes or **No**. The default value is **No**.

- **Yes**: e*Gate receives or creates the HL7 Application Acknowledgment and sends it to the e*Way, which forwards it to the external system.

- **No**: The e*Way creates and sends the HL7 Application Acknowledgment directly to the external system.

Additional Information

This parameter is used by Inbound e*Ways only.

If the e*Way decides to generate and subsequently send a Canned NAK to the external application, then the value of this parameter is ignored.

Forward External Acks to e*Gate Server

Description

This e*Way normally sends the appropriate e*Gate (DGP) Acknowledgment to e*Gate when an HL7 Application Acknowledgment is received. However, sometimes it is necessary to also forward the *contents* of the HL7 Application Acknowledgment to e*Gate (as Data). To accomplish this, you set the value of this parameter to **Yes**.

(The forwarded HL7 Application Acknowledgment is sometimes used in conjunction with an Inbound HL7 e*Way that expects the HL7 Application Acknowledgment to come from e*Gate.)

Required Values

Yes or **No**. The default value is **No**.

- **Yes**: The e*Way forwards HL7 Application Acknowledgments from the external system to e*Gate for processing.

- **No**: HL7 Application Acknowledgments from the external system are not forwarded to e*Gate by the e*Way.

Additional Information

This parameter is used by Outbound e*Ways only.

3.2.10. Recourse Action

The **Recourse Action** parameters determine the action the e*Way will take when operations occur outside the configured constraints.

Action on No Response

Description

Determines the action the e*Way will take if no ACK is received from the external system in the allotted time. The amount of time is determined by the **Communication Setup** parameter **Time to Wait for a Response**.

Required Values

Exit, **Resend**, or **Reset**. The default is **Reset**.

- **Exit**: The e*Way terminates its connection with the external system and shuts down.

- **Resend**: The e*Way sends a DGP NAK to e*Gate, then receives back the same Event and resends it to the external system.

- **Reset**: The e*Way closes its connection with the external system and goes through the connection scenario.

Additional Information

This parameter is used by Outbound e*Ways only. The logic used by this function is as follows:

1 Wait for a response.

2 If timed out, increase **no-response-counter.**

3 If **no-response-counter** is greater than **Max No Response**, then take action (step 5); else continue.

4 Take action on no response (**Resend** or **Reset** or **Exit**); if **Resend**, resend and go to step 1; else continue.

5 Take action on **Max No Response** (**Reset** or **Exit**).

Action on Max No Response

Description

Determines the action the e*Way takes if it has tried to send an Event to the external system a maximum number of times and not received any response (HL7 Application Acknowledgement) from the external system. The maximum number times the e*Way sends an Event without receiving a response is determined by the **Communication Setup** parameter **Max No Response**.

Required Values

Exit or **Reset**. The default is **Reset**.

- **Exit**: The e*Way terminates its connection with the external system and shuts down.

- **Reset**: The e*Way closes its connection with the external system and goes through the connection scenario.

Additional Information

This parameter is used by Outbound e*Ways only.

Action on Max Failed Read Retry

Description

Determines the action the e*Way takes after it has reached the empty read limit set by the **Max Empty Read Retry** parameter.

Required Values

Exit or **Reset**. The default is **Reset**.

- **Exit**: The e*Way terminates its connection with the external system and shuts down.

- **Reset**: The e*Way closes its connection with the external system and goes through the connection scenario.

Additional Information

This parameter is used by Inbound e*Ways only.

Action on Nak Received

Description

Determines the action the e*Way takes when it receives an HL7 Application NAK from the external system.

Required Values

Resend, **Reset**, or **Skip Message**. The default is **Resend**.

- **Resend**: The e*Way sends a DGP NAK to e*Gate, then receives back the same Event and resends it to the external system.

- **Reset**: The e*Way closes its connection with the external system and goes through the connection scenario.

- **Skip Message**: The e*Way logs the rejected Event and marks it as done. An ACK is sent to the external system.

Additional Information

This parameter works in conjunction with the parameter **Action on Max Nak Received**.

Note: *The **Action on NAK Received** and **Action on Max NAK Received** parameters should **not** both be set to **Skip Message**.*

Action on Max Nak Received

Description

Determines the action the e*Way takes when the maximum number of HL7 Application NAKs have been received from the external system, as set by the **Max NAK Receive Retry** parameter.

Required Values

Exit, **Reset**, or **Skip Message**. The default is **Skip Message**.

- **Exit**: The e*Way terminates its connection with the external system and shuts down.
- **Reset**: The e*Way closes its connection with the external system and goes through the connection scenario.
- **Skip Message**: The e*Way logs the rejected Event and marks it as done. A DGP ACK is sent to e*Gate to receive the next Event.

Additional Information

This parameter is ignored when the **Action on Nak Received** parameter is set to **Reset**.

Note: *The **Action on NAK Received** and **Action on Max NAK Received** parameters should **not** both be set to **Skip Message**.*

Action on Max Nak Sent

Description

Determines the action the e*Way takes when the e*Way has sent the maximum allowed number of NAKs to the external system, as set by the **Max NAK Send Retry** parameter.

Required Values

Exit or **Reset**. The default is **Exit**.

- **Exit**: The e*Way terminates its connection with the external system and shuts down.
- **Reset**: The e*Way closes its connection with the external system and goes through the connection scenario.

Additional Information

This parameter is used by Outbound e*Ways only.

Implementation

4.1 Inbound Functionality

4.1.1. Data Flow

Inbound to e*Gate, the TCP/IP HL7 e*Way receives HL7 messages from the external system. Each message is stripped from its envelope and verified to ensure it contains HL7 Data only.

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

The HL7 data is sent to e*Gate, and the e*Way waits indefinitely for a DGP ACK or NAK from e*Gate. As determined by the e*Way's configuration, one of two things will happen:

- e*Gate will send a DGP ACK or NAK to the e*Way. The e*Way then will generate an HL7 ACK or NAK and send it to the external system.

- e*Gate will send an HL7 ACK or NAK to the e*Way, which will format and send it to the external system.

Figure 3 on page 37 illustrates the flow of data for an inbound e*Way. A description of the diagram follows.

Figure 3   Data Flow for an Inbound e*Way
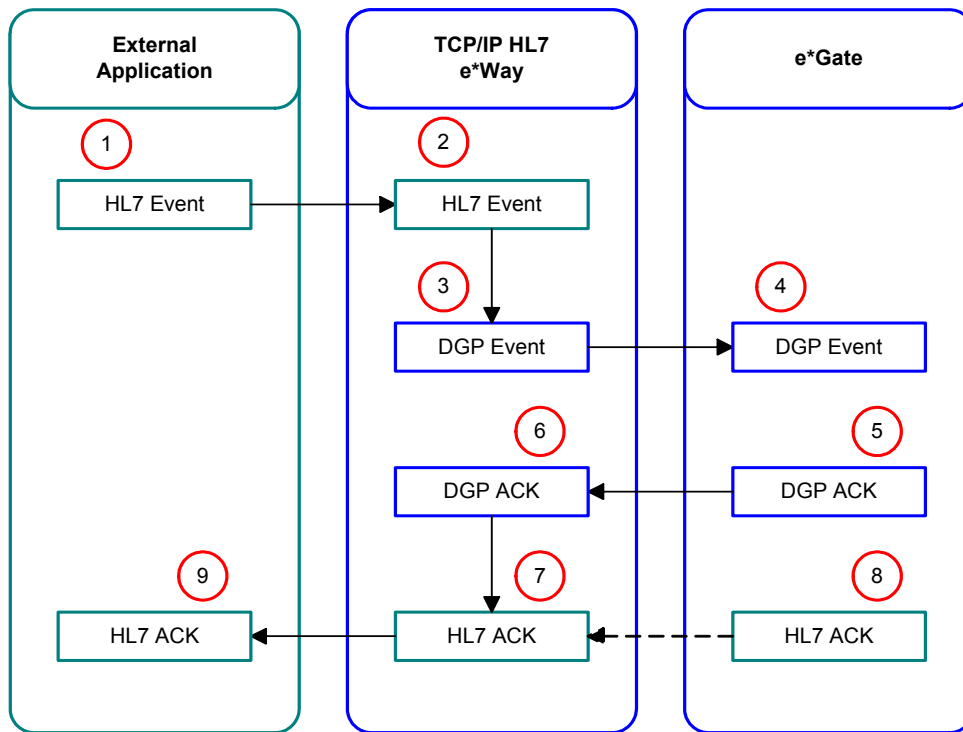
Data flow for an inbound e*Way

1   The external system sends the data to the e*Way.

2   The e*Way receives the data as an HL7 Event.

3   The e*Way strips the envelope from the Event and adds the appropriate status string to identify the Event.

4   The e*Way sends the Event to the next component in the e*Gate system according to the status tag.

5   e*Gate generates a positive or negative acknowledgment (DGP ACK or DGP NAK) and sends it to the e*Way.

6   The e*Way receives the acknowledgment from e*Gate.

7   The e*Way creates an HL7 ACK or NAK, adding the appropriate envelope for the external system.

8   Alternatively, e*Gate creates an HL7 ACK or NAK and send it to the e*Way.

9   The e*Way sends the HL7 ACK or NAK to the external system.

4.1.2. Event Verification

Event verification begins with reading the Event from the external system. The Event 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 an Event fails the read verification, it will be considered bad data. If read by an inbound e*Way, this failure will cause the e*Way to generate a Canned HL7 NAK. (An outbound e*Way will ignore the Event and log 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, and the Block Checksum and Block Size. The e*Way will behave as described above if the HLLP envelope verification fails.

After stripping the Event envelope, the actual contents of the HL7 data are examined to determine the type of Event. An incoming Event can be one of the following:

- HL7 Data

- HL7 ACK

- HL7 NAK

- Non-HL7 Data

As a minimum requirement, the HL7 Data component must start with "MSH," and the length must be equal to, or greater than, the number of fields in MSH plus three (3). The number of fields in MSH is obtained from the configuration file, and can be used for Block Size verification. If an Event fails the minimum requirement, it will be labeled as non-HL7 data.

The HL7 Data component is further processed by extracting the HL7 encoding characters from offset 4, then extracting all the fields in the MSH segment and storing them in an internal data structure. The e*Way tries to determine whether the HL7 Event is an HL7 ACK or an HL7 NAK by looking for the following string in the Event:

```
<HL7 Segment Delimiter>MSA<HL7 Field Separator>
```

The HL7 **Segment Delimiter** is a configuration parameter and the HL7 **Field Separator** is extracted from the incoming Event.

4.1.3. Acknowledgment Processing

e*Way Generates HL7 Acknowledgment

In this scenario, the e*Way generates an HL7 ACK after receiving a DGP ACK from e*Gate; otherwise, it generates an HL7 NAK. The HL7 ACK or NAK is placed in the proper envelope and sent to the external system. This scenario is illustrated in **Figure 3 on page 37**.

Note: *The expected message types (Event Types) from the external system are HL7 **data** only. The e*Way will ignore incoming HL7 ACK or HL7 NAK responses. It will generate a canned HL7 NAK on non-HL7 data. The expected response types from e*Gate are DGP ACK and DGP NAK. The* e*Way *will ignore all other input.*
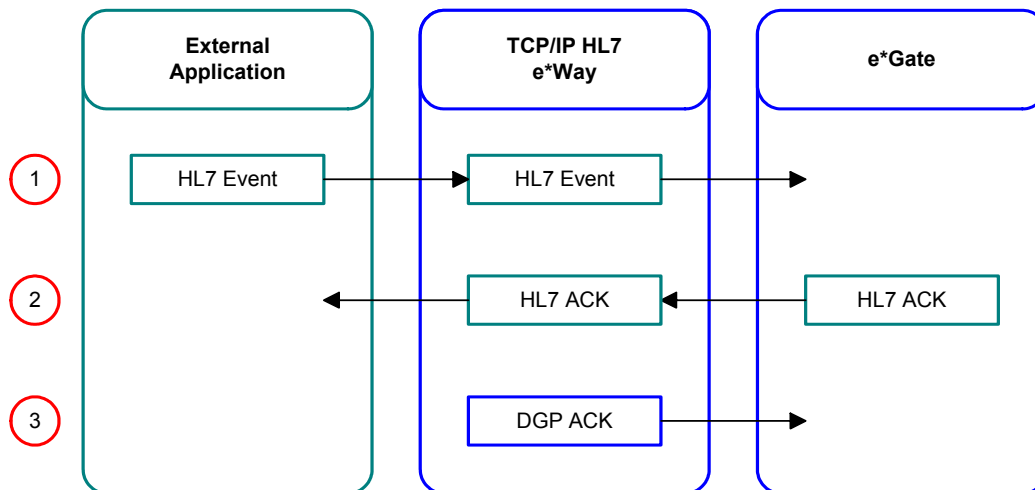
e*Gate Sends HL7 Acknowledgement

In this scenario, the e*Way will not react to the DGP ACK or NAK, and will wait until it receives either an HL7 ACK or NAK from e*Gate. The HL7 acknowledgment will either

be generated by e*Gate itself, or forwarded to e*Gate by a sending e*Way that has received an HL7 acknowledgment from an external system.

After verifying that the acknowledgement is an HL7 ACK or NAK, the e*Way will send a DGP ACK to e*Gate, place the HL7 ACK or NAK into the proper envelope, and send it to the external system. Anything other than an HL7 response will cause the e*Way to generate a warning and log the non-HL7 response in its log file. Figure 4 illustrates this scenario.

Figure 4 e*Gate Sends HL7 Acknowledgement

Note: *The expected message types from the external system are HL7* ***data*** *only. The e*Way will ignore incoming HL7 ACK and NAK responses. It will generate a Canned HL7 NAK on non-HL7 data. The expected acknowledgement types from e*Gate are DGP ACK, DGP NAK, HL7 ACK, or HL7 NAK. The e*Way will ignore all other messages.*

Canned HL7 NAK

A Canned HL7 NAK is created when a read error occurs, or when an Event cannot be identified as an HL7 message. The MSH segment in an HL7 Event is used to create the HL7 ACK/NAK, therefore a bad message should not be processed. There are five essential fields that must be present in the MSH segment of an HL7 message:

1 Sending Facility

2 Sending Application

3 Receiving Facility

4 Receiving Application

5 HL7 version number

The first four fields above are referred to as DataGate Facility (e*Gate), DataGate Application (e*Gate), External Facility, and External Application in the e*Way's configuration file.

The HL7 version number is the version of HL7 to which the HL7 Events will be formulated. The version number will appear in the configuration file as either 2.1, 2.2, or 2.3.

In order to build the Canned HL7 NAK, the number of fields in the MSH segment also must be specified in the configuration file.

The e*Way assumes its own HL7 encoding characters "|^~\&" when building the canned HL7 NAK.

4.1.4. Recourse Action

Recourse action is a configurable option. The only condition under which an inbound e*Way will take recourse action is:

- Excessive HL7 NAKs to the external system

Note: *The e*Way has internal counters that keep track of all error conditions.*

Excessive HL7 NAKs to the External System

This condition happens when the number of outgoing HL7 NAKs to the external system goes beyond the maximum allowed. The e*Way will respond in one of two ways:

- **RESET:** The e*Way will close its connection to the external system and go through the connection scenario at a later time, based on the **Time to Wait Before Attempting Connection** parameter.

- **EXIT:** The e*Way will terminate its connection to the external system and shut itself down.

4.2 Outbound Functionality

4.2.1. Data Flow

In the outbound direction, the e*Way receives HL7 Events from e*gate. Each Event is verified to ensure it contains HL7 data only. Legitimate HL7 data coming from e*gate is enveloped into its configured format and sent to the external system. HL7 acknowledgments and Non-HL7 data from e*Gate also are read and acknowledged. If journaling is set, these Events are placed in the journal file.

The e*Way waits for a configurable number of seconds (response timeout) for an incoming HL7 ACK or NAK from the external system. After receiving an HL7 response from the external system, the e*Way strips the Event from its envelope and verifies its integrity.

Any Non-HL7 acknowledgment received from the external system causes the e*Way to resend the same Event. If the incoming response is an HL7 ACK or NAK, the e*Way may do either of the following, as dictated by its configuration:

- Forward the external system's HL7 response to e*Gate
- Send a DGP ACK or NAK to e*Gate

Figure 5 illustrates the flow of data for an outbound e*Way. A description of the diagram follows.

Figure 5 Data Flow for an Outbound e*Way
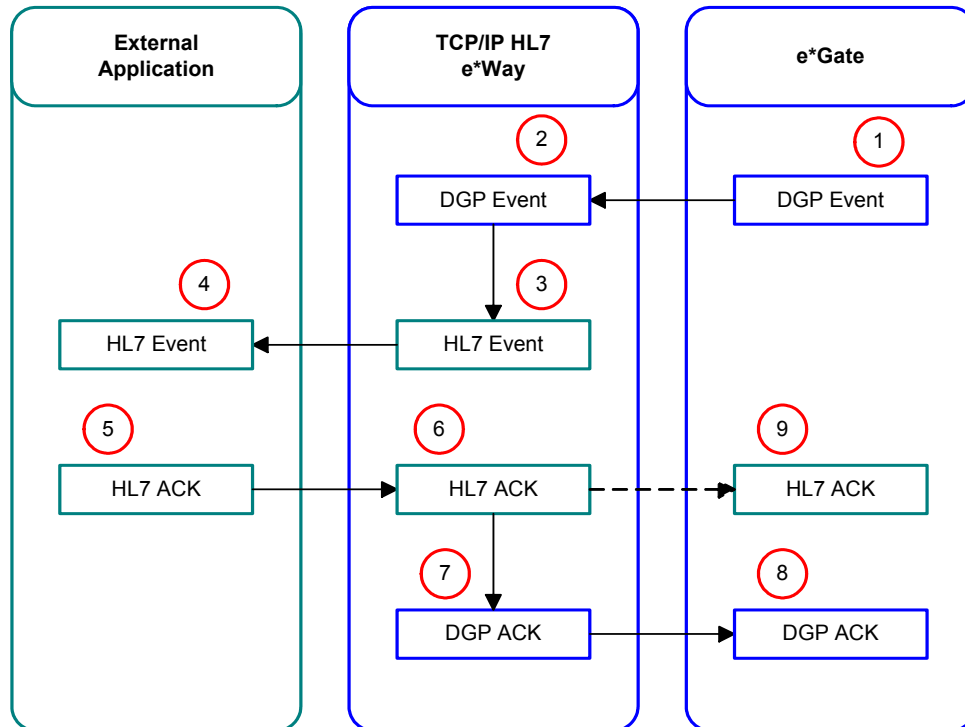
Data flow for an outbound e*Way

1 e*Gate sends the Event to the e*Way.

2 The e*Way receives the Event from e*Gate.

3 The e*Way adds the status string and appropriate envelope to the Event.

4 The e*Way sends the enveloped HL7 Event to the external system based on the status string.

5 The external system generates an HL7 ACK or NAK and sends it to the e*Way.

6 The e*Way receives the acknowledgment from the external system.

7 The e*Way strips the envelope from the acknowledgment and adds the appropriate status string to identify the acknowledgment.

8 The e*Way sends the DGP ACK or NAK to e*Gate.

9 Alternatively, the e*Way forwards the HL7 ACK or NAK to e*Gate.

4.2.2. Event Verification

Verification of e*Gate-outbound Events is the same as in the inbound scenario, with the exception that if an Event fails the read verification, the e*Way will ignore the Event and log a warning, reporting the nature of the problem to the log file.

4.2.3. Acknowledgment Processing

Receiving and Processing HL7 Responses

When configured as an outbound e*Way, it will send HL7 Events to the external system and expect HL7 acknowledgments from the external system. If the e*Way is not configured to forward these acknowledgments to e*Gate, it will process them for validity to distinguish between an HL7 ACK versus an HL7 NAK and, accordingly, send e*Gate either a DGP ACK or DGP NAK. This scenario is illustrated in **Figure 5 on page 41**.

Forwarding External HL7 Responses to e*Gate

In this scenario, the e*Way receives an HL7 ACK from the external system and then sends an HL7 ACK to e*Gate. e*Gate responds by sending a DGP ACK to the e*Way, and the e*Way sends a DGP ACK to e*Gate to acknowledge receipt of the response. e*Gate then sends the next Event. Figure 6 illustrates this scenario:

Figure 6   e*Way Forwards HL7 ACK
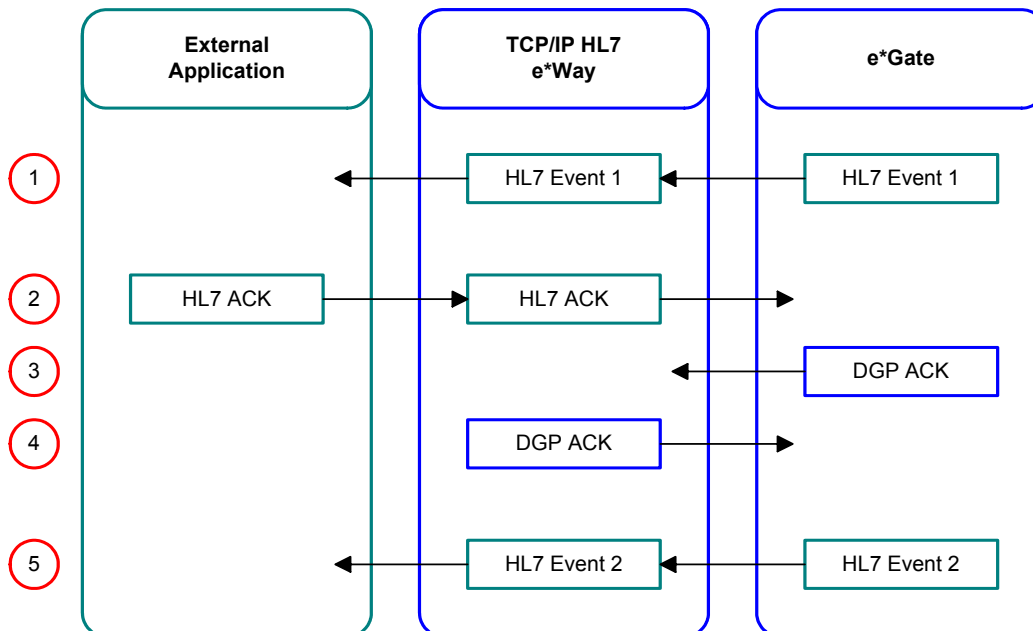
If the e*Way receives an HL7 NAK, however, it sends an HL7 NAK to e*Gate, and e*Gate resends the same Event. Figure 7 illustrates this scenario:

Figure 7 e*Way Forwards HL7 NAK
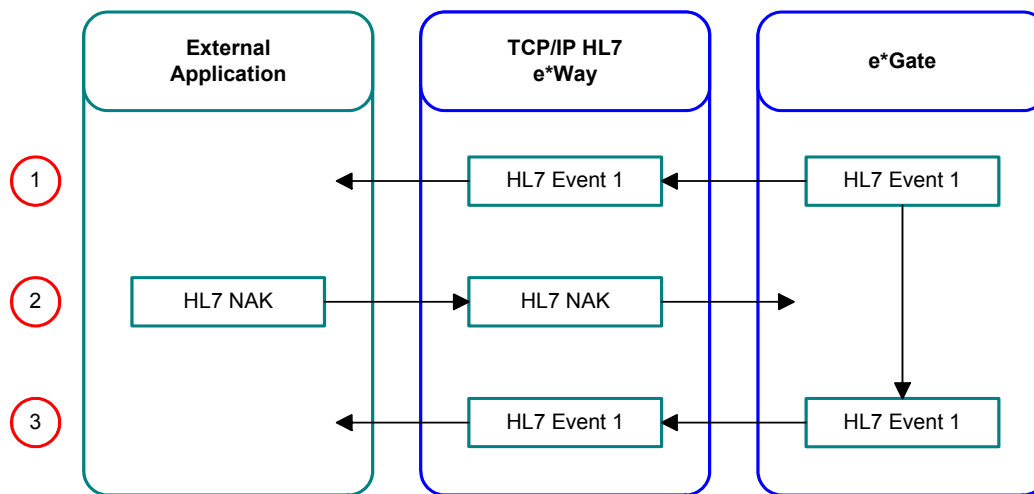
Note: *The expected message types (Event Types) from the external system are HL7 ACKs*
*and HL7 NAKs. The e*Way will ignore all other data. The expected Event types*
*from e*Gate are HL7 Data, DGP ACKs, and DGP NAKs. The e*Way will ignore all*
other Events.

4.2.4. Recourse Action

Recourse action is a configurable option. The conditions under which an outbound
e*Way will take recourse action are:

- Receiving an HL7 NAK from the external system
- Excessive HL7 NAKs from the external system
- The e*Way times out waiting for an external HL7 acknowledgment
- Excessive timeouts or failed read attempts

Note: *The e*Way has internal counters that keep track of all error conditions.*

Receiving an HL7 NAK from the External System

The e*Way can react in one of the following ways, upon its receipt of an HL7 NAK from
the external system:

- **RESEND:** The e*Way will send a DGP NAK to e*Gate to receive the same Event to
  be resent to the external system.
- **SKIP MESSAGE:** The e*Way will log the rejected Event and send a DGP ACK to
  e*Gate to receive the next Event. If journaling is on, the skipped Event will be
  journaled.
- **RESET:** The e*Way will close its connection with the external system and go
  through the connection scenario at a later time based on the **Time to Wait Before
  Attempting Connection** parameter.

Excessive HL7 NAKs from the External System

The e*Way can react in one of three ways when the number of incoming HL7 NAKs exceeds the maximum allowed. The **Max NAK Receive Retry** count is a configurable parameter.

- **SKIP MESSAGE:** The e*Way will log the rejected Event, and send a DGP ACK to e*Gate to receive the next Event. If journaling is on, then the skipped Event will be journaled.

- **RESET:** The e*Way will close its connection to the external system and go through the connection scenario at a later time based on the **Time to Wait Before Attempting Connection** parameter.

- **EXIT:** The e*Way will terminate its connection to the external system, and shut itself down.

e*Way Times Out Waiting for a Response

This condition happens when the e*Way times out waiting for the incoming HL7 ACK/ HL7 NAK from the external system. The **Time to Wait for a Response** is a configurable parameter. A timeout scenario will cause the e*Way to behave in one of three ways:

- **RESEND:** The e*Way will send a DGP NAK to e*Gate to receive the same Event to be resent to the external system. This recourse action can only be used in conjunction with HL7 sequence numbering. The e*Way will automatically change this option to RESET if the sequence numbering option is turned off in the configuration file.

- **RESET:** The e*Way will close its connection to the external system and go through the connection scenario at a later time, based on the **Time to Wait Before Attempting Connection** parameter.

- **EXIT:** The e*Way will terminate its connection to the external system and shut itself down.

Excessive Timeouts (on Responses from External System)

This condition happens when the number of timeouts waiting for a response goes beyond the maximum allowed timeouts, and applies to outbound e*Ways only. The e*Way will behave in one of two ways:

- **RESET:** The e*Way will close its connection to the external system and go through the connection scenario at a later time.

- **EXIT:** The e*Way will terminate its connection to the external system and shut itself down.

4.3   General Functionality

4.3.1. Non-blocking I/O

The non-blocking I/O feature will prevent the e*Way from locking up when an attempt to read or write data blocks. This will enable the e*Way to continue its operation in case of any communication errors.

If the read attempt fails for a configurable number of times, the e*Way will exit or reset its connection to the external system, based on its configuration.

In the event of a failed write, it is able to resume its write operation to pick up where it previously left off until the entire Event is successfully sent.

Without this feature, there is a possibility the e*Way will lock up when a read or write failure occurs, and will be unresponsive to all external events, including requests from the user or the e*Gate monitor (for status).

4.3.2. Internal Buffer Allocation Logic

This buffer is allocated one time at start-up and is used for storing the incoming Events from the external system. The size of this buffer is configurable.

There are two parameters that specify the **Initial Size** and **Absolute Max Size** of the internal buffer. The e*Way keeps track of the size of this buffer by remembering its current size. If the size of the incoming Event is larger than the current size of the external buffer, it will resize the current size of the buffer, increasing it in increments of 4096 bytes. This will happen each time an overflow occurs. The e*Way will continue resizing the buffer until the buffer size equals **Absolute Max Size**.

The e*Way will not resize the message beyond its **Absolute Max Size** if another overflow occurs. It will continue reading and ignoring the data from the external until all data is read. When the e*Way is finished reading, it will generate a Canned HL7 NAK and send it to the external system.

4.3.3. Support for HL7 Sequence Numbering Protocol

The e*Way can be configured to use HL7 sequence numbering. For more details on HL7 sequence numbering, refer to Appendix C (Lower Layer Protocols) of the *HL7 Implementation Support Guide* for the HL7 version you are using.

Before using HL7 sequence numbering, the sequence field number should be specified in the e*Way's configuration file. The sequence numbering file will open at start-up. If the sequence numbering file does not exist, an empty one will be created.

The sequence numbering file is updated on the inbound e*Way when the e*Way generates the HL7 ACK (this process is transparent to the user). Sequence numbering is updated on the outbound e*Way when the e*Way receives the HL7 ACK from the external system.

If you want to force the e*Way to resynchronize its sequence number with the external system, you must suspend the e*Way, remove the sequence number file, and activate the e*Way. A reload of configuration files is not necessary.

If you want to change the sequence number at runtime, you must suspend the e*Way, edit and save the sequence number in the sequence number file through the e*Way GUI, and finally reactivate the e*Way.

Note: *The minimum HL7 sequence number is 1. The maximum HL7 sequence number is 2 billion. Sequence number 0 is used to query the receiving system's expected sequence number. Sequence number -1 tells the sender to start at any sequence number.*

4.3.4. Journaling Unprocessed Events

The e*Way may be configured to journal unprocessed Events from e*Gate or SKIPPED Events (destined for the external system). An unprocessed Event is Non-HL7 data received from e*Gate. SKIPPED Events are those which get continuously NAKed by the external system and thus are SKIPPED (if configured).

The format of the journal file is the same as that used by e*Gate. This will help you resend those Events after you have examined and modified them.

Journaling is activated when you specify the **Journal File Name** in the configuration file.

4.3.5. Suspend, Reload, Activate Logic

When the e*Way is asked to suspend, it will close its journal file and sequence number file (provided these were configured for use). This will allow you to modify these files.

On reload, the e*Way will drop its connection and attempt connection at a later time.

On activate, the sequence number file and the journal file are opened again (provided the newly loaded configuration parameters have set sequence numbering and journaling to be active).

It is recommended you do not change the direction of the e*Way during runtime.

4.4 Sample Configuration

The TCP/IP HL7 e*Way includes a sample (or default) configuration file named **stcdgwtcpiphl7.def**, which is found in the **eGate/Client/configs/stcdgwtcpiphl7** directory (see **"Files/Directories Created by the Installation" on page 11**). This file should be used as a template for the actual configuration file you set up for your implementation.

4.5   Troubleshooting

4.5.1. Runtime Assumptions and Conflict Resolution

The TCP/IP HL7 e*Way assumes certain situations and configurations to operate properly, and in some situations will automatically correct conflicting configuration options (see **"Conflict Resolution" on page 48**).

Internal Buffer

1   During a read from the external system, if more bytes are read than the value in the **Initial Size** parameter (**Internal Buffer Configuration** section), the e*Way will resize its internal buffer in increments of 4096 bytes in order to store more data. The e*Way will not resize its internal buffer beyond the value in the **Absolute Max Size** parameter.

2   In the case of a buffer overflow, the e*Way will read and discard the remainder of the Event. It will then generate a NAK and send it to the external system.

Non-Blocking I/O

The TCP/IP HL7 e*Way is using non-blocking I/O for reading and writing to and from the external.

▪ If the TCP/IP HL7 e*Way is blocked in the middle of a write, it can resume sending the remainder of the Event in its next try.

▪ If the TCP/IP HL7 e*Way fails during read in the middle, it will assume a failed read and act according to the **Action on Max Failed Read Retry** parameter in the **Recourse Action** section.

Connection and Reconnection Logic

▪ The TCP/IP HL7 e*Way will connect to the external immediately at start up.

▪ The amount of time the TCP/IP HL7 e*Way will wait between connection resets is configurable. You set this value in the **Time to Wait Before Attempting Connection** parameter in the **Communication Setup** section.

▪ The TCP/IP HL7 e*Way will try indefinitely to make a connection.

▪ The TCP/IP HL7 e*Way will reset the connection when its configuration files are reloaded.

4.5.2. HL7 Application Layer Assumptions

1   The HL7 **Segment Delimiter** is configurable, whereas, the HL7 Event encoding characters are extracted from the Event.

2   When generating a Canned HL7 NAK, we assume the following encoding characters: **|** (vertical bar), **^** (caret), **~** (tilde), **** (back slash), and **&** (ampersand).

3   The TCP/IP HL7 e*Way assumes that all HL7 Events start with "MSH" followed by the HL7 encoding characters. The encoding characters are assumed to follow MSH immediately. It is also assumed that the length of the incoming Event will be at least 3 plus ("max fields in MSH segment") bytes.

4   The maximum length of each field in "MSH" is 255 bytes. Any field with a length larger than 255 bytes will be truncated.

5   The incoming Event is treated as either an HL7 ACK or an HL7 NAK if the e*Way can find the following string of bytes in the Event: **<HL7 Segment Delimiter>MSA<HL7 Field Separator>**.

6   In the **HLLP** envelope, the size of the **Length** field is assumed to be 5 bytes of a zero-filled decimal.

7   If the e*Way is configured to use **HLLP** and the **Absolute Max Size** of the buffer is larger than 99,999, it will reduce the **Absolute Max Size** to 99,999.

8   The **Length** component in the **HLLP** envelope is a five-digit number and as a result it imposes a maximum of 99,999 bytes on HL7 messages.

9   In the **HLLP** envelope, the size of the **Block Checksum Character** is assumed to be 3 bytes of zero-filled decimal and is calculated by XOR'ing all bytes from one byte of the start block character up to and including the Length field.

10 The envelope Start of Block, End of Block and End of Data delimiter characters must be unique. Additionally, the Start of Block and End of Data delimiter characters must be different from the HL7 **Segment Delimiter**.

Possible Deadlock Situations

Possible deadlock situations include:

1   Outbound e*Way receives HL7 Data from the external connection:

This may be due to a configuration error in which the e*Way is set up as Outbound instead of Inbound. It is also possible that the Outbound e*Way is connected to an ill-behaving inbound communication program.

2   Outbound e*Way receives HL7 ACK with a sequence number different from its expected sequence number (ESN):

This problem can occur due to excessive amounts of timeouts from the external, or subsequent recourse action taken by the e*Way, which forces the e*Way to reset its connection to the external. It can also be caused by the e*Way and the external communication program starting up with non-synchronized sequence numbers in their respective sequence number files.

The situation can be corrected by suspending the e*Way and manually editing the contents of the sequence file, or by deleting the sequence file. Once this is completed, the e*Way can be reactivated.

Conflict Resolution

In some cases, the TCP/IP HL7 e*Way will automatically change the value of some parameters to resolve conflicts. The following table lists the conflicts that are

automatically resolved by the e*Way. All parameters are written in the form of **Section: Parameter**. Values are in *italics*.

Conflict	Resolution
General Settings: **Data Direction** = *Inbound* and **Acknowledgment Generation**: **Forward External Acks to DataGate Server** = *Yes*	**Acknowledgment Generation**: **Forward External Acks to DataGate Server** is changed to *No*.
General Settings: **Data Direction** = *Outbound* and **Acknowledgment Generation**: **DataGate Server Sends Application Acknowledgment** = *Yes*	**Acknowledgment Generation**: **DataGate Server Sends Application Acknowledgment** is changed to *No*.
Internal Buffer Configuration: **Initial Size** value is greater than **Internal Buffer Configuration**: **Absolute Max Size** value.	**Internal Buffer Configuration**: **Absolute Initial Size** is set to *+8192* above the value of **Internal Buffer Configuration**: **Initial Size**.

4.6 Error Messages and Corrective User Actions

Error messages appear in the log file. If any of the following problems occur, the e*Way will automatically shut down.

4.6.1. Startup

Configuration File Fails to Load

Message in log file:

```
SCCFG_X_LoadConfigFile() failed loading file <filename>. Check
previous log messages.
```

Description:

The configuration file failed to load due to a bad file name.

Possible action:

Check the configuration file name for accuracy in your e*Gate port table. Access the e*Gate "define port" dialog box. Additionally, you must check for the existence and read permissions of the file being accessed.

Non HLLP or MLLP HL7 Envelope Type

Message in log file:

```
PARAM: <HL7 envelope type> has unrecognizable value : <bad value>
```

Description:

The only possible values for the HL7 envelope are HLLP and MLLP.

Possible Action:

The configuration file must have a valid value (HLLP or MLLP) for the HL7 Envelope Type.

HL7 Version Number Incorrect

Message in log file:

```
PARAM: <HL7 version> has invalid length : [length]
```

Description:

The size of the HL7 version number is incorrect.

Possible Action:

The length of the HL7 version number must be exactly 3 bytes and the HL7 version number must be one of the following: 2.1, 2.2, or 2.3.

Conflict Between Envelope Markers and/or HL7 Segment Delimiter

Message in log file:

```
Configuration CONFLICT: Envelope characters conflict. Check previous
log messages.
```

Description:

The same values have been assigned to either the Start of Block, End of Block, or End of Data characters or the HL7 segment delimiter.

Possible Action:

Make sure the Start of Block, End of Block, and End of Data characters and/or the HL7 segment delimiter have unique values.

Configuration File Fails to Load

Message in log file:

```
SCCFG_X_LoadConfigFile() failed loading file <filename>. Check
previous log messages.
```

Description:

The configuration file failed to load due to a bad file name.

Possible Action:

Check the configuration file name for accuracy in your e*Gate port table. Access the e*Gate "define port" dialog box. Additionally, you must check for the existence and read permissions of the file being accessed.

4.6.2. Communication Problems with e*Gate

e*Way is Unable to Connect to e*Gate

Message in log file:

```
Connect to (port number) '127.0.0.1' failed: trying 'locathost' this
time.

Connect to 'localhost' failed; trying 'local' this time.

Couldn't connect to port (port number) on host 'hostname': (error-
number) :error-msg
```

Description:

The e*Way is unable to connect to e*Gate. The first two messages suggest that e*Gate may not be up and running. The last message indicates that an incorrect port number may have been used in attempting a connection to e*Gate or that e*Gate is not running.

Possible Action:

See e*Gate's trouble shooting procedure in the *e*Gate System Administration and Operations Guide*.

Failed Communication Between e*Gate and the e*Way

Message in log file:

```
receiveMessage () failed. Error: [error-number]; msg: <error msg>
```

Description:

Reading error on the e*Gate socket: [e*Gate socket number]. This usually occurs when e*Gate has crashed or was terminated without a proper shutdown.

Possible action:

Unless you determine why e*Gate was terminated, there is not much you can do to correct this situation. Under these circumstances, the e*Way will close any open connections or files.

4.6.3. Runtime Resource Limitations

Insufficient Memory

Messages in log file:

```
allocMessage () failed. Error: [error-number]; msg <error-msg>
```

```
dupMessage( ) failed. Error: [error-number], msg: <error msg>

Unable to allocate memory for SC_G_ExternalMessage of [number of
bytes]. Error is [system error]
```

Description:

Insufficient memory to accommodate the size of the Event.

Possible action:

Check memory usage by all other processes.

e*Way is Unable to Resize the Internal Buffer

Message in log file:

```
resizeMessage () failed. Unable to resize buffer! Will send DGP NAK to
DataGate.
```

Description:

The e*Way is unable to resize the internal buffers to accommodate an Event larger than the value in the **Absolute Max Size** parameter. A DGP NAK will be sent to e*Gate.

Possible action:

It is possible that you have run out of memory. Check memory usage of other system processes and restart the e*Way.

Event Size Exceeds Maximum Buffer Size

Message in log file:

```
Datagate message length = [length], max allowed message length =
[length]
```

Description:

The size of e*Gate Event exceeds the maximum length allowed. The e*Way is sending a DGP NAK to e*Gate.

Possible action:

If the e*Way is processing Events larger than you expected, you need to increase the value of the **Absolute Max Size** parameter.

4.6.4. Runtime Problems with the Sequence Numbering File

e*Way Unable to Access Sequence Numbering File

Message in log file:

```
fopen() failed for creating. Error: [error-number; msg: <error-msg>
Unable to open  sequence file <file-name> for read/write.
```

Description:

The e*Way is unable to open the sequence numbering file. The file could be missing or the permissions on the file will not allow the fopen() function to open the file.

Possible Action:

Make sure that the sequence numbering file exists.

e*Way Unable to Read Sequence Numbering File

Message in log file:

```
fseek ( ) failed. Error: [error-number]; msg : <error-msg>
```

Description:

The e*Way is unable to read the sequence numbering file. Again, the permissions on that file may be set incorrectly.

Possible Action:

Ensure the sequence numbering file has read permissions.

e*Way Unable to Update the Sequence Numbering File

Message in log file:

```
fflush () failed. Error: [error-number]; msg : <error msg>
```

Description:

The e*Way is unable to update the sequence numbering file. This rarely happens and is usually due to physical damage to the disk, or the disk may be full.

Possible Action:

First, determine whether or not the disk is full and remedy that situation. If not, check the hard disk for problems. Check the *e*Gate System Administration and Operations Guide*, or get help from your system administrator.

4.6.5. Failure in Loading Required Monk Files

Error messages are varied. If any of the four required Monk functions are not properly loaded, the TCP/IP HL7 e*Way will generate an error message in its log file and shut down.

4.6.6. Missing Dependencies for a Particular Envelope

When certain parameters are not set or have incorrect values, some adjustments are made automatically on the user's behalf (see **Conflict Resolution** on page 48); otherwise, the e*Way will generate an error message or shut down.

Index