

HL7 BRIDGE REQUIREMENTS, INSTALLATION, & CONFIGURATION GUIDE



Version 1.0



**Scientific Technologies
Corporation**



**Scientific Technologies Corporation
67 E. Weldon Avenue
Phoenix, AZ 85012**

Original: 1.0.0

Revisions:

Scientific Technologies Corporation (STC) provides this material "as is." The programmatic and technical staff used their best efforts to prepare and test this material.

©2007 by Scientific Technologies Corporation (STC). All rights reserved. Other trademarks, service marks, products, logos, or services are trademarks of their respective holders. Trademarked names have been used throughout this manual. Rather than insert a trademark (™) symbol where required, we state that we are using the names only in an editorial fashion, with no intention of infringement.

Document Number: HL7-RIC-1.0.0 – 07.06.07



TABLE OF CONTENTS

1	Introduction	1
	Senders, Listeners, and Connections	2
	TCP/IP LISTENER AND SENDER	3
	File Listener and Sender	4
	HTTPS Sender	5
2	System Requirements, Installation, & Configuration	7
	System Requirements.....	7
	Installation & Configuration	7
	Windows 2000/XP	7
	Linux and Solaris.....	9
	AIX	10
	IBM JVM Specific Instructions	10
	Configuration Files	11
	HL7Bridge.Config.....	11
	HI7bridge.log.level	12
	HL7bridge.cert.keystore.....	12
	Properties Files	12
	HTTP/HTTPS Support	18
	Troubleshooting	19



LIST OF FIGURES

Figure 1-1: Connection Layout.....	3
Figure 2-1: File-to-File Properties	14
Figure 2-2: File-to-HTTP Properties.....	15
Figure 2-3: File-to-HTTPS Properties	15
Figure 2-4: File-to-TCPIP Properties	16
Figure 2-5: TCPIP-to-File Properties	16
Figure 2-6: TCPIP-to-HTTP Properties.....	17
Figure 2-7: TCPIP-to-TCPIP Properties.....	17

LIST OF TABLES

Table 1-1: HL7 Communication Messaging Components	2
Table 2-1: Connection Configuration Properties.....	13



1 INTRODUCTION

Scientific Technologies Corporation's (STC's) HL7 Bridge version 1.0 is a simple application that allows systems to connect securely to other applications across the Internet. The HL7 Bridge may be used to:

- Allow local messaging systems to integrate to outside systems using a local messaging network or protocol.
- Transfer messages securely to external systems across the Internet.
- “Bridge the gap” between systems and modern web enabled applications.

The task of electronic messaging between computer systems has been compared to a conversation between two people which has three components: meeting place, language, and vocabulary. For example, if I go to lunch in the park and meet my friend Bob and we discuss particle physics; then we can say the meeting place was the park, the language was English, and the vocabulary was physics. This conversation would not have worked if I had not shown up at the same time as Bob, or if Bob only spoke French and I only spoke English, or if I did not know anything about physics.

Electronic communication (messaging) has these three same issues in common with human communication. Two computer systems must agree on a meeting place (transport), a language to speak (message format or structure), and the vocabulary to use (content of message). Many HL7 systems were designed to work within a local network and cannot meet or connect with HL7 systems outside their local network. This is where the HL7 Bridge comes in.



Table 1-1: HL7 Communication Messaging Components

Message Component	Human	Electronic
Meeting Place – Transport	Park	TCP/IP
Language – Format / Structure	English	HL7
Vocabulary – Content	Physics	ORU

The HL7 Bridge solves this problem by acting as a switchboard operator that connects parties using different HL7 systems. The internal HL7 system may use an internal protocol and the external HL7 system may use a secure protocol appropriate for the Internet.

The HL7 Bridge assumes that both systems speak the same language (HL7) and use the save vocabulary. The HL7 Bridge will not read or modify messages, message structure, or message vocabulary. This simplicity allows the HL7 Bridge to focus exclusively on transporting messages.

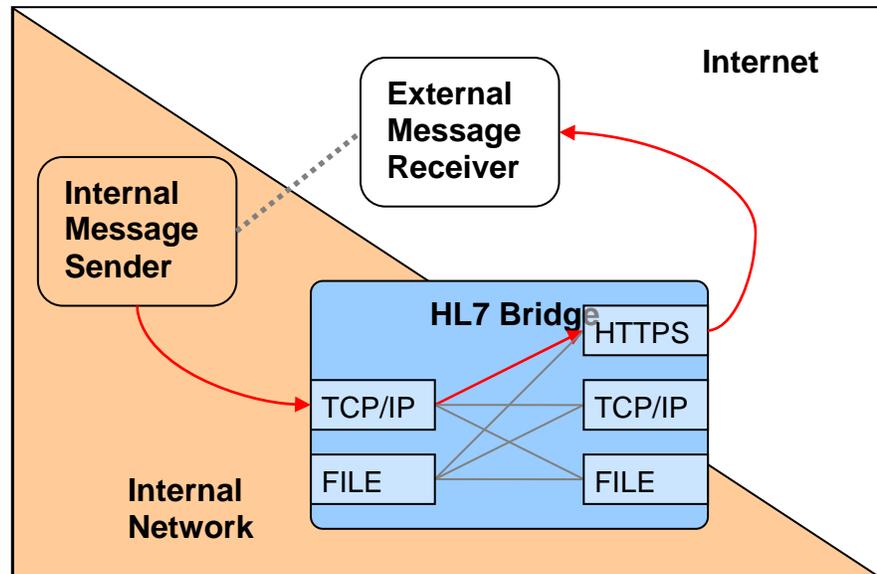
SENDERS, LISTENERS, AND CONNECTIONS

The HL7 Bridge has three (3) sending interfaces and two (2) listening interfaces. When a sender is paired with a listener, this is called a connection. The HL7 Bridge may be configured with multiple connections each one representing a different set of sending and receiving systems that wish to speak to each other. For this reason, the HL7 Bridge should be thought of as a point-to-point connector and not as a general messaging router.

The diagram below shows how the HL7 Bridge may be used to connect an Internal Message Sender using its own internal TCP/IP protocol directly to an External Message Receiver that uses secure HTTPS. The shaded connecting lines within the HL7 Bridge show all the other connections available.



Figure 1-1: Connection Layout



A connection is defined by a simple text file that is saved in the HL7 Bridge configuration directory. The HL7 Bridge reads this directory when starting up and creates a connection for each file. There are six (6) different kinds of connections (2 senders X 3 listeners) that can be configured.

The HL7 Bridge is a synchronous connector in that it makes the originating system wait until the HL7 Bridge has contacted the receiving system and returned a response. If the HL7 Bridge is unable to connect to the receiving system, it will report the problem back to the originating system in the form of a negative acknowledgment (ACK) message. The HL7 Bridge will not attempt to resend.

TCP/IP LISTENER AND SENDER

The HL7 Bridge supports a TCP/IP standard called Minimal Lower Layer protocol (MLLP or MLP) defined by HL7 in a document titled, "Transport Specification: MLLP, Release 1. (Author Rene Spronk. © 2003)."

According to this document:

"The MLLP protocol is a minimalistic OSI -session layer framing protocol. It is assumed that the MLLP protocol will be used only in a network environment. Most of the details of error detection and correction are handled by the lower levels of any reasonable



network protocol (e.g., TCP/ IP, SNA) and do not require any supplementation.”

“HL7 content is enclosed by special characters to form a block. The block format is as follows: < SB> dddd< EB> < CR>.”

The document further defines <SB> as ASCII <VT> or <0x0B>, dddd as the data being sent, <EB> as ASCII <FS> or <0x1C>, and <CR> as <0x0D>.

MLLP is widely used by HL7 systems even if not specifically called MLLP. For more information please see the document, “HL7 Implementation Guide version 2.3.1, Appendix C “Lower Layer Protocols,” section C.4.3.”

FILE LISTENER AND SENDER

The HL7 Bridge may be configured to “read from” and “write to” file directories on the local system.

If configured to listen for new requests, the HL7 Bridge must have two directories specified: one to hold the request files, and the other to hold the resulting responses. The HL7 Bridge will read the file in the request directory, send it to the associated sender, wait for a response, and write the response with the same file name into the response directory.

The HL7 Bridge reads both directories at regular intervals and compares them. If a file exists in the request directory and not the response, it sends the request file and writes the response. To resend a request, simply delete the appropriate response file.

If the HL7 Bridge is unable to send the request, it will write a standard HL7 negative acknowledgement (ACK) explaining the problem. The HL7 Bridge will not automatically resend the request at a later time. To resend, delete the file with the negative acknowledgement.

If configured as a sender, the HL7 Bridge will save requests received from the associated listener to a specified directory. Each request session will generate a new file. The HL7 Bridge will then report back to the sender with a generic message (ACK) indicating the success, or if not configured correctly, the inability to see the message.



HTTPS SENDER

The HTTPS sender follows a standard called “Secure Transport” which was defined by a group called the “Committee on Immunization Registry Standards for Electronic Transactions (CIRSET)” and can be found at <http://www.cirset.org>.

This standard defines a secure method of submitting HL7 messages via HTTPS. Four parameters (USERID, PASSWORD, FACILITYID, and MESSAGEDATA) are submitted as an HTTPS POST operation and the web server responds with an HL7 message. For more information please see the CIRSET document.



[This page intentionally left blank.]



2 SYSTEM REQUIREMENTS, INSTALLATION, & CONFIGURATION

This section lists the System Requirements and the Installation and Configuration steps.

SYSTEM REQUIREMENTS

The following requirements are necessary for the HL7 Bridge:

1. Modern computer server or workstation with Windows, LINUX/UNIX, or other modern Operating System (OS) installed.
2. Latest Java 1.4 JRE installed.
3. Computer must be connected via intranet and internet to the appropriate systems that wish to use the HL7 Bridge.

INSTALLATION & CONFIGURATION

This section includes installation and configuration instructions for various operating systems.

WINDOWS 2000/XP

Perform the following steps for the Windows 2000/XP Operating Systems.

1. Save the “hl7bridge_install.zip” file sent via email on the desktop.
2. Create a directory on the desktop called “bridge”
3. Unzip the “hl7bridge_install.zip” file in the “bridge” directory
4. There will be four directories in the folder:
 - Aix
 - Nt
 - Linux
 - Solaris



5. Create the following path:

```
C:\Program Files\Scientific Technologies Corporation\HL7 Connection Bridge
```
6. Create the following directories in the C: drive or the root directory:

```
C:\hl7bridge\requests
```

```
C:\hl7bridge\responses
```
7. Copy the content of the “nt” directory from the unzipped “bridge” directory on the desktop, and paste the content of the “nt” directory in the “HL7 Connection Bridge” folder located at

```
C:\Program Files\Scientific Technologies Corporation\HL7 Connection Bridge
```

The “HL7 Connection Bridge” directory should now have four folders (“bin,” “lib,” “logs,” and “Props”) and two files (“hl7brige.config,” and “readme.html.”).
8. Navigate to the path below:

```
C:\Program Files\Scientific Technologies Corporation\HL7 Connection Bridge\props
```
9. Open the CHIRP_Profile_file.properties and make sure the “listener.request.dir” and the “listener.response.dir” paths are correct and make sure forward slashes (/) are used in the path names. Make sure the path reflects those shown in Step 6 above. If a file is being sent, make sure the “sender.send-to.http-address” is pointing to the correct server; in other words, make sure the URL is correct.
10. Navigate to the path below:

```
C:\Program Files\Scientific Technologies Corporation\HL7 Connection Bridge\nt\bin
```
11. Double click the “install-service.bat” from the “bin” directory. This will install the HL7 Bridge Service
12. Navigate to the “Control Panel” and select the following:
 - Performance and Maintenance



- Administrative Tools
 - Services
13. Double-click Services and start the newly installed “HL7 Bridge Service.”
- Note:** If the Service does not start, refer to the section titled, “Troubleshooting” for instructions.
14. The installation is done. The bridge can now be tested to see if it is working properly.

LINUX AND SOLARIS

Perform the following steps for the Linux and Solaris Operating Systems:

1. Check Permissions
 - Make sure that the bin/hl7bridge and bin/wrapper files have the +x file permission.
2. Configure the Connection Property File
 - Check the Bridge Connection Properties in the Props directory to ensure they are configured properly for your system.
 - Refer to the section titled, “Configuration Files.”
3. JVM Instructions
 - If using IBM JVM, refer to the section below titled, “IBM JVM Specific Instructions.”
4. Start the Service
 - Start the hl7bridge daemon.
 - `#!/bin/hl7bridge start`
 - Usage: `./hl7bridge { console | start | stop | restart | dump }`

Note: If the Service does not start, refer to the section titled, “Troubleshooting” for instructions.



AIX

Perform the following steps for the AIX Operating Systems:

1. Check Permissions
 - Make sure that the bin/hl7bridge and bin/wrapper files have the +x file permission.
2. Configure the Connection Property File
 - Check the Bridge Connection Properties in the Props directory to ensure they are configured properly for your system.
 - Refer to the section titled, “Configuration Files.”
3. JVM Instructions
 - If using IBM JVM, refer to the next section titled, “IBM JVM Specific Instructions.”
4. Start the Service
 - Start the hl7bridge daemon.
 - #./bin/hl7bridge start
 - Usage: ./hl7bridge { console | start | stop | restart | dump }

Note: If the Service does not start, refer to the section titled, “Troubleshooting” for instructions.

IBM JVM SPECIFIC INSTRUCTIONS

Perform the following steps for the IBM JVM:

1. The HL7 Connection Bridge is tested working for **Java version 1.4.2, J2RE 1.4.2 on IBM AIX.**
2. To use the HL7 Connection Bridge to connect via HTTPS, the **java.security file (/usr/java14/jre/lib/security/java.security)** needs to be modified.
 - There appears to be a less known issue with IBMJSSEProvider for the custom truststore implementation, but IBMJSSEProvider2 solves that issue. The security.provider, ssl.SocketFactory.provider and



ssl.ServerSocketFactory.provider properties in java.security file should be modified as shown below:

```
The list of security provider and their order should be changed as follows:
#
# List of providers and their preference orders (see above):
#
security.provider.1=com.ibm.jsse2.IBMJSSEProvider2
security.provider.2=com.ibm.jsse.IBMJSSEProvider
security.provider.3=com.ibm.crypto.provider.IBMJCE
security.provider.4=com.ibm.security.jgss.IBMJGSSProvider
security.provider.5=com.ibm.security.cert.IBMCertPath

Also, the default SSLSocketFactory provider should be changed to use JSSE2 implementation as follows:
#
# Determines the default SSLSocketFactory and SSLServerSocketFactory
# provider implementations for the javax.net.ssl package.If, due to
# export and/or import regulations, the providers are not allowed to be
# replaced, changing these values will produce non-functional
# SocketFactory or ServerSocketFactory implementations.
#
ssl.SocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl
ssl.ServerSocketFactory.provider=com.ibm.jsse2.SSLServerSocketFactoryImpl
```

CONFIGURATION FILES

The following files may need certain settings changed.

HL7BRIDGE.CONFIG

The HL7Bridge.Config file is located in the root of the HL7Bridge installation directory. It contains two property file settings that you may want to change:

- HL7bridge.log.level – to change the logging level of reporting.
- HL7bridge.cert.keystore – to point to the location of the client certificates for secure site (HTTPS).



HL7BRIDGE.LOG.LEVEL

This setting is used to set the logging level. The logging level is the amount of reporting detail that is happening. The default setting is “standard.” Other values include:

- Verbose
- Minimal
- Never

Note: After changing the logging level, stop and restart the "HL7 Bridge for STC IWeb" service in the services control panel.

HL7BRIDGE.CERT.KEYSTORE

This setting is used to set the keystore location containing the client certificates for secure sites (HTTPS).

You also need to set the value of the hl7bridge.cert.keystore property found in the hl7bridge.config file that is located in the root of the HL7Bridge installation directory.

This property points to the location of the keystore that contains the client certificates for secure sites (HTTPS). Therefore, in order to do transmission of an HL7 file to a URL using the HTTPS protocol, that site's certificate must be included in this keystore.

Note: ELR transmission will fail if a client's certificate expires. At that time, a new certificate for that site must be downloaded and inserted into the keystore.

Instructions are located in the section titled, “HTTPS Support.”

PROPERTIES FILES

The HL7 Bridge is capable of handling multiple connection types. Each connection is defined as a listener and a sender. The listener waits for incoming messages and gives them to the sender to handle. When the sender gets a response, it replies back to the listener so the listener can reply back to the original sender. A connection is defined by a **.properties** file and should be placed in the **/props** directory.



The **/props** directory contains seven example connection configuration files named according to the configuration. These files show many possible combinations of listener and sender but may be disabled. All of the properties are listed in the table, followed by sample configuration files.

Table 2-1: Connection Configuration Properties

Property	Example	Format	Explanation
listener.title	Test Link	Text	A human readable title for this connection link
listener.enabled	true	true false	When set to true this listener will listen for requests
listener.type	TCP/IP	TCP/IP FILE	Specifies the type of listener
listener.file.ext	.hl7, .txt	comma separated extensions	For listeners of type FILE. Comma separated list of file extensions to be processed (case insensitive). If no extensions are specified, all file extensions will be processed.
listener.listen-on-port	4353	Number	The port number to start listening on for requests (TCP/IP only)
listener.accept-from	*	* ip-address	Indicates which computer can connect to this link, use * for all (TCP/IP only)
listener.request.dir	C:/requests	Directory	The directory where the listener should look for new HL7 files, use forward slashes (FILE only)
listener.response.dir	C:/responses	Directory	The directory where the listener should put the response messages, use forward slashes (FILE only)
listener.check_interval_in_seconds	15	Number	The number of seconds the Listener should wait between checks for new HL7 files (FILE only)
sender.type	HTTP	HTTP FILE TCP/IP	How the message received should be forwarded
sender.send-to.http-address	http://ir.state.us/HL7Server	HTTP URL	The HTTP URL of the real-time HL7 server (HTTP, HTTP_RAW only)
sender.account.username	jdoe	String	The username supplied by the system administrator (HTTP only)
sender.account.password	jdoe	String	The password supplied by the system administrator (HTTP only)
sender.destination.dir	C:/new requests/	Directory	The directory where the incoming hl7 should be saved (FILE only)



Property	Example	Format	Explanation
sender.send-to.address	10.0.1.0	IP Address Machine Name	The address of the machine ready to receive HL7 messages (TCP/IP only)
sender.send-to.port	4353	Number	The port number to send the message to (TCP/IP only)
sender.send.filename	false	true false	When set to true, FILENAME parameter will be included in posts to HTTP senders. Only applicable when using a FILE listener.
sender.https.verifyhostname	false	true false	Pertains only to connections using Https protocol. When set to true, will only allow a connection to a website whose hostname matches the hostname on the client installed certificate (in keystore). Even valid hosts may not have their hostname set correctly on their certificate so it is best to keep this at its default value of false if making Https connections.

Depending on the connection you are setting up, refer to one of the configuration files below. If you are trying to set up a TCP/IP to HTTPS connection, use the configuration file shown in Figure 2-6: TCPIP-to-HTTP Properties below and change the properties appropriately.

Figure 2-1: File-to-File Properties

```
#-----
# LISTENER
#   The listener listens for incoming HL7 messages
#-----
listener.title=Test Link A
listener.enabled=false
listener.type=FILE
listener.request.dir=C:/temp/hl7bridge/A/requests
listener.response.dir=C:/temp/hl7bridge/A/responses
listener.check_interval_in_seconds=15

#-----
# SENDER
#   The sender sends the HL7 message on to its destination
#-----
sender.type=FILE
sender.destination.dir=C:/temp/hl7bridge/B/requests
```



Figure 2-2: File-to-HTTP Properties

```
#-----  
# LISTENER  
#   The listener listens for incoming HL7 messages  
#-----  
listener.title=Test Link G  
listener.enabled=false  
listener.type=FILE  
listener.request.dir=C:/temp/hl7bridge/G/requests  
listener.response.dir=C:/temp/hl7bridge/G/responses  
listener.check_interval_in_seconds=15  
  
#-----  
# SENDER  
#   The sender sends the HL7 message on to its destination  
#-----  
sender.type=HTTP  
sender.send-to.http-  
address=http://nathan.phx.stchome.com:8081/iweb_test/HL7Server  
sender.account.username=johnny  
sender.account.password=johnny
```

Figure 2-3: File-to-HTTPS Properties

```
#-----  
# LISTENER  
#   The listener listens for incoming HL7 messages  
#-----  
listener.title=Test Link H  
listener.enabled=false  
listener.type=FILE  
listener.request.dir=C:/temp/hl7bridge/H/requests  
listener.response.dir=C:/temp/hl7bridge/H/responses  
listener.check_interval_in_seconds=15  
  
#-----  
# SENDER  
#   The sender sends the HL7 message on to its destination  
#-----  
# HTTP is type to use for HTTPS or HTTP protocol  
sender.type=HTTP  
sender.send-to.http-address=https://dreams.stchome.com/dreams/hl7login  
sender.account.username=superuser  
sender.account.password=foopassword  
sender.send.filename=false  
sender.https.verifyhostname=false
```



Figure 2-4: File-to-TCPIP Properties

```
#-----  
# LISTENER  
#   The listener listens for incoming HL7 messages  
#-----  
listener.title=D Test Link  
listener.enabled=false  
listener.type=FILE  
listener.request.dir=C:/temp/hl7bridge/D/requests  
listener.response.dir=C:/temp/hl7bridge/D/responses  
listener.check_interval_in_seconds=15  
  
#-----  
# SENDER  
#   The sender sends the HL7 message on to its destination  
#-----  
sender.type=TCP/IP  
sender.send-to.address=127.0.0.1  
sender.send-to.port=4352
```

Figure 2-5: TCPIP-to-File Properties

```
#-----  
# LISTENER  
#   The listener listens for incoming HL7 messages  
#   read-timeout is defined in seconds (0 to disable)  
#-----  
listener.title=Test Link C  
listener.enabled=false  
listener.type=TCP/IP  
listener.listen-on-port=4351  
listener.accept-from=*  
listener.read-timeout=0  
  
#-----  
# SENDER  
#   The sender sends the HL7 message on to its destination  
#-----  
sender.type=FILE  
sender.destination.dir=C:/temp/hl7bridge/D/requests
```



Figure 2-6: TCPIP-to-HTTP Properties

```
#-----  
# LISTENER  
#   The listener listens for incoming HL7 messages  
#   read-timeout is defined in seconds (0 to disable)  
#-----  
listener.title=F Test Link  
listener.enabled=false  
listener.type=TCP/IP  
listener.listen-on-port=4353  
listener.accept-from=*  
listener.read-timeout=0  
  
#-----  
# SENDER  
#   The sender sends the HL7 message on to its destination  
#-----  
sender.type=HTTP  
sender.send-to.http-  
address=http://nathan.phx.stchome.com:8081/iweb_test/HL7Server  
sender.account.username=johnny  
sender.account.password=johnny
```

Figure 2-7: TCPIP-to-TCPIP Properties

```
#-----  
# LISTENER  
#   The listener listens for incoming HL7 messages  
#   read-timeout is defined in seconds (0 to disable)  
#-----  
listener.title=E Test Link  
listener.enabled=false  
listener.type=TCP/IP  
listener.listen-on-port=4352  
listener.accept-from=*  
listener.read-timeout=0  
  
#-----  
# SENDER  
#   The sender sends the HL7 message on to its destination  
#-----  
sender.type=TCP/IP  
sender.send-to.address=127.0.0.1  
sender.send-to.port=4353
```



HTTP/HTTPS SUPPORT

The HL7Bridge supports **HTTP** and **HTTPS** connections transparently. (HTTP sender type is used for both protocols). If a target URL contains the HTTPS protocol, the client certificate for that website will first have to be added to the Java Keystore (keystore/cacerts) that contains all trusted sites and certificates for sites that communicate over SSL. Steps to follow on how to obtain a certificate and how to add that certificate into the Bridge's keystore are listed below.

Note: If you already have a client .P12 or .PFX cert, install it into your browser (right-click, choose Install)

1. Connect to the desired secure site using Internet Explorer.
2. Double-click the **lock** icon in the bottom right-hand side of IE.
3. Click the **Certification Path** tab.
4. Select the desired client certificate.
5. Click the **View Certificate** button. This will pop up a new Certificate dialog.
6. Click on the **Details** tab on the new Certificate Dialog.
7. Click the **Copy to File** button.
8. Export the certificate as **Base64** encoded **X.509** to a file on your local machine.
9. Import the **X.509** certificate file into the keystore using keytool and the **trustcacerts** option. For example:

```
keytool -import -alias sun40key -file  
c:\cert_stuff\michigan\sun40.cer -trustcacerts -keystore  
c:\hl7bridge\keystore\cacerts -v
```

Note: "keytool" is a java tool that comes with the java JDK. Java version 1.4.2 is required for transparent SSL as described in this document.

10. The default "keystore" password is: changeit
11. To verify that your certificate was added successfully to the key store, issue this keytool command to list all trusted certs in the keystore:



```
keytool -list -keystore c:\hl7bridge\keystore\cacerts
```

TROUBLESHOOTING

If the “Service” fails to start, the problem is recorded in the error file, “<Program Installation Folder>/logs/wrapper.log. If there is a problem listening or handling a connection, it will be noted in the *-err.txt file of the same name as the connection’s property file.

Make sure the receiving system is up and running. If the messages are to be forwarded via HTTP to STC’s IWeb application, copy the URL from the property file and using a browser, make sure you can see the HL7 Service from the same machine that the HL7 Bridge is installed.

[This page intentionally left blank.]