HL7Kit Pro

User's Manual, version 1.4

HL7 KIT IS AN INTEGRATION ENGINE SPECIALLY DESIGNED
FOR BUSY HEALTHCARE IT PROFESIONALS.

You are 20 minutes away from HL7 integration!

RZ Software Services
2/1/2011

RZ – Software Services
19/8 Yishaayahu St.
Tel-Aviv 62494, Israel

www.hl7kit.com
info@roniza.com

HL7Kit User's Manual

RZ – Software Services
19/8 Yishaayahu St.
Tel-Aviv 62494, Israel

www.hl7kit.com
info@roniza.com

Overview

HL7KitPro is an interface engine and interface design tool for HL7 messaging. Use HL7KitPro to define and implement HL7 interfaces with no programming, simply using drag-and-drop to map the attributes.

HL7Kit Pro includes four applications

- HL7Sender is a HL7 message editor that can send messages
- HL7Receiver is a GUI HL7 Listener
- HL7Mapper is a GUI interface design tool
- HL7Service is a background process HL7 listener

Features

Feature	HL7FreeSender	HL7Kit	HL7Kit Pro
Edit HL7 Messages	Yes	Yes	Yes
Convert HL7 to XML	Yes	Yes	Yes
Validate Message Structure	Yes	Yes	Yes
Send Messages over TCP/IP	Yes	Yes	Yes
Configurable Protocol Definition	Yes	Yes	Yes
Receive Messages over TCP/IP	No	Yes	Yes
Launch custom application	No	Yes	Yes
System Events Full Logging	No	Yes	Yes
Database Integration	No	No	Yes
Dynamic Rules Definition	No	No	Yes
GUI Interface Definition Tool	No	No	Yes
Hot Folder Integration[1]	No	No	Yes
Multiple destinations[2]	No	No	Yes

Applications

The HL7Kit products include the following applications:

	HL7FreeSender	HL7Kit	HL7KitPro	HL7Kit Runtime
HL7sender	+	+	+	
HL7Reciever		+	+	
HL7Mapper			+	
HL7Runtime			+	+

[1] New in version 1.3
[2] New in version 1.4
[3] New in version 1.4 – In order to integrate with other database engines, like MySQL for example, it is possible to define

Installation Instructions

System Requirements

HL7Kit is tested on the following Operating Systems:

- Windows XP
- Windows 2003 Server
- Windows 2003 Server x64
- Windows 2008 Server
- Windows 2008 Server x64
- Windows Vista
- Windows Vista x64
- Windows 7

.NET Framework 2.0 or later

Database Engines[3]

- Microsoft SQL Server 2005
- Microsoft SQL Server 2005 Express Edition
- Microsoft SQL Server 2008
- Microsoft SQL Server 2008 Express Edition

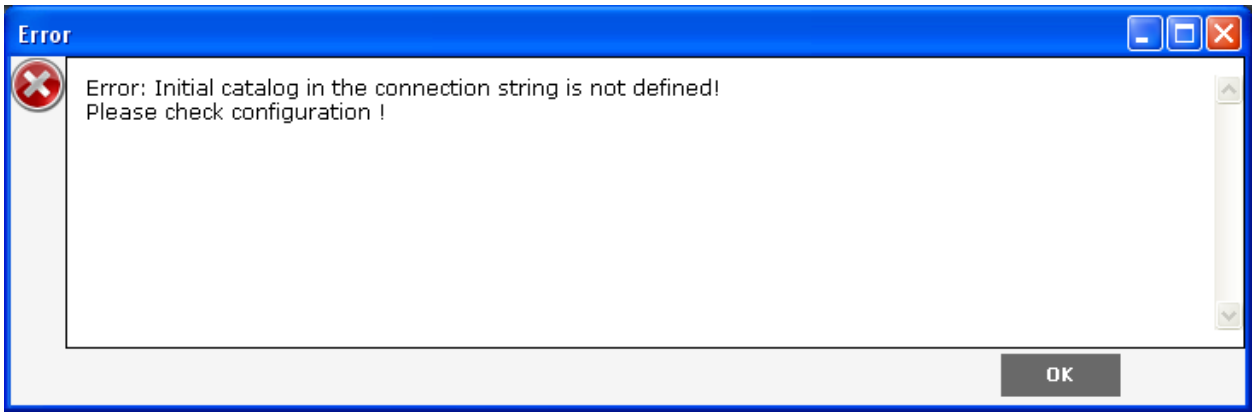
Installing HL7Kit Pro

Double Click Setup.exe and follow the instructions.

After installation is complete, start the HL7Mapper application.

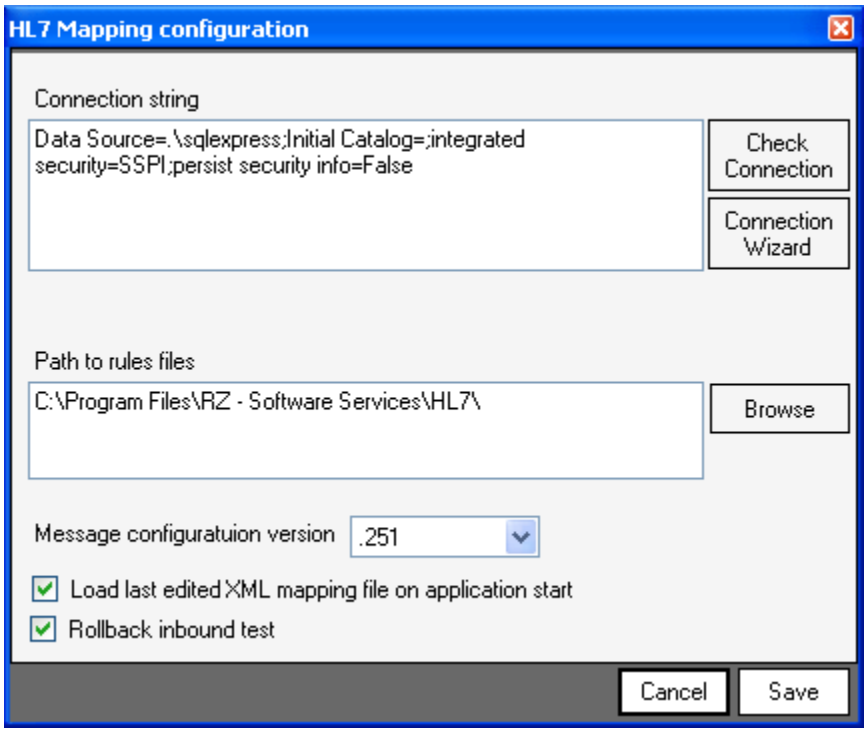
When the HL7Mapper program first runs, you'll probably be prompted to configure the database connection as in the following screenshot.

[3] In order to integrate with other database engines, like MySQL for example, it is possible to define linked tables between SQL Express and the application database.



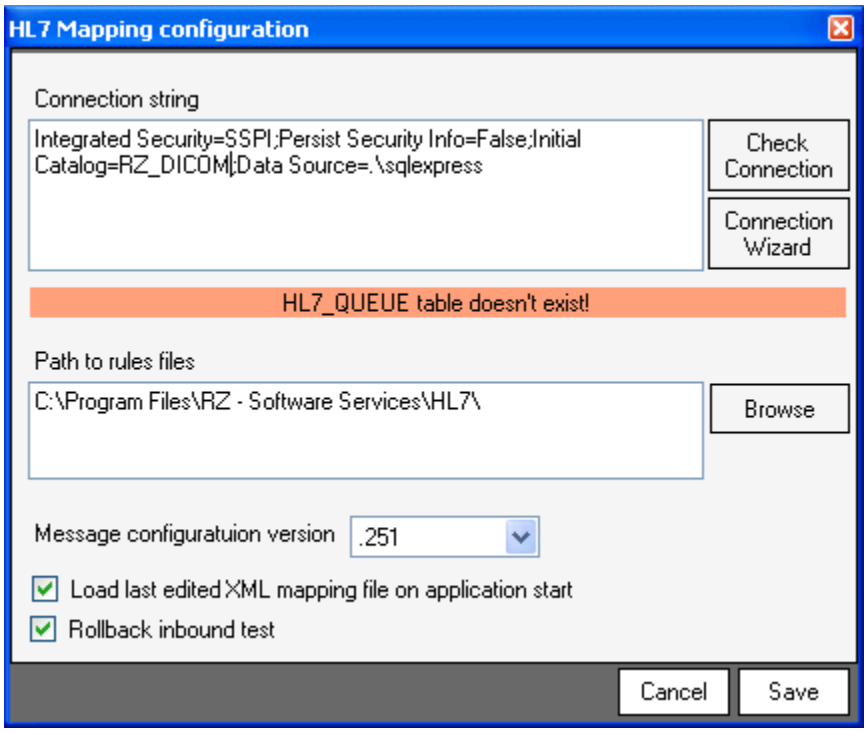
Click OK.

After clicking OK, the application configuration screen is displayed.

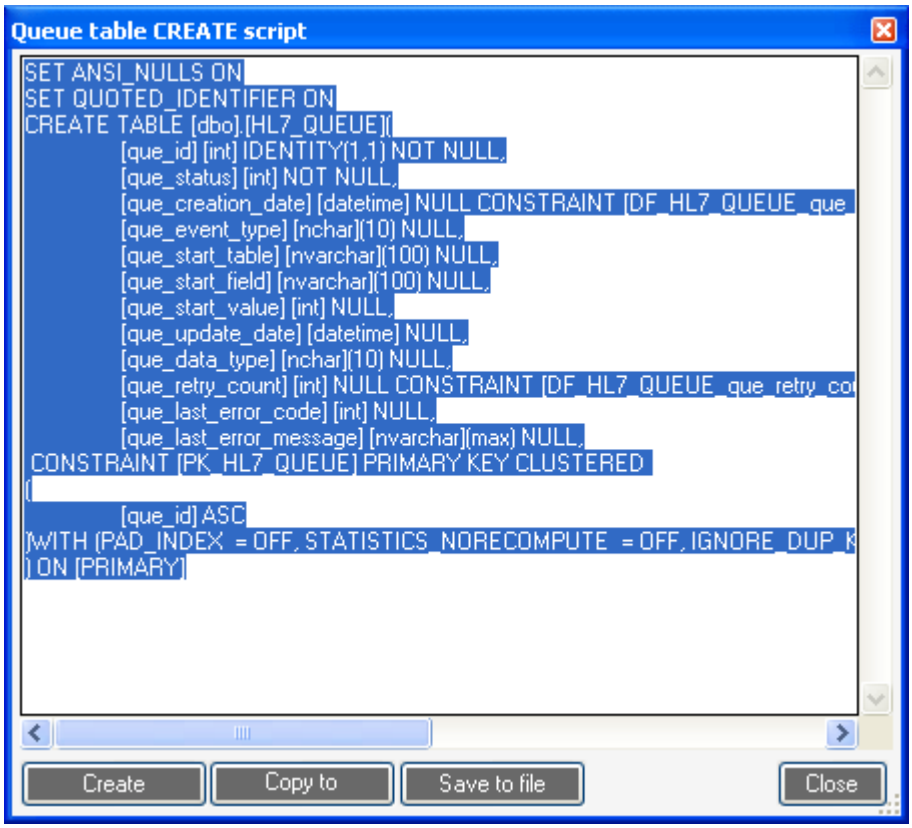


Use the connection string text box to configure the database connection or click the connection wizard to use the UML utility.

Once the connection string is set properly, the application will attempt to find the HL7_QUEUE table that is used for sending outbound messages.



Click the salmon colored stripe to create the table.



Either click ‘create’ or copy the code and run it manually.

Installing HL7 Runtime Service

Double click Setup.exe and follow the instructions.

After installation, the HL7 Runtime Service is configured to start automatically with your system but is not yet running.

To start the service without restart, do one of the following:

1. Using the HL7 Service configuration tray icon 

    a. Double click the tray icon
    b. Click the start button at the top left.

    OR

2. Start the HL7Service from the services control panel
    a. From the start menu select run
    b. Type in services.msc
    c. Find HL7 Runtime Service
    d. Start it

    Or

3. Start HL7Service from command line
    a. From the start menu select run
    b. Type cmd
    c. In the command window type:
    net start HL7Service

Installing HL7Kit

Double click the setup application and follow the instructions.

Installing HL7FreeSender

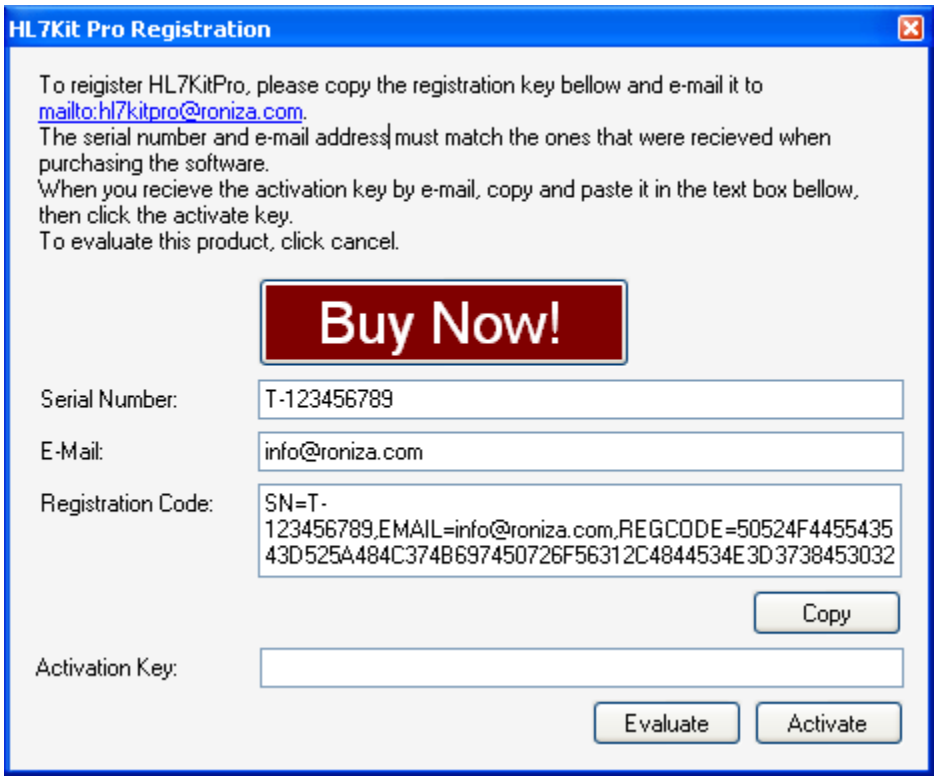
Double click the setup application and follow the instructions.

Activation

HL7KitPro and HL7Kit are protected using activation keys. The evaluation copies are limited by the number of messages that they can process.

Activating the product removes the evaluation copy limitations.

The registration form is displayed when the HL7Mapping application or HL7Receiver are started.



To activate your product:

1. Key in the serial number (or transaction id) and e-mail address from your receipt.
2. Click the Copy button.
3. Paste the copied text into the body of an e-mail message and e-mail it to hl7kitpro@roniza.com
4. We will e-mail you back with an activation key.
5. Copy the activation key and paste it into the Activation key text box.
6. Click Activate.

Evaluation

To evaluate HL7Kit and HL7Kit Pro, simply click the Evaluate button in the registration form.

Applications

HL7Sender

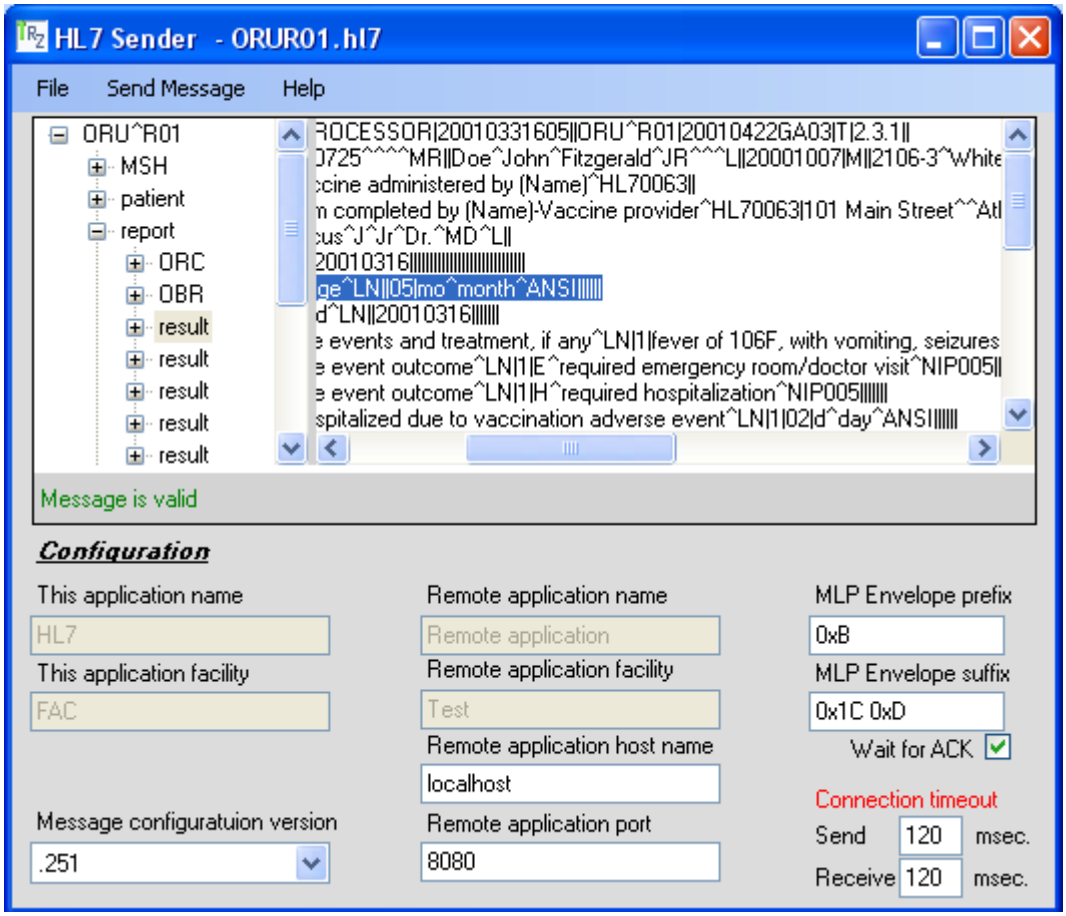


Figure 1: HL7Sender Main Application Form

HL7Sender is a utility for sending HL7 messages through the network. It is a single form application can read, write, validate, convert to XML and send HL7 messages.
The upper part of the form shows the message in a rich text control where it can be edited. Every HL7 segment starts in a new line. The best way to edit the message is to cut and paste segments from other messages (you can open multiple instances of the program) or fill text between two | signs (field separator).
At the lower part of the form there are a few text fields where the user can set the LLP network protocol elements, the TCP/IP protocol attributes (destination host name or IP address, port number and timeouts) and disable or enable ACK's.
The two message structure text configuration files (segDefs and msgRules) are located in the installation folder and can be edited manually if necessary. You can duplicate these files with different suffixes in order to use multiple 'flavors' of the HL7 version.

Viewing Message Content

From the File menu select Open HL7 File and select a file to open.

The message content is shown on the right panel.

When the message structure is valid, a green message is shown on the upper right side of the screen.

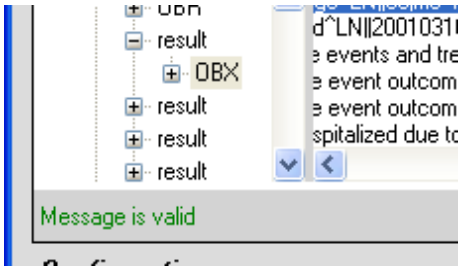


Figure 2: Valid message indicator

If the message is not valid, a red message is shown on the upper right side of the screen describing the problem.

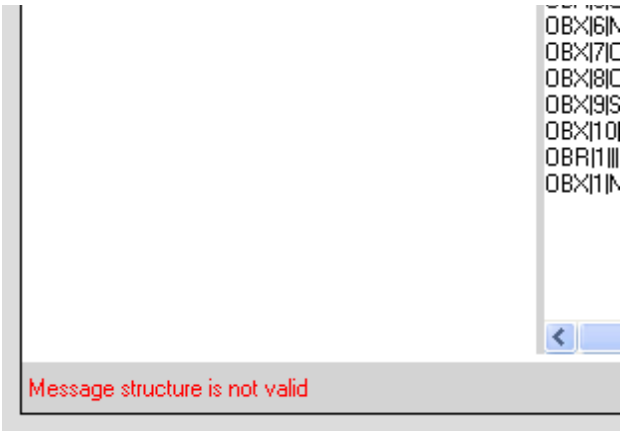


Figure 3: Invalid Message Indicator

In the above example, a wrong segment name is present.

To fix that, simply edit the message content and remove the offending segment.

Sending Messages

Use HL7Sender to send messages stored in HL7 Files. Open a file and the click the 'Send Message' menu button.

Editing Messages

Open a file and edit its content in the HL7 Message text box.

Use the message structure left panel to navigate through the message segments and fields.

Modifying Fields' Values

In the message structure left panel click on the field name.

The field text is highlighted in the message text panel.

Use the keyboard to change the field value.

Adding Segments

In the message structure click on the segment you would like to insert a segment before.

Click <Home>.

Click <Enter>.

Key in the 3 letters segment name followed by the | sign.

The new segment is added to the message structure panel.

Convert HL7 to XML

From the File menu select Save As.

Choose XML from the "save as type" combo box.

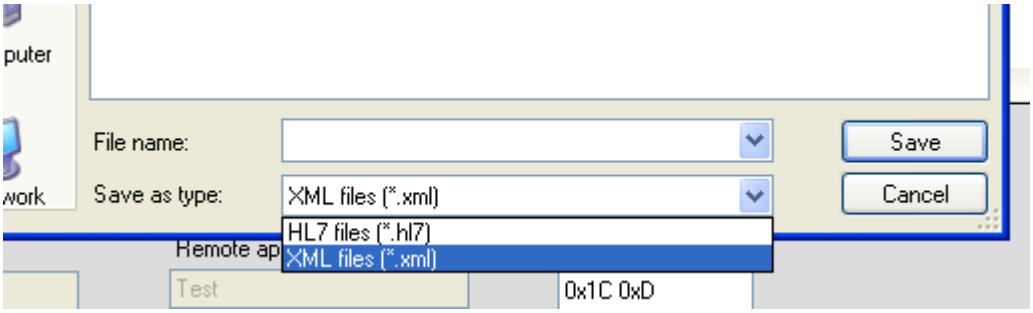


Figure 4: Save as XML

HL7Sender main form

Hl7Sender Menu

The File Menu

Open HL7 File

Browse and select an HL7 file. The file content is loaded to the HL7 Message text box.

Save to HL7 File

Save the context of the HL7 Message text box to a HL7 file or XML file.

The Send Message Menu Button

Clicking the Send Message menu button sends the content of the HL7 Message text box through the network to the remote application.

The Help Menu

About HL7 Sender

This item displays the about form.

HL7 Message Structure Tree

The HL7 Message Structure tree control displays the message structure as a tree. Each logical part of the message is represented as node holding a collection of segment nodes. Each segment node is a collection of fields. The node's names are the field name.

When clicking on a message structure node, the corresponding message text is selected in the HL7 Message text box.

HL7 Message text box

The HL7 Message text box is the main editing area of the HL7Sender. It displays the HL7 Message and enables editing of its content. Every segment is displayed in a new line. When editing a message be careful not to break segments.

The configuration Area

Message Configuration Version

Use this drop down to select the HL7 version to work with. See creating new configurations for explanation how to add HL7 versions to this drop down.

Remote Application Host Name

Use this text box to set the host name or the IP address of the computer that you want to send the message to.

Remote Application Port

Use this text box to set the port number that the application that you want to send the message to listens on.

MLP Envelope Prefix

Use this text box to set the LLP prefix using Hexadecimal notation. See LLP for more information.

MLP Envelope Suffix

Use this text box to set the LLP suffix using Hexadecimal notation. See LLP for more information.

Connection Timeout

Send and receive network timeouts in milliseconds. When HL7 Sender performs network activities and there's no response from the peer for the specified period, the activity is stopped and an error message is displayed.

HL7Reciever

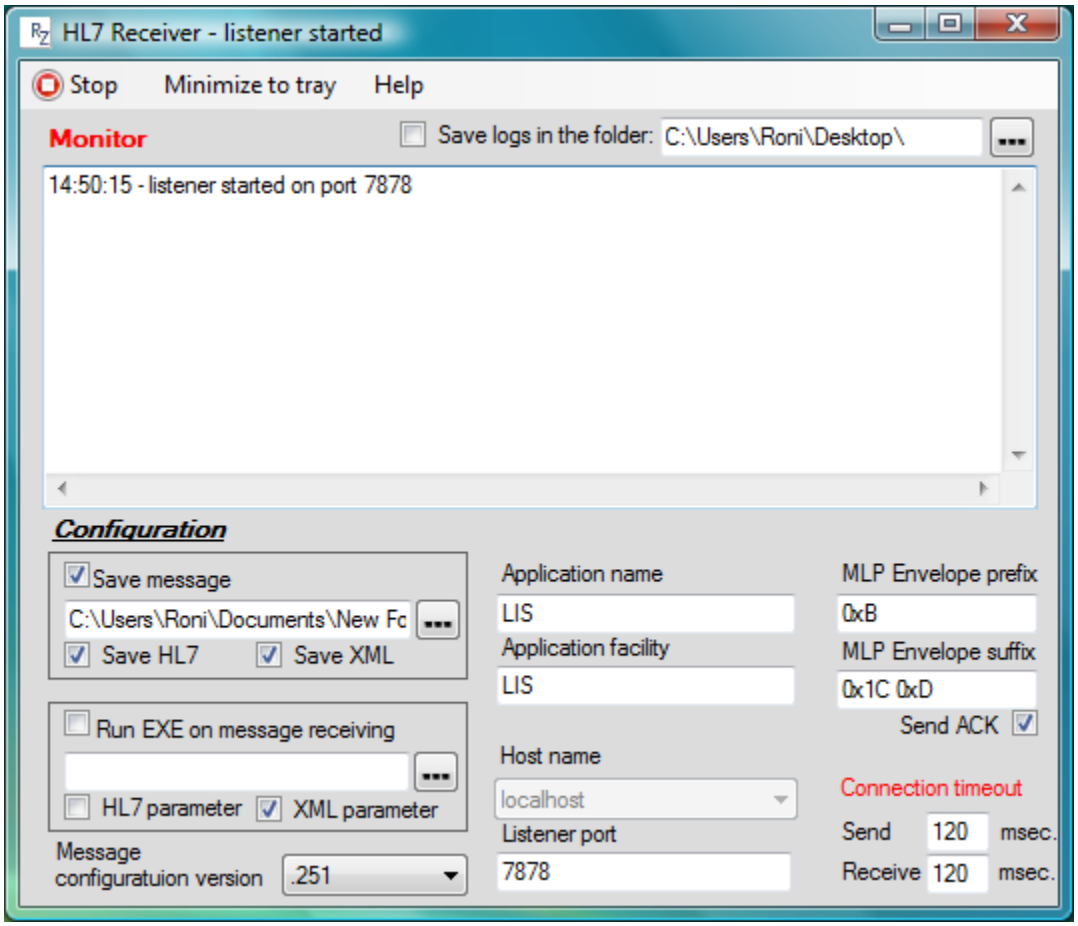


Figure 5: HL7Receiver Main Application Form

HL7Receiver is a utility for receiving HL7 messages through the network (AKA HL7 Listener). The receiver listens on a network port and waits for incoming connections. Every incoming message is validated and can be saved on the local hard disk. A configurable application or script can be launched after receiving a message. The messages can be stored either as they are in HL7 or as an XML file.

HL7Receiver Main Form

HL7 Receiver Menu

Stop/Start

Click this button to change the HL7 network listener state. By default, the listener is active and messages are accepted.

Event Log

HL7Reciever creates an event log that can be used for monitoring and trouble shooting. To view the event log:

1. From the start menu select Control Panels/Administrative Tools/Event Viewer
2. Look at RZHL7 log.

The event log is not available in the evaluation copy.

Text Log File

Check the "Save logs in the folder" checkbox to save text log files in a specific folder.

The text log is not available in the evaluation copy.

XML's

When the save XML checkbox is checked, HL7Reciever creates an XML file for every valid inbound message. The XML file structure is determined according to the message rules definition in the configuration files.

The XML file creation is not available in the evaluation copy. To view the XML structure use the HL7Sender save as XML option.

HL7Mapper

Overview

The HL7 Mapping Application is an authoring tool for the HL7Service mapping rules. The rules are stored in a XML file that is used by the HL7Service and defines how to process messages.

Configuration

At the first run, a configuration form is displayed. The mapping application must connect to a SQL Server Database before it can be used. Use the connection wizard to configure your database connection.

After configuring the database connection, create the HL7_QUEUE table if it doesn't exist already. Use the

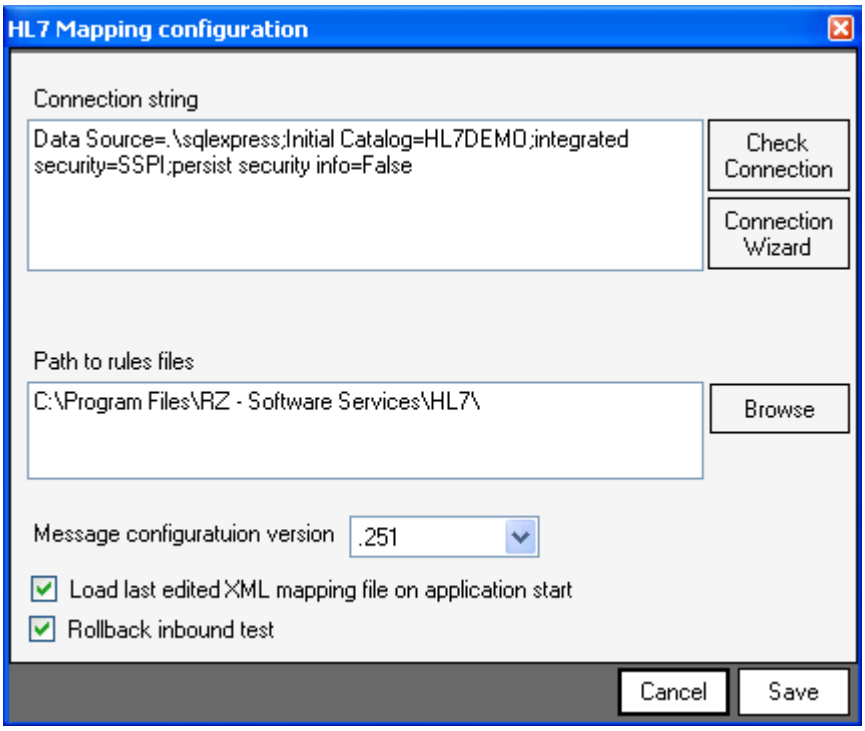


Figure 6: HL7Mapper Configuration Form

Connection String Text Box

The connection string text box shows the database connection string. It can be edited manually or using the connection wizard.

Check Connection Button

The check connection button is used to test the connection string.

Connection Wizard

The connection wizard button displays a connection utility (UDL) that can be used to configure the database connection.

Path to Rule Files Text Box

The path to rule files text box is used to edit the name of the folder in which the message definition files are stored. These files define the HL7 message parsing.

Browse Button

The browse button is used to set the path to rule files.

Message Configuration Version Combo Box

The message configuration Version Combo Box displays that message definition files suffix. If there's more than one set of message definition files in the folder that is selected in path to rules files.

Load Last Edited XML Mapping File …

If this check-box is set the last edited file will be opened when the application starts.

Rollback inbound text check-box

If set, than the data that is inserted when performing an inbound test is rolled-back. Uncheck this check-box to leave the data in the database. This is useful when validating the rules.

Cancel Button

The cancel button closes the configuration form without saving.

Save Button

The save button saves the configuration and then closes the form.

HL7Mapper Main Form

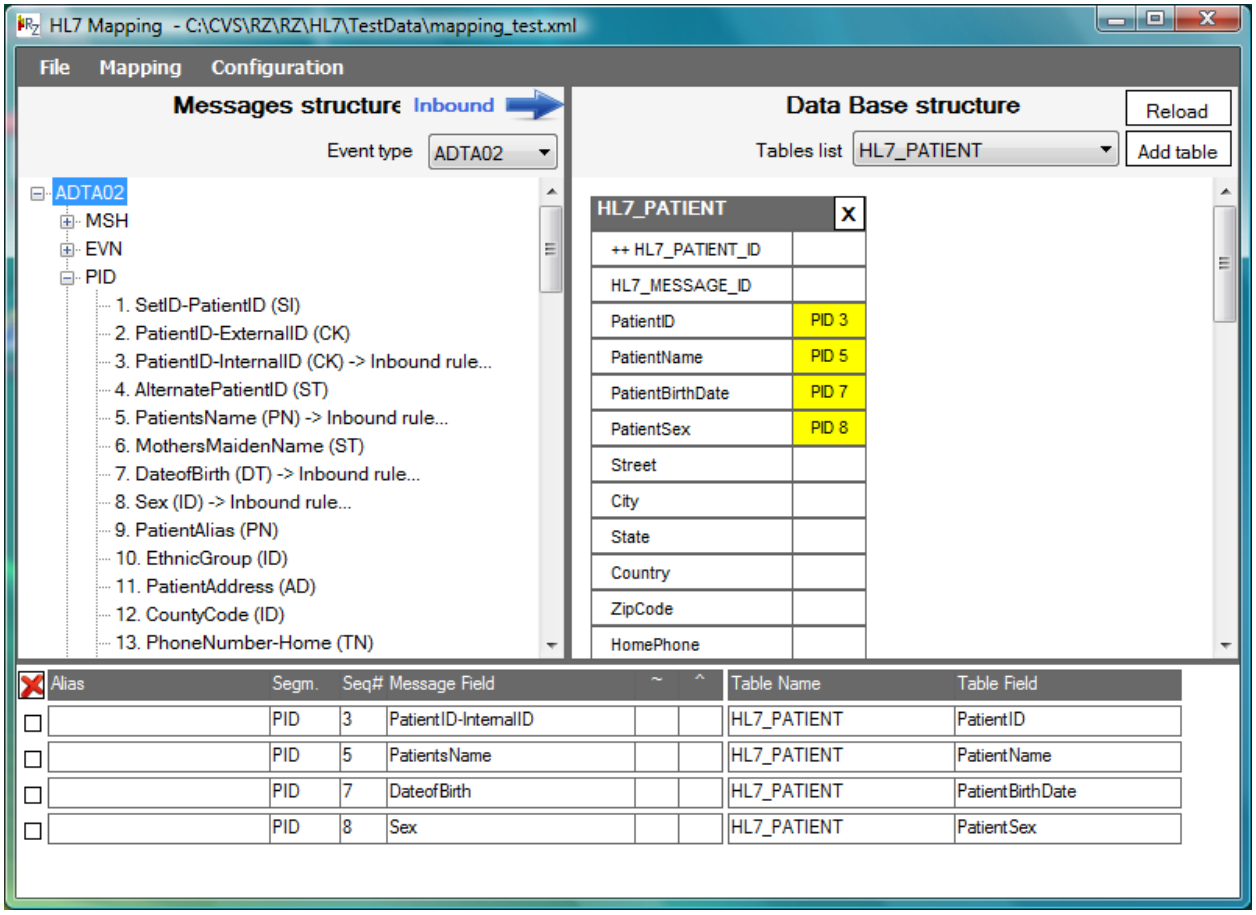


Figure 7: HL7Mapper Main Form

The HL7Mapper Main Form is divided into three panels. The left panel displays the hl7 message structure, the right panel displays database tables and the bottom panel displays the mapping rules between the message and the tables.

Inbound Message Rules

Inbound messages are messages that are sent from an external system to the HL7Kit Runtime Service.

Creating a simple Inbound Rules

1. In the main view, select the event type of the incoming message from the event-type combo-box.
   The message structure is displayed on the left panel.
2. Select a table from the table list combo box and click the 'Add Table' button.
   The table structure is displayed on the right panel.
3. Click a field on the right panel and drag it over a column name in the table.
4. Repeat step 3 for other fields.
5. Hover over a database column name to see it's data type.
6. Hover over a HL7 Message field name to see it's mapping target.
7. Continue mapping all required fields. Make sure all not-null columns are mapped.
8. When you're done mapping choose 'Inbound Test' from the mapping menu.
9. Select a HL7 message with the same event type to test.
10. If the test fails, the SQL error will be shown.
11. Repeat steps 3 – 9 until the test is successful.

Testing Inbound Rules

In order to test an inbound rule:

1. From the mapping menu select 'Inbound Test'
2. Open a message file with the corresponding event-type
3. Review the test result
   If an error occurs the message shows the SQL error.

Creating a Parent-Child Mapping Rule Example

This example explains how to populate database tables with parent-child structure from a HL7 message.

In this example we'll use the ORU^R01 message. This message contains a report with optionally many observations.

We will map the data into two tables with parent-child relation:

1. Reports
2. Observations

HL7Kit identifies tables' relations using foreign-keys. Matching primary and foreign keys are displayed using the same background color in the table layout.

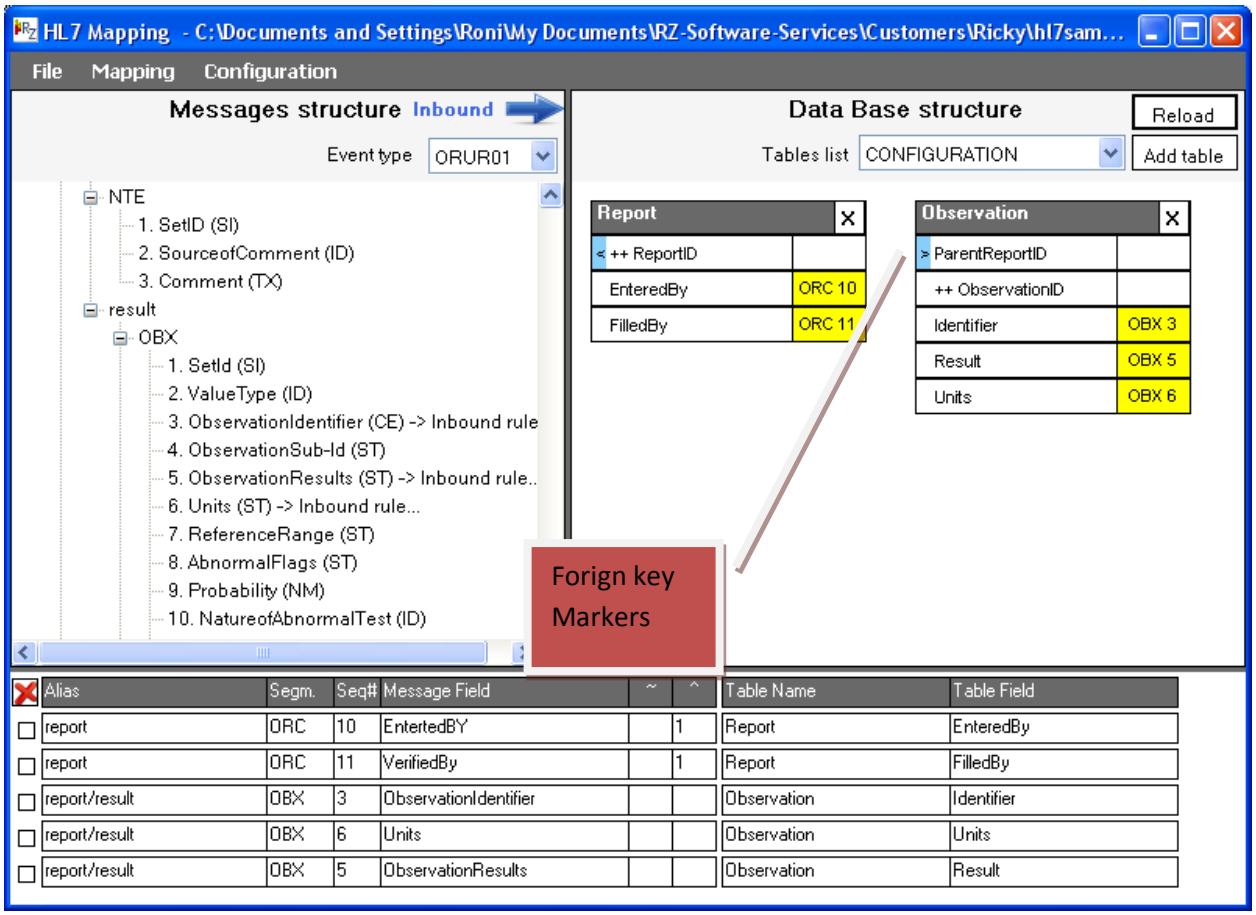


Figure 8: Matching primary (left) and foreign (right) keys marked with light blue

Sub-Components and Repetitions

Sometimes, it is required to set part of a field in one column and another in other column. For example first and last name are divided by a ^ sign in the same field.

To do this, follow this example that demonstrates breaking PID 5 into first and last name:

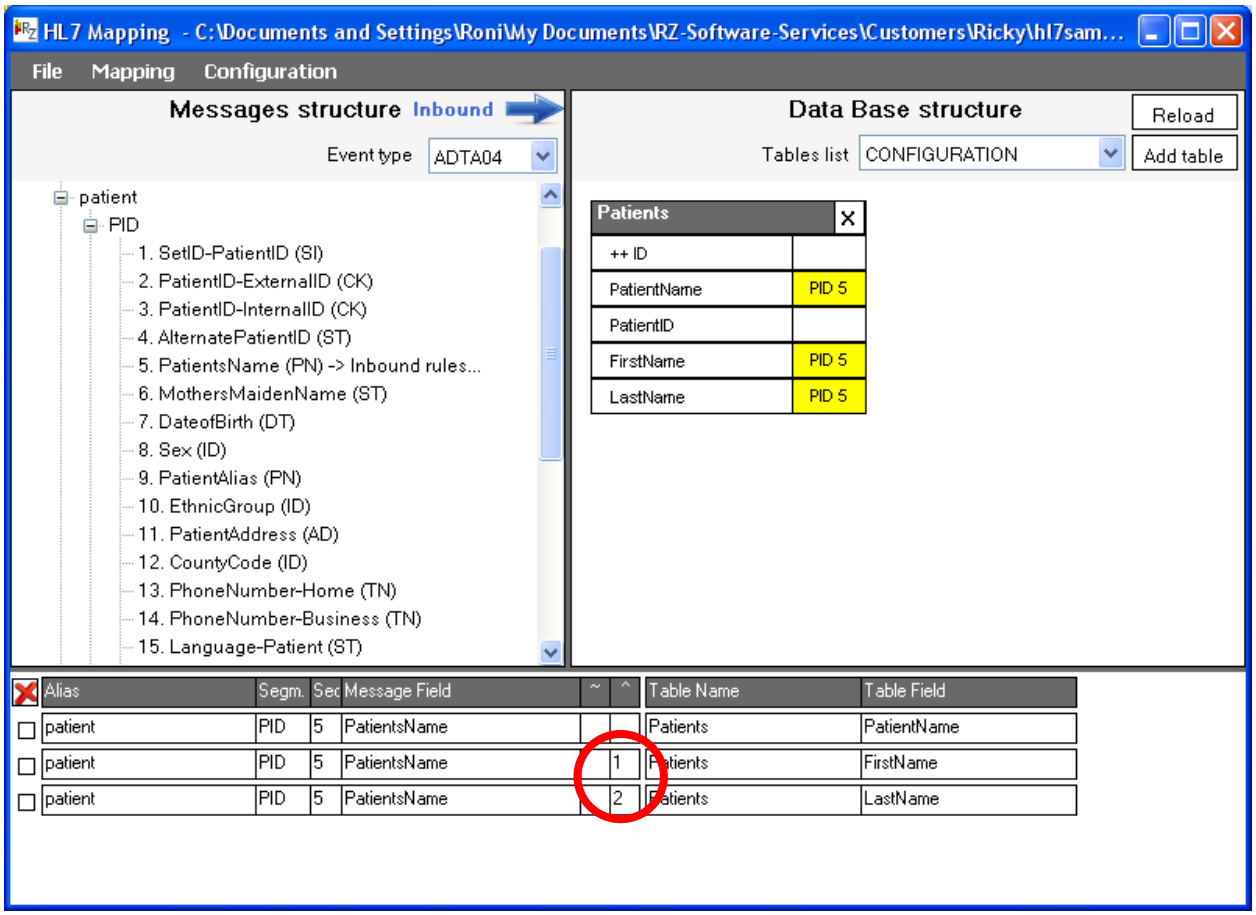


Figure 9: Break Sub-Components into separate fields

1. Before starting, uncheck the rollback test data in the configuration form so that you can see the results in the database
2. Select Event Type ADT^A04
3. Add the patients table to the database panel
4. Drag PID 5 on PatientName field
5. Drag PID 5 this time over first name
6. In the rules panel edit the rules that maps first name and set the sub-component (^) value to 1
7. Drag PID 5 this time over last name
8. In the rules panel edit the rules that maps last name and set the sub-component (^) value to 2
9. Run the inbound test

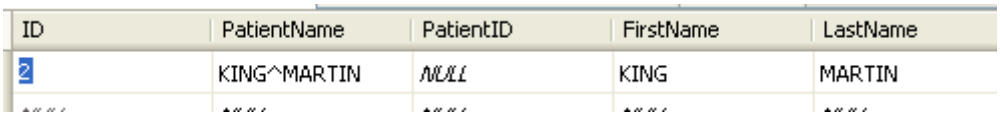


Figure 10: First and last name separated in database

The same can be done for repetitions (~ separator) for example when multiple phone numbers are present.

Outbound Message Rules

Outbound messages are messages that are sent from HL7Kit to an external system.

Creating a Simple Outbound Rule

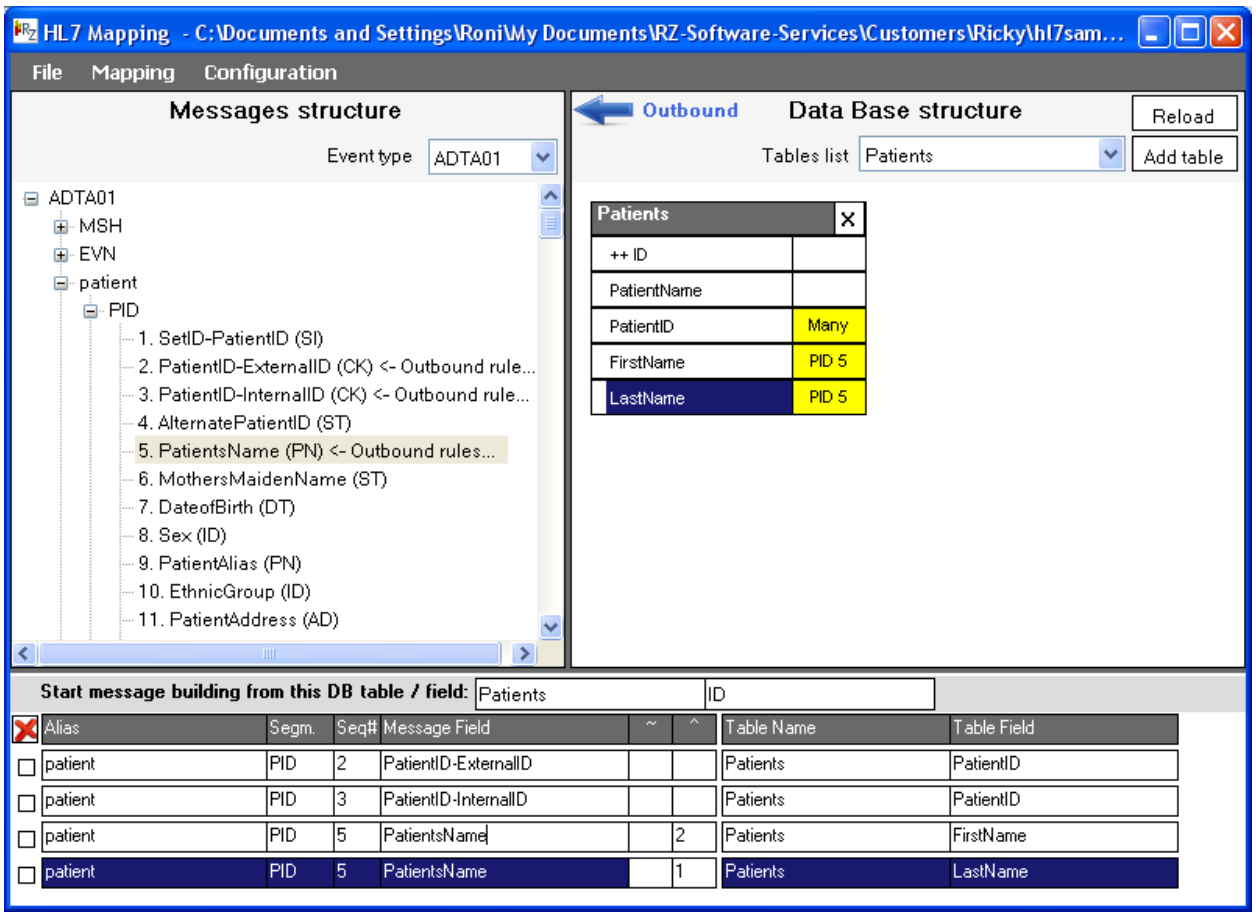


Figure 11: Outbound rule with sub-components mapping

Outbound messages differ from inbound messages because they have to be triggered from the database. For this reason the 'Start message building from DB table/field is required'.

To set the start table/field simply drag a field from the database panel on this bar.

Follow these steps:

1. Drag the patient ID on the start table/field bar.
2. Drag patient id column on PID2
3. Drag patient first name column on PID5

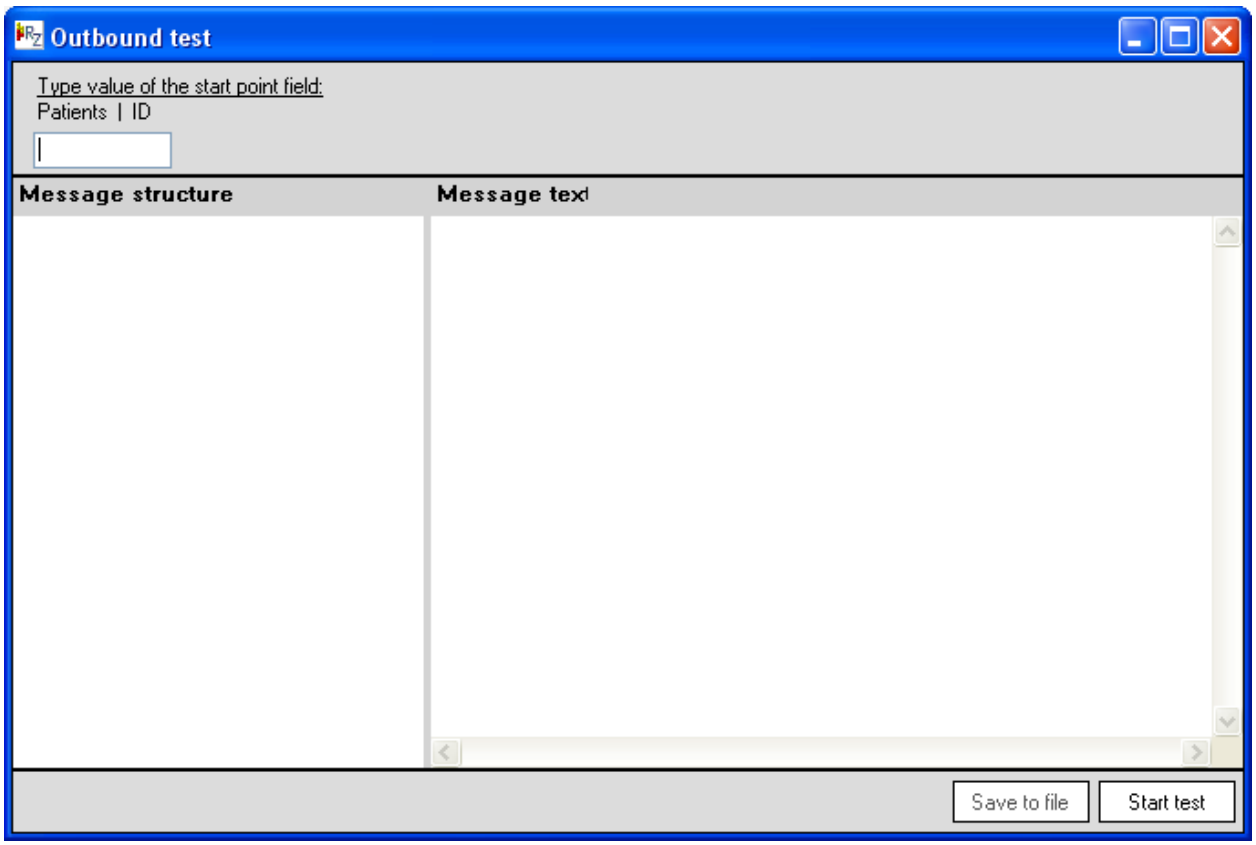
4. Set the ^ field of the rule to 2 (and click outside the text box)
5. Drag patient last name column on PID5
6. Set the ^ field of the rule to 1

To set a constant value to a field double click the field on the left panel and set the constant value.

Testing Outbound Rules

After the rule is defined, select 'Outbound Test' from the Mapping menu.

he



In the start point field type the value of the field in the database that you would like to create message from. In our example this would be one of the ID's in the Patients table.

Creating a Parent-Child Outbound Rule

Creating a parent-child outbound message rule is similar to the process of parent-child inbound rule with one exception: you must choose the parent table key that is used to link to the child tables so that HL7Kit can start the processing from the correct place.

Database Panel Symbols

Table columns in the database panels have the following symbols:

<	Primary Key. The background color matches the color of the foreign key.

>	Foreign Key. The background color matches the color of the primary key.
++	Identity Column

HL7Runtime

The HL7 Service is a background process that starts and stops automatically with the operating system. When installed, the service is responsible for receiving inbound messages and for sending outbound messages according to the defined mapping rules.

HL7 Runtime Configuration

The HL7 Runtime Configuration Control panel is installed together with the HL7 Runtime Service. When logging in, the control panel is started and minimized to the system tray area.

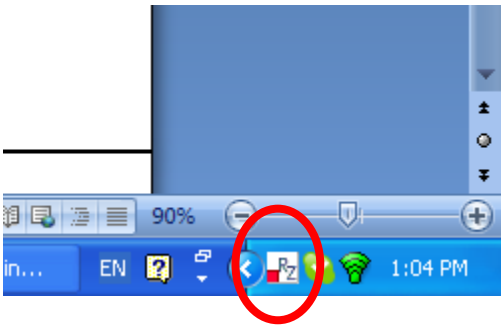


Figure 12: System Tray Icon

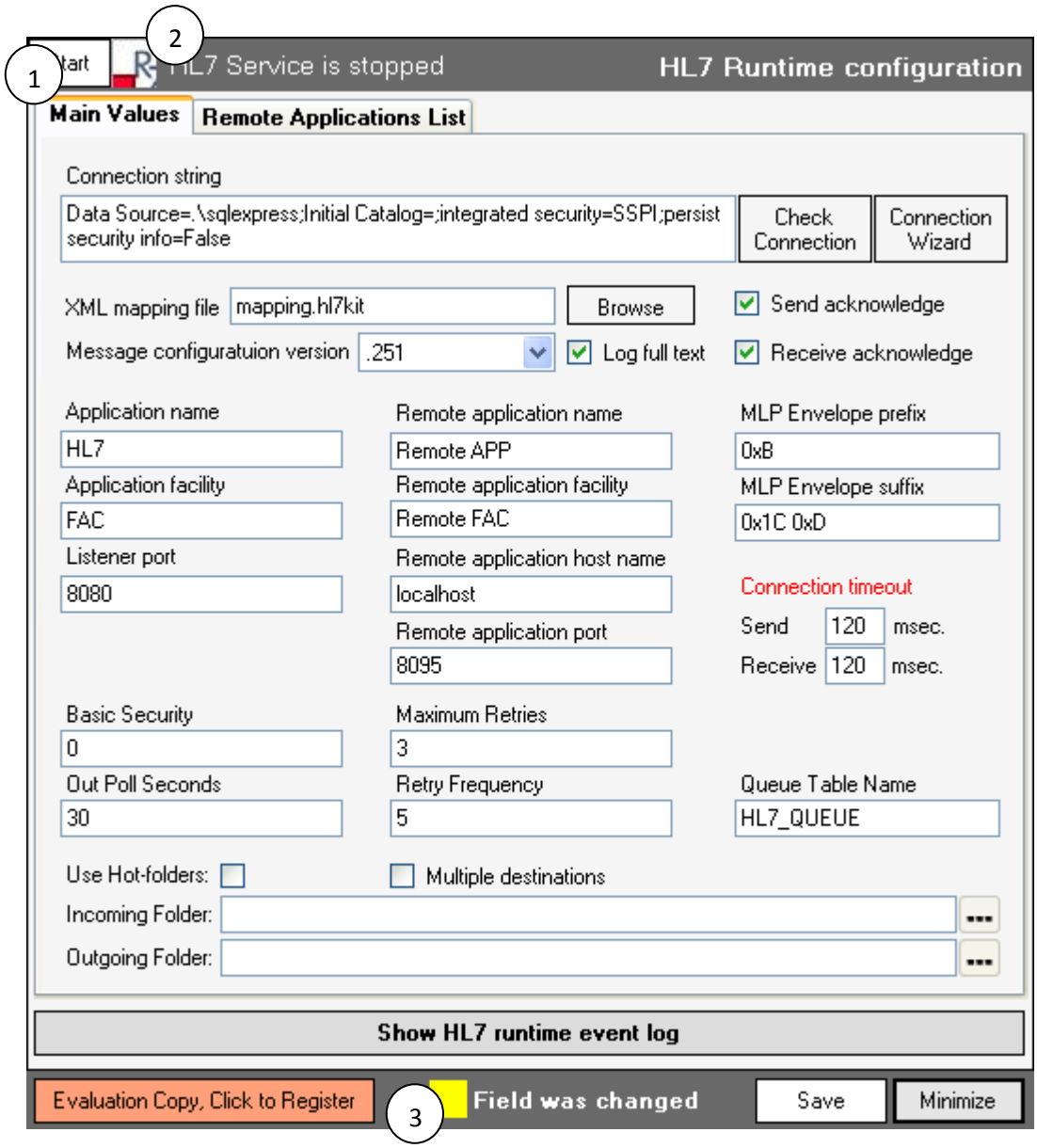


Figure 13: Service Configuration Control Panel – Main Values Pane

Service Start/Stop Button (1)

This button sends a stop or start message to the runtime service.

Service Status Indicator (2)

This indicator shows the service status:

1. Red: Stopped
2. Green: Running
3. Neutral:

Connection String Text Box
See Connection String Text Box.

Check Connection Button
See Check Connection Button.

Connection Wizard Button
See Connection Wizard.

XML Mapping File Text Box and Browse Button
This is the mapping file that the runtime service uses for processing messages. Use the browse button to select a file. The mapping file is created using the HL7Mapper application.

Message configuration Version
See Message Configuration Version.

Send Acknowledge Check-Box[4]
When checked, send HL7 Acknowledge message for every inbound message.

Receive Acknowledge Check-Box[4]
When checked, expect a HL7 Acknowledge message for every outbound message.

Log Full Text Check-Box
When checked, the complete message body of inbound and outbound messages is logged in the event log. See Monitoring and Logging.

Application Name Text Box
This is the application name of the runtime service (your application). This value is sent in MSH 2 (Sending Application) of outbound messages and expected in MSH 4 (Receiving Application) of inbound messages. This value is used for verification of the messages source and target when Basic Security is set.

Application Facility Text Box
This is the facility name of the runtime service (your application). This value is sent in MSH 3 (Sending Facility) of outbound messages and expected in MSH 5 (Receiving Facility) of inbound messages. This value is used for verification of the messages source and target when Basic Security is set.

Listener Prot
This is the network port that the runtime service is using for inbound messages. Make sure that this port is not used by other applications.

---

[4] With multiple destinations, this control is disabled and the feature is controlled from the remote applications pane.

Remote Application Name Text Box[4]

This is the application name of the peer application (the application you integrate with). This value is sent in MSH 4 (Sending Application) of outbound messages and expected in MSH 2 (Receiving Application) of inbound messages. This value is used for verification of the messages source and target when Basic Security is set.

Remote Application Facility Text Box[4]

This is the facility name of the peer application (the application you integrate with). This value is sent in MSH 5 (Sending facility) of outbound messages and expected in MSH 3 (Receiving facility) of inbound messages. This value is used for verification of the messages source and target when Basic Security is set.

Remote Application Host Name (or IP Address) [4]

This is the host name or IP Address of the peer application (the application you integrate with) The runtime service sends outbound messages to this network address.

Remote Application Port[4]

This is the network port that the peer application (the application you integrate with) uses. The runtime service sends outbound messages to this port of the remote application host.

MLP Envelope Prefix

This is the hex notation of the MLP envelope prefix characters that the runtime service uses for inbound and outbound messages.

MLP Envelope Suffix

This is the hex notation of the MLP envelope suffix characters that the runtime service uses for inbound and outbound messages.

Connection Timeout Text Boxes

These are the connection timeouts for inbound and outbound connections in seconds.

Basic Security Flag

This flag defines how the values in MSH 2, MSH 3, MSH 4, and MSH 5 is validated against the values of Application Name, Facility Name, Remote Application Name and Remote Facility resp. Use the following values:

- 0 – No validation, accept all messages regardless of the values in MSH 2-5
- 1 – Validate Sending – For inbound messages, check that sending application and facility is the defined remote application and facility.
- 2 – Validate Receiving – For inbound messages, check that application name and facility is the defined local application and facility.
- 3 – Validate Both – For inbound messages, check both sending and receiving application and facility.

Out Poll Seconds

The outbound messages queue poll period. The runtime service will poll the HL7-QUEUE table every this number of seconds and looks for new messages to send.

Maximum Retries

The runtime service will try sending a message this number of times before marking it as error message. Retries will be performed only if the message is syntactically valid but the remote application didn't receive it.

Retry Frequency

This is the number of seconds that the runtime service will wait before attempting another retry.

Queue Table Name

This is the name of the outbound messages queue table. You can change the default table name if for example you already have a table named HL7_QUEUE in your database.

Use Hot Folders Checkbox

When checked, the integration is based on files sharing instead of TCP/IP networking. Two folders are used for integration, one for incoming messages and one for outgoing messages.

Incoming Folder

This text box specifies the directory that is monitored for incoming messages. The service will pick every file with suffix "hl7" and process it. The service deletes files after reading them. Other files with no "hl7" suffix are ignored.

Outgoing Folder

This text box specifies the directory where the service saves outgoing messages. Message files are named with timestamp and "hl7" suffix.

Multiple Destinations Checkbox

Starting in version 1.4, a single HL7Kit runtime can integrate many HL7 Applications. When checked, configuration of multiple applications is enabled. See Multiple Destinations.

Show HL7 runtime event log

This button shows a report with the last 100 HL7 Service Runtime log entries from the system event log. See Monitoring and Logging.

Save Button

This button saves the current configuration. After clicking save, you can click cancel to minimize the panel to the tray.

Minimize Button

This button hides the control panel. To restore, double click the tray icon.

Changes Indicator

When edits are made, the modified controls and the save button are highlighted in yellow.

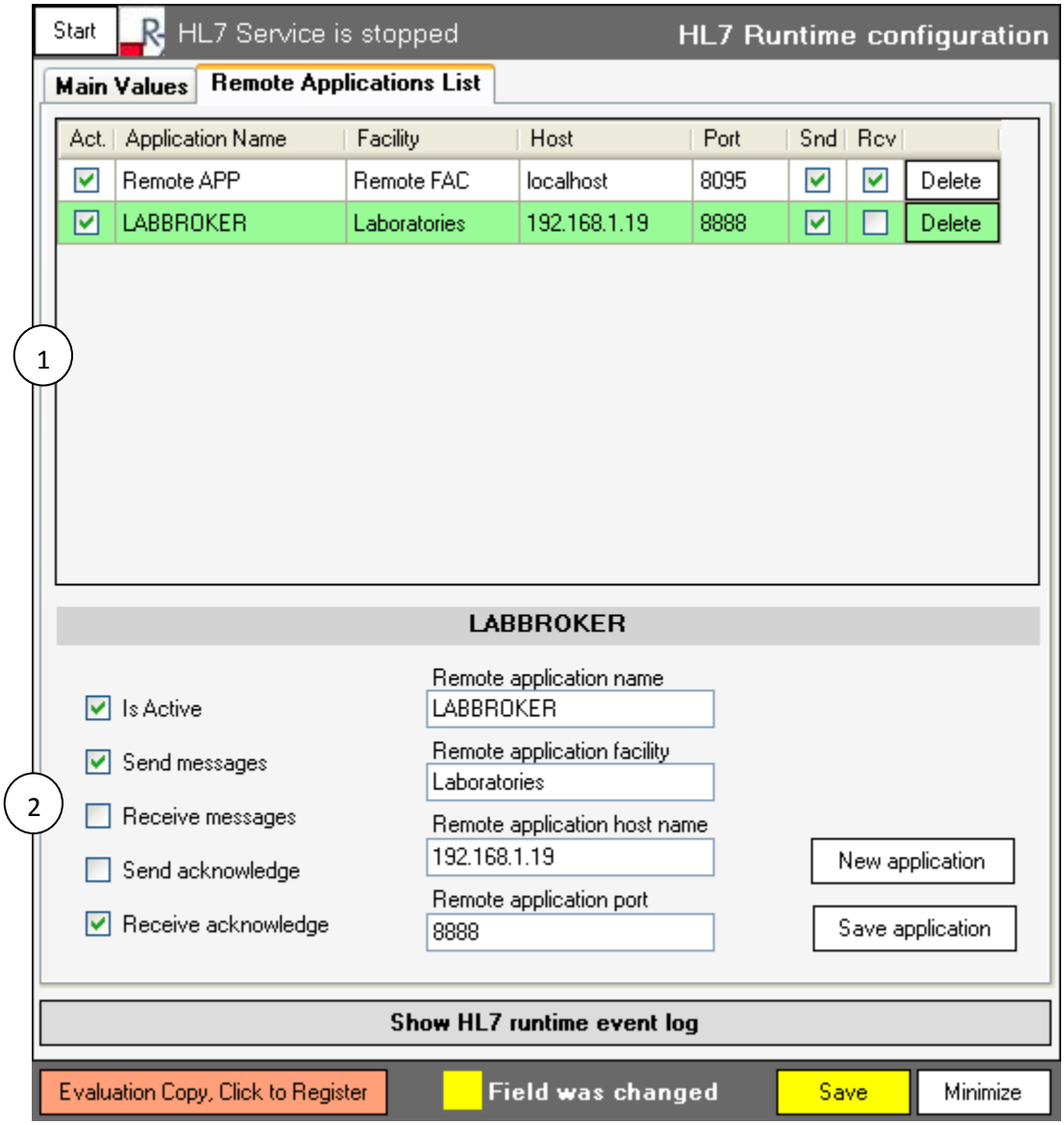


Figure 14: HL7Kit Runtime Control Panel - Remote Application List Pane

Remote Application List (1)

The remote application list displays all the applications that the kit is configured to integrate with.

Click an application line to highlight it and edit the application information is the edit area of this pane.

The Edit Area (2)

The controls in the edit area (1) show the information of the highlighted application in the remote applications list (2).

Is Active Checkbox

When this checkbox is marked, messages will be sent to and received from the selected application.

By default, every outbound message is sent to all active remote applications, unless the integrator explicitly sets the destination in the queue table entry.

Send Messages Checkbox

When this checkbox is set, messages will be sent to the selected application.

Receive Messages Checkbox

When this checkbox is set, messages received from the selected application will be processed.

Send Acknowledge Checkbox

When checked, send HL7 Acknowledge message for every inbound message from the selected application.

Receive Acknowledge Checkbox

When checked, expect a HL7 Acknowledge message for every outbound message that is sent to the selected application.

Remote Application Name Text Box

This is the application name of the selected peer application. This value is sent in MSH 4 (Sending Application) of outbound messages and expected in MSH 2 (Receiving Application) of inbound messages. This value is used for verification of the messages source and target when Basic Security is set.

Remote Application Facility Text Box

This is the facility name of the selected peer application. This value is sent in MSH 5 (Sending facility) of outbound messages and expected in MSH 3 (Receiving facility) of inbound messages. This value is used for verification of the messages source and target when Basic Security is set.

Remote Application Host Name (or IP Address)

This is the host name or IP Address of the selected peer application. The runtime service sends outbound messages intended to the selected application to this network address.

Remote Application Port

This is the network port that the selected peer application uses. The runtime service sends outbound messages to this port of the selected remote application host.

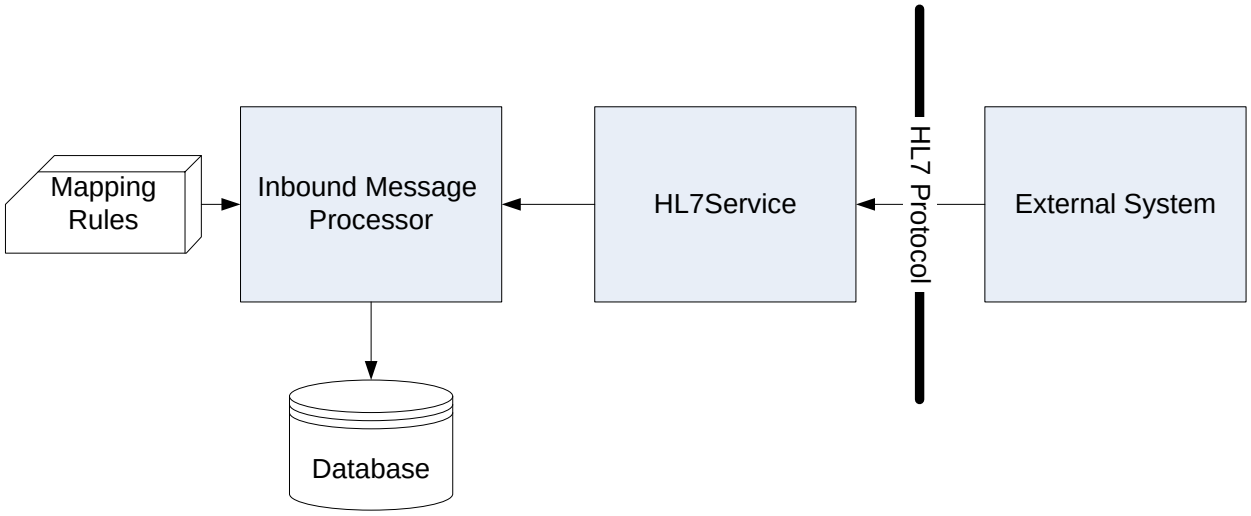
New Application Button

This button clears the edit area in order to enable user input of a new remote application.

Save Application Button

This button saves the new application configuration or the changes to an existing application configuration.

Inbound Messages Processing

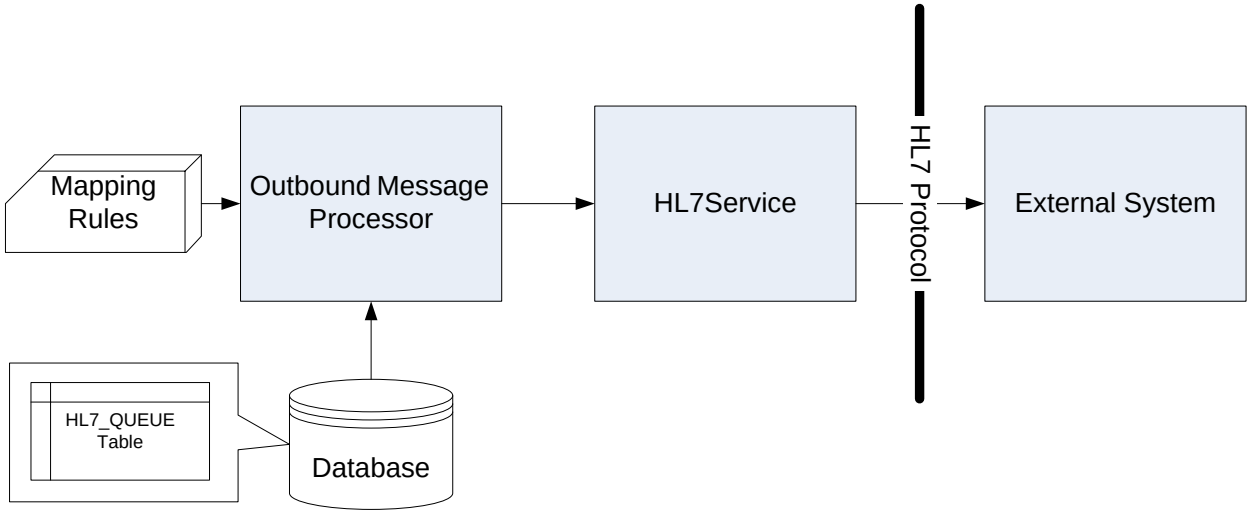


When an external application sends a HL7 message to the HL7 Service, the message is processed according to the inbound mapping rules defined for its event type and records are inserted into the appropriate database tables.

The processing of the message is done based on the message structure definition and the foreign keys (relationships) between the database tables.

The hierarchy of the HL7 message structure should match the hierarchy of the database tables.

Outbound Message Processing



Outbound message processing is triggered by inserting or updating a record in the HL7_QUEUE database table. The HL7 service polls the HL7_QUEUE table every few seconds. If it finds record with status ready, it activates the outbound mapping rules and sends the message to the remote system.

The outbound messages queue

The HL7_QUEUE Table

HL7Kit runtime service uses the outbound queue for sending outgoing messages.

When defining an outbound rule, the mapping application can create a trigger on the database tables participating in the rule so that when new data is inserted, a new queue record is created.

Alternatively, the queue record can be inserted programmatically by your application.

Column Name	Data Type	Allow Nulls	Description
que_id	int	No	The id of this queue entry. Referenced in log records.
que_status	int	No	The status of the
que_creation_date	datetime	Yes	The date and time the record was created on.
que_event_type	nchar(10)	Yes	The HL7 Event type (e.g. ADTA01) that should be sent. This value must match one of the rules in the mapping definition file.
que_start_table	nvarchar(100)	Yes	The table name that is the root table of the outbound message rule.
que_start_field	nvarchar(100)	Yes	The primary key or ID field that identifies the message in the start table.
que_start_value	int	Yes	The id value of the record in the start table that holds the message data to be sent.
que_update_date	datetime	Yes	The last time the queue record was updated. Automatically updated by the runtime service.
que_data_type	nchar(10)	Yes	Not in use.
que_retry_count	int	Yes	Send retries counter.
que_last_error_code	int	Yes	The last retry error code.
que_last_error_message	nvarchar(MAX)	Yes	The last retry error message
que_app_name	nvarchar(100)	Yes	The remote application Name to send the message to. If inserted with NULL value, the runtime service will duplicate the record for all active remote application and manage one queue record for each destination.

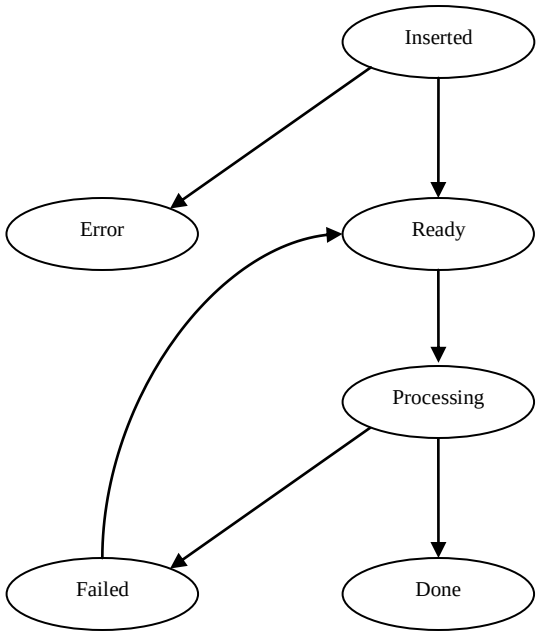
que_app_facility	nvarchar(100)	Yes	The remote application facility. See que_app_name.

Queue Statuses

The HL7_QUEUE status column can have the following values:

Status	Name	Meaning
1	Inserted	This is the initial status, when a new record is created.
2	Ready	The message is ready for sending. The runtime service picks up records with status ready and sends them.
3	Error	A final state for messages that could not be processed or sent.
4	Processing	Intermittent state while the message is being sent.
5	Done	The message has been sent successfully.
6	Failed	Failed to send message. The service will retry sending failed messages until the retry count is exhausted.

Queue State machine



Sending messages to multiple destinations

If the destination application in the HL7_QUEUE record is NULL, the message the runtime service duplicates the record for every active remote application that it is configured to send messages to. If the destination application is not NULL, the message is sent only to the specified application.

Queries

HL7 Queries are not supported in this version of HL7Kit.

Multiple Destinations

Starting at version 1.4, a single HL7Kit Pro runtime service can integrate with many remote HL7 Applications. The remote applications are configured using the Remote Application List (1).

Each remote application can be configured for sending, receiving or sending and receiving messages.

When a new message is generated, the runtime service sends it to every active application in the list that is configured to send messages to.

When a new message is recived and the basic security checkbox is set, the runtime service validates the source application name with the active applications that are configured to receive application from.

Maintenance

Monitoring and Logging

HL7Kit uses the built in windows event log for logging. This makes it very easy to use advanced monitoring tools built into your windows operating system.

Both HL7Receiver and the HL7 Runtime Service write into the RZHL7Pro log file.[5]

Comment: HL7Kit's evaluation copy does not write into the event log.

HL7 Version Configuration

Message Rules

The message and segment rule files are only slightly different than the HL7 demonstration implementation that is used for example in the IHE test tools. The major difference is the ability to define hierarchical structure using aliases and thus making the definition for XML structure as well.

[5] Starting in version 1.4 the source for the log file is RZHL7Pro. Versions 1.3 and older used the source RZHL7.

Editing the message rules

```
ACK:MSH MSA [ ERR ]

QRYQ01:MSH QRD [QRF] [DSC]

DSRQ03:MSH QRD [QRF]{DSP}[DSC]

ADTA01:MSH EVN \ PID <NK1> PV1 [PV2] / patient
<OBX><AL1><DG1><PR1><GT1> <IN1 [IN2] [IN3]>[ACC] [UB1] [UB2]
```

Figure 15: Message Rules Definition File. A hierarchy group alias is highlited.

The message rules configuration file msgRules.251 is using a similar notation to the one used in the HL7 Standard documentation.

Every line in the file represents one message definition.

The first element is the event type.

When the event type is followed by a semicolon and another event type that means that the structure of the message is defined by the event type on the left is identical to the one with event type on the right. In the example above ADT^A03 is defined to have an identical structure as ADT^A02.

When the event type is followed by a colon, a list of segment names and special grouping characters follow.

The special grouping characters are:

Open group	Close group	Meaning	Mandatory	Repeating
{	}	1 or more occurrences	Yes	Yes
(	)	Exactly one of	Yes	No
\	/	Exactly one	Yes	No
[	]	0 or 1 occurrences	No	No
<	>	0 or more occurrences	No	Yes

Aliases

Elements that are not special characters or a valid segment names are regarded as aliases. Aliases are used to group segments into entities. For example, in the ADT^A01 an alias called ‘patient’ is defined. Note that the alias name comes AFTER the segments group.

Segments Definitions

Editing the segment definitions

The segment definitions can be edited using a simple text editor like notepad.

Every segment is defined as a list of fields.

Field Definition

```
2 ID REQ NO_RPT AcknowledgementCode     /* 1  Acknowledgement Code
*/

20 ST REQ NO_RPT MessageControlID         /* 2  Message Control ID
*/
```

A field is defined using the following attributes:

1	Length	A number stating the maximum data length of the field
2	Data type	A two-characters field type code that defines the data type of the field. Valid field types are: • ST • TX • FT • NM • DT • TM • TS • PN • TN • AD • ID • SI • CM • CK • CN • CQ • CE
3	Mandatory	A flag that defines if this field is mandatory or can be left blank: • OPT – Optional • REQ - Required
4	Repetition	A flag that define the repetition of the field • NO_RPT – One value • NO_MAX – Multiple values
5	Field name	A string that defines the field name for XML representation. The field name must not include any spaces. CaMeL notation is common.
6	Comment	Free text for documentation enclosed by /* */

Creating new configurations

Configuration files are stored in the application folder. The configuration is stored in two files called:

- msgRules.251
- segDefs.251

To create a new configuration simply copy an existing configuration and change its suffix.

Examples

This section includes two detailed example projects, one for processing incoming messages and one for sending outgoing messages.

Every example contains:

1. Readme file with detailed step by step instructions,
2. SQL Scripts for creating the databases and inserting data into the tables,
3. HL7Kit Mapping Definition file,
4. HL7 message file to use in the example.

The example projects can be downloaded from the product support page at http://www.hl7kit.com/support.html.

Link to inbound example: http://www.hl7kit.com/Dist/HL7_ADTA01_INBOUND_EXAMPLE.zip

Link to outbound example: http://www.hl7kit.com/Dist/HL7_ORUR01_OUTBOUND_EXAMPLE.zip

Inbound Message Example Project

This project demonstrates processing of patient registration event ADT^A01. The message, included in the example, is sent using the HL7Sender to the HL7 runtime service and the information from the message is inserted to the patients table.

Outbound Message Example Project

This project demonstrates creating a report message OR^R01. The project includes a SQL script that whenever run, a message is generated by the HL7 runtime service and sent to the HL7Receiver.

Annexes

Concepts and Vocabulary

HL7 File

HL7 File is a file that stores a HL7 message data. HL7 File is not a text file. Every segment in the file is separated by a segment separator character that is usually 0xD. Do not edit HL7 Files using a text editor such as Notepad as it will replace the segment separators with newlines.

LLP

The Lower Layer Protocol (LLP), sometimes referred to as the Minimal Lower Layer Protocol (MLLP), is the standard for transmitting HL7 messages via TCP/IP. Since TCP/IP is a continuous stream of bytes, the wrapping protocol (i.e. headers and trailers) is required for communications code to be able to recognize the start and the end of each message. The Lower Layer Protocol is the most common mechanism for sending unencrypted HL7 via TCP/IP over a local area network, such as those found in a hospital.

An HL7 message must be wrapped using a header and trailer (also called footer) to signify the beginning and end of a message. These headers and footers are typically non-printable characters that would not be shown in the actual content of an HL7 message.

MLLP

MLLP is an acronym for Minimal Lower Level Protocol. See LLP.

ACK

The HL7 mechanism used for acknowledgment of delivery and/or processing. When a system receives a message, it is expected to respond with the matching ACK message. The ACK message carries a status code.

The handling of ACK's can be configured so as to expect or not ACK responses and to send ACK's or not to. and