



Fall 2010-09-19

MirthConnect howto document


Sebastien Pardon

Dublin Institute of Technology, seb.pardon@gmail.com

Damon Berry

Dublin Institute of Technology, damon.berry@dit.ie

Follow this and additional works at: <http://arrow.dit.ie/teapotrep>

 Part of the [Computer Engineering Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Pardon, Sebastien and Berry, Damon, "MirthConnect howto document" (2010). *Reports*. 2.
<http://arrow.dit.ie/teapotrep/2>

This Report is brought to you for free and open access by the tPOT: People Oriented Technology at ARROW@DIT. It has been accepted for inclusion in Reports by an authorized administrator of ARROW@DIT. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie, brian.widdis@dit.ie.



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 License](#)



Table of Contents

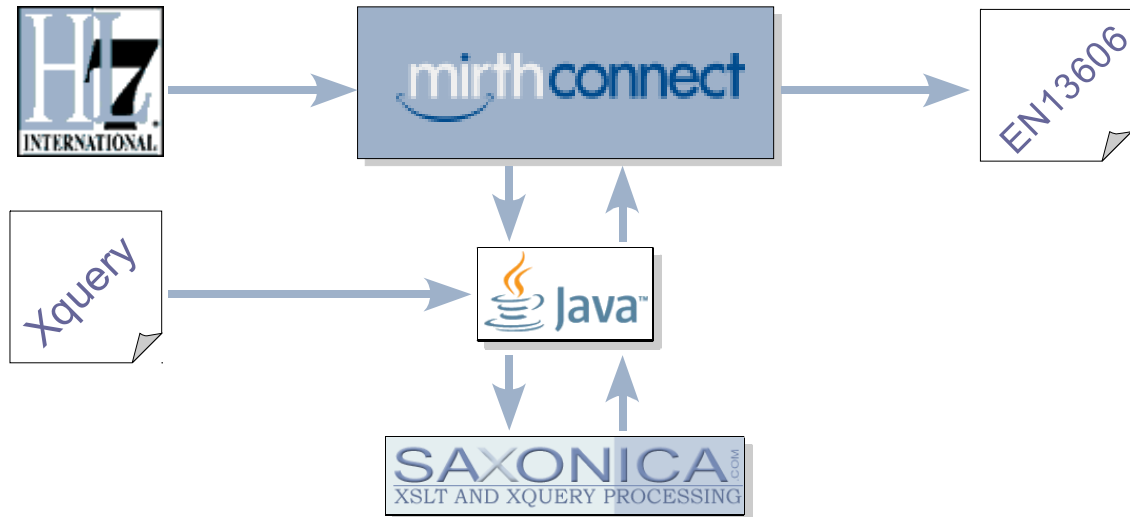
Introduction.....	2
Requisites.....	3
Installation.....	3
Configuration.....	5
HL7v2 to simple xml channel.....	6
Simple xml to EHRCOM xml.....	11
Deployment.....	16

Illustration Index

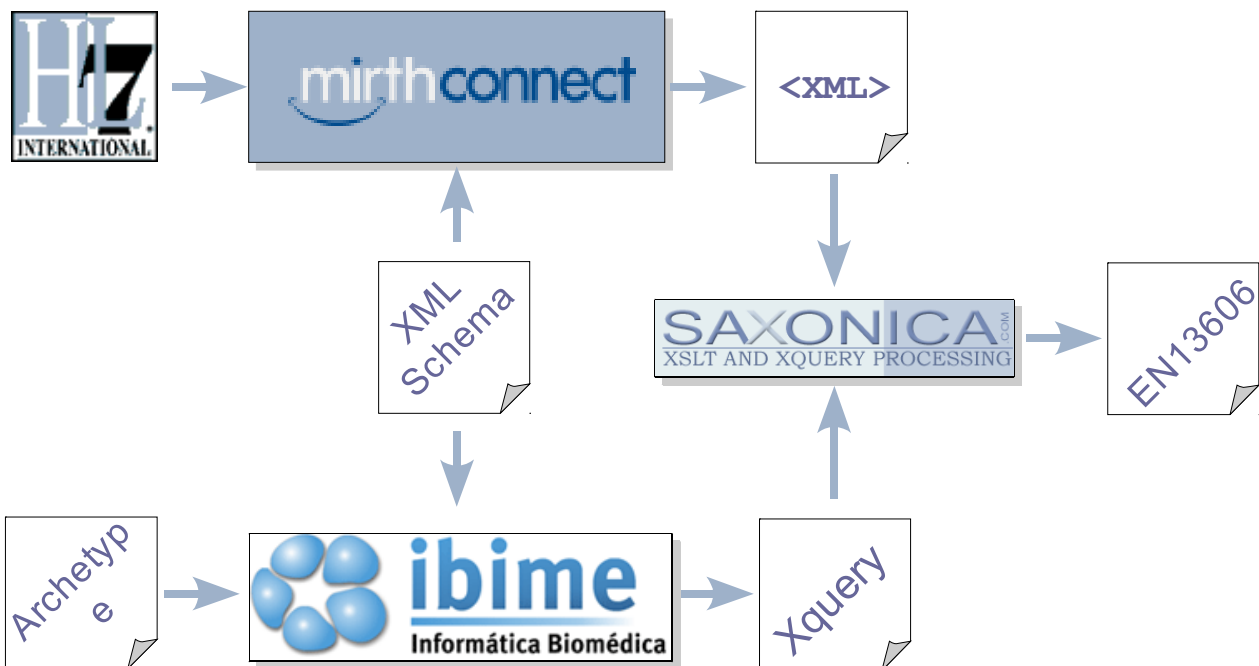
Illustration 1: Mirth Server Window.....	3
Illustration 2: Mirth Connect Administrator - Java Web Start.....	4
Illustration 3: Opening webstart.....	4
Illustration 4: Mirth Connect Administrator - Login.....	5
Illustration 5: Mirth Connect menu.....	5
Illustration 6: Channel Tasks menu.....	5
Illustration 7: HL7v2 to simple xml - summary.....	6
Illustration 8: HL7v2 to simple xml - source.....	7
Illustration 9: HL7 Message Templates.....	8
Illustration 10: HL7 Message Trees.....	8
Illustration 11: Reference tab panel.....	9
Illustration 12: HL7v2 to simple xml - transformer.....	9
Illustration 13: HL7v2 to simple xml - destinations.....	10
Illustration 14: simple xml to EHRCOM xml - summary.....	11
Illustration 15: simple xml to EHRCOM xml - source.....	12
Illustration 16: simple xml to EHRCOM xml - destinations.....	13
Illustration 17: XML Message Templates.....	13
Illustration 18: XML Message Trees.....	13
Illustration 19: simple xml to EHRCOM xml - transformers.....	15
Illustration 20: Channels.....	16
Illustration 21: Dashboard.....	17
Illustration 22: Dashboard after message processing.....	18
Illustration 23: Status Tasks contextual menu.....	19
Illustration 24: Send Message Window.....	19

Introduction

This document aims at describing the steps involved in installing and configuring Mirth Connect so as to produce an EN13606-compliant record extract from an HL7v2 message using the LinkedEHR-produced xquery.



Drawing 1: Process flow



Drawing 2: Data flow

Requisites

- Mirth Connect: you can download the software at <http://www.mirthcorp.com/community/downloads>. The version used for this manual is 1.8.2.4472.1351.
- The Saxon library for Xquery parsing: you can download the software at <http://saxon.sourceforge.net/>. The version used in this manual is Home Edition 9-2-0-5j
- The LinkEHR Xquery. For this manual we used the mapping of simple xml for a Blood Pressure Archetype
- The xquery jar file containing the Converter class. This is the extension to Mirth Connect that bridges the channel transformers with the Saxon library

Installation

- Install Mirth Connect. Refer to the Mirth Connect website and manual for information regarding the installation procedure. This document refers to the Mirth Connect installation directory as \$MIRTH_HOME\$
- Locate the Saxon library (saxon9he.jar) and open it with a Zip-file manager. Remove the services/ directory from the META-INF/ directory. This directory contains configuration information that conflict with Mirth Connect.
Copy the modified jar file to the \$MIRTH_HOME\$/lib/custom/ directory
- Copy the xquery jar file to the \$MIRTH_HOME\$/lib/custom/ directory
- Start the Mirth Server or service (depending on the choice you made during the Mirth Connect installation procedure). If you chose to use the server, the following window should appear.

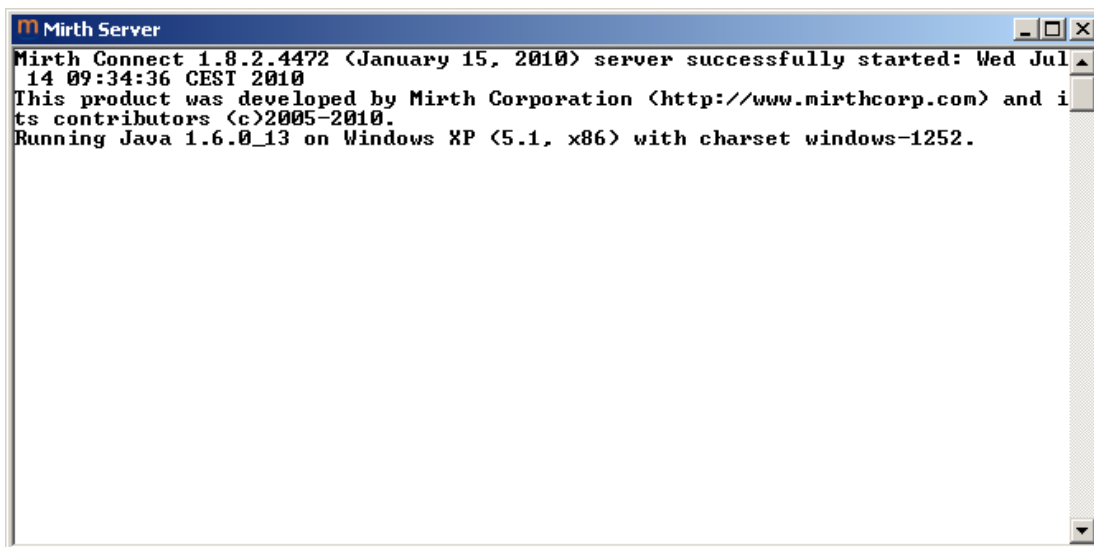


Illustration 1: Mirth Server Window

- Start a web browser and go <http://localhost:8080/> this will display the Mirth Connect Administrator Java Web Start page where you will be able to launch the Administrator.



Mirth Connect Administrator - Java Web Start

Overview of Web Start

Java Web Start is a framework developed by Sun Microsystems that enables launching Java applications directly from a browser. Unlike Java applets, Web Start applications do not run inside the browser.

Click the big green button below to launch the Mirth Connect Administrator using Java Web Start.

[Launch Mirth Connect Administrator](#)



Illustration 2: Mirth Connect Administrator - Java Web Start

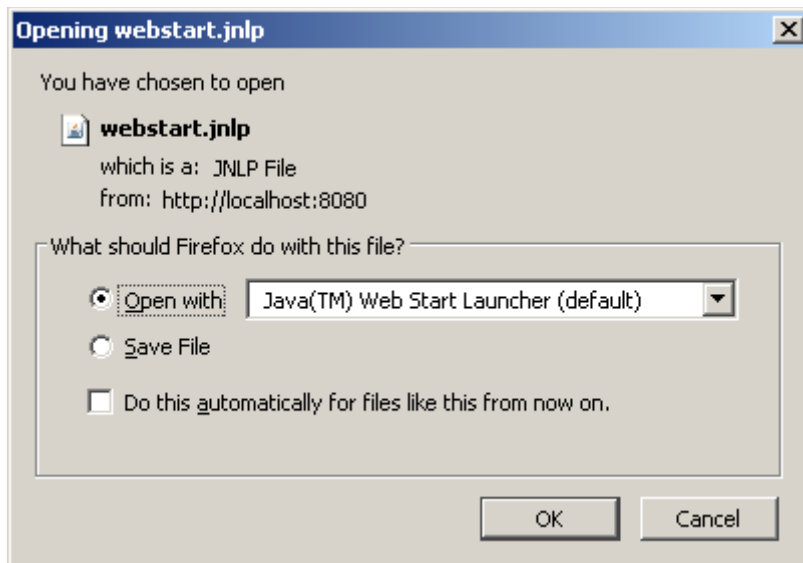


Illustration 3: Opening webstart

Configuration

This section focuses on the creation of channels that will parse an HL7v2 message file from the [c:\ehrland](#) directory and produce an XML record extract to the [c:\ehrland](#) directory.

In order to do so we will create 2 channels:

- HL7v2 to simple xml: This channel will parse the HL7 message file and map it to the xml format used during the LinkEHR mapping procedure. This xml message (referred to as “simple xml”) will then be sent to the next channel
- simple xml to EHRCom xml: This channel will receive the simple xml message and invoke the xquery converter class to produce an EN13606-compliant xml message that will then be written to the target directory.

Log on to the Mirth Connect Administrator. The default username and password are admin/admin.

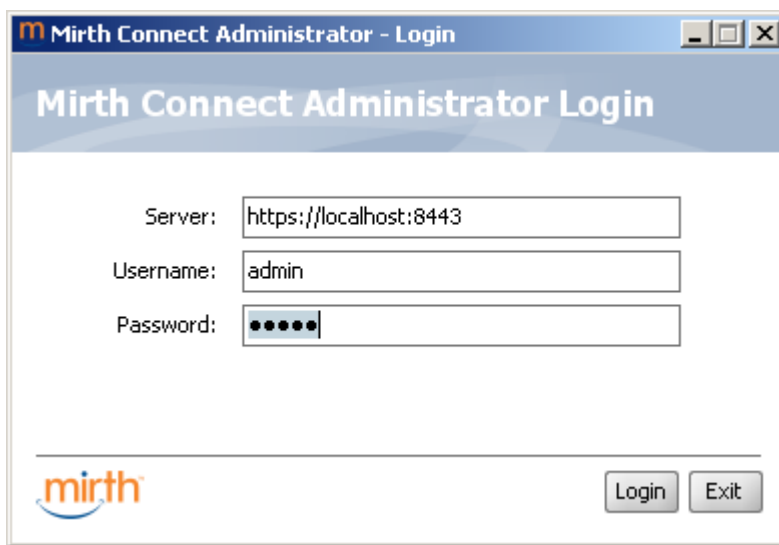


Illustration 4: Mirth Connect Administrator - Login

The administrator presents an empty dashboard. Select **Channels** on the top-left-hand Mirth Connect menu and then **New Channel** from the Channel Tasks menu.

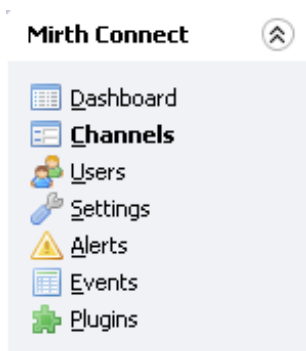


Illustration 5: Mirth Connect menu

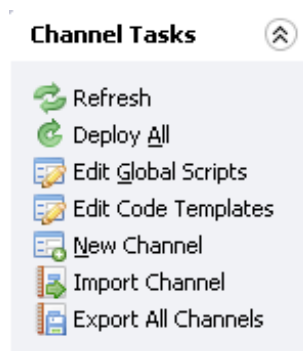


Illustration 6: Channel Tasks menu

HL7v2 to simple xml channel

Create a new channel named “**HL7v2 to simple xml**” of incoming data type “**HL7 v2.x**”. Make sure the channel is **enabled** and **synchronized**.

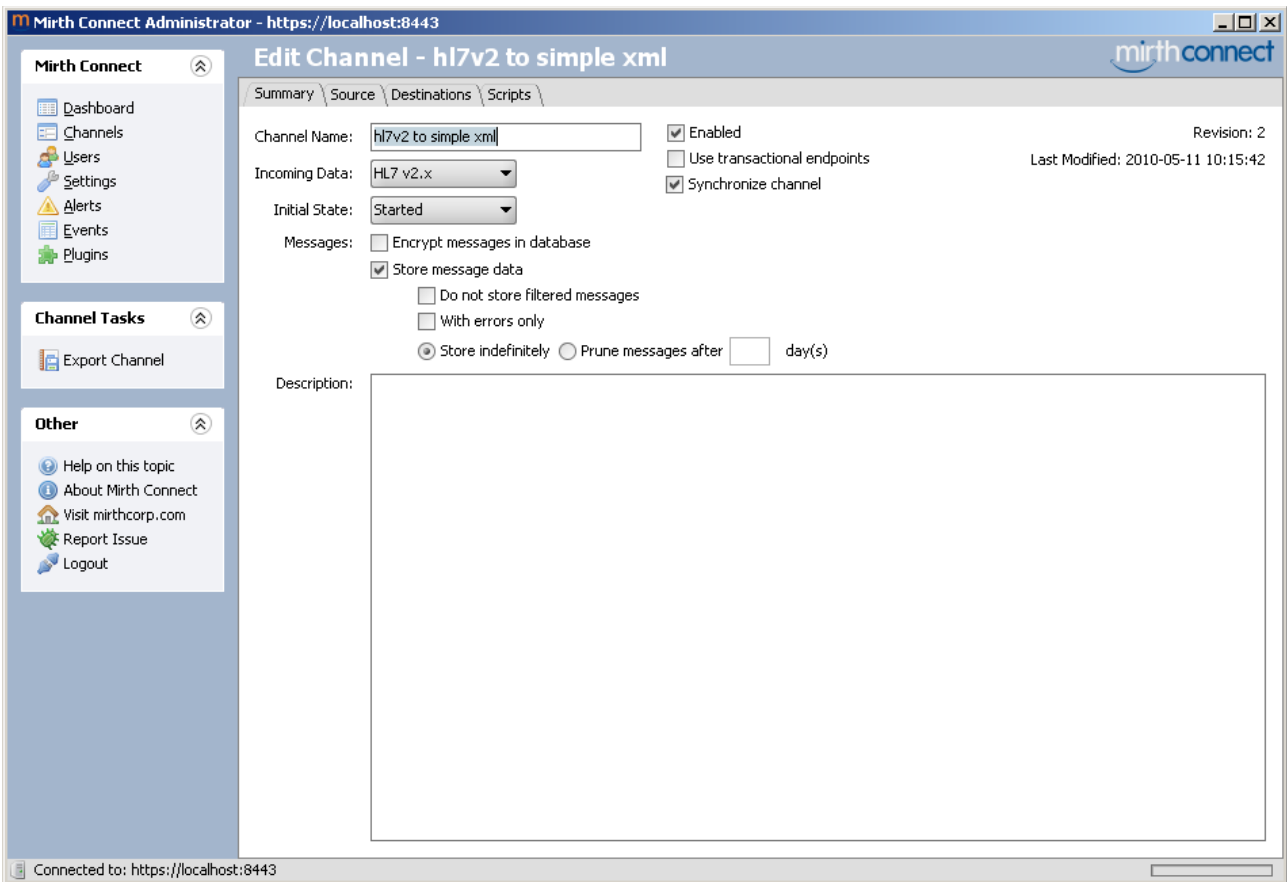


Illustration 7: HL7v2 to simple xml - summary

As the channel will read HL7 message file (with extension .hl7) from the [c:\ehrland](#) directory, on the **Source** tab panel select the **File Reader** connector type, insert [c:/ehrland](#) as the directory and ***.hl7** as the file pattern. You can use the **Test Read** button to control if Mirth Connect is able to access the directory. Make sure that the **Delete File After Read** option is set to **Yes**.

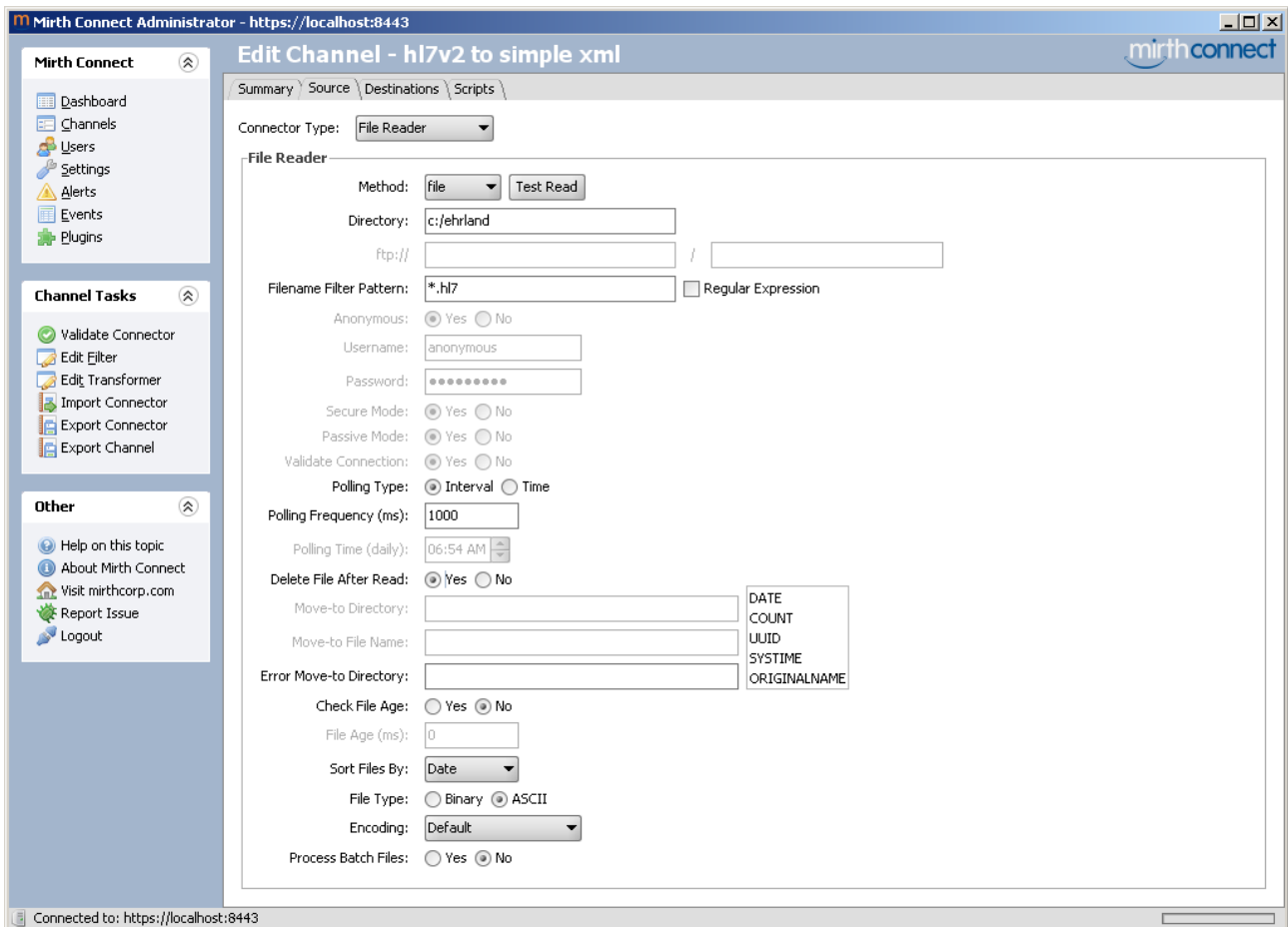


Illustration 8: HL7v2 to simple xml - source

As the destination will be a channel we have yet to create, on the Destinations tab create a New Destination (from the **Channel Tasks** menu) of type **Channel Writer**. We will leave the channel name blank for the moment.

In the template section insert the xml data structure that was used for the LinkEHR xquery mapping procedure:

```
<?xml version="1.0" encoding="UTF-8" ?>
<data>
  <patient id="">
    <systolic></systolic>
    <diastolic></diastolic>
    <position>1</position>
    <date></date>
  </patient>
</data>
```

In order to perform the mapping from an HL7v2 message to our xml format we will need to use some transformers. In the **Channel Tasks** menu select **Edit Transformer**. On the right-hand side in the **Message Templates** tab panel, insert an example HL7v2 message as shown in Illustration 9.

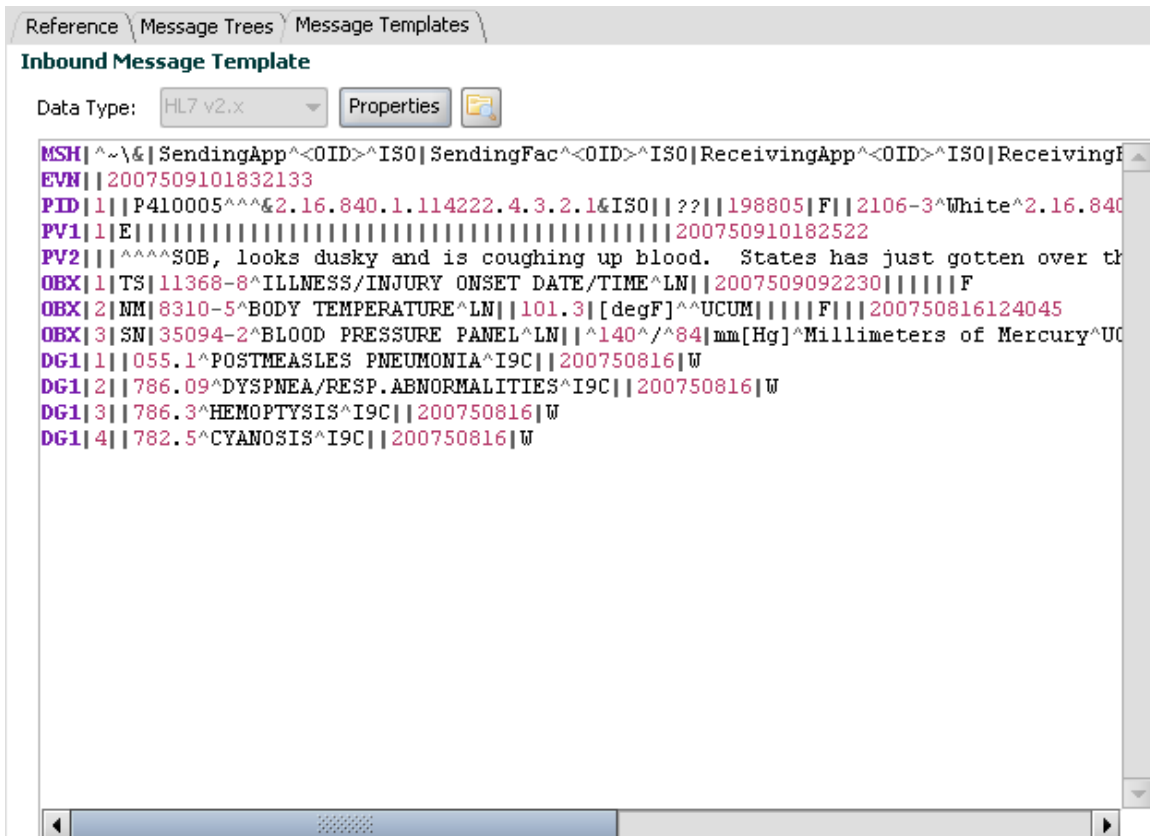


Illustration 9: HL7 Message Templates

Mirth Connect will automatically interpret the message and will provide its hierarchical representation in the **Message Trees** tab panel as shown in Illustration 10.

This panel can be used to easily create a transformer.

In the Transformer Task menu select **Add New Step** of type **Mapper** to be added to the **Channel Map**.

For the patient identifier we can use the **pid** variable name. In the Message Tree pane, drag the child node of the **PID . 3 . 1** node and drop it into the Mapping section. It should automatically write

```
msg['PID']['PID.3']['PID.3.1'].toString()
```

Repeat the procedure with a step called **systolic** referring to the child node of **OBX . 5 . 2**.

Repeat the procedure with a step called **diastolic** referring to the child node of **OBX . 5 . 4**.

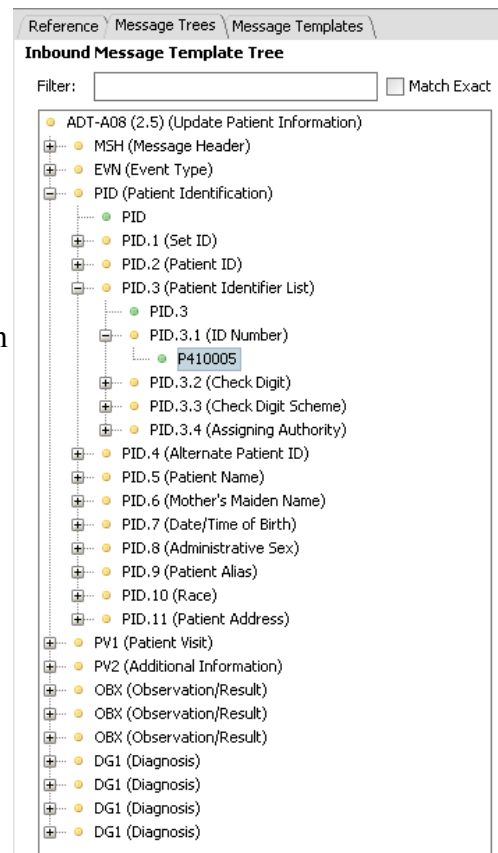


Illustration 10: HL7 Message Trees

As the date format used by the simple xml differs from the date format used by the HL7 message, we will need to perform some type of conversion on top of the mapping.

Create a new step called **event_date**. In the Reference tab panel filter the various options with the **Date Functions** value. Drag the **Convert Date String** function into the mapping section. We will need to convert the date from a yyyyMMdd format to a dd-MM-yyyy one. Finally from the Message Tree Panel we can drag the child node of **EVN.2.1** and drop it as the last variable of the conversion procedure.

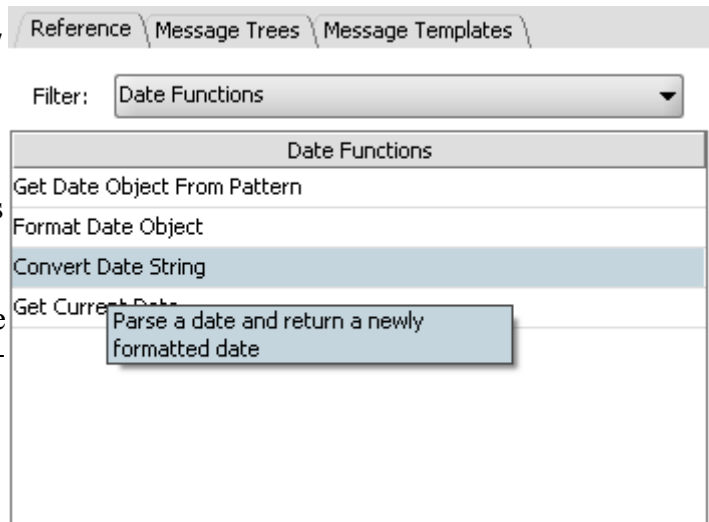


Illustration 11: Reference tab panel

The final result of the mapping should be:

```
DateUtil.convertDate('yyyyMMdd', 'dd-MM-yyyy', msg['EVN']['EVN.2']
['EVN.2.1'].toString())
```

Here is a summary of the 4 variables:

pid	msg['PID']['PID.3']['PID.3.1'].toString()
systolic	msg['OBX'][2]['OBX.5']['OBX.5.2'].toString()
diastolic	msg['OBX'][2]['OBX.5']['OBX.5.4'].toString()
event_date	DateUtil.convertDate('yyyyMMdd', 'dd-MM-yyyy', msg['EVN']['EVN.2']['EVN.2.1'].toString())

The following picture shows the Transformer section after completion of the 4 mapping variables.

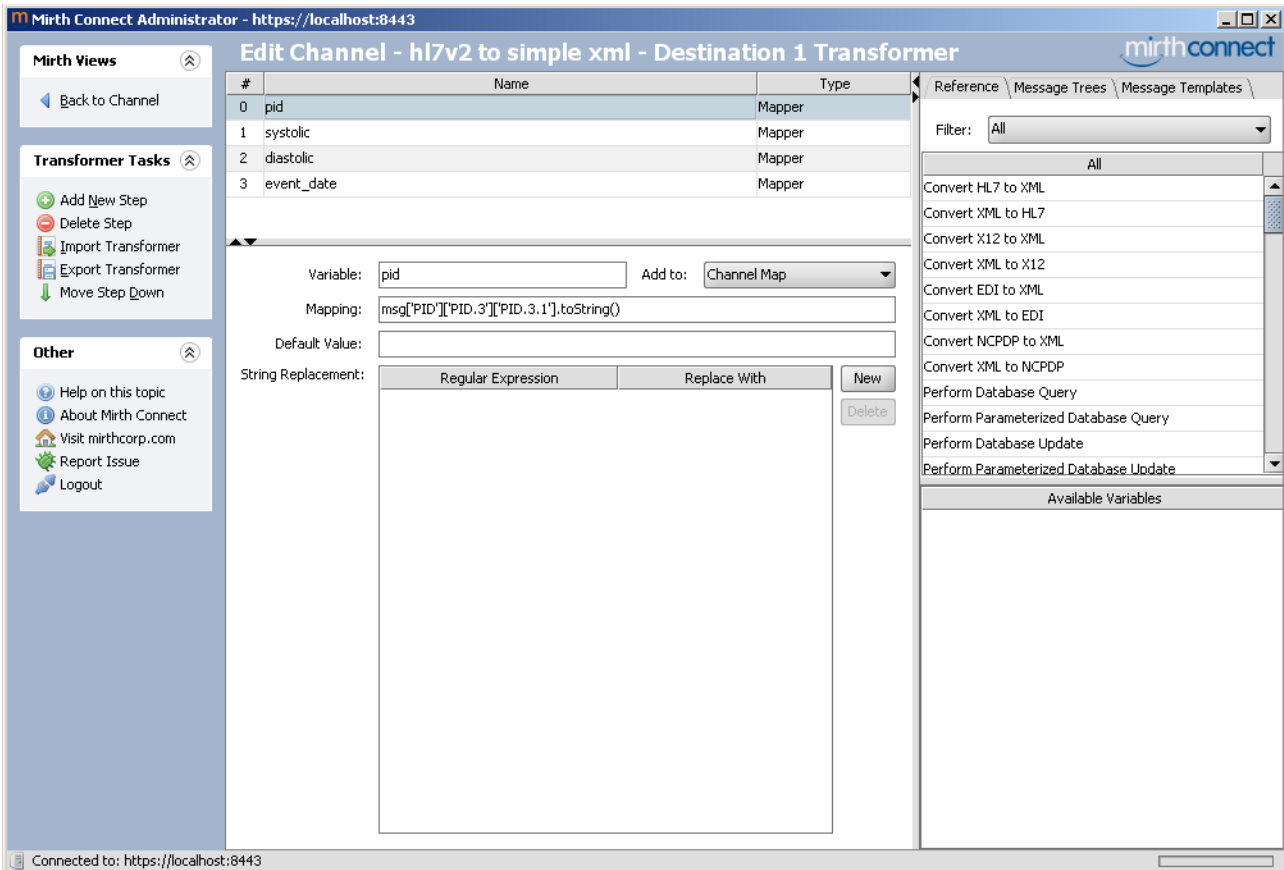


Illustration 12: HL7v2 to simple xml - transformer

In the **Mirth Views** menu on the left-hand side, go back to the channel.

In the **Destination Mappings** section on the bottom-right-hand of the Destinations tab panel we can see the 4 variables previously created. These variables can be dragged and dropped into the **Template** area to complete the xml message.

For example, dragging and dropping the **pid** variable should result in **\${pid}**. The final template should be

```
<?xml version="1.0" encoding="UTF-8" ?>
<data>
  <patient id="${pid}">
    <systolic>${systolic}</systolic>
    <diastolic>${diastolic}</diastolic>
    <position>1</position>
    <date>${event_date}</date>
  </patient>
</data>
```

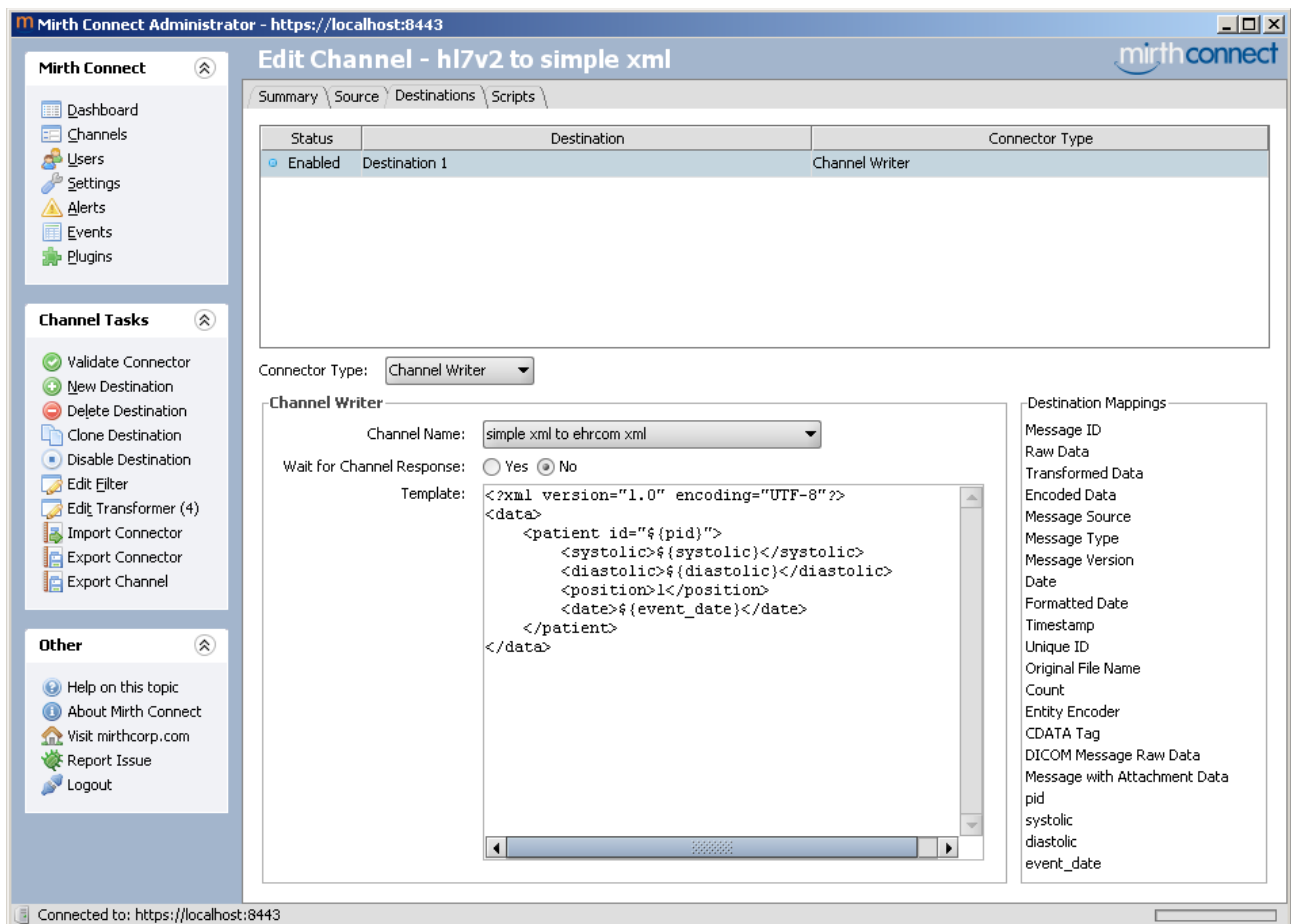


Illustration 13: HL7v2 to simple xml - destinations

You can save the channel in the **Channel Tasks** menu and go back to the Channel list by clicking **Channels** in the **Mirth Connect** menu.

Simple xml to EHRCOM xml

Create a new channel named “**simple xml to EHRCOM xml**” of incoming data type “**XML**”. Make sure the channel is **enabled** and **synchronized**.

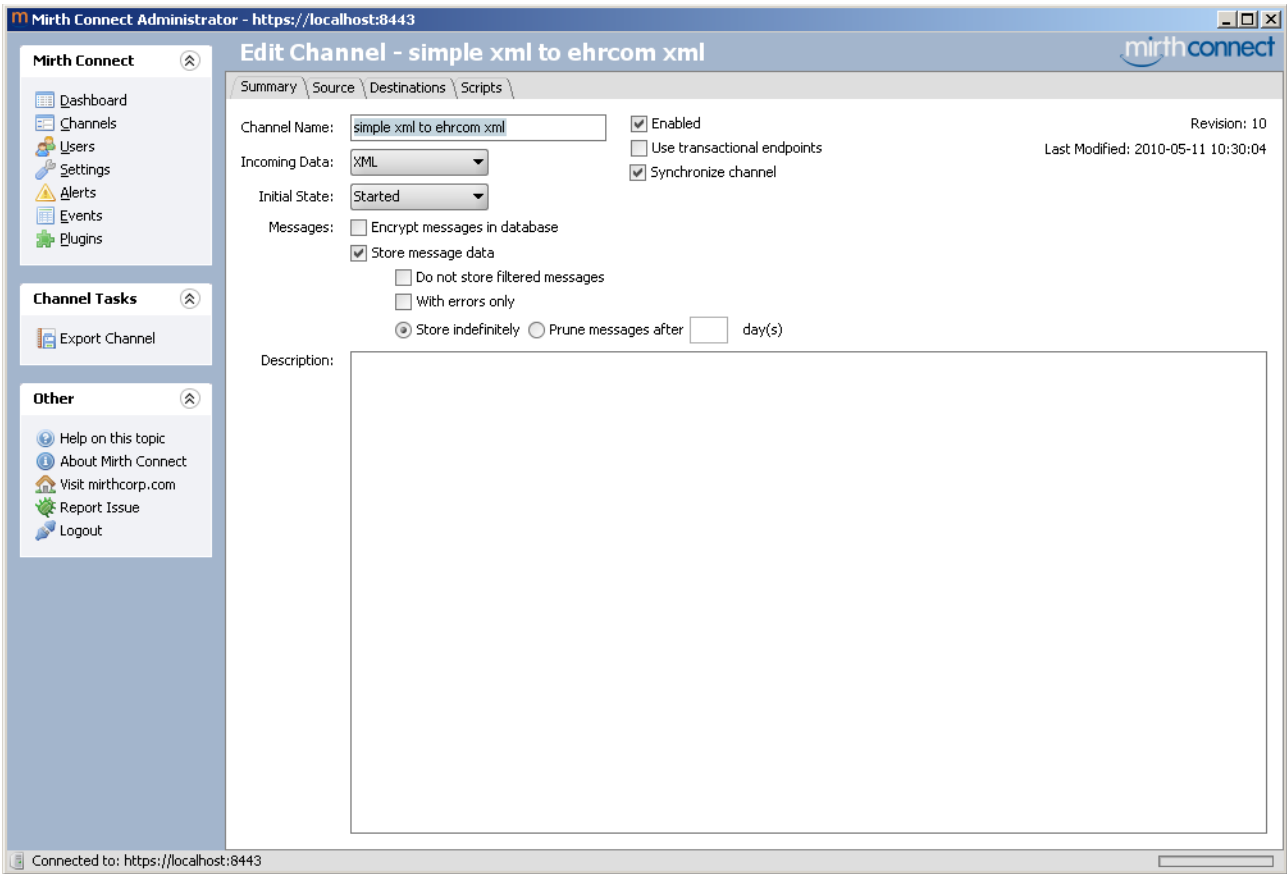


Illustration 14: simple xml to EHRCOM xml - summary

As the channel will read a message provided by another channel, on the **Source** tab panel select the **Channel Reader** connector type.

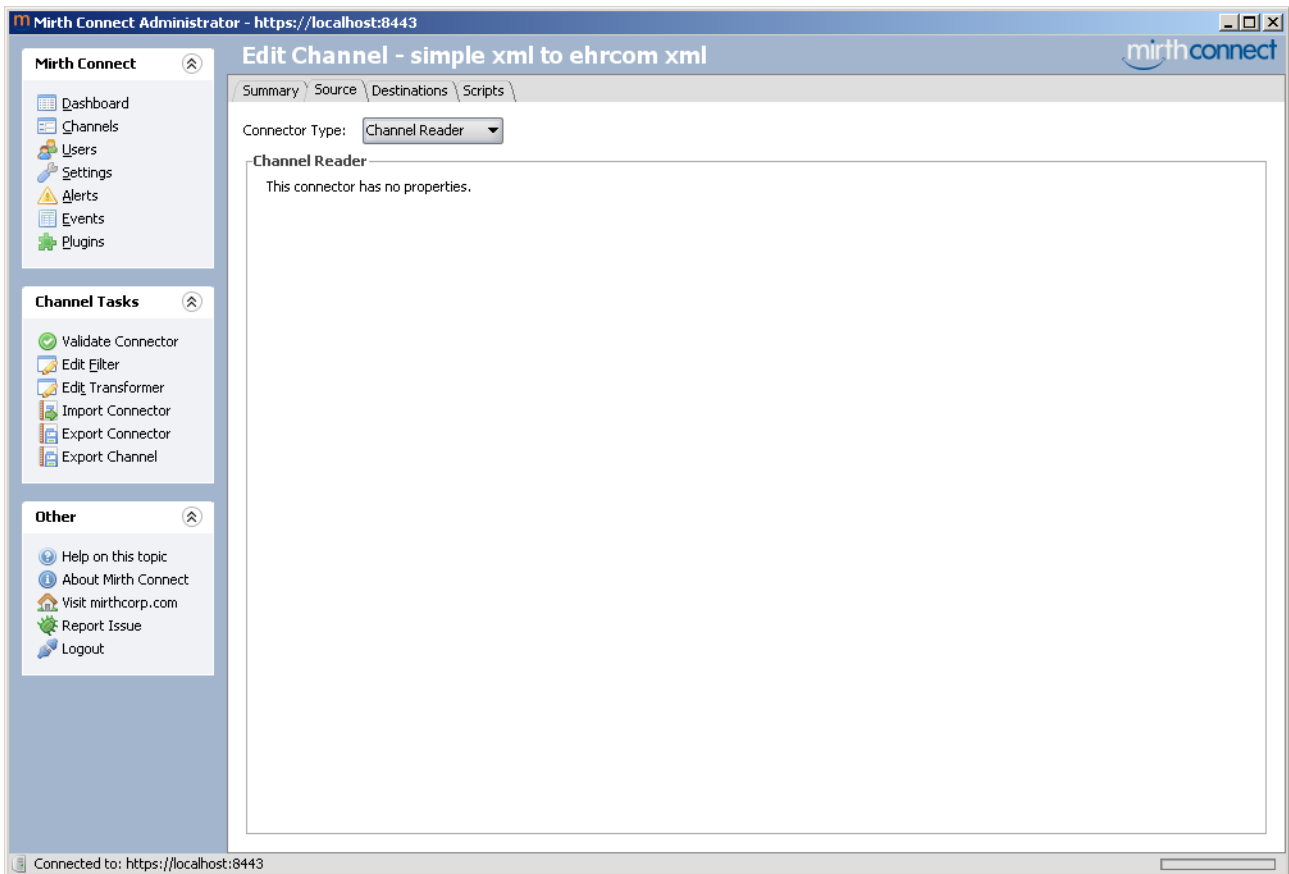


Illustration 15: simple xml to EHRCom xml - source

As the channel will write the EN13606-compliant record extract file (with extension .xml) to the [c:\ehrland](#) directory, on the **Destinations** tab panel create a new destination (from the **Channel Tasks** menu) of type File Writer. Insert [c:/ehrland](#) as the directory. We will fill the other fields once we've created the transformers. You can use the button to control if Mirth Connect is able to access the directory.

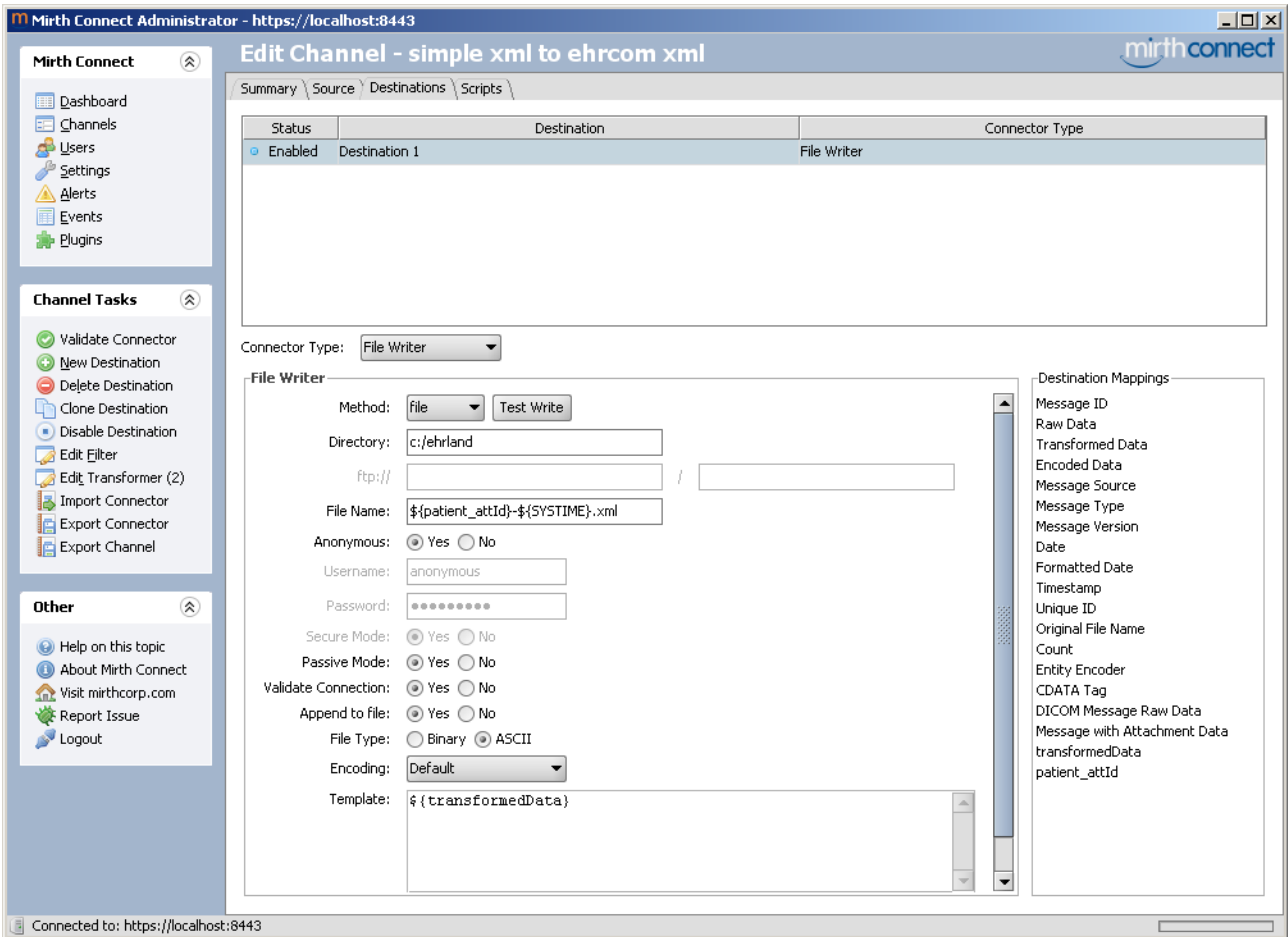


Illustration 16: simple xml to EHRCom xml - destinations

In the **Channel Tasks** menu select **Edit Transformer**. On the right-hand side in the **Message Templates** tab panel, insert a simple xml message as shown in Illustration 17. Mirth Connect will automatically interpret the message and will provide its hierarchical representation in the **Message Trees** tab panel as shown in Illustration 18.



Illustration 17: XML Message Templates

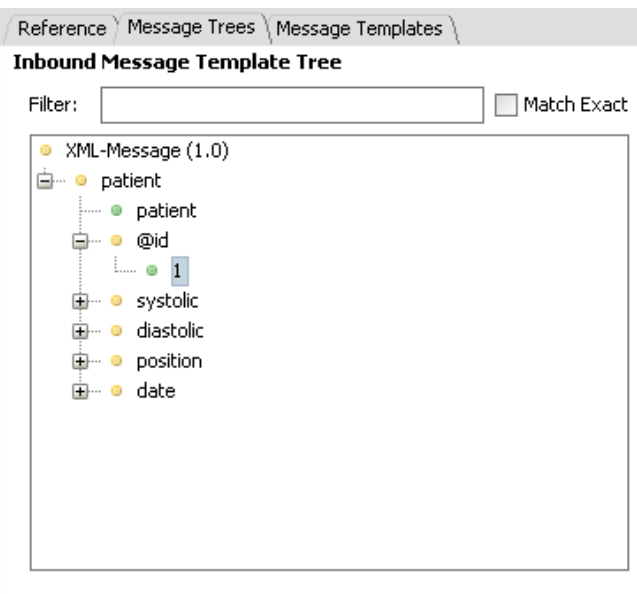


Illustration 18: XML Message Trees

In the Transformer Task menu select **Add New Step** of type **Mapper** to be added to the **Channel Map**.

For the patient identifier we can use the **patient_attId** variable name. In the Message Tree pane, drag the child node of the **@id** node and drop it into the Mapping section. It should automatically write

```
msg['patient']['@id'].toString()
```

For the transformer handling the data conversion from simple xml to EN13606, as it refers to an external java class, we will need to write the mapping by hand. Create a new step of type **Mapper** to be added to the **Channel Map**.

The external class is called `Converter` from the package `xquery`. On its interface it presents a single function:

```
public static String convert(String in, String file)
```

- The `in` argument represents the simple xml message.
- The `file` argument represents the location of the LinkEHR-produced xquery mapping file.
- The returned value is the converted message (EN13606-compliant record extract).

Assuming the xquery mapping file location is c:/ehrland/xquery/CEN-EN13606-ENTRY.Blood_pressure.v1.xquery, and given that the processed message in Mirth Connect can be accessed by calling the `msg` variable, in the Mapping field we should write:

```
Packages.xquery.Converter.convert(msg,'c:/ehrland/xquery/CEN-EN13606-ENTRY.Blood_pressure.v1.xquery')
```

Here is a summary of the 2 variables:

patient_attId msg['patient']['@id'].toString()

transformedData Packages.xquery.Converter.convert(msg,'c:/ehrland/xquery/CEN-EN13606-ENTRY.Blood_pressure.v1.xquery')

The following picture shows the Transformer section after completion of the 2 mapping variables.

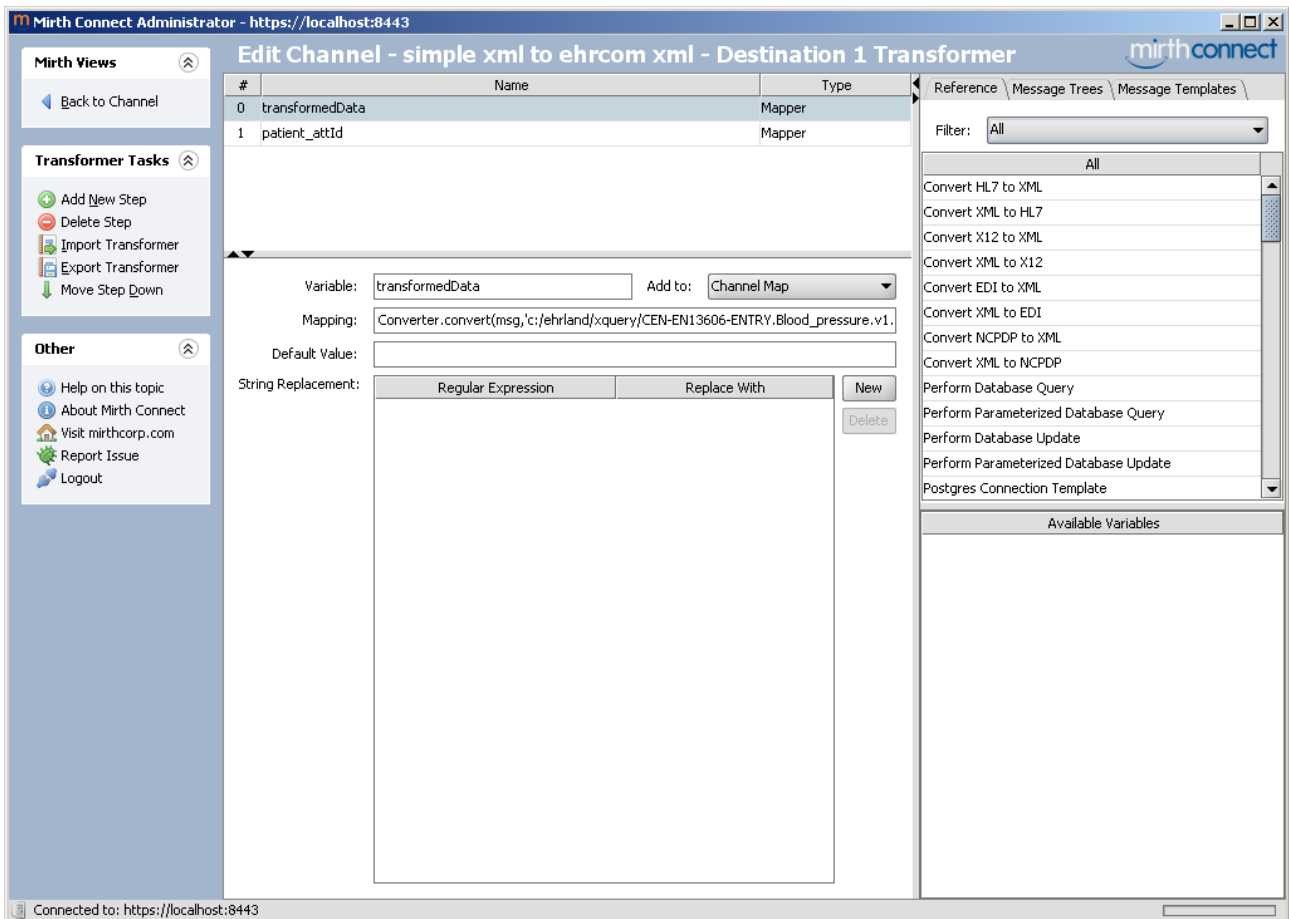


Illustration 19: simple xml to EHRCom xml - transformers

We can go back to the Destinations tab panel and fill the remaining fields.

From the **Destination Mappings** section on the bottom-right-hand side of the window, drag the transformedData variable and drop it into the **Template** field.

Drag the patient_attId variable and drop it into the **File Name** field. Repeat the operation with the Timesamp variable so as to form the file name

$$\${patient_attId}-\${SYSTEMTIME}.xml$$

You can save the channel in the **Channel Tasks** menu and go back to the Channel list by clicking **Channels** in the **Mirth Connect** menu.

Select the **HL7v2 to simple xml** channel, go to the **destinations** tab panel and select **simple xml to EHRCom xml** as **Channel Name**. Save the channel in the **Channel Tasks** menu and go back to the Channel list by clicking **Channels** in the **Mirth Connect** menu.

The configuration is complete; we are now ready to deploy the channels.

Deployment

In the **Channels Tasks** menu, select **Deploy All**.

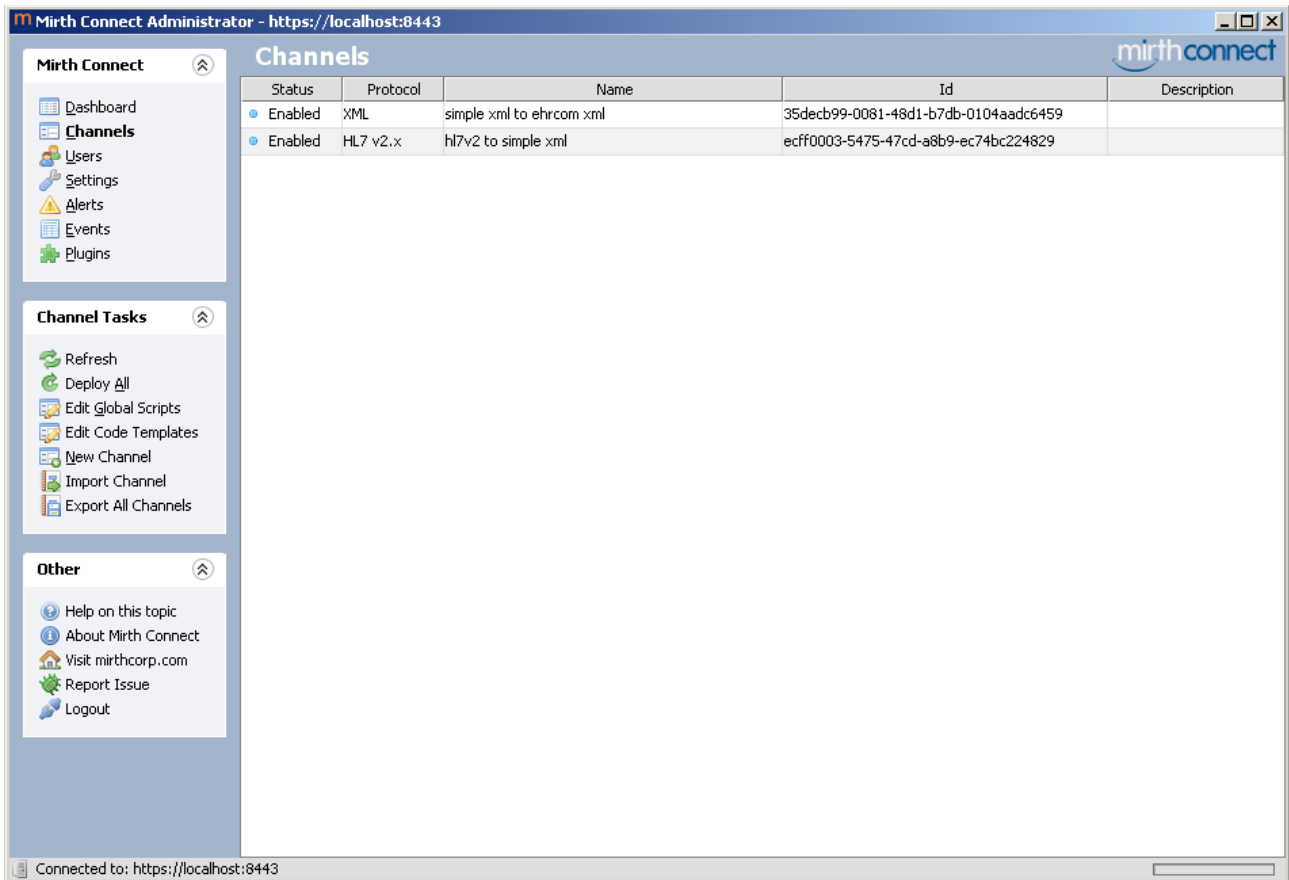


Illustration 20: Channels

Mirth Connect should switch to the Dashboard and deploy the channels as shown in Illustration 21.

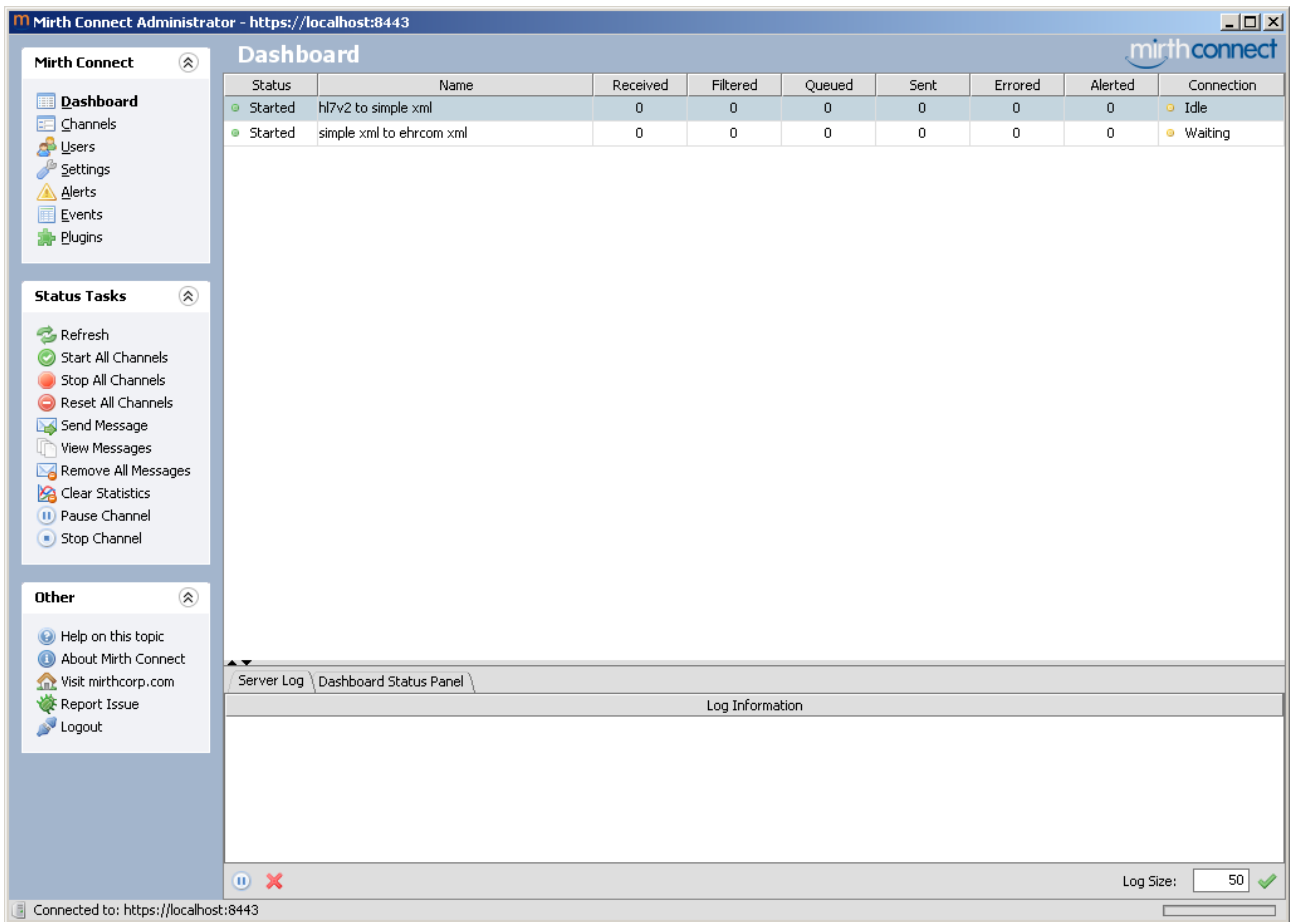


Illustration 21: Dashboard

Copy an HL7v2 message file with a **.hl7** extension in the [c:\ehrland](#) directory . Mirth Connect should detect, process and delete the file after a few seconds, generating an xml file in the same directory. The content of the xml file is an EN13606-compliant record extract containing some of the information of the original HL7 message.

The dashboard should reflect the event by updating the **Received** and **Sent** columns as shown in Illustration 22.

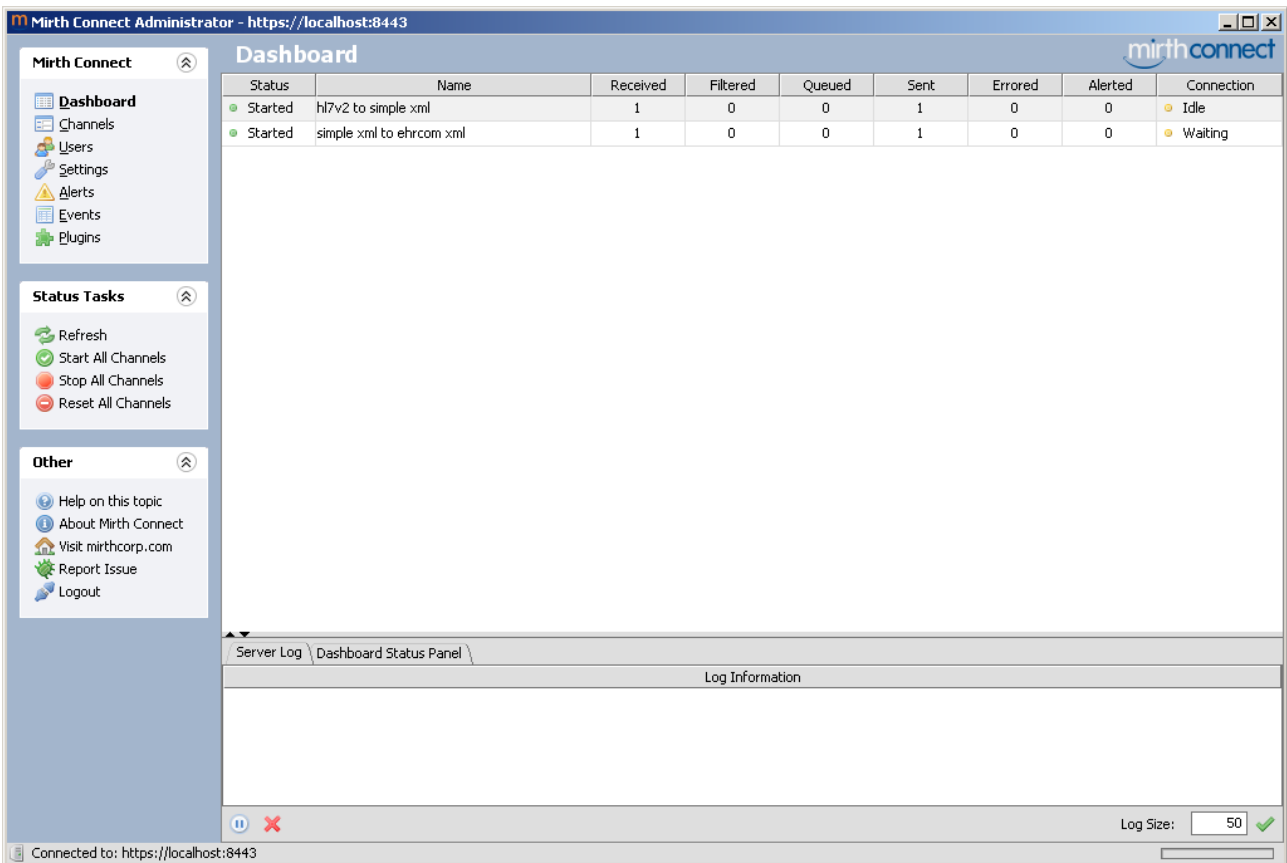


Illustration 22: Dashboard after message processing

It is also possible to test the channels directly from the Mirth Connect Administrator. From the dashboard select the **HL7v2 to simple xml** channel and in the **Status Tasks** menu select **Send Message**. You can perform the same operation by right-clicking on the channel; the **Status Tasks** menu will be presented as a contextual menu as shown in Illustration 23.

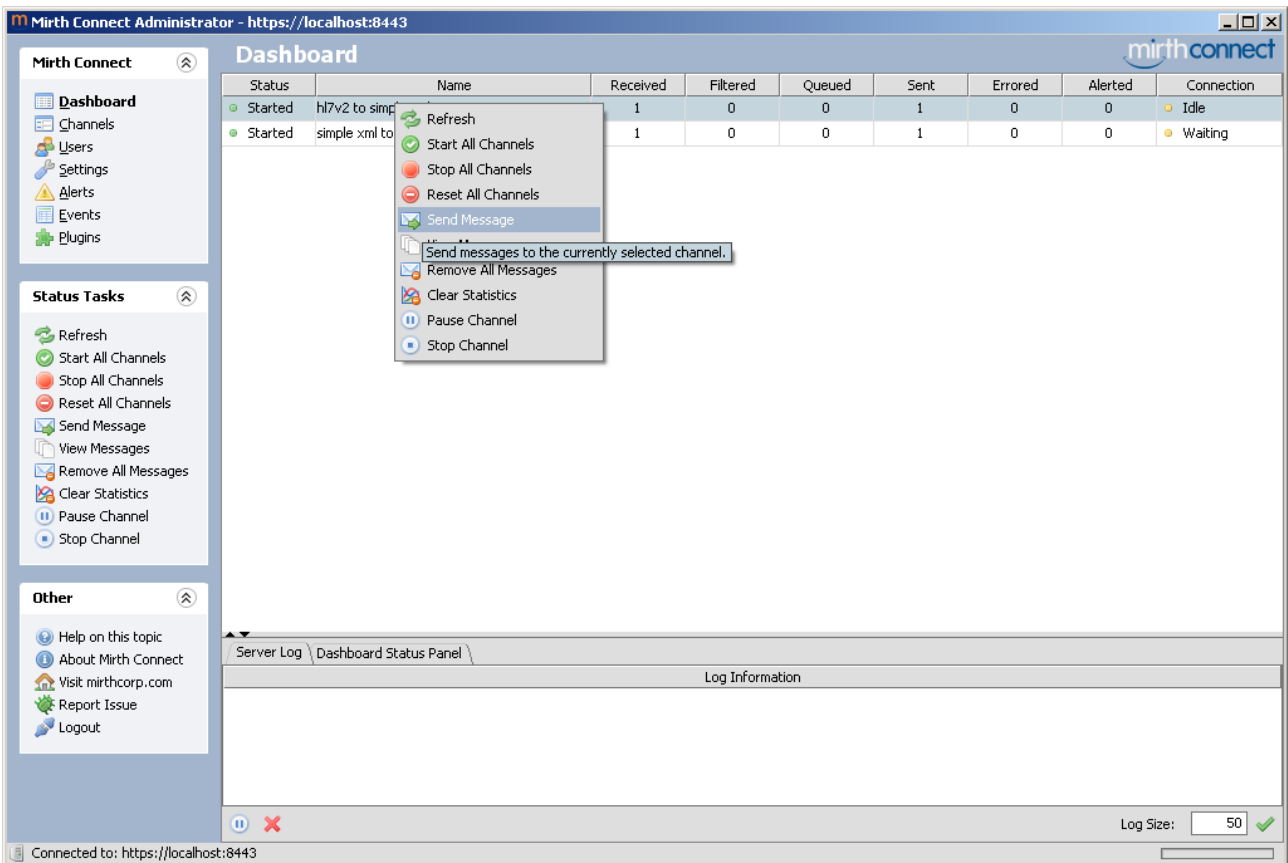


Illustration 23: Status Tasks contextual menu

A Message window will appear. Copy and paste an HL7v2 message into the text area and click the **Process Message** button. The message will bypass the source connector and be processed by the channel as if it was the content of a file with a **.hl7** extension from the <c:\ehrland> directory.

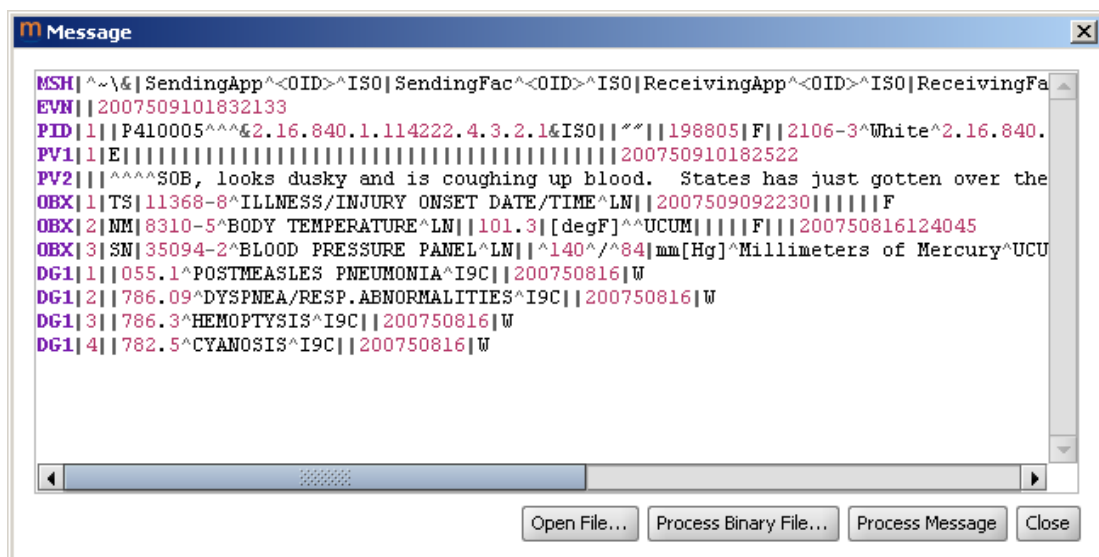


Illustration 24: Send Message Window