Mirth® Connect

3.4 User Guide

April 19, 2016

Contents

1. Introduction to Mirth Solutions 4
2. About Mirth Connect 4
3. Getting Started with Mirth Connect 6
    3.1 Downloading and Installing Mirth Connect 6
    3.2 About the Mirth Connect Server Manager 10
    3.3 Launching the Mirth Connect Administrator 12
4. The Fundamentals of Mirth Connect 14
    4.1 About Mirth Connect Panels 15
    4.2 Using Channels 18
        4.2.1 About Channel Tasks and Group Tasks 18
        4.2.2 About the Edit Channels Interface 19
        4.2.3 Creating a Channel 26
        4.2.4 Editing a Channel 27
        4.2.5 Editing Data Types 28
        4.2.6 Editing Dependencies 30
        4.2.7 Changing a Channel's Status 31
        4.2.8 Deploying Channels 31
        4.2.9 About Channel Grouping 33
        4.2.10 Managing Code Templates 37
            4.2.10.1 Configuring the Code-Template Library Table 38
            4.2.10.2 Creating a Library or a Code Template 39
            4.2.10.3 Editing a Library or a Code Template 41
            4.2.10.4 Importing/Exporting a Library or a Code Template 43
            4.2.10.5 Deleting a Library or a Code Template 45
    4.3 About Messages 46
        4.3.1 Managing Messages 46
            4.3.1.1 Viewing Messages 46
            4.3.1.2 Searching for Messages 50
            4.3.1.3 Reprocessing Messages 54
            4.3.1.4 Importing/Exporting a Message 56
            4.3.1.5 Sending a Message 60
            4.3.1.6 Removing a Message 61
            4.3.1.7 Working with Message Statistics 62
        4.3.2 About the Message-Processing Flow 64
        4.3.3 About Data Types 66
        4.3.4 Setting Message Templates 72
        4.3.5 About Filters 77
        4.3.6 About Transformers 79
        4.3.7 About Connectors 84
            4.3.7.1 Selecting a Connector Type 85
            4.3.7.2 Viewing Connector Statistics 85
            4.3.7.3 About File Connectors 86
            4.3.7.4 About TCP Connectors 89
            4.3.7.5 About Database Connectors 90
            4.3.7.6 About Channel Connectors 91
            4.3.7.7 About HTTP Connectors 91
            4.3.7.8 About Web Service Connectors 93
            4.3.7.9 About JavaScript Connectors 97
            4.3.7.10 About the Mirth Results Connector 98
            4.3.7.11 About JMS Connectors 98
            4.3.7.12 About the Document Writer Connector 98
            4.3.7.13 About Email Connectors 99
            4.3.7.14 About DICOM Connectors 100
            4.3.7.15 About Serial Connectors 103
        4.3.8 Messaging Tools 103

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

4.3.9 Advanced Messaging Topics . . . 103
4.3.9.1 About Message Attachments . . . 104
4.3.9.2 About Advanced Message Routing . . . 106
5. Mirth Connect and JavaScript . . . 107
5.1 About JavaScript . . . 107
5.2 Using JavaScript in Mirth Connect . . . 113
5.3 Using the JavaScript Editor . . . 119
5.4 JavaScript Transformers . . . 122
5.5 JavaScript Filters . . . 123
5.6 JavaScript Connectors . . . 123
5.7 Global and Channel Scripts . . . 125
5.8 About the User and Client APIs . . . 126
6. Mirth Connect Operations . . . 130
6.1 About Mirth Connect Architecture . . . 131
6.2 About Server/Connection Logs and Global Maps . . . 131
6.3 About Import/Export Tasks . . . 134
6.4 About the Command-Line Interface . . . 134
6.5 Managing Users . . . 135
6.6 About Settings . . . 136
6.6.1 About the Server Page . . . 136
6.6.2 About the Administrator Page . . . 137
6.6.3 About the Configuration Map Page . . . 138
6.6.4 About the Database Tasks Page . . . 139
6.6.5 About the Resources Page . . . 139
6.6.6 About the Data Pruner Page . . . 140
6.7 About Alerts . . . 141
6.8 About Events . . . 145
6.9 Managing Extensions . . . 146
7. Commercial Extensions . . . 147
7.1 About the Advanced Alerting Plugin . . . 148
7.2 About the Advanced Clustering Plugin . . . 151
7.3 About the ASTM E1381 Transmission Mode Plugin . . . 152
7.4 About the ASTM E1394 Data Type Plugin . . . 152
7.5 About the Channel History Plugin . . . 153
7.6 About the Email Reader Plugin . . . 155
7.7 About the LDAP Authorization Plugin . . . 156
7.8 About the Message Generator Plugin . . . 157
7.9 About the SSL Manager Plugin . . . 158
7.10 About the Serial Connector Plugin . . . 171
7.11 About the User Roles Plugin . . . 172
7.12 About the Mirth Results Connector Plugin . . . 172
8. Mirth Connect FAQs . . . 173

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

The *Mirth Connect 3.4 User Guide* helps acquaint you with the features and functions of the Mirth Connect software appliance. Click on a section link to learn more about specific aspects of Mirth Connect:

- Introduction to Mirth Solutions – a brief orientation of the Mirth Solutions mission, using open-source software, and what to expect in this guide
- About Mirth Connect - an overview of the MC software appliance including the data types it supports
- Getting Started with Mirth Connect - installing MC, using the MC Server Manager, and launching the MC Administrator
- The Fundamentals of Mirth Connect - an explanation of channels and message processing
- Mirth Connect and Java Script - an explanation of Java Script and its affiliation with MC
- Mirth Connect Operations - how MC works, including descriptions of its operational elements
- Commercial Extensions – for purchase from NextGen® Healthcare
- Mirth Connect FAQS – frequently asked questions about MC.

Introduction to Mirth Solutions

The Mirth Solutions Mission

Mirth Solutions help many of the nation's largest, most respected healthcare entities streamline their care-management processes to satisfy the demands of a regulatory, competitive healthcare industry. With Mirth Solutions, NextGen's goal is to provide the healthcare community a secure, efficient, cost-effective means of sharing health information. The natural product of this aim is a corps of applications – which includes Mirth Connect – flexible enough to manage patient information, from small practices to large HIEs, so our customers and users can work confidently and effectively within the healthcare-delivery system.

Using Commercial OSS (Open-Source Software)

Because the Mirth Solutions appliance suite is the leading open-source toolset for patient-related transactions, NextGen customers benefit from the contributions and testing resources of scores of healthcare professionals. Issues are resolved quickly, and community input is adapted to make Mirth Solutions more helpful and user friendly. If you are hesitant about using open-source software, be assured that NextGen fully backs its entire open-source suite with:

- support services to match every need level and budget
- professional services to complete your integration project quickly and correctly
- hosting services that offer you HIPAA (Health Insurance Portability and Accountability Act)-grade security
- an array of physical and virtual appliances to save you time and resources so you can deploy on a standardized, reliable platform.

About This Guide

If you have taken the Mirth Connect certification training, the information in this guide will likely be familiar to you because its content reflects portions of the training course.

The guide begins with procedures that instruct you how to download and install Mirth Connect and launch the Mirth Connect Administrator. These are followed by details about channels and messages (the two basic elements of MC), and Mirth Connect's relationship with JavaScript. A section about Mirth Connect Operations includes such topics as the server log, connection monitoring, alerts, and data pruning. To conclude, answers to a broad range of frequently asked questions should resolve many practical matters not covered in the guide itself.

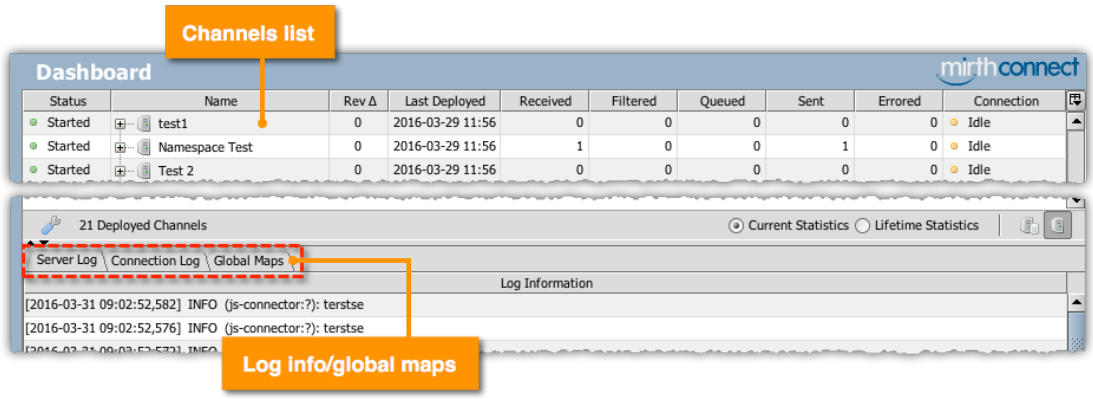
About Mirth Connect

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Like an interpreter who translates foreign languages into the one you understand, Mirth Connect translates message standards into the one your system understands. Whatever "foreign" system sends you a message, MC's integration capabilities expedite:

- filtering – MC reads a message's parameters and passes the message to or stops it on its way to the transformation stage
- transformation – MC converts the incoming message standard to another standard (e.g., HL7 to xml)
- extraction – MC can "pull" data from and "push" data to a database
- routing – MC makes sure messages arrive at their assigned destinations.

Users manage and develop channels (message pathways) via the interface known as the *Administrator*:



Channels are deployed to the MC server where the essential transport connectors are created. MC supports a variety of connectors that are detailed in the About Connectors section.

The Healthcare Interoperability Challenge/Solution

Most often, patient data is exchanged via computer systems (e.g., a doctor's office sends patient records to a hospital, a clinic sends a prescription request to a pharmacy). Such communication is not foolproof. Data can be delayed or lost, and privacy is not always assured, making the transaction less efficient and reliable than it could be. Contributing factors include:

- protocol conflicts between sites
- mismatched versions of record-keeping software
- costly software licensing
- HIPAA privacy and security
- incompatible data due to varied software and communication methods
- lack of control and flexibility related to software use.

MC's flexibility shows in this work-flow diagram, in which a lab's data system sends an HL7 message to MC via an MLLP (minimal lower-layer protocol). MC inserts patient data into an EHR (electronic health record) database, creates a PDF (portable document format) file, and sends an email message with the PDF file attached.

In the next scenario, MC reads patient data from a hospital's EMR (electronic medical record) system. With elements mapped in its own channel, MC generates an HL7 message and sends it to a client for outpatient care. Multiple

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

configurations are available depending on how the channel is constructed.

Getting Started with Mirth Connect

The stand-alone instance of Mirth Connect comes with an installer for Windows, Linux, and Mac OS X.

> If you have purchased a Mirth appliance, disregard the download/install procedures.

This section consists of these related subjects:

- Downloading and Installing Mirth Connect
- About the Mirth Connect Server Manager
- Launching the Mirth Connect Administrator.

Downloading and Installing Mirth Connect

System Requirements

The Mirth Connect Server is a fully standalone application that does not require any sort of application server. The MC Server and MC Administrator are cross-platform applications that require an Oracle JRE. Supported JRE versions appear in this table:

	2.2.3	3.0.0	3.0.1	3.0.2	3.0.3	3.1.x	3.2.x+
JRE 1.6	X	X	X	X	X	X	—
JRE 1.7	X	X	X	X	X	X	X
JRE 1.8	X	—*	—*	X	X	X	X
*JRE 1.8 is not supported by Mirth Connect 3.0.0 and 3.0.1 due to a bug in one of our third-party libraries (MIRTH-3207)							

The Mirth Connect Server requires a database for its configuration and message store. For quick deployment, development, and testing, MC already includes an embedded database (Apache Derby). For production, the latest version of MC supports these databases:

- PostgreSQL 8.3+
- MySQL 5.0+
- Oracle 10gR2+
- SQL Server 2005+

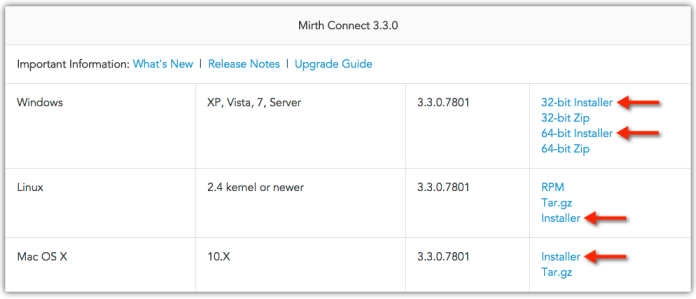
These database requirements apply only to what is used for the MC Server's configuration and message store; they do not pertain to the databases with which MC can interface.

Downloading/Installing the Software

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

The Mirth Connect installer (for Windows, Linux, and Mac OS X) automatically upgrades the previous version of the software installed on your system.

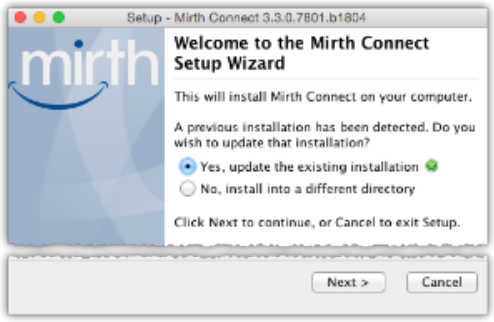
1. Click this link: https://www.mirth.com/Products-and-Services/Mirth-Connect, and, in the **GET STARTED!** section at the bottom of the page that appears, under **DOWNLOAD Mirth Connect Now**, click the **DOWNLOAD NOW** button.
2. On the information form that appears, enter all required fields (and others as desired), and click the **SEND** button.
3. On the **Downloads** page, click the appropriate installer link for your system.



4. When the download is complete, double-click the download file, then double-click the **Mirth Connect Installer** icon that appears.

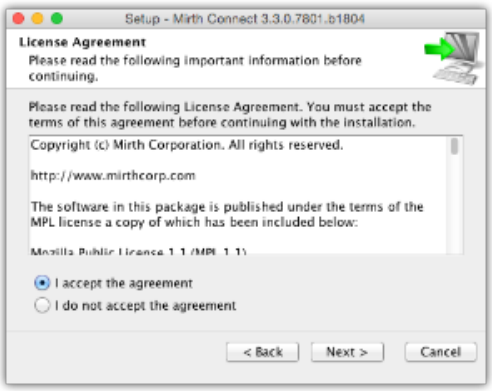
5. On the **Mirth Connect Setup Wizard**, click the **Next >** button.

> **Yes, update the existing installation** is the default selection. If, however, you want to install MC elsewhere on your system, before clicking the **Next >** button, select **No, install into a different directory**, and install MC in your desired location.



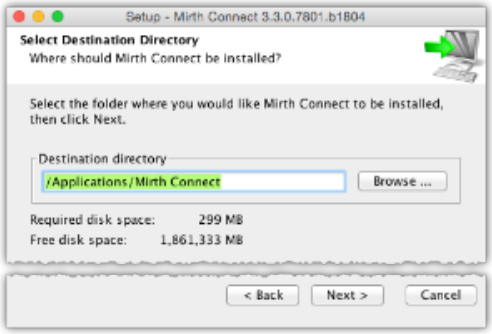
6. In the **License Agreement** window, read the licensing terms, select **I accept the agreement**, then click the **Next >** button.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

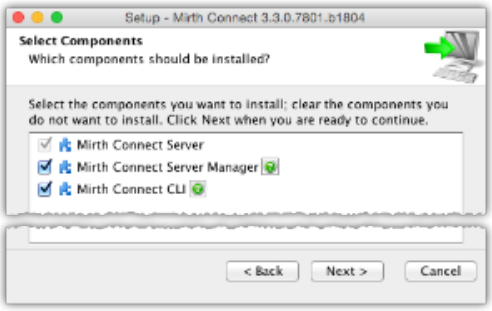


7. In the **Destination directory** section, click the **Browse...** button, find and select the folder in which you want to install MC, then click the **Next >** button.

> (See the graphic in Step 5) If you selected the **Yes…** radio button because you are updating the Mirth Connect application that is already installed on your system, the graphic in this step does not appear. If, however, you selected the **No…** radio button because this is your first installation of Mirth Connect, this graphic appears so you can select the folder in which you want to install the application.



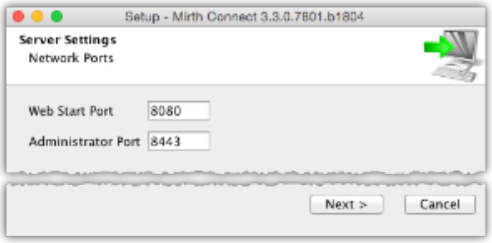
8. On the **Select Components** dialog, select the MC components you want to install, and click the **Next >** button.



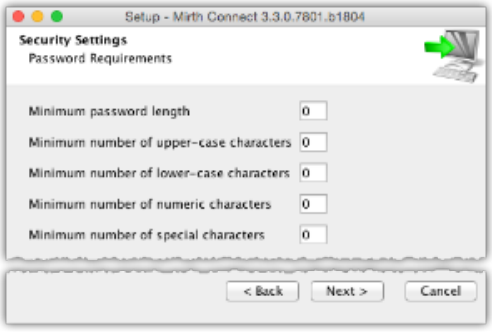
> The **Mirth Connect Server** component is greyed out because it is not an option, but you can select/deselect **Mirth Connect Server Manager** or **Mirth Connect CLI**.

9. On the **Server Settings** – **Network Ports** dialog, enter the port values as needed, and click the **Next >** button.

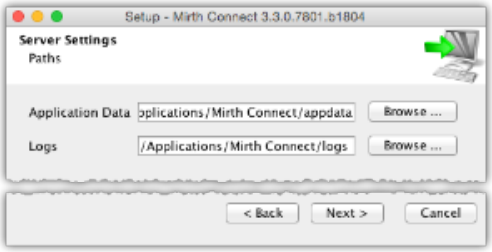
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



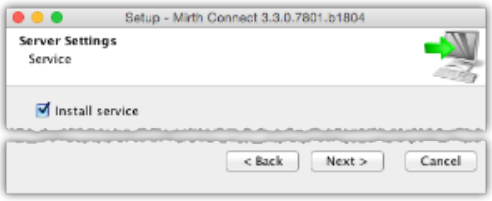
10. On the **Security Settings** – **Password Requirements** dialog, set your password parameters, and click the **Next >** button.



11. On the **Server Settings** – **Paths** dialog, set the server-settings paths for the Application Data and Logs, and click the **Next >** button.

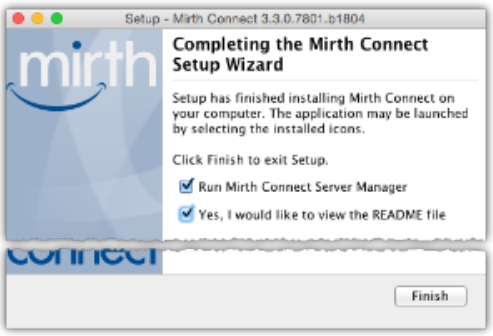


12. On the **Server Settings** – **Service** dialog, click the **Next >** button to install MC.

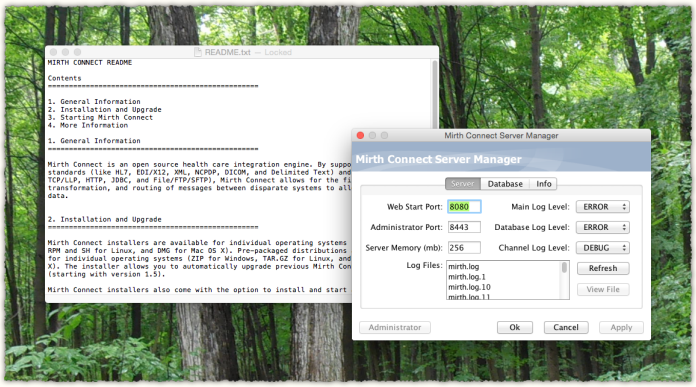


13. Once MC is installed, the **Completing the Mirth Connect Setup Wizard** dialog appears; select/deselect the options as desired, and click the **Finish** button.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Depending on the options you chose in the **Setup Wizard**, the **MC Server Manager** or the **README** file – or both – appear.



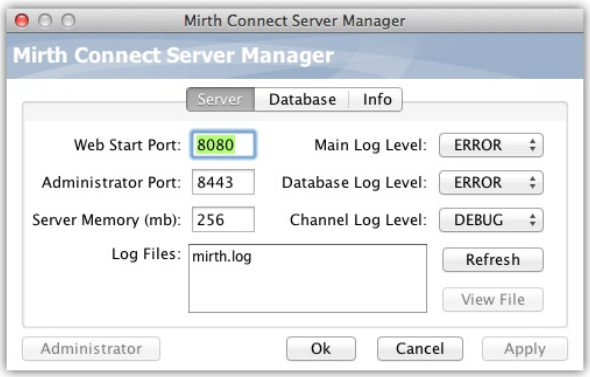
About the Mirth Connect Server Manager

Windows and Mac OS X versions install a **Mirth Connect Server Manager**, an application that resides in the system tray (Windows) or in the **Applications** folder > **Mirth Connect** folder (Mac). The **MC Server Manager** dialog consists of **Server**, **Database**, and **Info** pages:

Server Page

Here you can set ports, server memory, log levels, and view log files.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Function	Description
Web Start Port	Accesses the Java Web Start page for the MC Administrator (default port: *8080*)
Administrator Port	Used by the MC Administrator to communicate with the MC Server (default port: *8443*)
Server Memory (mb)	The server's maximum available memory
Log Level	(Main, Database, Channel) These drop-down menus feature *ERROR*, *WARN*, *INFO*, *DEBUG*, and *TRACE* log levels; depending on the log level, messages of that and all lower levels pass into that level's log when the system logs a certain-level message
Refresh	Updates the list of **Log Files** if any were added/removed since the previous refresh
View File	Select a file in the **Log Files** window to enable this button, which displays the selected file
Administrator	(PC only) Opens the MC login page (inactive on Macs; see Launching the Mirth Connect Administrator)
Ok	Saves your changes then exits the Server Manager
Cancel	Exits the Server Manager without saving your changes
Apply	Applies changes to the field and drop-down settings but does not exit the Server Manager; inactive unless you change any page settings

Database Page

You can manage MC's internal database on the Server Manager's **Database** page.

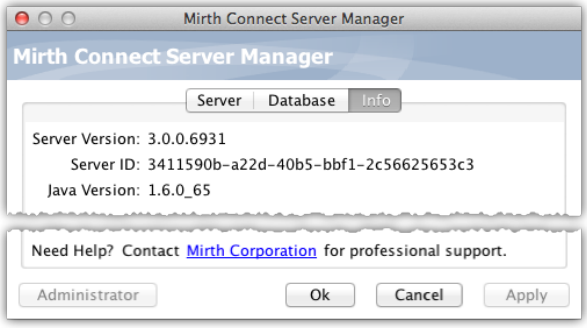
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Function	Description
Type	Select the database on which MC will store data
URL	The JDBC (Java Database Connectivity) URL string, which changes with your selection in the Type menu
Username/Password	The user's unique personal identifier/access code

> MC's default database, **Apache Derby**, is included only to help you set up quickly. Because it is not a production-level database, Mirth Corporation strongly recommends that you do ***not*** use Derby for production.

Info Page

The **Info** page shows the MC server and Java versions as well as the server ID and a link to the Mirth Corporation website.



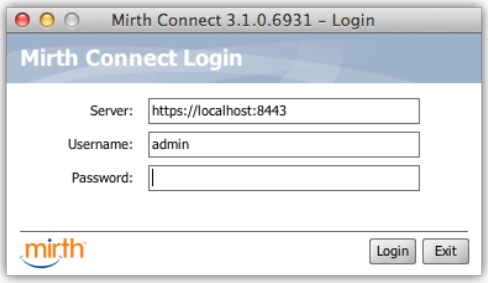
Launching the Mirth Connect Administrator

The MC Administrator is used for all channel development and is a UI composed of menu and view panels that change depending on focus or selected views. Views differ for various tasks (Dashboard, Channel, Settings, etc.).

The first time you log in to the MC Administrator, the **Mirth Connect Login** dialog appears. (The default username

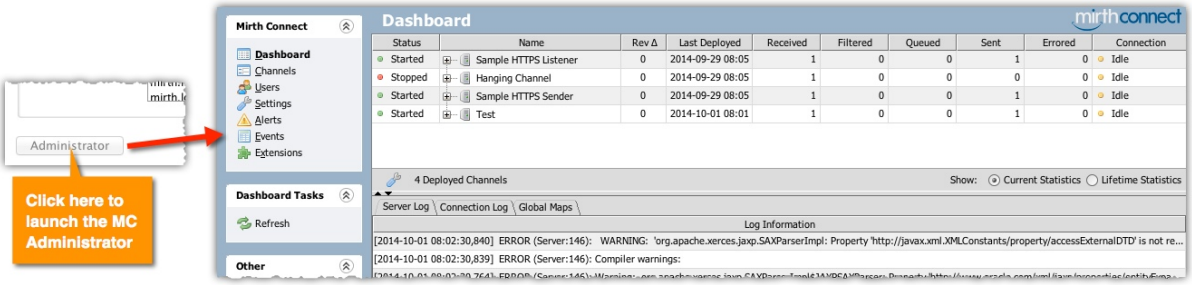
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

and password are both *admin*.)



When you click the **Login** button (previous graphic), the **Welcome to Mirth Connect** dialog appears, on which you will create a default user. Even if you want to keep the default credentials, you must complete all required (*) fields on this dialog.

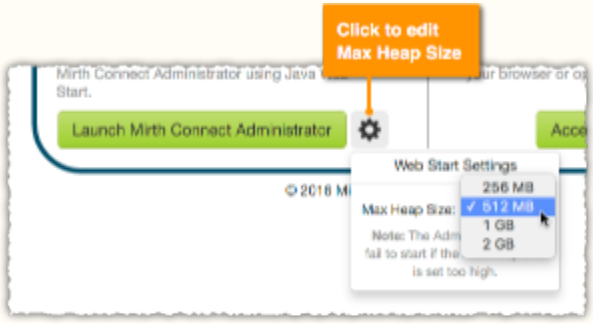
On PC and Linux platforms, you can launch the MC Administrator via the **Administrator** button in the bottom-left corner of the **MC Server Manager** – **Server** page.



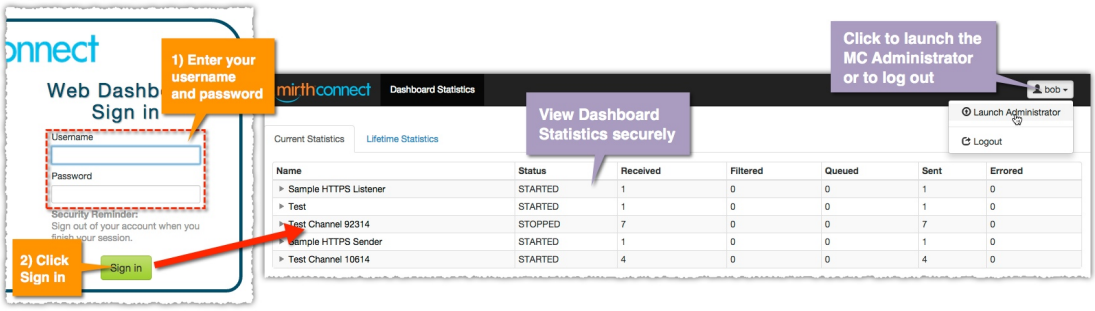
To launch the MC Administrator on a Mac, in the address field of your browser, type *localhost:* followed by the Web Start Port number. (See the **Server Manager** – **Server** page; default: *8080*.) Follow the directions on the Web Start page to launch the Administrator.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Before downloading the administrator, you can select a heap size (256 MB, 512 MB, 1 GB, 2 GB) by clicking the Gear icon next to the Launch button. Edit this setting if client-side operations for your instance use large quantities of memory. (The Administrator may not start if you set your Max Heap Size too high.)



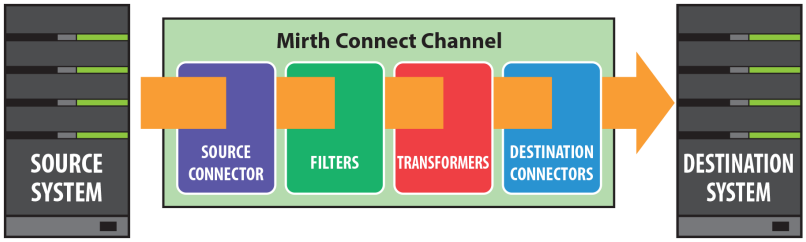
If you click the **Access Secure Site** button for **Web Dashboard Sign in** (previous graphic), the page changes, prompting you to enter a **Username** and **Password** (next graphic). Use **Web Dashboard Sign In** if you want to view your read-only MC statistics securely. You can log in to MC from this page by clicking the user button in the top-right corner of the page and selecting **Launch Administrator**.



The Fundamentals of Mirth Connect

A Mirth Connect channel, with its own filters and transformers, is the medium for the source and destination systems. Messages flow into a source connector, pass through a pre-processor script, are converted to XML, pass through filters (that may be on the source side), then through any transformers. The transformed message is encoded, then flows to the destination connector.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



An **Edit Channel** page features all elements needed to configure a new or reconfigure an existing channel via its **Summary**, **Source**, **Destination**, and **Scripts** tabs. (See About the Edit Channels Interface for details on each of these pages.)

This section explains the relationship between channels and messages. Click a link to learn about these elements, which are vital to working in MC:

- About Mirth Connect Panels
- Using Channels
- About Messages.

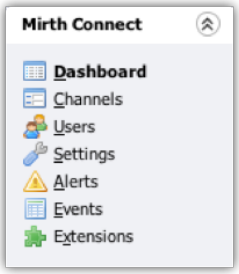
About Mirth Connect Panels

The **Tasks** panel example in this section pertains to the **Dashboard** function in the **Mirth Connect** panel. The other functions in the **MC** panel have their own **Tasks** panels with corresponding tasks. To learn about other elements in the Mirth Connect panel, click one of these links:

- About Channel Tasks and Group Tasks
- Managing Users
- About Settings
- About Alerts
- About Events
- Managing Extensions.

About the Mirth Connect Panel

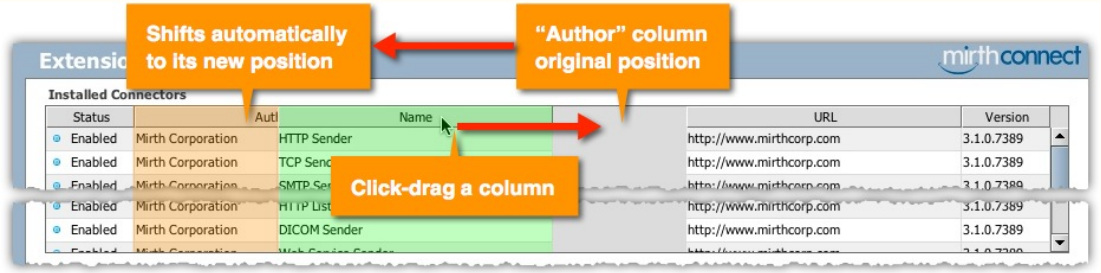
The functions in this panel are the same for all MC pages. Clicking one of these functions opens its corresponding page.

Panel	Function	Description
	Dashboard	Shows deployed channels, and tracks their deployment status (*Started/Stopped/Paused/Halted*) and message statistics (messages received, filtered, queued, sent, etc)
	Channels	Shows all channels (regardless of deployment status), including each's channel status (*Enabled/Disabled*), data type, name, Id, etc
	Users	Shows system users, including such details as username, first/last name, and email address
	Settings	Shows system settings on **Server**, **Administrator**, and **Data Pruner** tabs; other tabs appear for any installed commercial extensions (e.g., **LDAP Authorization**, **SSL Manager**)

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

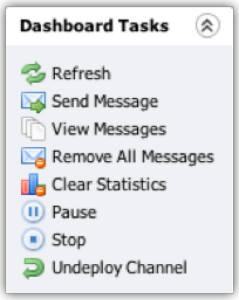
	Alerts	Shows system alerts, including status, description, and Id
	Events	Shows system events, including date/time, name, user at the time of the event, event outcome, and IP address
	Extensions	Shows all system connectors and plugins installed on your system, including each's status, name, author, URL, and version

When you click any function (except **Settings**) in the **Mirth Connect** panel, a list/table relative to the function appears. In each of these, you can rearrange the columns by clicking a column header and dragging the column to the desired position. As you drag a column, the next columns automatically slide in the opposite direction, to make room for the column you are moving.



About the Tasks Panel

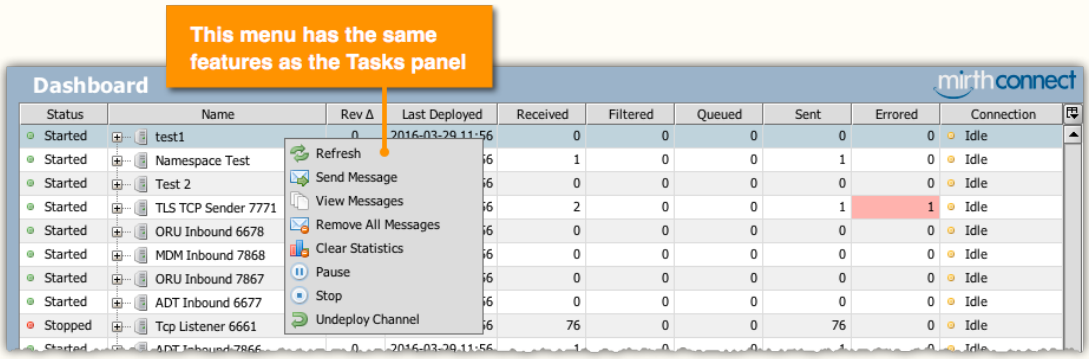
The tasks in a particular **Tasks** panel change with your selection in the **Mirth Connect** panel and your selections on the task's corresponding page.

Panel	Function	Description
	Refresh	Updates the values on the channels list
	Send Message	Sends a message to the selected channel
	View Messages	Displays the selected channel's messages
	Remove All Messages	Removes all messages from the selected channel
	Clear Statistics	Reveals a popup on which you can select the statistics you want to clear from the selected channel
	Pause	Pauses the selected channel; this task appears in the panel if the channel's status is *Started*
	Start/Stop/Halt	**Start**: Starts the selected channel; appears in the panel if the channel's status is *Paused/Stopped*; **Stop**: Stops the selected channel (after the current message is processed); appears in the panel if the channel's status is *Started/Paused*; **Halt**: Stops the selected channel (the channel stops immediately, leaving the message incomplete; when the channel is restarted, processing resumes from the last checkpoint); appears in the panel while a channel is being *Started/Paused/Stopped*

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

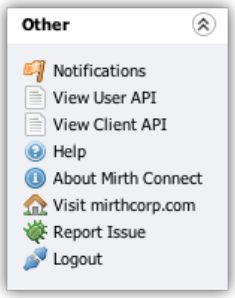
	Undeploy Channel	Removes the selected channel from deployment; the channel will not reappear on the **Dashboard** page until it is deployed via the **Channels** page

instead of moving back and forth between the **Tasks** panel and the work area, you can place the pointer in the appropriate spot on the work area and right-click (PC)/control+click (Mac) the mouse to reveal a menu – at the pointer location – with the same tasks as the **Tasks** panel. This shortcut works for all pages accessed via the functions in the **Mirth Connect** panel. You need to place the pointer on the page's work area to enable the shortcut. Placing the pointer elsewhere (e.g., in a margin or on a greyed-out element or list header) does not enable the shortcut.



About the Other Panel

The functions in this panel are the same for all MC pages and do not affect the workings of the MC Administrator; rather, these functions connect you to sources you may need as you work in MC.

Panel	Function	Description
	Notifications	Opens the **Notifications** dialog on which Mirth posts software updates, commercial-extension releases, critical bugs and workarounds, et al.; you can archive the notifications and set whether or not to show them at login
	View User API	Opens a page on which you can view documentation for the Mirth Connect user API
	View Client API	Opens a page on which you can view documentation for the Mirth Connect client API
	Help	Opens the Mirth Connect wiki, which contains information about working in MC
	About Mirth Connect	Opens a popup containing technical information about MC (e.g., server id, software build date, Java version)
	Visit mirthcorp.com	Opens the Mirth Corporation website
	Report Issue	Opens a page on which you can create a report about an issue concerning MC

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

	Logout	Automatically logs you out of MC; opens the MC login dialog by which you can log in to or exit MC

Using Channels

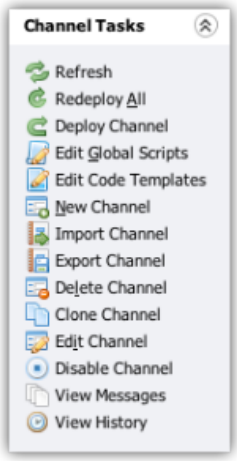
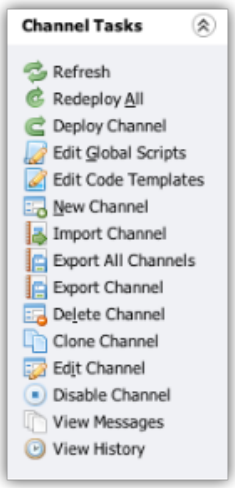
Channels are the conduit between a source system and one or more destination systems. This section consists of information that acquaints you with and instructs you how to work with channels. Click a link to view its topic:

- About Channel Tasks and Group Tasks
- About the Edit Channels Interface
- Creating a Channel
- Editing a Channel
- Editing Data Types
- Changing a Channel's Status
- Deploying Channels.

About Channel Tasks and Group Tasks

This section consists of two tables that explain the **Channel Tasks** and **Group Tasks** panels on the **Channels** page.

About the Channel Tasks Panel

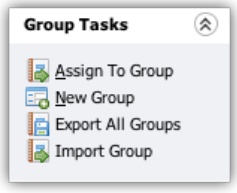
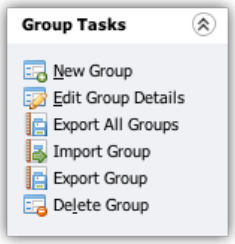
Panel		Function	Description
 *Which tasks appear in the panel depends on the number of channels selected on the Channels List. The above graphic shows all tasks available when one channel is selected in Groups mode.*	 *The above graphic shows all tasks available when one channel is selected in Channels mode.*	Refresh	Updates the data in the Channels List
		Redeploy All	Undeploys all channels and redeploys all *enabled* channels
		Deploy Channel	Deploys the selected channel(s)
		Edit Global Scripts	Edit all scripts that are not channel specific
		Edit Code Templates	Create/manage templates to be used in JavaScript throughout Mirth Connect
		New Channel	Create a channel
		Import Channel	Import a channel from an XML file
		Export All Channels	Export all channels to an XML file
		Export Channel	Export the selected channel to an XML file
		Delete Channel	Removes the selected channel from the list
		Clone Channel	Clone the selected channel

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

		Edit Channel	Modify the selected channel
		Enable Channel*	Enables the selected channel (*only available if you select a disabled channel)
		Disable Channel*	Disables the selected channel (*only available if you select an enabled channel; a disabled cannot be deployed)
		View Messages	Shows the messages of the selected channel
		View History	View previous revisions of the selected channel

About the Group Tasks Panel

For instructions on how to perform the actions in this panel, see About Channel Grouping.

Panel		Function	Description
 *The above tasks appear when you select one or more channels*	 *The above tasks appear when you select a channel group*	Assign To Group	Assign the selected channel(s) to a group
		New Group	Create a new channel group
		Edit Group Details	Change the channel group's name/description
		Export All Groups	Export all channel groups to XML files
		Import Group	Import a channel from an XML file
		Export Group	Export the selected channel group to an XML file
		Delete Group	Delete the selected channel group

About the Edit Channels Interface

[When you create a channel, you use the tabs (Summary, Source, Destination, Scripts) on the **Edit Channel** page to configure the channel.

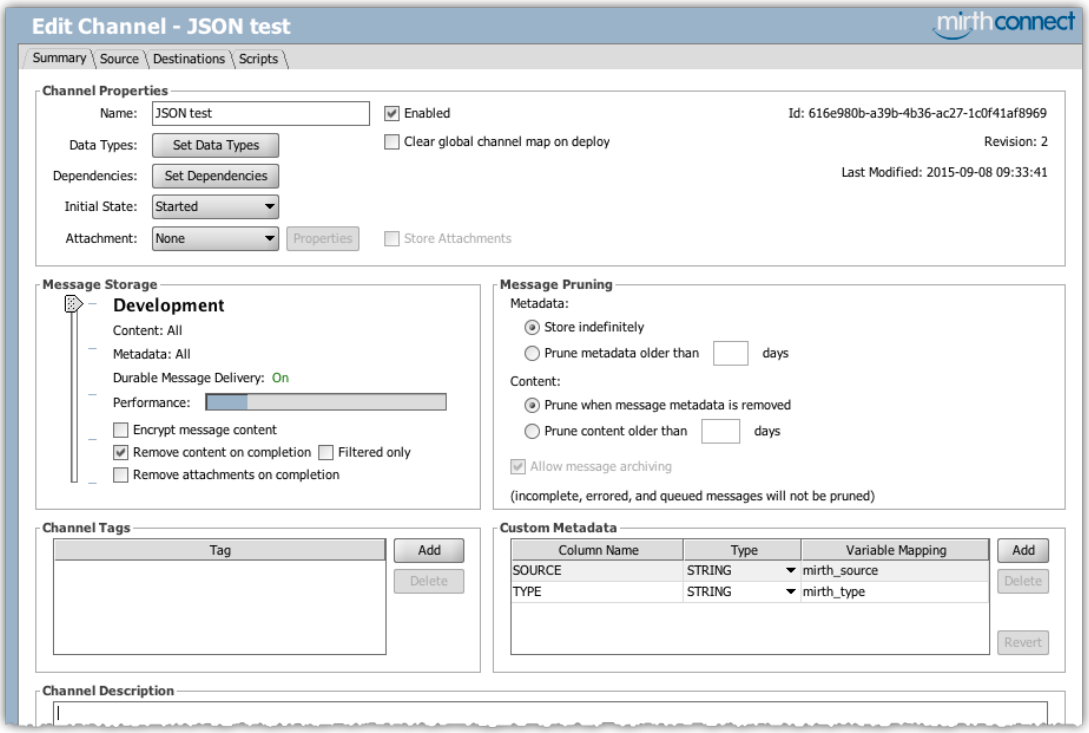
To access an **Edit Channel** page, in the **Mirth Connect** panel, click:

- **Channels**, then double-click the desired channel (or)
- **Channels** > desired channel – **Channel Tasks** panel > **Edit Channel** (or)
- **Channels** – **Channel Tasks** panel > **New Channel**.

Summary Page

A channel's basic properties are defined on this page.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Channel Properties

Feature	Description
Name	The name you give the channel
Data Types	Configure the inbound/outbound properties, and set the data type for the source/destination connectors
Dependencies	Configures the custom libraries and code templates to be included on specific connectors or on the channel scripts; the Channel Dependencies dialog has three tabs: **Code Template Libraries** (contains available code template libraries that you can select as desired), **Library Resources** (select which context(s) to include library resources in: channels, channel scripts, source/destination connectors), and **Deploy/Start Dependencies** (*This channel depends upon*: Your channel depends on these channels; when your channel is *started/resumed*, it verifies that the dependencies are started or prompts you to verify whether or not they should be started; if your channel is *deployed*, it prompts you about whether or not the channel dependencies should be redeployed in order of dependencies first; *This channel is depended upon by*: These channels are dependent on your channel; when your channel is stopped/paused/undeployed/deployed, it prompts you about whether or not the dependents should also take this action (if needed).
Initial State	Sets the state of the channel (*Started/Paused/Stopped*) when first/next deployed
Attachment	Sets the handler used to extract attachments from a message
Properties	(*Regex/JavaScript/Custom*) Lets you set attachment properties (*Regex/Custom*) or script properties (*JavaScript*)
Store Attachments	(Select an attachment type) If checked (default), attachments are stored in the database and are available during reprocessing; if not, attachments are extracted but not stored

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Enabled	If checked (default), the channel can be deployed
Clear global channel map on deploy	If checked (default), clears the global channel map on single channel deploys and full redeploys

Message Storage

Feature	Description
Slide Bar	The selected level (*Development/Production/Raw/Metadata/Disabled*) determines what message content and metadata is stored
Performance	(Status bar) Shows the level of system performance for the selected message-storage level (the lower the slide bar, the lesser the content storage, so the better the performance)
Encrypt message content	(*Development/Production/Raw*) If checked, message content stored in the database is encrypted, meaning that a message can be viewed, but its content is not subject to searching
Remove content on completion	(*Development/Production/Raw*) If checked, message content is removed after a message is processed (excludes errored/queued messages)
Filtered only	(*Development/Production/Raw*) If checked, only content for filtered connector messages is removed (the **Remove content on completion feature** must be checked to enable this associated feature)
Remove attachments on completion	(*Development/Production/Raw*) If checked, attachments are removed after a message is processed (excludes errored/queued messages)

Message Pruning

Feature	Description
Metadata	*Store indefinitely*: message metadata is kept as long as this feature is selected; *Prune metadata older than ? days*: message metadata older than the set number of days is eliminated when the Data Pruner runs
Content	*Prune when message metadata is removed*: message content is deleted when message metadata is deleted; *Prune content older than ? days*: message content older than the set number of days is eliminated when the Data Pruner runs
Allow message archiving	(Select a *Prune metadata/content older than…* feature to enable) If checked, messages in this channel are archived on your system before being pruned from MC (the archiving feature must also be enabled in the Data Pruner settings)

Channel Tags

Feature	Description
Tag	A list of the channel's tags that can be used for filtering
Add/Delete	*Add*: lets you create/assign a tag for the channel; *Delete*: removes the selected tag

Custom Metadata

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Feature	Description
Add/Delete	*Add*: lets you add a column for custom metadata to the channel's messages; *Delete*: removes the selected custom metadata Custom Metadata columns are indexed in Mirth Connect's database, meaning that search results on values in these columns are returned especially quickly. (For details, see Searching for Messages.)
Revert	Returns you to the previously saved custom metadata settings if you have added/deleted custom metadata but have not saved the changes

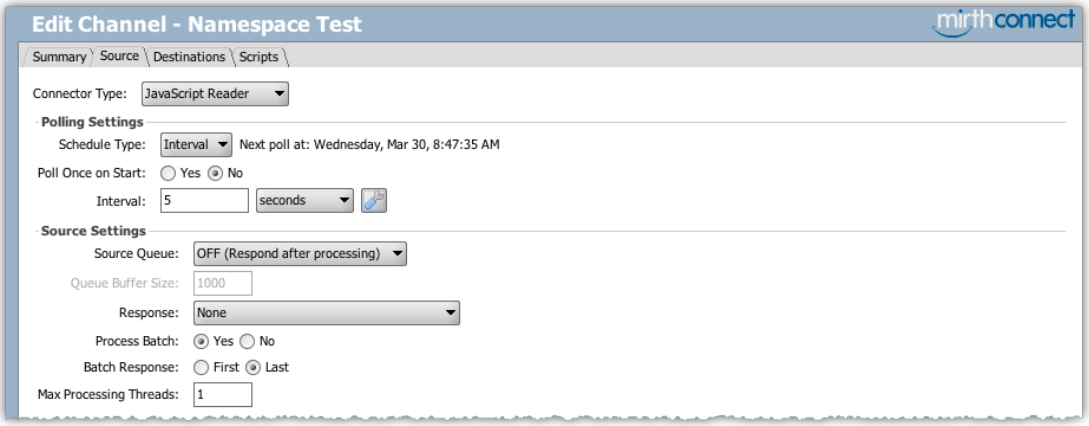
Channel Description

Feature	Description
Text box	Enter information pertinent to the channel

Source/Destinations Pages

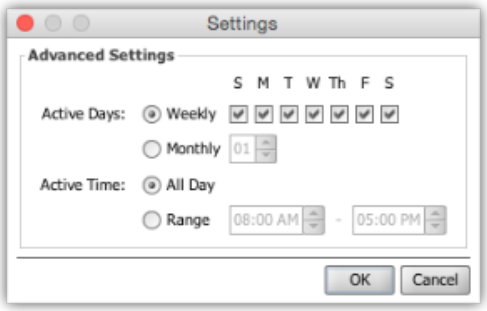
A channel's filters and transformers are defined on these pages.

Source Page



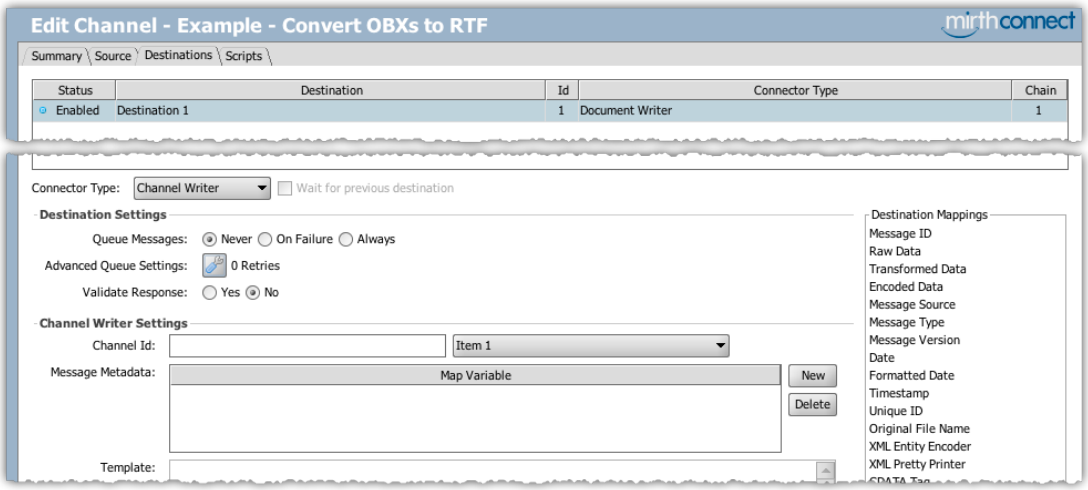
Feature (Source page)	Description
Connector Type	Sets the type of source connector, which in turn determines how the channel receives data

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

(Reader) Polling Settings	**Schedule Type** – Select: *Interval* (polls each "n" units of time); *Time* (polls once per day at the specified time); *Cron* (polls at the specified cron expression(s); **Poll Once on Start** – Select **Yes** to poll immediately one time on start (all subsequent polling follows the specified schedule): **Interval** – The specified repeating time interval in *hours/minutes/seconds/milliseconds* (for milliseconds, units must be less than 24 hours of time); Wrench icon – opens a Settings dialog, by which you can set active polling days and active polling time: 
(Reader) Source Settings	**Source Queue** – *OFF*: processes the message before sending the response; *ON*: queues messages and instantly sends a response (cannot use response from destinations); **Queue Buffer Size** – The source queue's buffer size; the entered value indicates the most messages that can be held in memory if Source Queue is set to ON; **Response** – Selects an auto-generated response, a destination's response, post-processor return value, or response map variable as the response to be sent; select *None* if you do not want to send a response; **Process Batch** – Select *Yes* to enable batch processing; batch messages are only supported if the source connector's inbound properties contain a batch section; **Batch Response** – Each message in the batch contains its own response that is generated via the selected **Response** method; select the response from the batch's first or last message to be sent to the originating system; receives one message, processes multiple messages; **Max Processing Threads** – This value indicates the most messages that can process through the channel at one time (if this value is >1, message order is not guaranteed
Reader/Listener Settings	Depending on the connector type, there can be few, many, or no configurable settings in this section

Destinations Page

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Feature (Destinations page)	Description
Destinations Table	Displays the channel's destination connectors; *Status*: the connector's operational state (Enabled/Disabled); *Destination*: the destination connector's name; *Id*: the connector's unique identifier; *Connector Type*: e.g., Channel Writer, DICOM Sender; *Chain*: indicates which chain each destination executes under; each chain executes *simultaneously*, and each destination in a chain executes *sequentially*
Connector Type	Determines how data is sent from the channel
Wait for previous destination	(Disabled for the channel's original destination or if the channel has one destination) If checked, the destination connector is in the same chain as the previous destination and does not do any processing until the previous destination finishes processing

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

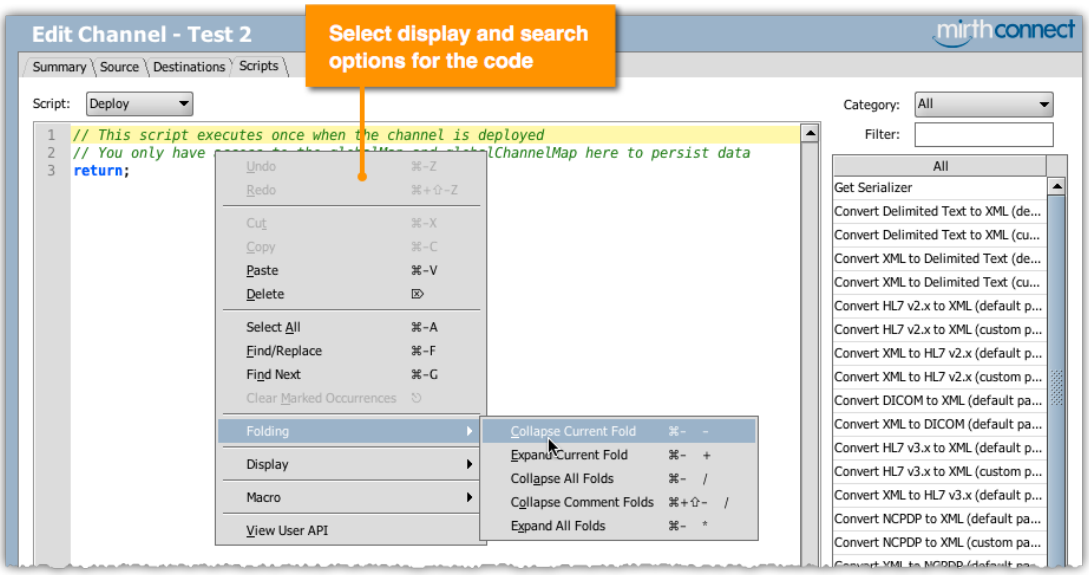
Destination Settings	**Queue Messages** – *Never*: the destination queue is disabled, so messages are not queued; *On Failure*: MC tries sending messages before queuing them; *Always* : messages are always queued; **Advanced Queue Settings**: click the Wrench icon to reveal the **Settings Dialog** which contains these features: *Retry Count Before Queue/Error* – The user-specified number of times the connector tries to send a message before it is queued/errored; *Retry Interval (ms)* – (Enabled if *Attempt First* or *Always* is selected) The timespan (in ms) between message-send attempts; *Rotate Queue* – If Yes, when any message fails to be sent from the queue, the connector "rotates" the message tp the end of the queue and tries to send the next message; *Regenerate Template* – If Yes, regenerates the template and other connector properties by replacing variables each time the connector tries to send the message from the queue; *Include Filter/Transformer* – (Enabled if Regenerate Template is selected) If Yes, the filter and transformer are re-executed before each queue send attempt; *Queue Threads* – (Enabled if *Attempt First* or *Always* is selected) The number of threads that will read from the queue and dispatch messages simultaneously (message order is not guaranteed if this value > 1); *Thread Assignment Variable* – When using multiple queue threads, this map variable determines how to assign messages to specific threads; *Queue Buffer Size*: The destination queue's buffer size; the entered value shows the most connector messages that can be held in memory simultaneously when queuing; **Validate Response** – Select Yes to validate the response; responses can only be validated if the response transformer's inbound properties contains a **Response Validation** section; if validation fails, the message is marked as *Queued* or *Errored*
Writer/Sender Settings	Depending on the connector type, there can be few or many configurable settings in this section

Scripts Page

A channel's scripts are defined on this page. You run the script when the channel is deployed, shut down, and before (preprocessor) or after (postprocessor) a message is processed. Right-click/contol+click in the JavaScript Editor to reveal the context menu, which you can use to select display and search options for the code in the editor.

> The JavaScript Editor also appears on any page where JavaScript is used (e.g., **Edit Transformer** pages if JavaScript is the chosen step type, **Edit Global Scripts** pages, **Edit Code Template** pages). For details on the JavaScript Editor, see Using the JavaScript Editor.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Feature	Description
Script	Determines the channel's script type (*Deploy/Undeploy/Preprocessor/Postprocessor*)
Category	Determines the category (e.g., *Conversion Functions*, *Channel Functions*, *Map Functions*) of reference-code template shown in the window below the **Filter** field; select *All* to show all ref-code templates in the window
Filter	Enter part of a filter type to restrict the window display to filters that match your entry
List	Displays the ref-code templates/categories matching the selection on the **Category** bar

Creating a Channel

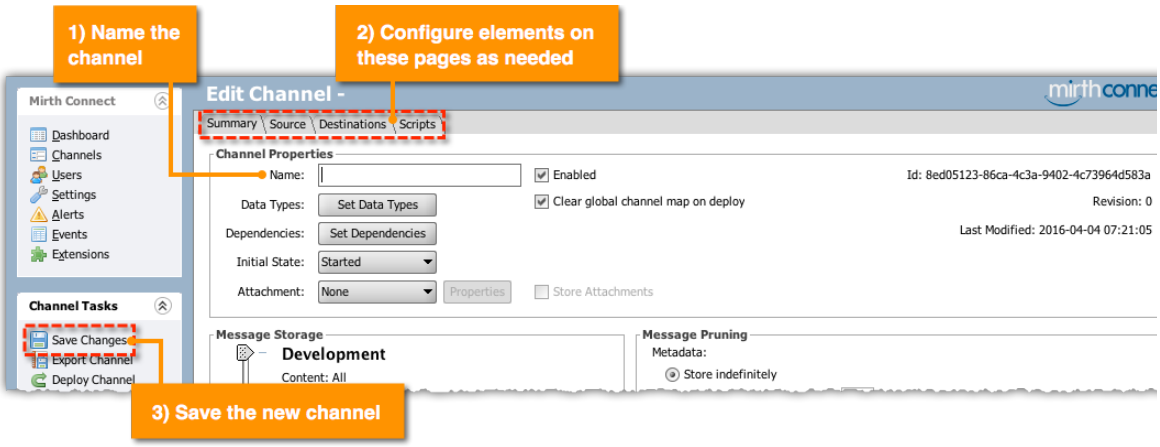
This procedure instructs you how to create a channel. (For details on the new/edit channel pages, see About the Edit Channels Interface.)

1. In the **Mirth Connect** panel, click **Channels**.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

2. On the **Channels** page – **Channel Tasks** panel, click **New Channel**.
3. On the **Edit Channel** – **Summary** page, name the channel, and configure items on this and other Edit Channel pages (**Source**/**Destinations**/**Scripts**) as needed.

> You do not have to save your changes before moving from one **Edit Channel** page (**Summary**/**Source**/**Destinations**/**Scripts**) to another. If, however, you try to leave the **Edit Channel** pages without saving your changes, a **Select an Option** dialog appears, asking *Would you like to save the channel changes?* Click **Yes** to save your changes.



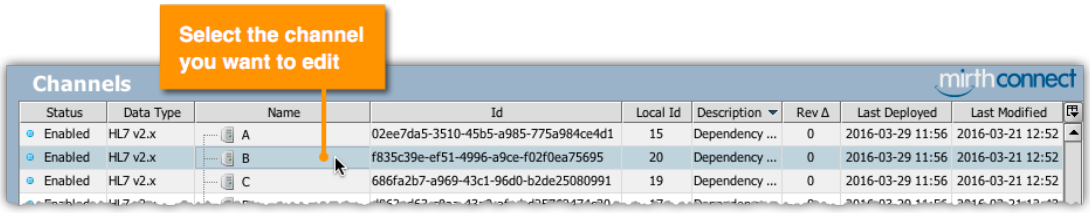
4. In the **Channel Tasks** panel (previous graphic), click **Save Changes**.

> You can automatically assign your new channel to a channels group by putting the **Channels** page in Groups mode (click the Groups button in the bottom-right corner of the page), then selecting the desired channels group before you click the **New Channel** task. When you save your new channel, it will appear in the selected group.

Editing a Channel

This procedure instructs you how to edit a channel. (For details on the edit channel pages, see About the Edit Channels Interface.)

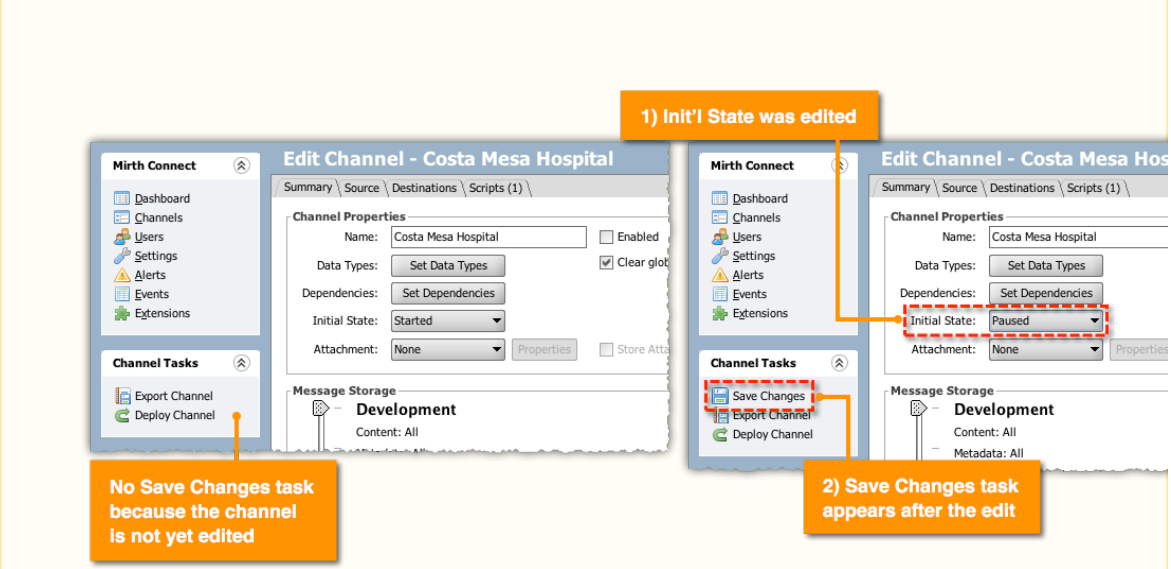
1. In the **Mirth Connect** panel, click **Channels**.
2. On the **Channels** page, select the channel you want to edit.



3. In the **Channel Tasks** panel, click **Edit Channel**.

> The **Edit Channel** page appears with the name of the selected channel in the header. In this example, the left graphic shows that the channel's **Initial State** is *Started*; the right graphic shows that the **Initial State** was edited to *Paused*. Notice that the **Save Changes** function does not appear until a change is made.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



4. Make your changes, and in the **Channel Tasks** panel (previous graphic), click **Save Changes**.

> You do not have to save your changes before moving from one **Edit Channel** page (**Summary**/**Source**/**Destinations**/**Scripts**) to another. If, however, you try to leave the **Edit Channel** pages without saving your changes, the **Select an Option** dialog appears, asking *Would you like to save the channel changes?* Click **Yes** to save your changes.

> **Important**
> For your saved edits to take effect, you need to redeploy the channel. See *Redeploying a Channel* in the Deploying Channels section.

Editing Data Types

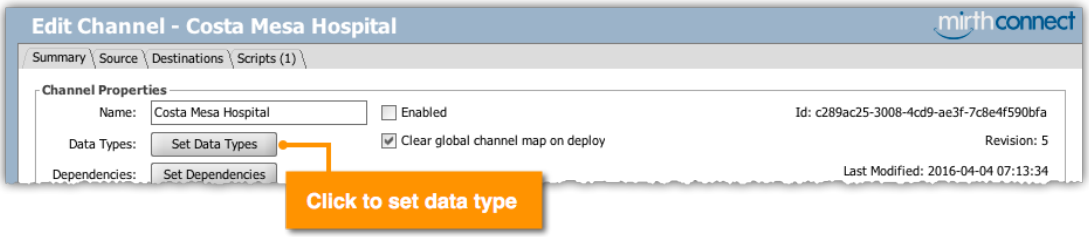
This section consists of procedures that instruct you how to edit the data types for one connector and for multiple connectors (a process called "bulk" editing).

Editing One Connector's Data Types

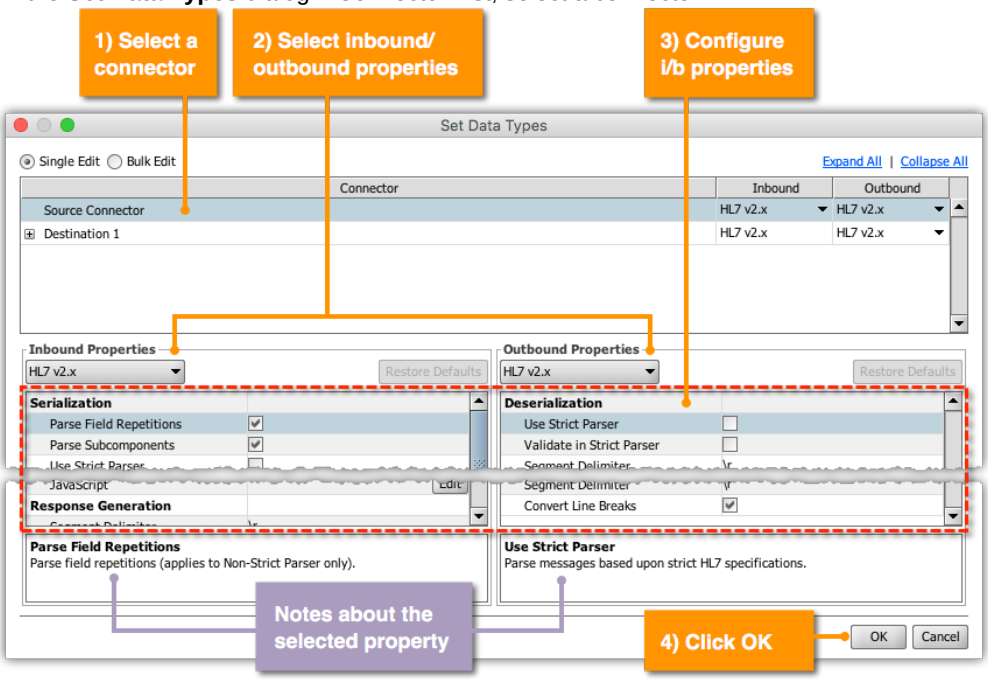
This procedure instructs you how to edit a channel's data types, which is done on an **Edit Channel** page and may include changing data-type properties.

1. In the **Mirth Connect** panel, click **Channels**.
2. Select a channel, and in the **Channel Tasks** panel, click **Edit Channel**.
3. On the **Summary** tab – **Channel Properties** section, click the **Set Data Types** button.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

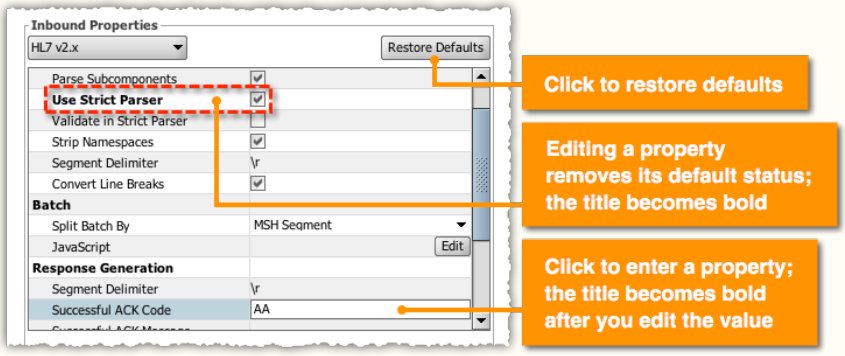


4. On the **Set Data Types** dialog – Connector List, select a connector.



5. In the **Inbound Properties** section, click the Data Type bar (previous graphic), and select a data type, then select its properties. (*DICOM* and *Raw* data types do not have properties.)

When you first select a data type, e.g., *HL7 v2.x*, its default properties appear in the inbound/outbound properties sections. If you check/uncheck any boxes or otherwise select/change any of the defaults, the property is no longer a default, and its name appears in bold text. Any changes to the defaults activate the **Restore Default** button above the properties list. Click this button to restore the default settings.



6. Repeat the previous step for the data type's **Outbound Properties** (graphic, Step 4), and click the **OK** button.
7. In the **Channel Tasks** panel, click **Save Changes**.

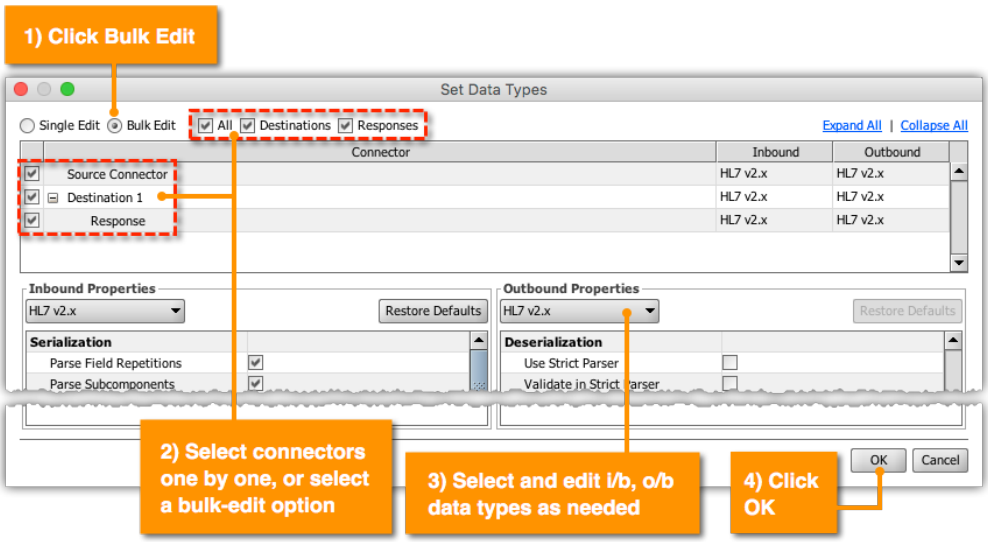
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Bulk Editing Data Types

This procedure instructs you how to edit the data types of more than one of a channel's connectors. With bulk editing, you can edit the data types of all of a channel's connectors or any combination of connectors as needed.

1. In the **Mirth Connect** panel, select **Channels**.
2. On the **Channels** page, select a channel, and in the **Channel Tasks** panel, click **Edit Channel**.
3. On the **Edit Channel** page, and click the **Data Types** bar.
4. In the top-left corner of the **Set Data Types** dialog, click the **Bulk Edit** button.

> This action reveals the **All**/**Destinations**/**Responses** checkboxes next to the **Bulk Edit** button. Checking **All** selects all connectors on the list (Source, Destination, and Response); checking **Destinations** selects all destination connectors; checking **Responses** selects all responses. If you do not want to include all of any connectors, on the Connector List, check the boxes of the connectors you do want to include in your edits.



5. Either check the **All** box or, on the Connector List, check the desired connectors.
6. Select in-/outbound data types, and edit their properties as needed, then click the **OK** button.

Editing Dependencies

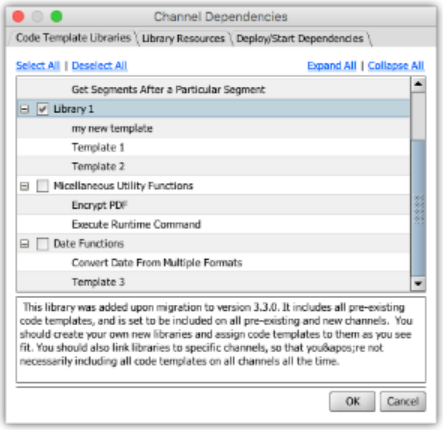
Dependencies are elements a channel depends on to function as desired. This procedure instructs you how to set a channel's dependences.

1. In the **Mirth Connect** panel, click **Channels**, and on the Channels List, select a channel.
2. In the **Channel Tasks** panel, click **Edit Channel**.
3. On the **Edit Channel** page – **Channel Properties** section, click the **Set Dependencies** button.

4. On the **Channel Dependencies** dialog, set the desired dependencies (via the **Code Template Libraries**,

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Library Resources, and **Deploy/Start Dependencies** tabs), and click the **OK** button.

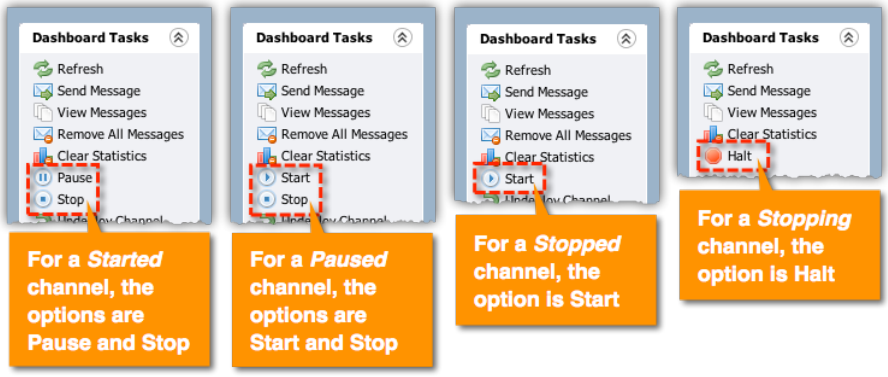


> If you edit a code-template library (selecting or deselecting a library), you will be asked to save the library after you click the **OK** button. These changes are tracked on the **Code Templates** page in the Code Templates table. After you save your changes, the edited library's **Revision** value increases by 1.

Changing a Channel's Status

You can *Stop/Start/Pause* deployed channels on the **Dashboard** page's Channels List. (Undeployed channels do not appear on the Channels List, and their status cannot be changed until they are redeployed.) A *Paused* channel does not receive new messages but keeps processing messages in its Source/Destinations queues. A *Stopped* or *Halted* channel cannot receive messages.

1. On the **Dashboard** page, select a channel.
2. In the **Dashboard Tasks** panel, click the desired task. (The panel's available tasks depend on the channel's status.)



3. In the **Channel Tasks** panel, click **Save Changes**.
4. In the **Mirth Connect** panel, click **Channels**, and in the **Channel Tasks** panel, click **Deploy Channel**.

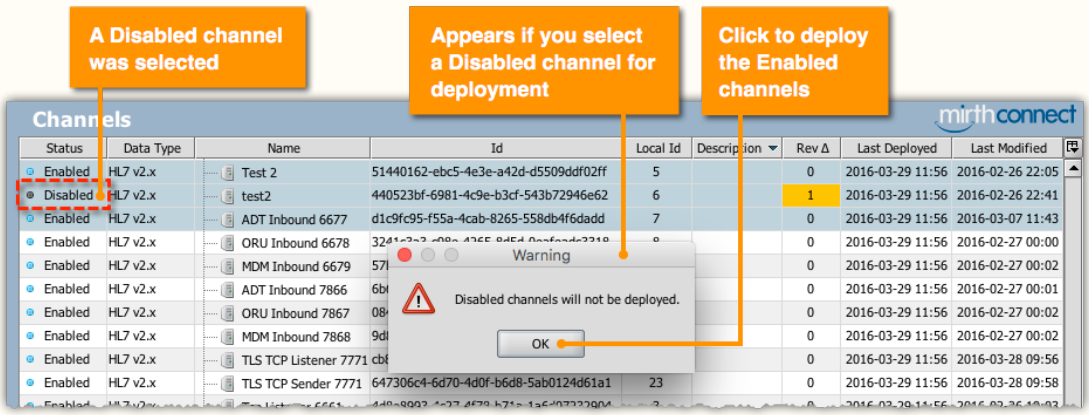
> The **Dashboard** page appears, showing the selected channel's new status.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Deploying Channels

This section consists of procedures that instruct you how to deploy and undeploy a channel, and redeploy an edited channel. (For details, see About the Edit Channels Interface.) Only *Enabled* channels can be deployed. Channels are *Enabled/Disabled* on the **Channels** page; channels are *Deployed* on the **Channels** page and *Undeployed* on the **Dashboard** page.

An existing channel must be *Enabled* before it can be deployed. (New channels are enabled by default.) If you select a *Disabled* channel for deployment, the **Deploy Channel** task does not appear in the **Channel Tasks** panel. If you select multiple channels for deployment, one or more of which is *Disabled*, when you click **Deploy Channel**, a **Warning** dialog appears, informing you that *Disabled channels will not be deployed*. When you click the **OK** button, the selected *Enabled* channels are deployed, and the **Dashboard** appears, displaying the newly deployed channels.



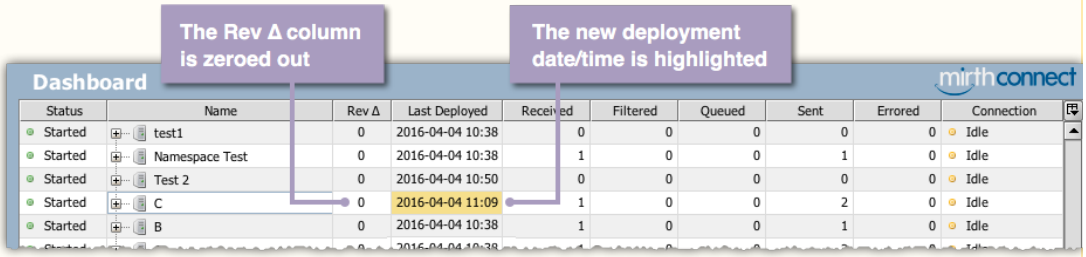
Deploying a Channel

New, edited, and undeployed channels need to be deployed (to the server) before you can use them. (To deploy multiple channels, select the desired channels, and click **Deploy Channel** in the **Channel Tasks** panel; MC does not provide a "Deploy Channels" task.) This procedure instructs you how to deploy a channel, new or otherwise.

1. In the **Mirth Connect** panel, click **Channels**.
2. Select a channel, and in the **Channel Tasks** panel, click **Deploy Channel**. (The next graphic, which represents the channel after editing but before redeployment, shows that edits were saved twice since the channel was last deployed.)

The **Dashboard** page appears, showing the newly deployed channel with deployment date/time highlighted and a zero in the **Rev** column. Deployment zeroes out the channel's Rev value, which represents how many times edits have been saved since the channel's previous deployment.

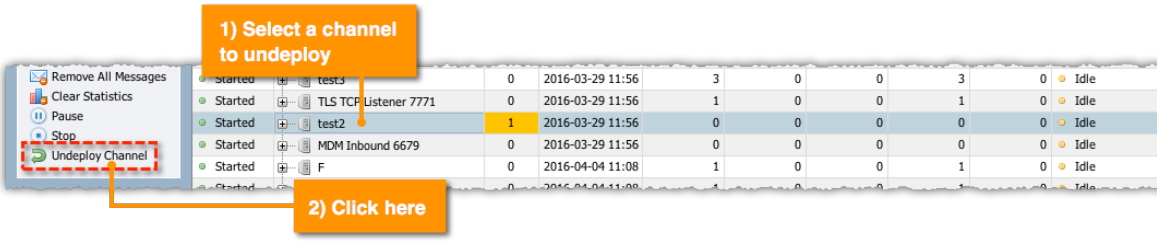
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Undeploying a Channel

If, for example, you no longer need a channel, but protocol dictates that you do not delete it, or if current needs do not include a channel that may be useful later, you can "undeploy" the channel. Such channels do not appear on the **Dashboard** page nor do they send/receive messages. You do not need to disable an undeployed channel, although to be deployed, a channel must be enabled.

1. On the **Dashboard** page, select the channel you want to undeploy.



2. In the **Dashboard Tasks** panel (previous graphic), click **Undeploy Channel**.

> The channel disappears from its previous position on the Channels List. (You can access the undeployed channel on the **Channels** page.)

Redeploying All Channels

While the **Deploy Channel** task does not appear in the **Channel Tasks** panel until you select one or more channels, the **Redeploy All** task always appears in this panel. Moreover, to use the **Redeploy All** task, you do not need to select all channels because by virtue of its function, the task includes all channels when it executes its function. When you click this task, all deployed channels are undeployed, then all *Enabled* channels are immediately redeployed. **Redeploy All** is convenient for instantly updating all deployed *Enabled* channels at once instead of taking the time to go down the Channels List, redeploying *Enabled* channels one by one.

> Although the task name implies that every channel will be redeployed, the general rule concerning channel deployment still applies; that is, only enabled channels can be deployed. (See this Note from earlier in the **Deploying Channels** section.)

About Channel Grouping

This tool helps you organize your channels into groups for better management, especially if you have a very large number of channels. For instance, you may find it advantageous to separate channels based on departments (radiology, oncology, laboratory) or on the source of their messages (general hospitals, psychiatric hospitals). Once you have established channel groups, you can export them to an external folder. You can also import a channel from

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

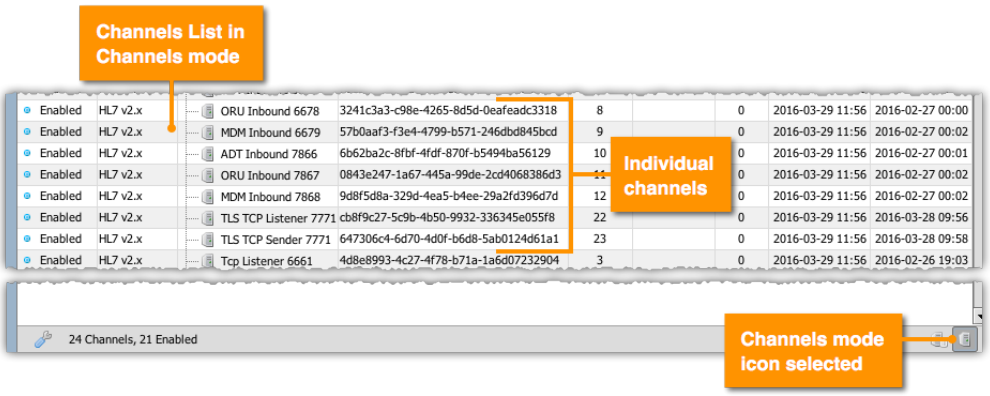
an external folder. Those tasks are performed from the Channels page. This section contains an explanation of grouping tasks and directions on how to create a channel group and how to add channels to a group.

About the Channels/Groups Modes

On the Channels page, you can display the Channels List in Channels mode or in Groups mode. You can switch between these modes via the Groups and Channels buttons in the bottom-right corner of the Channels page.

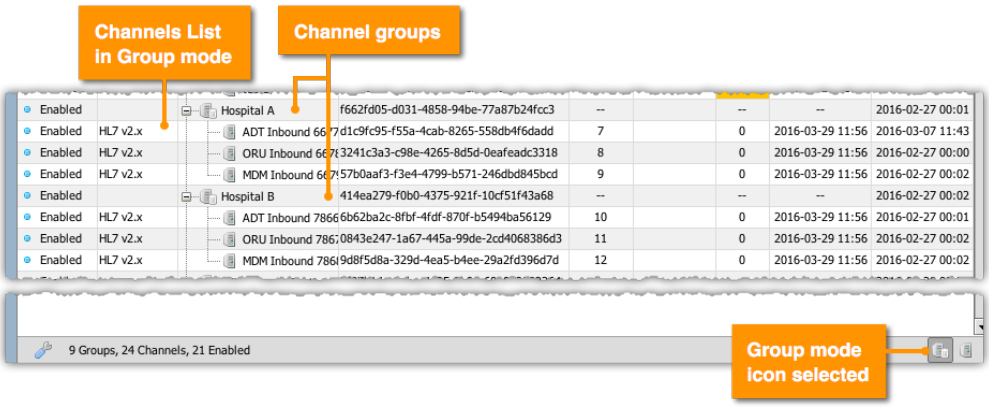
Channels Mode

In Channels mode you see individual channels irrespective of the channels group in which they might be.

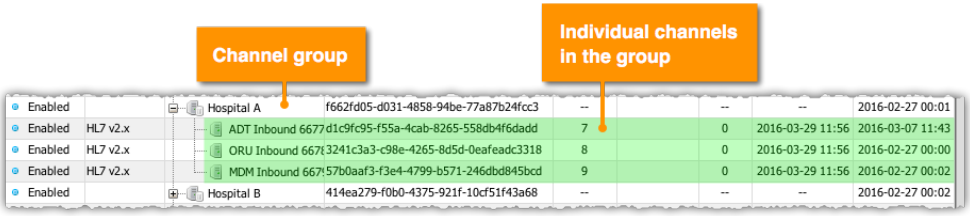


Groups Mode

In Groups mode you see the "parent" channel group and under each group, its "child" channels.



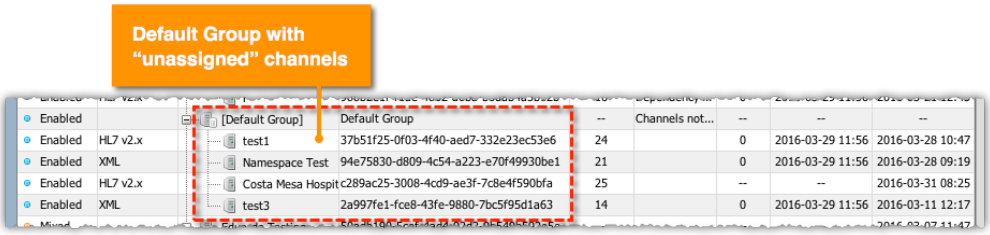
On the **Channels** page (in Groups mode), your channel groups appear with a +/- icon next to their names. Click the icon to expand (view individual channels) or collapse the group.



In Groups mode, you will see the **Default Group**, which contains all channels that have not been assigned to a

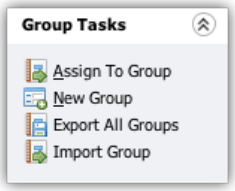
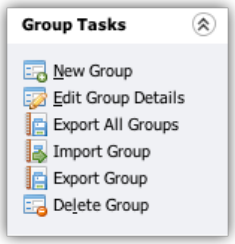
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

user-created channel group. If you create a channel but do not assign it to a group, that channel is automatically placed in the Default Group.



About the Group Tasks Panel

This table describes the various actions in the **Group Tasks** panel.

Panel		Function	Description
 The above tasks appear when you select one or more channels	 The above tasks appear when you select a channel group	Assign To Group	Assign the selected channel(s) to a group
		New Group	Create a new channel group
		Edit Group Details	Change the channel group's name/description
		Export All Groups	Export all channel groups to XML files
		Import Group	Import a channel from an XML file
		Export Group	Export the selected channel group to an XML file
		Delete Group	Delete the selected channel group
		Save Group Changes*	Saves edits to a group or saves a new group (*this task appears in the panel if you edit group details or add a new group)

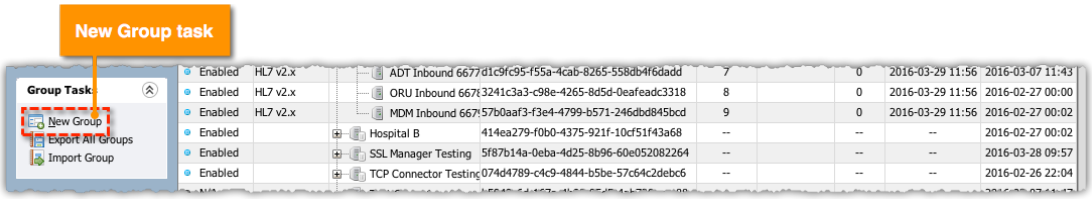
Creating a Channel Group

Perform this procedure once you have decided how you want to group a number of channels.

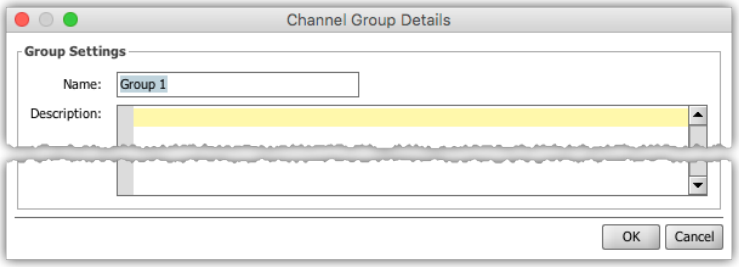
> You need to be in Groups mode to perform this procedure; the **Group Tasks** panel does not appear in Channels mode; click the Groups button in the bottom-right corner of the page if needed.

1. Navigate to the **Channels** page, and in the **Group Tasks** panel, click the **New Group** task.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



2. On the **Channel Group Details** dialog, in the **Name** field, enter a name for the group, and in the **Description** text box (if desired), enter a unique description of the group.



3. Click the **OK** button, and in the **Group Tasks** panel, click **Save Group Changes**.

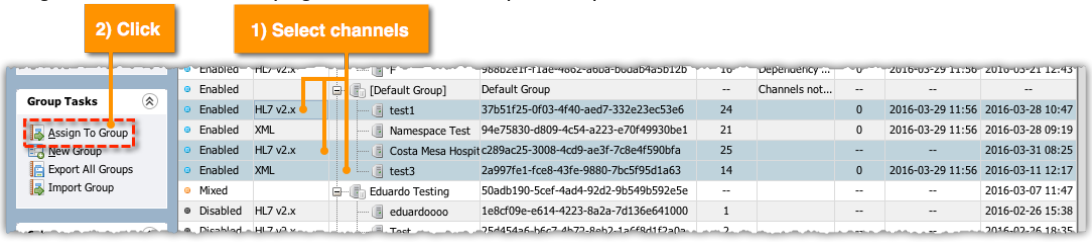
> The new channel group appears on the **Channels** page with a status of *N/A*. This status remains until you add channels to the group and save the changes, at which time the group status changes to *Enabled*.

Adding a Channel To a Group

This procedure instructs you how to add a channel to a group. You can add more than one channel to a group simultaneously. You cannot add the same channel to more than one group. If you add to another group a channel that is already in a group, that channel will no longer be in the previous group.

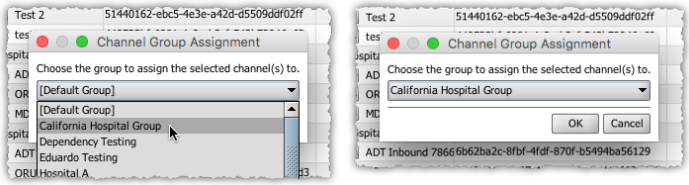
> You need to be in Groups mode to perform this procedure; the **Group Tasks** panel does not appear in Channels mode; click the Groups button in the bottom-right corner of the page if needed.

1. Navigate to the **Channels** page, and select one (or more) channels.



2. In the **Group Tasks** panel (previous graphic), click **Assign To Group**.
3. On the **Channel Group Assignment** dialog, click menu bar, and selected the desired channels group, then click the **OK** button.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



The channel(s) appear in the selected group.

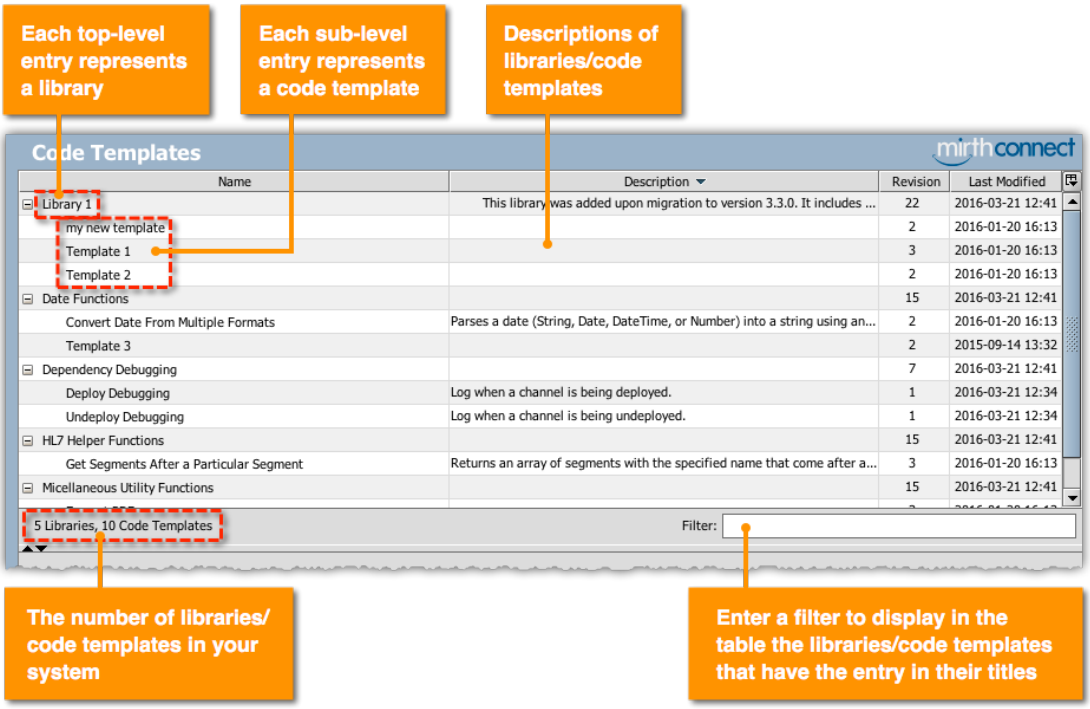
4. (Previous graphic) In the **Group Tasks** panel, click **Save Group Changes**.

Managing Code Templates

A code template is a user-defined function or snippet of code that can be used in channel JavaScript contexts (or the global scripts). One code template can be used in multiple channels, meaning that it lets the user organize common logic in one place for better overall management.

In Mirth Connect, code templates are organized into libraries, which are displayed on the **Code Templates** page via the **Mirth Connect** panel > **Channels** function – **Channel Tasks** panel > **Edit Code Templates** function.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



This section consists of procedures that instruct you how to management operations regarding code templates:

- Configuring the Code-Template Library Table
- Creating a Library or a Code Template
- Editing a Library or a Code Template
- Importing/Exporting a Library or a Code Template
- Deleting a Library or a Code Template.

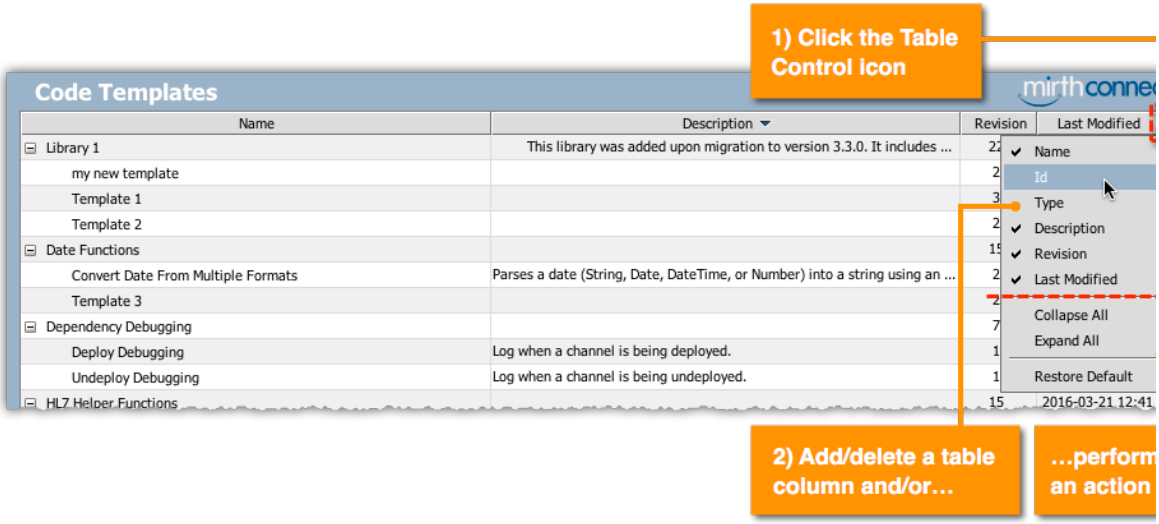
Configuring the Code-Template Library Table

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

This table lists code-template libraries and the code templates each library contains. (The list can be filtered to display certain libraries/code templates via the **Filter** field below the table.) By default, the table shows the **Name**, **Description**, **Revision**, and **Last Modified** columns; however, you can remove any of these defaults and add various other columns, and perform certain table functions as needed.

This procedure instructs you how to add/remove columns to/from the code-template/library table and how to perform a table function.

1. In the **Mirth Connect** panel, click **Channels**.
2. On the **Channels** page – **Channel Tasks** panel, click **Edit Code Templates**.
3. In the top-right corner of the **Code Templates** page, click the Table Control icon.



4. On the drop-down menu that appears (previous graphic), select the desired option to add/delete a table column and collapse/expand the libraries to manipulate the table display.

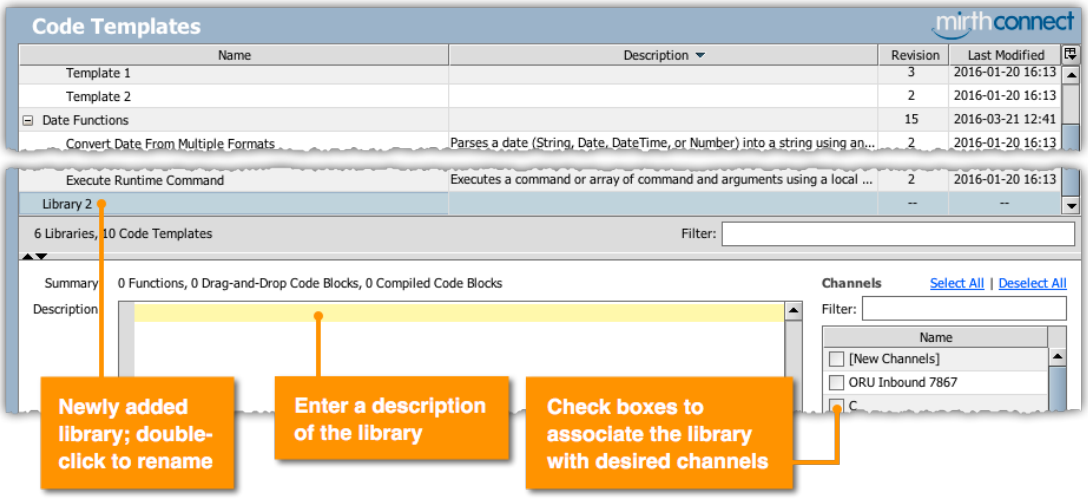
Creating a Library or a Code Template

Creating a Code-Template Library

This procedure instructs you how to create a code-template library. The new library will be empty. To add a code-template to a library, see the next procedure.

1. In the **Mirth Connect** panel, click **Channels**.
2. On the **Channels** page – **Channel Tasks** panel, click **Edit Code Templates**.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



3. In the **Code Template Tasks** panel, click **New Library**.

> The new library, with a default name (e.g., **Library 2**) appears on the code-templates table.

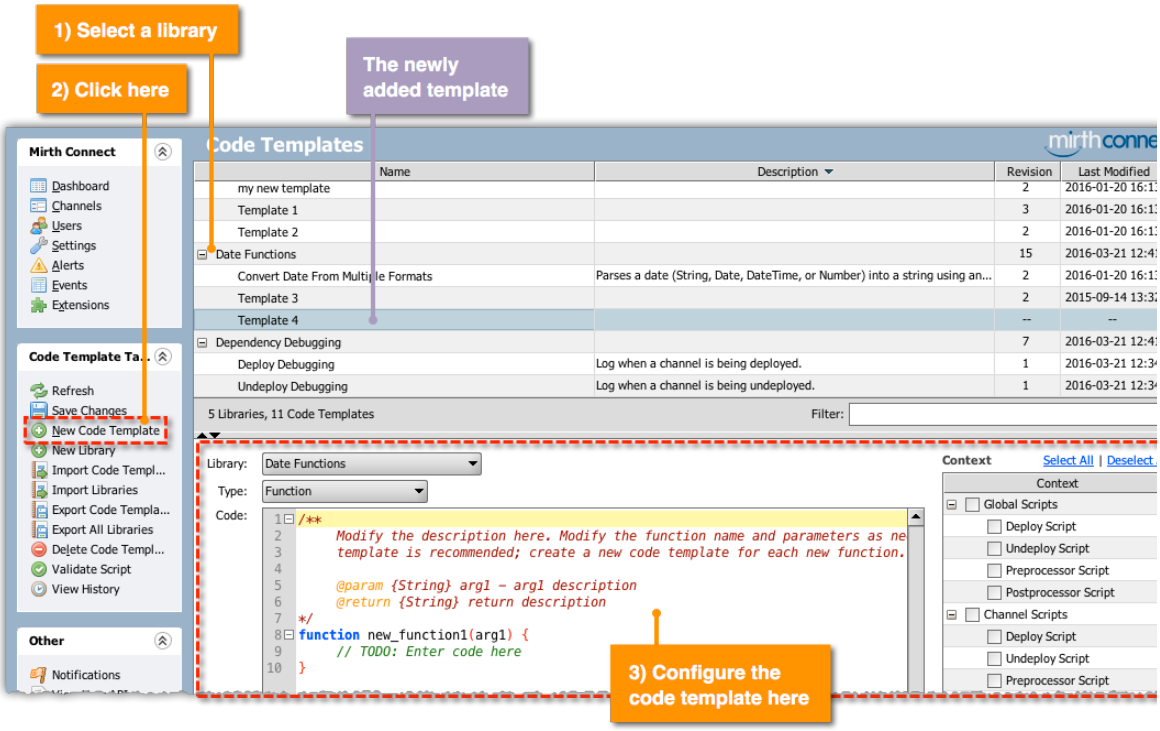
4. (Optional) Double-click the default library name to activate the field, and edit the name.
5. (Optional) In the **Description** box, add a description of the library.
6. In the **Channels** section, check the box of each channel with which you want to associate the library.
7. In the **Code Template Tasks** panel, click **Save Changes**.

Creating a Code Template

This procedure instructs you how to create a code template and add it to a library in the table.

1. In the **Mirth Connect** panel, click **Channels**.
2. On the **Channels** page – **Channel Tasks** panel, click **Edit Code Templates**.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



3. On the Code Templates table, select the library to which you want to add the new template.

> A code template cannot stand alone in the Code-Template Library table. If you click a blank area on the table, the **New Code Template** function does not appear in the **Code Template Tasks** panel. You can also select any code template in the desired library, and when you click **New Code Template**, the new template will be added at the end of the list of templates in that library.
>
> If you add the new code template to the wrong library, or if you decide to move the template to a different library, (on the bottom half of the page) click the **Library** bar, and on the drop-down menu, select the proper library.

4. In the **Code Template Tasks** panel, click **New Code Template**.
5. The new template, with a default name, appears within the selected library.
6. (Optional) Double-click the default code-template name to activate the field, and edit the name.
7. Using the elements on the bottom half of the page, configure the new code template.
8. In the **Code Template Tasks** panel, click **Save Changes**.

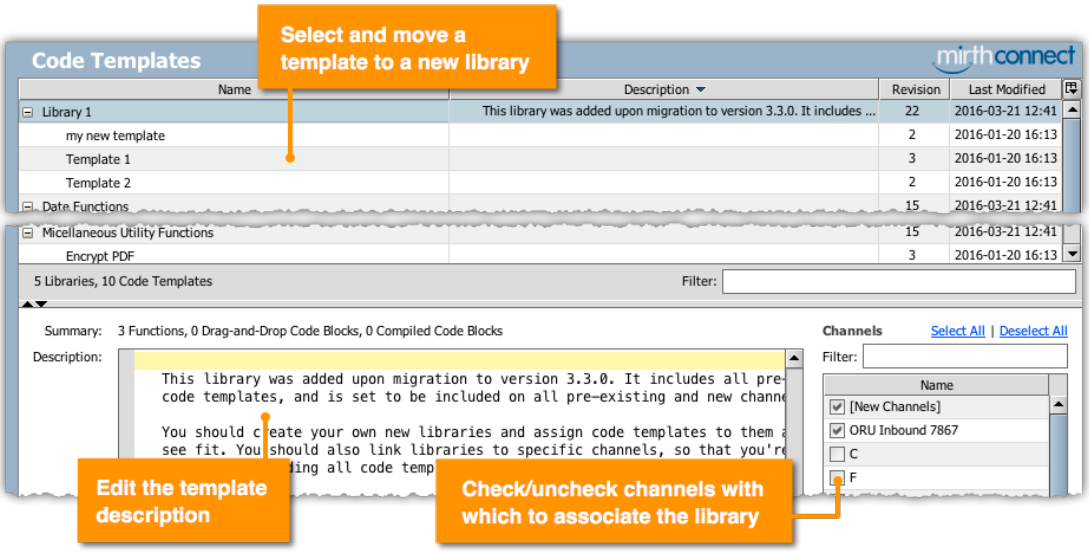
Editing a Library or a Code Template

Editing a Code-Template Library

Editing a code-template library may involve adding/removing a code template, adding/removing a channel to which it is associated, and adding/editing the library description.

1. In the **Mirth Connect** panel, click **Channels**.
2. On the **Channels** page – **Channel Tasks** panel, click **Edit Code Templates**.
3. In the **Code Templates** table, select the library you want to edit, and make your changes.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

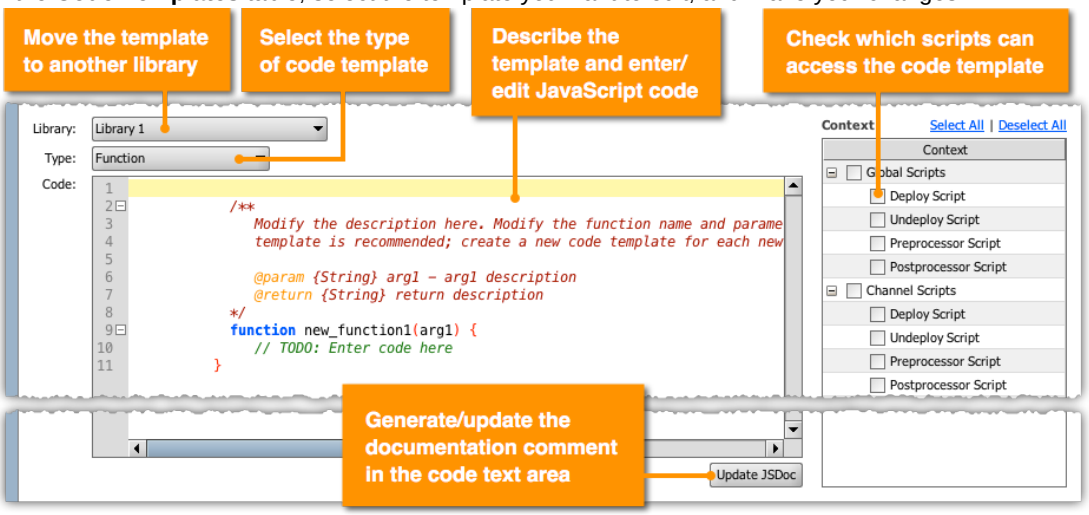


4. In the **Code Template Tasks** panel, click **Save Changes**.

Editing a Code Template

Editing a code template may involve moving the template to another library, changing the template type, generating/updating code, and changing which scripts can access the template.

1. In the **Mirth Connect** panel, click **Channels**.
2. On the **Channels** page – **Channel Tasks** panel, click **Edit Code Templates**.
3. In the **Code Templates** table, select the template you want to edit, and make your changes.



4. In the **Code Template Tasks** panel, click **Save Changes**.

> Other possible changes to a code template include: changing the name (double-click the name in the table, and enter the new name); and generating/updating the template's JavaScript code in the **Code** text box via the button below the box.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

> Moving a code template to another library does not constitute a change to the template. (The value in the Revision column will stay the same.) This action does, however, constitute a change to the library to which you move the template. (The value in the library's Revision column increases by 1.)

Viewing Library or a Code Template Histories

You can view the revision histories of code-template libraries and code templates if you purchase the Channel History plugin. Library and code-template histories are viewed via the same process as channel histories. For details on how to view these histories, see About the Channel History Plugin.

> The Channel History plugin requires a Silver-, Gold-, or Platinum-level support subscription. For technical and purchasing information about the Channel History plugin, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

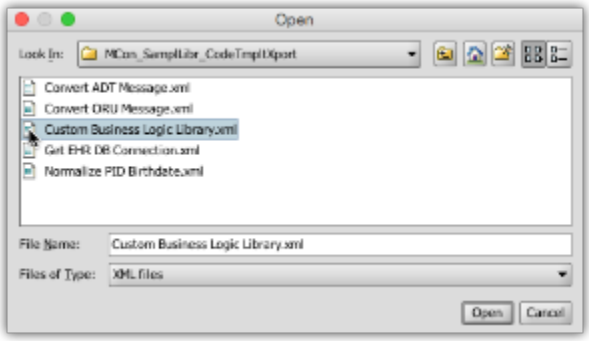
Importing/Exporting a Library or a Code Template

The procedures in this section instruct you how to import/export a code-template library. The process for importing/exporting code templates is the same; therefore, you may follow these same procedures when importing/exporting a code template but clicking **Import Code Template** or **Export Code Template** where appropriate.

Importing a Code-Template Library

This procedure instructs you how to import a code-template library into Mirth Connect from an external source.

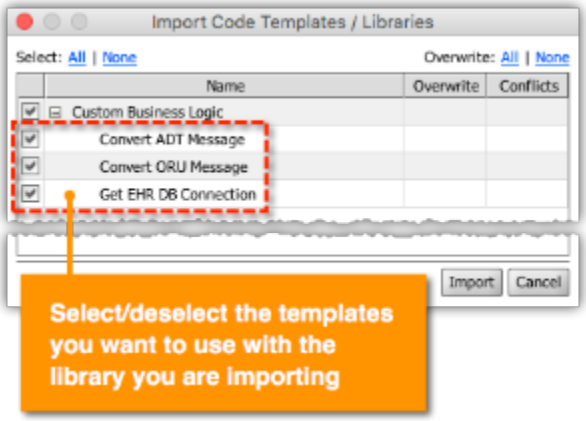
1. In the **Mirth Connect** panel, click **Channels**.
2. On the **Channels** page – **Channel Tasks** panel, click **Edit Code Templates**.
3. On the **Code Templates** page – **Code Template Tasks** panel, click **Import Libraries**.
4. On the **Open** dialog, navigate to and select the library you want to import, then click the **Open** button.



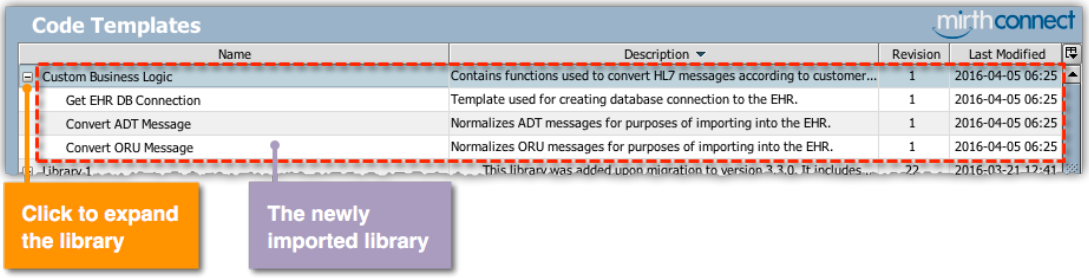
5. On the **Import Code Templates/Libraries** dialog, select/deselect the code templates you want to include in the imported library, and click the **Import** button.

> You can double-click any of the names on the list and rename the library/templates.

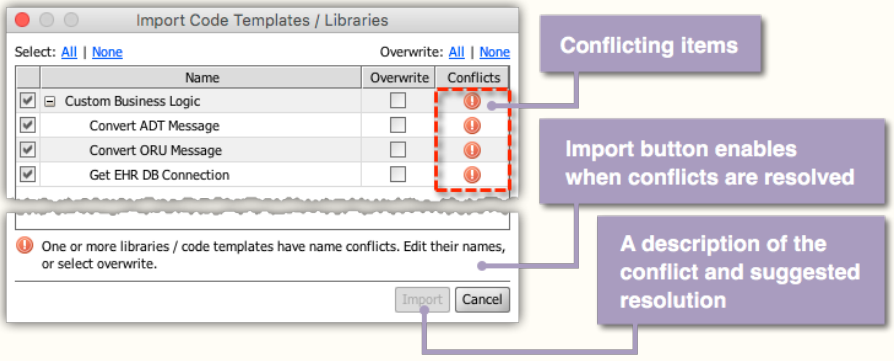
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



6. On the Code Templates table, click the "**+**" icon next to the newly imported library to expand the library and view its contents.



If you attempt to import a library/template that is already on the Code Templates table or that has an identical name, the **Import Code Templates/Libraries** dialog appears with conflict alerts for the offending items.



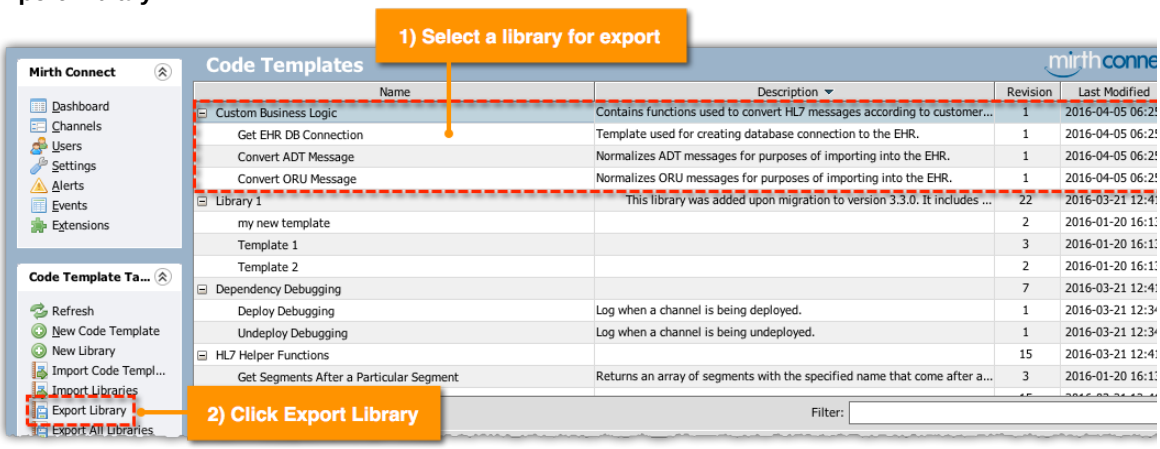
Below the list appears a description of the conflict(s) and a suggested resolution. If, for instance you are importing an updated version of the existing library/templates, you may check to overwrite any or all of the existing items, or as appropriate, you may double-click and rename the appropriate item(s).

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

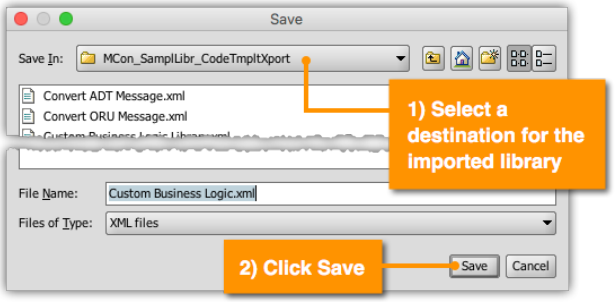
Exporting a Code-Template Library

This procedure instructs you how to export a code-template library from your system to an external location.

1. In the **Mirth Connect** panel, click **Channels**.
2. On the **Channels** page – **Channel Tasks** panel, click **Edit Code Templates**.
3. On the **Code Templates** page, select the desired library, and in the **Code Template Tasks** panel, click **Export Library**.



4. On the **Save** dialog, select a destination for the exported library, and click the **Save** button.



> An **Information** popup appears verifying that the library export was written your selected destination; click the **OK** button.
>
> If you attempt to export a library to a destination where that library or a library with an identical name already exists, a **Select an Option** popup appears that asks if you want to overwrite the file. Click the appropriate button (**Yes**/**No**).

> If, for example, you want to back up your libraries or move them to another instance of Mirth Connect on another server, you can export all code-template libraries in one action: (On the **Code Templates** page) In the **Code Template Tasks** panel, click **Export All Libraries**. In the **Save** dialog that appears, select a destination for the libraries, then in the **File Name** field, enter a name for the XML file that will contain the libraries, and click the **Save** button.
>
> Mirth Connect does not provide an Export All Code Templates function. Because every code template belongs to a code-template library, exporting all libraries also exports all code templates.

Deleting a Library or a Code Template

This procedure instructs you how to eliminate a code-template library or code template from the system.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

1. In the **Mirth Connect** panel, click **Channels**.
2. On the **Channels** page – **Channel Tasks** panel, click **Edit Code Templates**.

If you are deleting a code-template library:

1. On the Code Templates table, select the library you want to delete.

> **This action also removes all code templates in the library**. Make sure you do not need the library's code templates or move them to another library ***before*** you save your work.

2. In the **Code Template Tasks** panel, click **Delete Library** > **Save Changes**.

If you are deleting a code template:

1. On the Code Templates table, select the template you want to delete.
2. In the **Code Template Tasks** panel, click **Delete Code Template** > **Save Changes**.

About Messages

Mirth Connect lets healthcare sites communicate securely using electronically transmitted messages. Before a message is sent and after it is received, it undergoes various operations that ensure it arrives at the right place in the right format with the right information, then is logged in the right way with all the right statistical data recorded in the right categories. This section explains the messaging process via:

- Managing Messages – viewing, searching for, reprocessing, importing/exporting, sending, removing, and working with statistics
- About the Message-Processing Flow – the attributes of a message from source to destination
- Setting Message Templates – the function of inbound/outbound message templates
- About Transformers – maps, scripts, message builder, XSLT steps
- About Filters – variables that pass/stop and route messages
- About Connectors – the elements that link sources to destinations
- Messaging Tools – links to applications and tutorials you can use to improve your messaging capabilities
- Advanced Messaging Topics – attachments, and routing.

Managing Messages

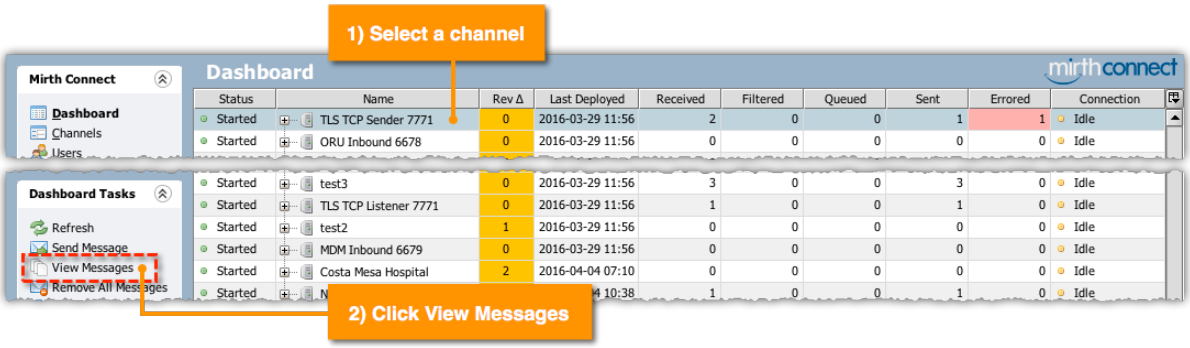
Managing messages in Mirth Connect involves:

- Viewing Messages
- Searching for Messages
- Reprocessing Messages
- Importing/Exporting a Message
- Sending a Message
- Removing a Message
- Working with Message Statistics.

Viewing Messages

You can view a channel's messages by selecting the channel on the **Dashboard** page and in the **Dashboard Tasks** panel, clicking **View Messages** (or you can double-click the channel). To view the messages of undeployed channels, see Viewing Messages on Undeployed Channels.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

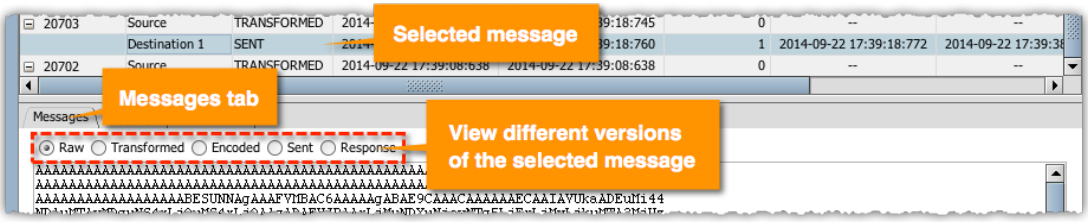


Channel Message Page Tabs

On the **Channel Messages** page, you can select a message and use the tabs in the lower portion of the page to view various message traits and elements.

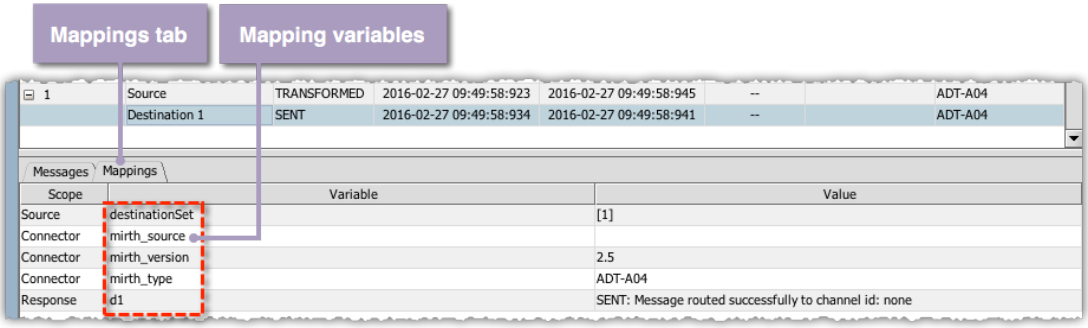
Messages Tab

Content options (raw, transformed, encoded, etc) appear based on the type of processing a message has undergone.



Mappings Tab

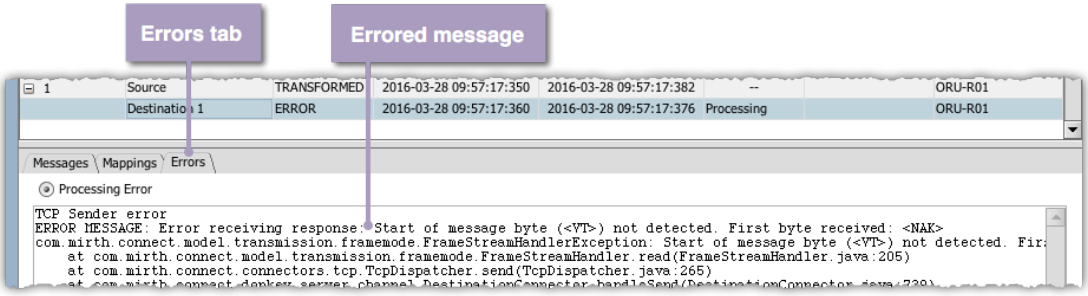
Select this tab to view mapping variables and associated data.



Errors Tab

This tab appears only if the selected message has an error.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



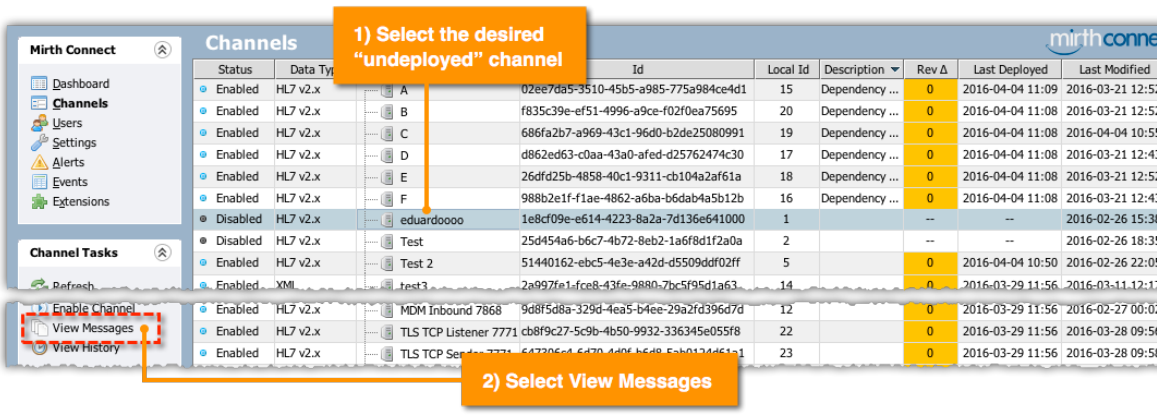
Attachments Tab

This tab appears only if the selected message has an attachment. Via functions in the **Message Tasks** panel, you can view the attachment, and you can export the attachment to the server or to your computer.

Viewing Messages on Undeployed Channels

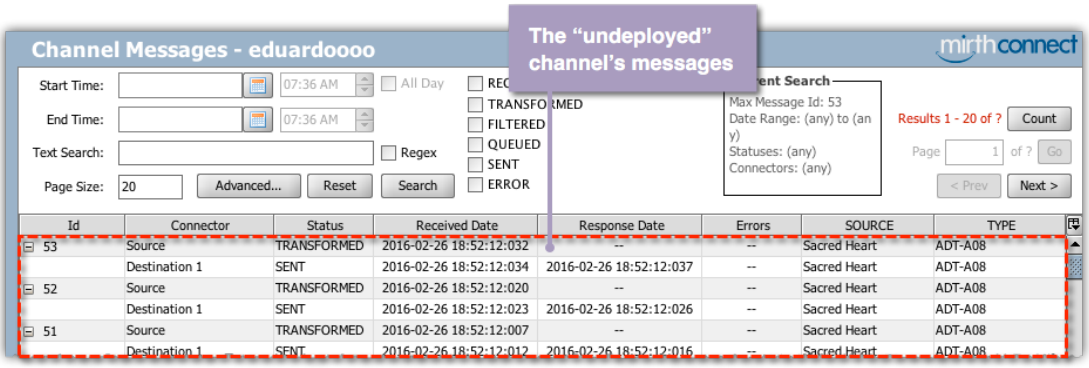
A channel does not have to be deployed for you to view the messages associated with it. (Undeployed channels have no data displayed in their **Rev** and **Last Deployed** columns.)

1. (Because undeployed channels do not appear on the **Dashboard** page) In the **Mirth Connect** panel, click **Channels**.



2. Select the desired undeployed channel (previous graphic), and in the **Channel Tasks** panel, click **View Messages** to view the undeployed channel's messages.

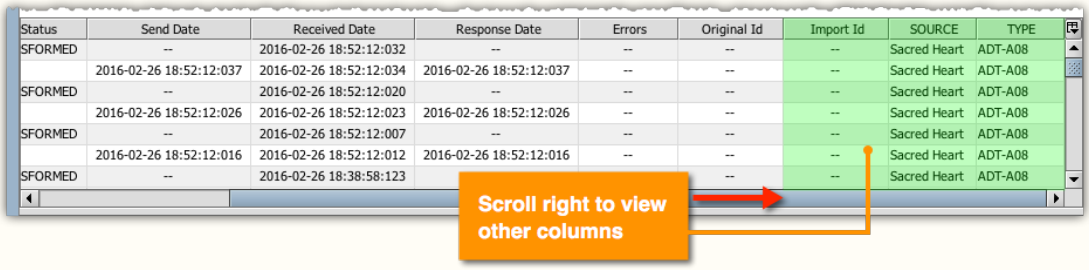
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Adding a Column to the Messages List

On a **Channel Messages** page, the Messages List features eight default columns: *Id*, *Connector*, *Status*, *Received date*, *Response Date*, *Errors*, *Source*, and *Type*. You can add columns to and delete columns from the list one at a time via the Column Options icon/menu in the list's column header.

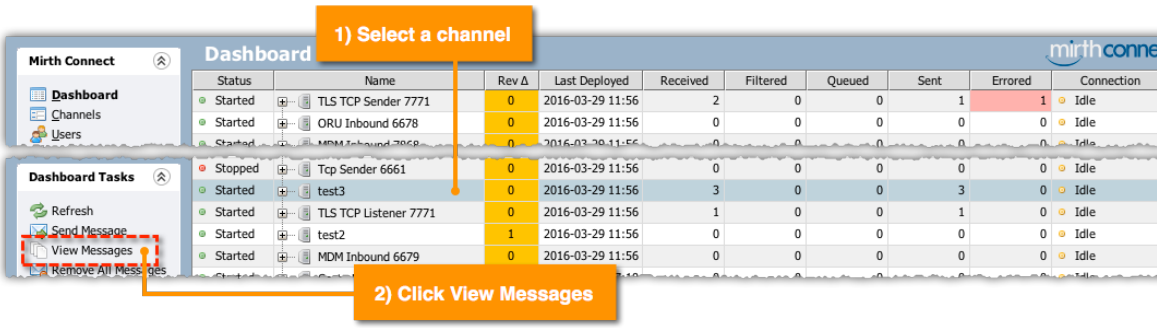
All Messages Lists for all channels have the same column configuration; that is, any columns added to/removed from the Messages List of one channel are added to/removed from the Messages List of all channels. You can add as many available columns as you like, though not all may be visible onscreen at once. If all columns cannot be viewed onscreen, a horizontal scroll bar appears below the Messages List. Scroll left or right to view the desired columns/data.



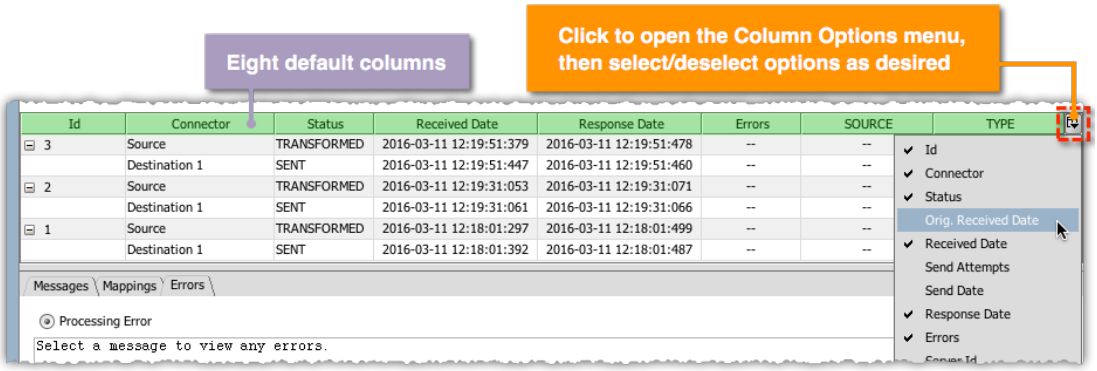
Deleting a column does not delete its data; the data reappears when you add the column back to the Messages List.

1. On the **Dashboard** page, select a channel, and in the **Dashboard Tasks** panel, click **View Messages**.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



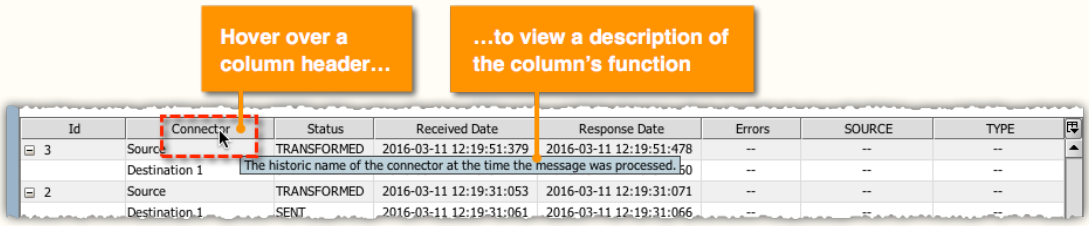
2. At the far-right end of the Message Lists's column header, click the Column Options icon, and on the menu, select an option to add to/remove from the column header.



Also on the Column Options menu are functions by which you can perform various list-related tasks:

- *Collapse All* – Shows only the Source Connector row of each message in the Messages List
- *Expand All* – Shows the Source Connector row *and* Destination row(s) of each message in the Messages List (as in the previous graphic)
- *Restore Default* – Reconfigures the Messages List to contain the eight default columns (see the introductory paragraph in this section).

You can pause the pointer on a column header to reveal a tool tip with a description of that column's function.



Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Searching for Messages

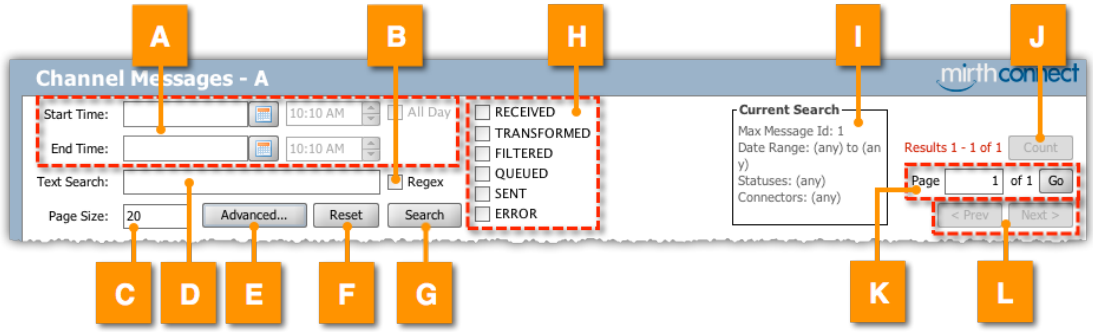
This topic consists of an overview of the search filters on a **Channel Messages** page and of a message-search procedure. MC's message-search feature is capable of a wide range of searches, from general in nature, using one or just a few filters, to finely grained, using many filters and including the **Advanced…** dialog, which offers a number of Id, content, and metadata filters.

Using Message Search Filters

Because messages have so many distinguishing elements, finding the right message (or group of messages) can be tedious, especially if a channel has thousands of messages; thus, MC provides numerous corresponding filters to help you configure a well-targeted search. This section acquaints you with those filters.

About the Message Search Filters

To access these filters, on the **Dashboard** page, select a channel, and in the **Dashboard Tasks** panel, click the **View Messages** task.



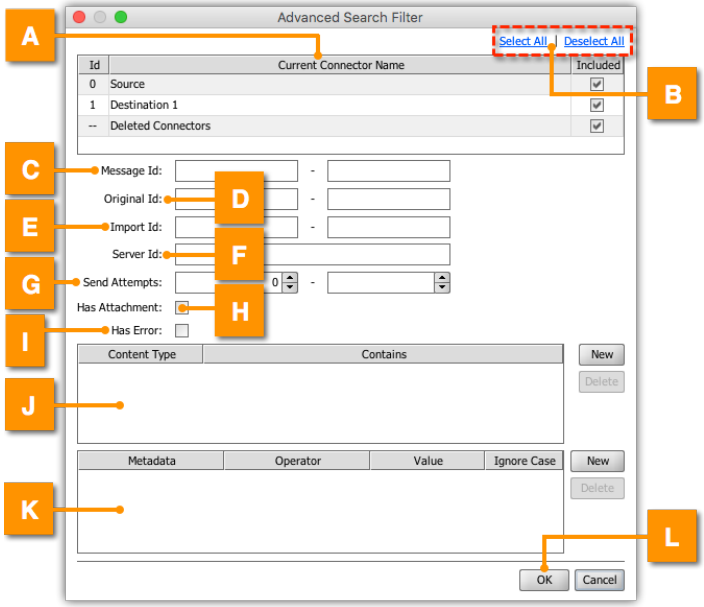
Option	Name	Description
A	Start/End Time	Use the Calendar icons to populate these fields with the start/end dates of the desired messages; use the Hour fields to restrict the search to a range of hours, or check the **All Day** box to include messages that started/ended at all hours of the day
B	Regex	Search all message content for a match to the regular expression pattern; Regex matching could be a very costly operation and should be used with caution, especially with large amount of messages; any message content that was encrypted by this channel will not be searchable; only supported on PostgreSQL, Oracle, and MySQL databases; when you check this box, you can use the **Text Search** field to enter a Regular Expression pattern to match message content, metadata, and connector names
C	Page Size	Shows the max amount of messages that will appear on a page
D	Text Search	Enter text (e.g., patient name, IP address) to include message content, metadata, and connector names containing the entered text; when you enter text in this field and click the **Search** button, the **Select an Option** dialog appears, informing you that *Text searching may take a long time, depending on the amount of messages being searched*, then asks *Are you sure you want to proceed?* Click the **Yes** or **No** button as desired
E	Advanced…	Reveals the **Advanced Search Filter** dialog consisting of more specific search-filter options
F	Reset	Removes all filters from the previous search

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

G	Search	Starts the message search
H	Status	Check these boxes to search for messages with *RECEIVED*, *TRANSFORMED*, *FILTERED*, *QUEUED*, *SENT*, and/or *ERROR* statuses
I	Current Search (Filters)	The filters you selected for your search
J	Count	Shows how many of the channel's messages match the search criteria
K	Page	(Multi-page lists) Activates when you click the **Count** button; enter a value (coinciding with the page range) in this field, and click the **Go** button to display the desired page
L	< Prev/Next >	(Multi-page lists) Click the **< Prev** button to go back one page or the **Next >** button to advance one page
The **Current Search** box displays various data relative to the most recent search.		

About the Advanced Search Filters

To access this dialog, click the **Advanced…** button on a **Channel Messages** page. (See Option D, previous section.)



Option	Name	Description
A	Current Connector Name	Check any/all of the displayed boxes to find messages with the selected connector type(s)
B	Select/Deselect All	Click to select/deselect all connectors on the **Current Connector Name** list
C	Message Id	Enter an Id in the first field to find the message with that Id, or enter different Id's in each field to find the messages in that range of Id's (e.g., for Message Id, enter *1* in the first field and *10* in the second field to find Messages 1-10)
D	Original Id	
E	Import Id	

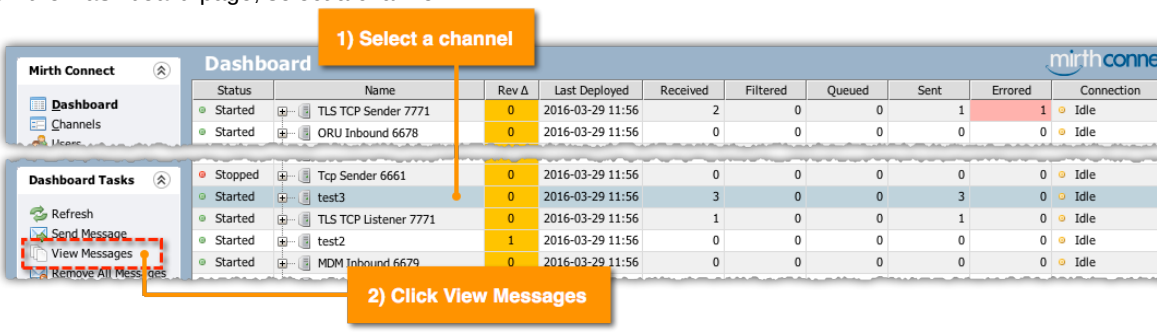
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

F	Server Id	Enter an Id to find all messages containing that server Id
G	Send Attempts	Enter min (first field)/max (second field) values to find messages with the number of send attempts in that range
H	Has Attachment	If checked, the search includes messages with an attachment
I	Has Error	If checked, the search includes messages with an error
J	Content Type	Click the box's **New** button to add a search filter, then click the row in the **Content Type** column, and select a menu option; double-click the row in the **Contains** column, and enter your search object (**Contains** refers to an element shared by certain of the channel's messages)
K	Metadata	Click the box's **New** button to add a search filter, then click the row in the **Metadata** column, and select a custom-metadata menu option; click the row in the **Operator** column, and select a search operator (=, *CONTAINS*, *STARTS/ENDS WITH*), which determines the metadata to search for; double-click the row in the **Value** column, and enter your search object; **Ignore Case**: if checked, the search is not case-sensitive
L	OK/Cancel	Click the **OK** button to save or the **Cancel** button to disregard your changes

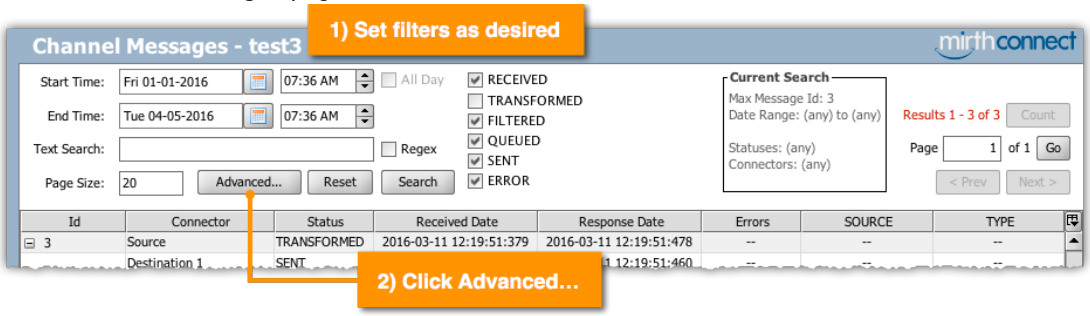
Searching For a Message

This procedure instructs you how to conduct a message search (including the **Advanced…** button) on a **Channel Messages** page.

1. On the **Dashboard** page, select a channel.

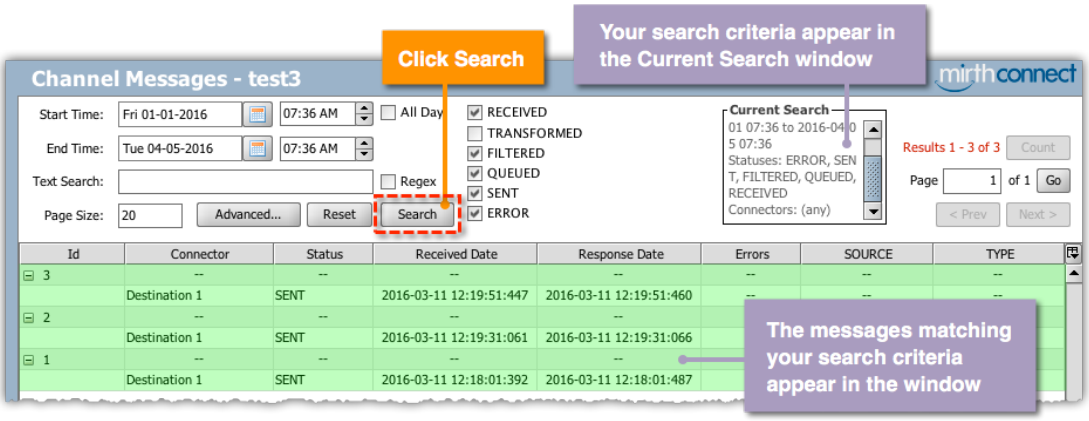


2. In the **Dashboard Tasks** panel (previous graphic), click **View Messages**.
3. On the **Channel Messages** page, select filters as desired, then click the **Advanced…** button.

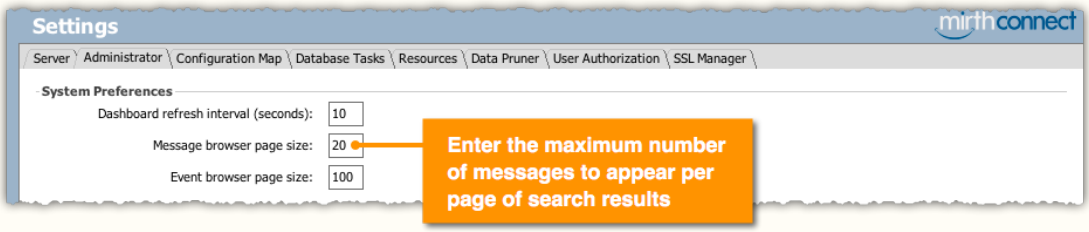


4. On the **Advanced Search Filter** dialog, fill in the variable fields and checkboxes as desired, and click the **OK** button.
5. On the **Channel Messages** page, click the **Search** button to reveal the selected message(s) **o**n the Messages List.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Search results are paginated by configurable page size (i.e., the most messages that can appear on a page). Change the default page size via the **Mirth Connect** panel – **Settings** > **Administrator** tab.



Reprocessing Messages

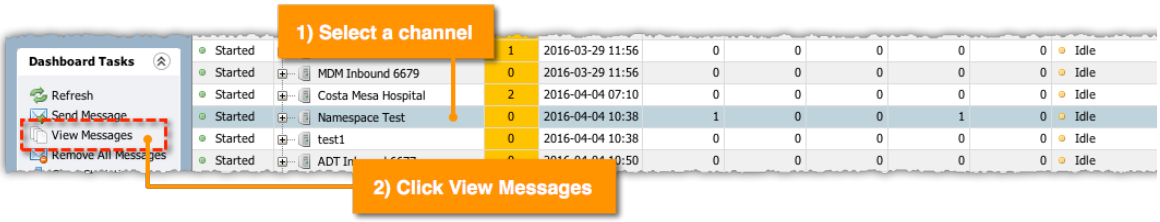
Messages are reprocessed for many reasons (e.g., a message is errored because: it is improperly configured; the destination channel is stopped, paused, or disabled), but the point is always to deliver the message successfully. Also, if multiple messages need reprocessing, you can conduct a search to collect the messages in question, then reprocess the results of your search; that is, reprocess all messages returned by your search. This section consists of procedures that instruct you how to reprocess a message and reprocess the results of a message search.

Reprocessing a Message

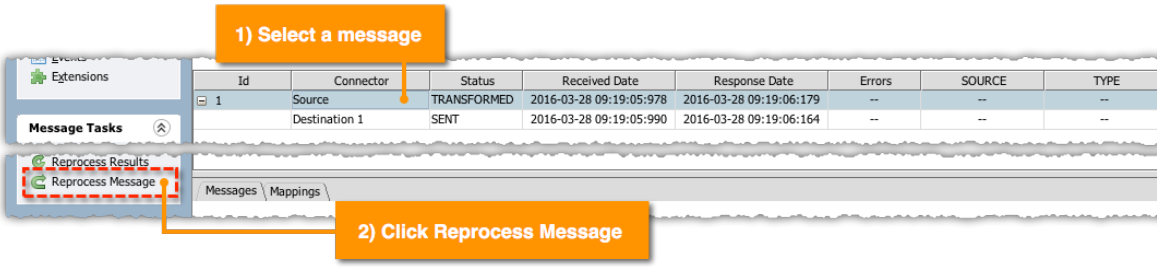
You can reprocess messages through a paused channel, and you can pause a channel during reprocessing without interrupting the job; however, if you *stop* a channel during reprocessing, the entire job is cancelled and must be redone. The **Reprocess Message** function reprocesses only the selected message. You cannot select multiple messages on the Messages List for reprocessing; to reprocess multiple messages, you need to conduct a search of the messages sharing the desired traits, and reprocess the results of your search; that is, the messages returned by your search. (See the next procedure: Reprocessing Search Results.)

1. On the **Dashboard** page, select a channel, and in the **Dashboard Tasks** panel, click **View Messages**.

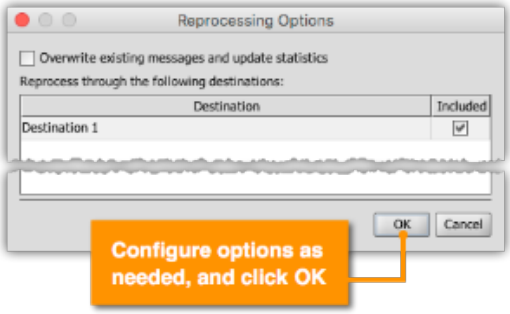
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



2. On the **Channel Messages** page, select the message you want to reprocess.



3. In the **Message Tasks** panel, click **Reprocess Message** (previous graphic).
4. On the **Reprocessing Options** dialog, select options as needed, and click the **OK** button.

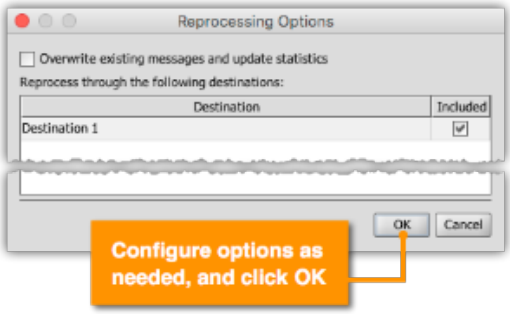


> If you check the **Overwrite existing messages and update statistics** box, the existing message is replaced and its statistics reset on the **Dashboard** page. If you do not check this box, the selected message is sent as a new message which appears in the channel's Messages List.

Reprocessing Search Results

This action simultaneously reprocesses *all* messages resulting from a message search. This procedure starts on a Channel Messages page after a message search.

1. Navigate to a **Channel Messages** page, and use the search filters to conduct a search for the desired messages. (See Searching for Messages if needed.)
2. In the **Message Tasks** panel, click **Reprocess Results**.
3. In the **Reprocessing Options** dialog, select options as needed, and click the **OK** button.



Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

If you check the **Overwrite existing messages and update statistics** box, the existing messages are replaced and their statistics reset on the **Dashboard** page. If you do not check this box, the messages are sent as new messages, which appear with those of the previous search on that channel's Messages List.

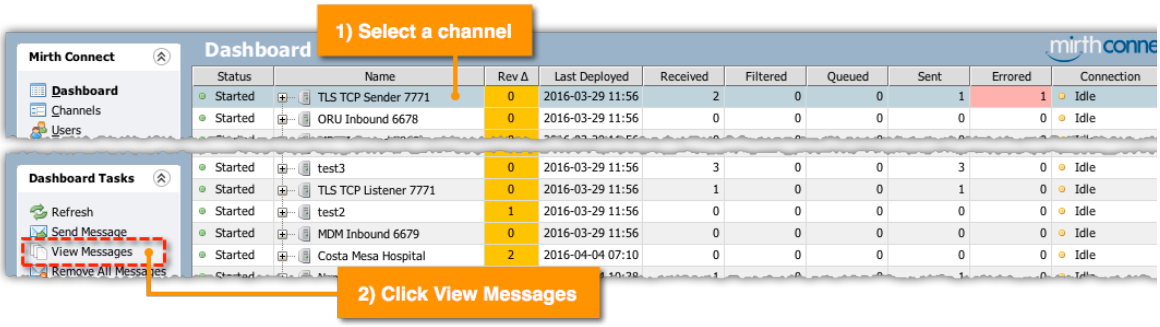
Importing/Exporting a Message

You can import messages into Mirth Connect and export messages from MC to be stored elsewhere. This section consists of procedures that instruct you how to import/export a message.

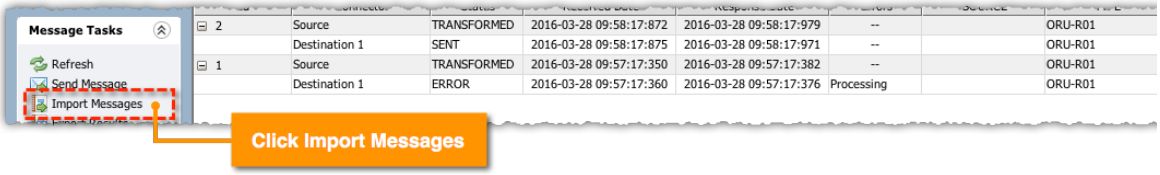
Importing a Message

This procedure instructs you how to import a message to a channel.

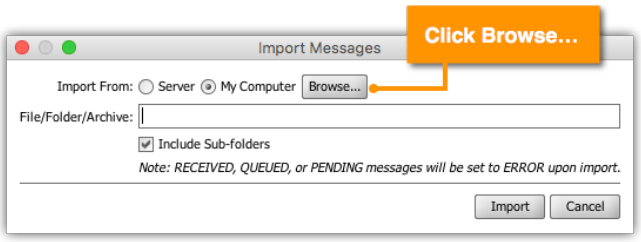
1. On the **Dashboard** page, select the channel into which you want to import the message, and in the **Dashboard Tasks** panel, click **View Messages**.



2. On the **Channel Messages** page – **Message Tasks** panel, click **Import Messages**.



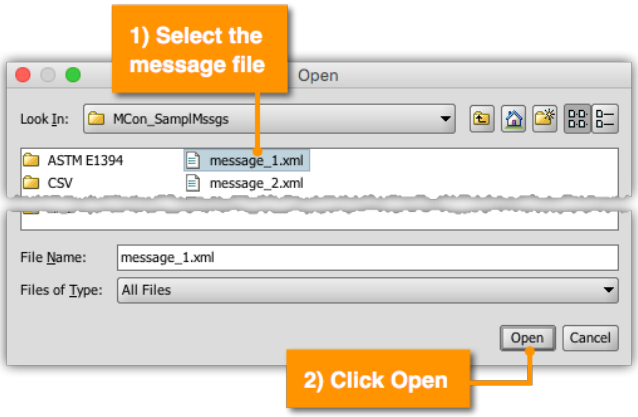
3. On the **Import Messages** dialog, click an **Import From** option (in this case, *My Computer*), and click the **Browse…** button.



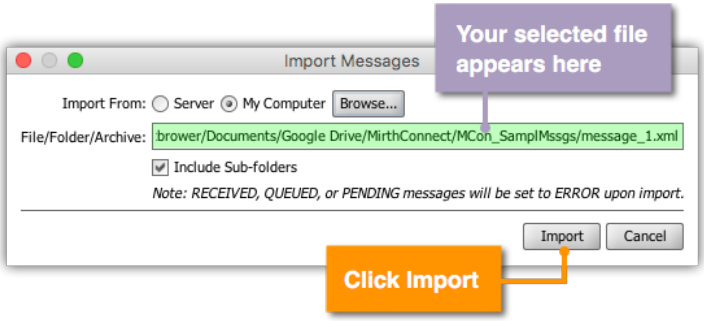
If you select the *My Computer* radio button and browse for the desired file, the **File/Folder/Archive** field is automatically populated with your selected destination when you click the **Open** button on the **Open** dialog in the next step. If, however, you select the **Server** radio button, you will need to populate the **File/Folder/Archive** field manually.

4. On the **Open** dialog, navigate to the folder containing the message file you want to import, select the file, and click the **Open** button.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



5. The file appears in the **File/Folder/Archive** field of the **Import Messages** dialog; click the **Import** button.

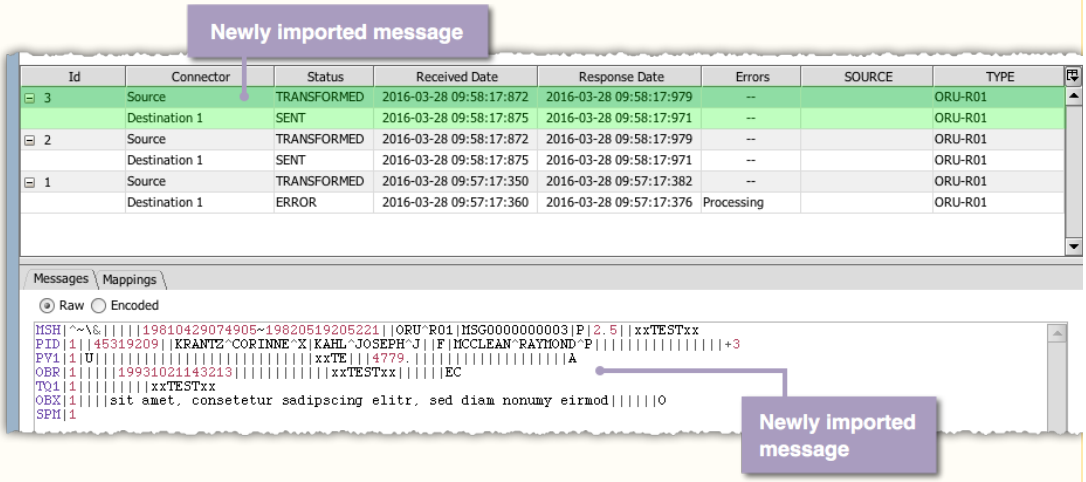


An **Information** dialog appears, informing you that the message was successfully imported. Click the **OK** button.

6. On the **Channel Messages** page, click the **Search** button to refresh the Messages List.

The message appears on the list with its contents in the **Messages** window at the bottom of the page.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



A unique identifier for each imported message appears in the **Import Channel Id** column. This alphanumeric code lets you readily identify messages that have been imported into the channel. If you import a message into the channel from which it originated, an Import Channel Id is not assigned, and the row in that column will show a dash (—), by which you can identify the message as original to the selected channel.

The **Import Channel Id** column (and others) can be chosen via the Column Options icon at the right-hand end of the Message Lists's column header. (For directions on how to create a column, see Viewing Messages.)

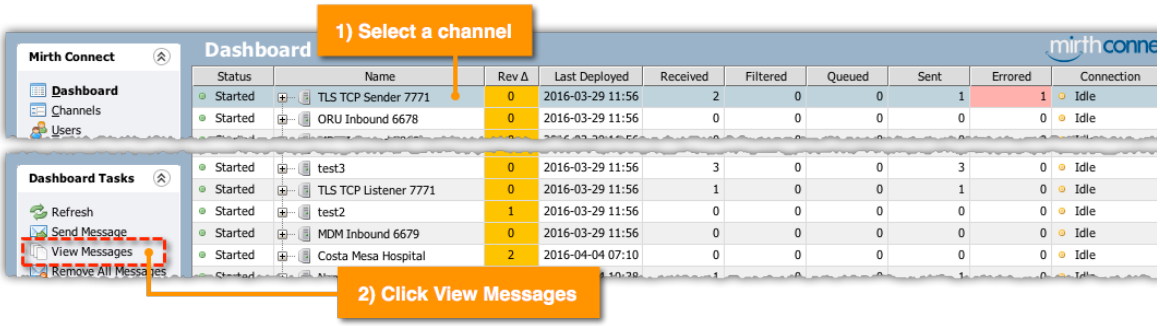
Exporting a Message

This procedure instructs you how to export message data from a channel to a storage site outside MC. The export function is **Export Results** (not, for instance, Export *Message*) because MC requires you to search for the message(s) you want to export; therefore, "Results" refers to the results of your search, whether for one or some other number of messages.

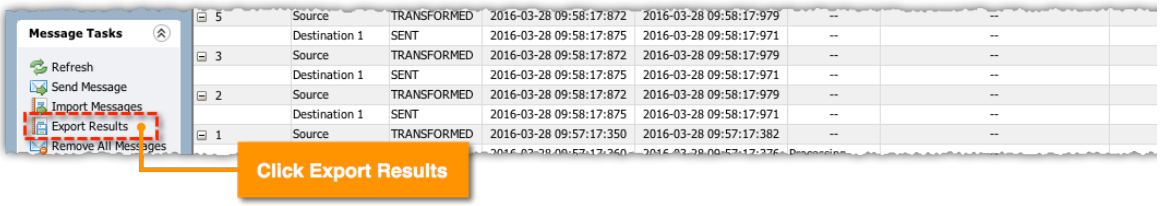
This procedure exports all messages in the selected channel. To export only certain messages, configure the search criteria to include only your desired messages. To export only one message, use that message's Id (from the **Id** column of the Messages List). Click the **Advanced…** button, and enter the Id in both **Message Id** fields on the **Advanced Search Filter** dialog, then click the **OK** button. On the **Channel Messages** page, click the **Search** button to reveal the desired message on the Messages List, and proceed from Step 2 of the export procedure.

1. On the **Dashboard** page, select the channel from which you want to export the messages, and in the **Dashboard Tasks** panel, click **View Messages**.

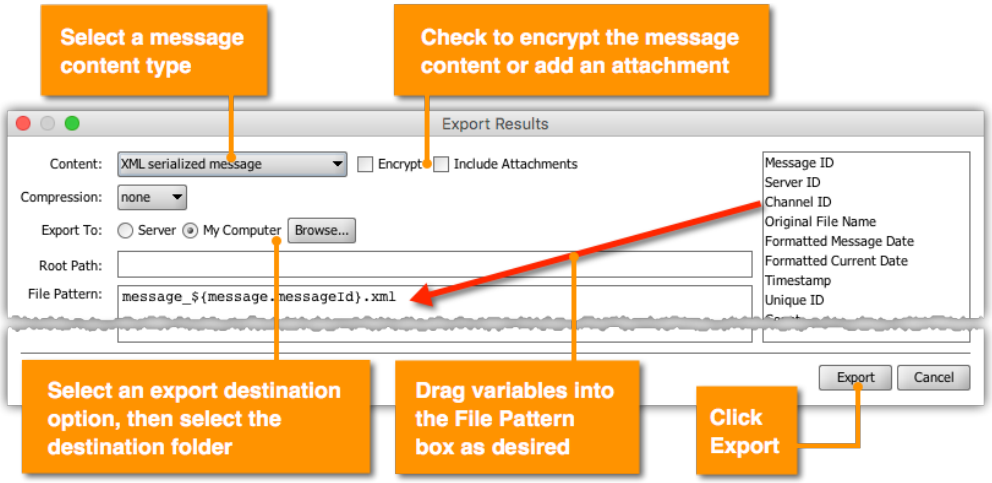
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



2. On the **Channel Messages** page – **Message Tasks** panel, click **Export Results**.



3. On the **Export Results** dialog, select parameters as needed, and in the **Export To** section, select a destination (in this case, *My Computer*), then click the **Browse…** button (*My Computer* option only).



If you select the *My Computer* radio button and browse for the desired file, the **File/Folder/Archive** field is automatically populated when you click the **Open** button on the **Open** dialog in the next step. If, however, you select the **Server** radio button, you will need to populate the **File/Folder/Archive** field manually.

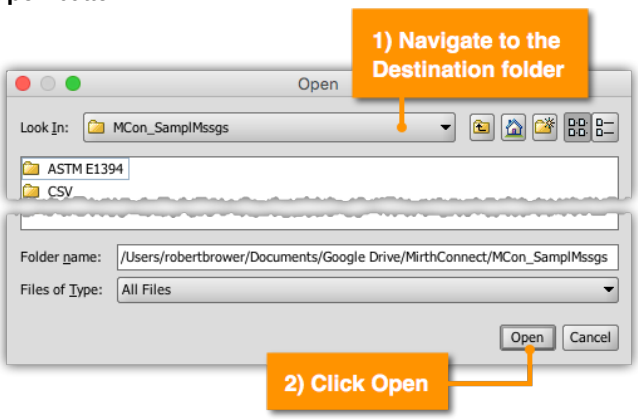
(Optional) On the **Export Results** dialog (previous graphic), you can check the **Encrypt** box to encrypt the message's content, and (XML serialized messages only) if you check the **Include Attachments** box, attachments to the message will be exported with the message. (Leave the box unchecked if you want to export the message without its attachments.)

(**File Pattern** box – previous graphic) Enter a file name (or folders). You may also click-drag variables from the list to the right into the **File Pattern** box (e.g., to create a separate file for each message, drag the *Message ID* variable into the box). If you do not use any variables, all exported

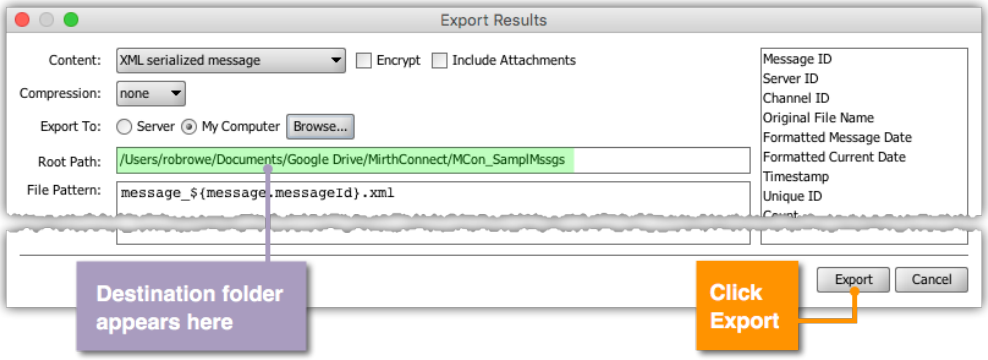
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

messages will be appended together into one file.

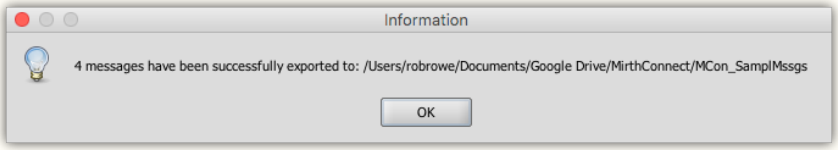
4. On the **Open** dialog, navigate to the folder into which you want to export the message files, and click the **Open** button.



5. The destination folder appears in the **Root Path** field of the **Export Results** dialog. Click the **Export** button to export the messages to the destination folder.



An **Information** dialog appears, informing you that the messages have been successfully exported.



Click the **OK** button, and go to the destination folder to view the exported messages.

Sending a Message

When you send a message; that is, when you click the **Send Message** function, you send a message to the selected channel, which processes the message and sends it to its destination. How the message is processed and where it is sent depends on the channel's configuration. (See Creating a Channel and Editing a Channel.)

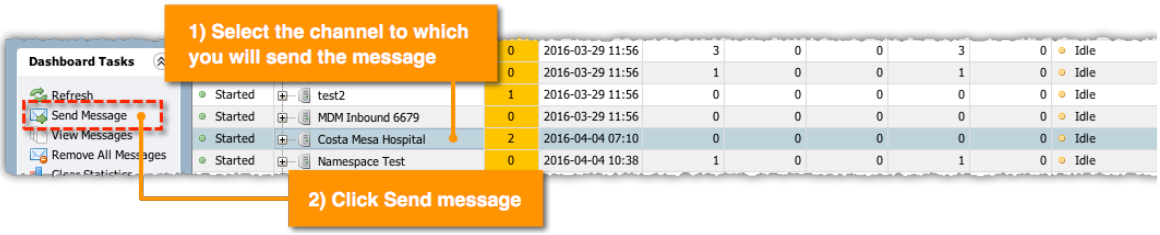
The **Send Message** function in the **Message Tasks** panel lets you test a channel without having a “live” incoming stream; that is, you can drop messages into a channel by opening or copying-pasting the contents of a file.

You can send a message manually to a *Paused* channel. You cannot send a message to a *Stopped*

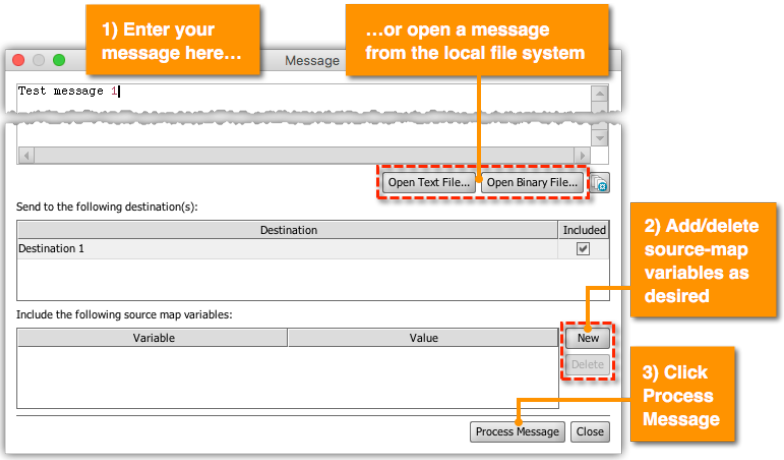
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

channel.

1. On the **Dashboard** page, select the channel to which you will send the message.



2. In the **Dashboard Tasks** panel, click **Send Message** (previous graphic) to reveal the **Message** dialog.



3. Enter a message manually, or click an **Open File** button (previous graphic) to automatically enter a message from the local file system.
4. Click the **Process Message** button (previous graphic) to send the message.
5. In the **Dashboard Tasks** panel, click **View Messages**.

> The **Channel Messages** page may show the completed message or may show it being processed. Greyed-out, italic type indicates the message is being processed according to the channel's configuration, which could direct it to store the message on your system or send it to an external destination. *RECEIVED* means the channel received and is processing the message you sent to it. *TRANSFORMED* means the channel processed the message; *SENT* means the channel sent the message to its destination.
>
> You may need to click the **Search** button to display the newly sent message.

> If you are on a **Channel Messages** page when you send a message, the *RECEIVED* message appears on the Messages List when you click the **Message** dialog's **Process Message** button. To view the completed message, click the **Search** button.

Removing a Message

In the **Message Tasks** panel, three functions enable you to remove messages or message-search results:

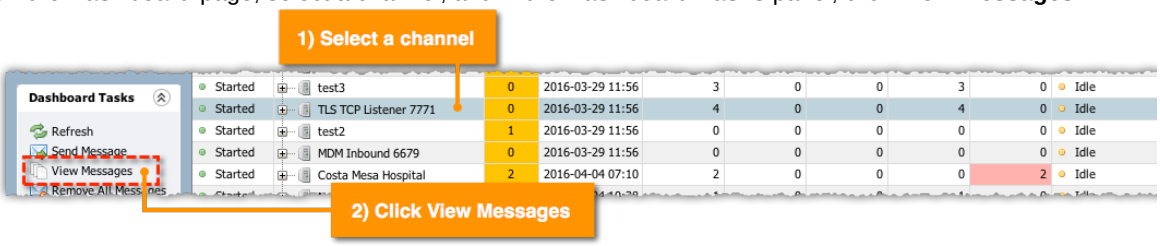
- **Remove All Messages** – removes all messages from the selected channel
- **Remove Results** – removes only messages found by a filtered search

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

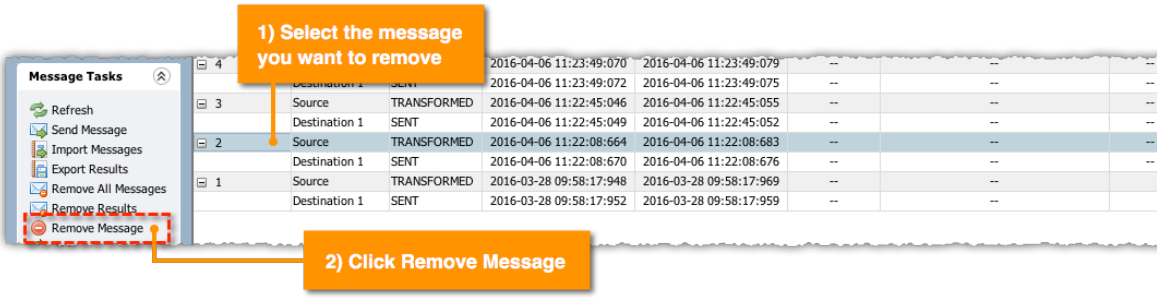
- **Remove Message** – removes only the selected message.

This procedure instructs you how to remove one message from a channel. (You could use this procedure to remove all messages or remove search results from a channel by clicking the appropriate function in the **Message Tasks** panel.)

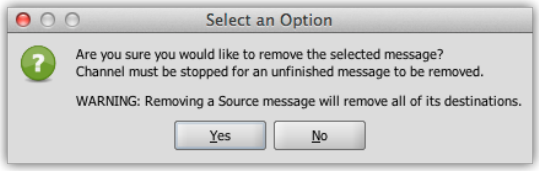
1. On the **Dashboard** page, select a channel, and in the **Dashboard Tasks** panel, click **View Messages**.



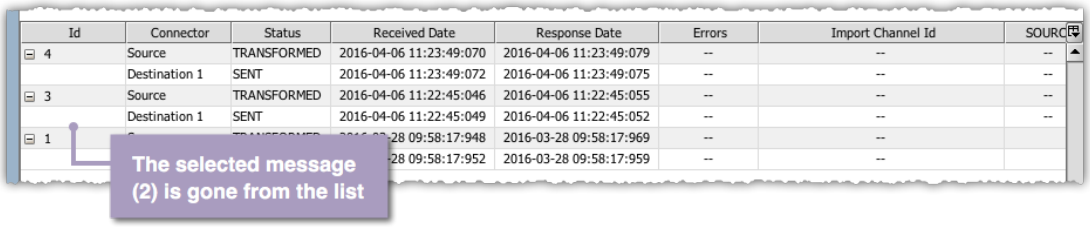
2. On the **Channel Messages** page, select the message you want to remove.



3. In the **Dashboard Tasks** panel (previous graphic), click **Remove Message**.



4. On the **Select an Option** dialog (previous graphic), click the **Yes** button, and see that the selected message (in this case, 2) has been removed from the list.



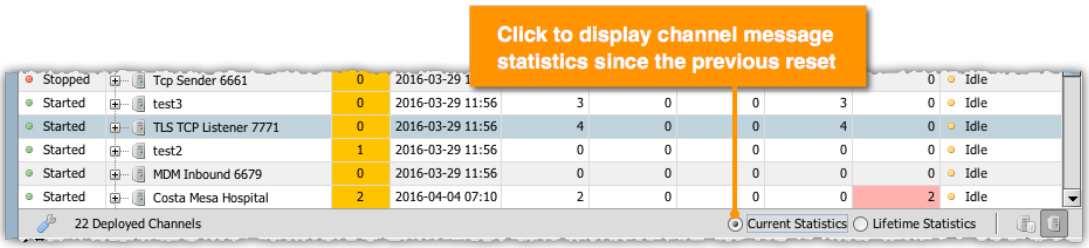
Working with Message Statistics

This section consists of a brief overview of channel message statistics and procedures that instruct you how to clear the Current and Lifetime message statistics of channels.

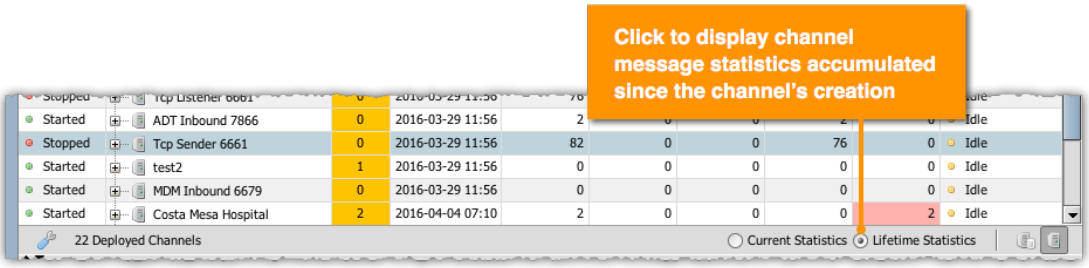
The **Dashboard** page shows the channels' cumulative message statistics, including how many of each channel's messages were: *Received/Filtered/Queued/Sent/Errored*. Statistics are classified as *Current* (having accumulated since the last reset of the channel's statistics), and *Lifetime* (having accumulated since the channel was created). Click the radio buttons below the Channels List to view **Current Statistics** (default) or **Lifetime Statistics**, which are shown in the next two graphics. Notice that some lifetime statistics are much higher than the same current statistics, indicating that the channel's current statistics have been cleared at least once since the channel was created.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Current Statistics



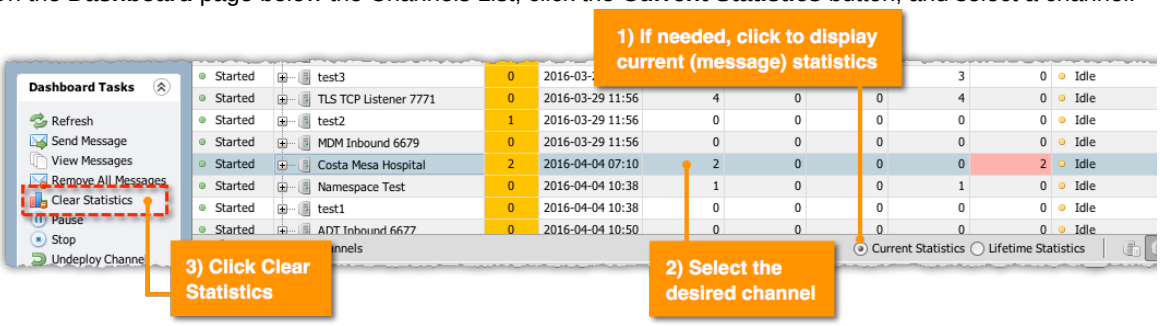
Lifetime Statistics



Clearing Current Message Statistics

This procedure instructs you how to clear the current message statistics of one channel. (You can clear the current statistics of multiple channels by selecting the desired channels and beginning this procedure at Step 2.)

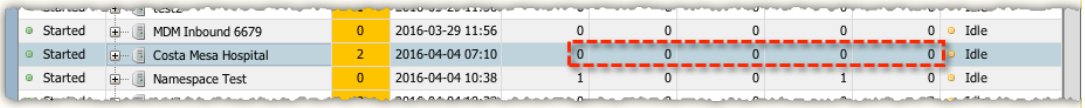
1. On the **Dashboard** page below the Channels List, click the **Current Statistics** button, and select a channel.



2. In the **Dashboard Tasks** panel, click **Clear Statistics** (previous graphic).
3. On the **Clear Statistics** dialog, check the boxes of the statistic types you want to clear (checking all boxes clears all message statistics from the channel), and click the **OK** button.

Note that the channel's current message statistics have been reset to zero.

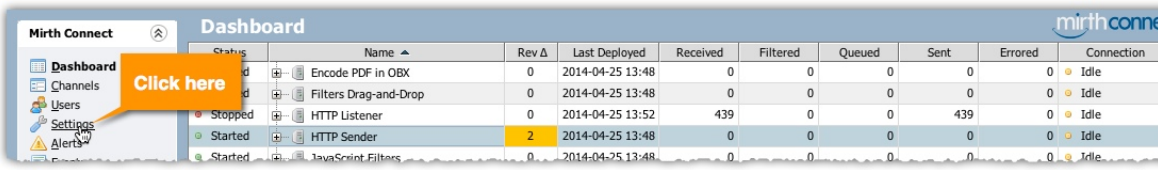
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Clearing Lifetime Message Statistics

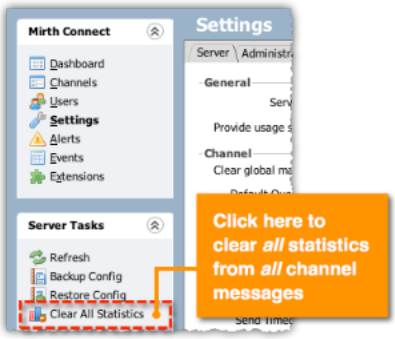
This procedure instructs you how to clear *all* message statistics from *all* channels. (Mirth Connect has no provision for clearing all lifetime message statistics from an individual channel or group of channels.)

1. On the **Dashboard** page – **Mirth Connect** panel, click **Settings**.

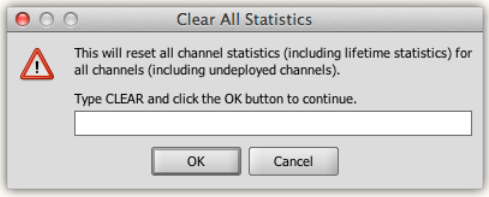


> Before performing Step 2, Mirth suggests that you thoroughly consider your reasons for clearing the lifetime message statistics. You cannot retrieve statistics that have been cleared via the **Settings** page.

2. On the **Settings** page – **Server** tab – **Server Tasks** panel, click **Clear All Statistics**.



When you click **Clear All Statistics**, a warning dialog appears that requires you to enter *CLEAR* in the available field before clicking **OK** to complete the task, which you should do *only* if you intend to clear *all* message statistics from *all* your channels.



About the Message-Processing Flow

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Fundamentally, Mirth Connect breaks the message process into a series of steps divided between a *Source* and a *Destination*. Each source and destination connector has various versions of processed messages (not all of which appear in each message):

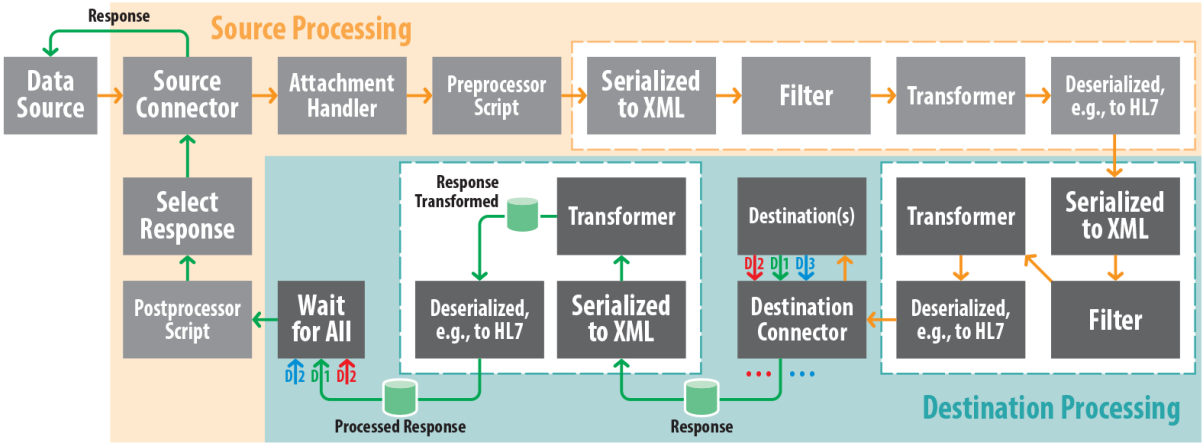
Source

- **Raw** – the state of the message as it enters the connector
- **Processed Raw** – the state of the message after passing through the pre-processor script
- **Transformed** – the message's intermediate XML representation, which exists only if a connector has filter or transformer steps (a channel that has no filters or transformers is called a *pass-through* channel)
- **Encoded** – the state of the message as it exits the transformer (includes changes made to the transformed data)
- **Response** – the message sent to the originating system

Destination

- **Raw** – the state of the message as it enters the connector; in the destination connector, it is the same as the source encoded
- **Transformed** – the message's intermediate XML representation, which exists only if a connector has filter or transformer steps (a channel that has no filters or transformers is called a *pass-through* channel)
- **Encoded** – the state of the message as it exits the transformer (includes changes made to the transformed data)
- **Sent** – the message/connector properties used by the destination connector to send messages to the outbound system
- **Response** – the message received from the outbound system after it sends the message via the destination connector
- **Response Transformed** – the intermediate XML representation of the response, which exists only if a destination connector has response-transformer steps
- **Processed Response** – the state of the response as it exits the response transformer (includes changes made to the transformed data)

The *raw* inbound message enters MC, is received by the source connector, is filtered and *transformed*, then *encoded* and sent to the destination connector. From the source connector, the raw inbound message can be passed through multiple destination connectors where it can again be influenced by filters and transformers before it is processed, encoded, and sent on.



Source Processing Steps

(Filters and transformers are optional steps that broaden the scope of message and, consequently, the task of managing it.)

1. A message is received by the source connector.
2. (Optional – an attachment must be present) The attachment handler extracts and/or stores any attachment

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

data.
3. The post-attachment-handler message is stored as *raw* content.
4. (Optional) The raw content runs through the preprocessor script and is stored as *processed raw* content.
5. The raw (or processed raw) content is serialized (converted) to XML.
6. (Optional – a filter must be present) The XML content runs through the filter.
7. (Optional – a transformer must be present) The XML content runs through the transformer.
8. (Optional – a filter/transformer must be present) The post-filter/-transformer message is converted to XML and stored as *transformed* content.
9. (Optional – a filter/transformer must be present) The XML content is encoded to the outbound data type (HL7, EDI, etc) and stored as *encoded* content.

> If no filter or transformer is run, the stored content in this step is *raw* (see Step 3), not *encoded*.

10. After all destination connectors are finished processing, the postprocesor is run.
11. A response is selected from auto-generated values, the destination-connector response, the postprocessor, or response-map values and is stored as the source connector's *response* content and returned to the data source, if needed.

Destination Processing Steps

1. The encoded content from the source connector is used by each destination connector as its *raw* content.
2. The raw content is serialized (converted) to XML.
3. (Optional – a filter must be present) The XML content is run through the filter (optional)
4. (Optional – a transformer must be present) The XML content is run through the transformer (optional)
5. (Optional – a filter/transformer must be present) The post-filter/-transformer XML content is stored as *transformed* content.
6. (Optional – a filter/transformer must be present) The converted XML is encoded to the outbound data type (HL7, EDI, etc.) and stored as the *encoded* content. If no Filter or Transformer was run, the Raw content is used as the Encoded content.

> If no filter or transformer is run, the stored content in this step is *raw* (see Step 1), not *encoded*.

7. The destination connector builds a message from all available previous content, stores it as *sent* content, and sends it to the destination.
8. The response is received by the destination connector and stored as the *response* content.
9. The response content is serialized (converted) to XML.
10. (Optional – a transformer must be present) The converted XML runs through the response transformer.
11. (Optional – a transformer must be present) The post-response-transformer XML is stored as *response transformed* content.
12. (Optional – a transformer must be present) The XML is encoded to the outbound data type (HL7, EDI, etc) and stored as *processed response* content.

About Data Types

Data types determine how to serialize (convert) inbound messages to/from XML so they can be easily modified in a transformer. Data types facilitate generating and validating responses, and batch processing.

Mirth Connect supports nine document standards or *data types*, as they are known in MC. This section details each of those data types.

- HL7 Version 2.x
- HL7 Version 3

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

- XML
- NCPDP
- EDI/X12
- DICOM
- Delimited Text
- Raw
- ASTM E1394 (commercial).

HL7 Version 2.x

Developed by Health Level 7, a non-profit healthcare-standards organization, this is an ANSI (American National Standards Institute) standard for healthcare-information exchange by hospitals, labs, pharmacies, and other healthcare sites. HL7 2.x, created in 1987 (and much more widely used than the newer HL7 version 3), is currently supported by every major health-information-systems vendor in the US. (HL7 2.x versions consist of: 2.1, 2.2, 2.3, 2.3.1, 2.4, 2.5, 2.5.1, and 2.6, all of which are backward compatible.) HL7 is a text-based standard, suing, non-standard encoding of message segments and data fields, for example:

```
MSH|^~\&|SENDAPP|SENDFAC|RECAPP|RECFAC|200812091126|SECURITY|
ADT^A01^ADT_A01|MSG00001|P|2.5|
EVN|A01|200812091126||
PID|1|PATID1234|PATID1234^5^M11^ADT1^MR^SHM~123456789^^^USSSA^SS||
    TESTLAST^TESTFIRST^E^IJR.II||19580704|M||C|
    17628 AVOCADO AV^^TUSTIN^CA^92780|OC|(714) 838-1234|(714) 838-5678|
    S|[PATID12345001^2^M10^ADT1^AN^A|123456789|U1234567^CA|
NK1|1|TESTLAST^TESTSPOUSEFIRST^K|WI^WIFE|||NK^NEXT OF KIN
PV1|1|I|2000^2012^01||||001122^SLAGMONT^LYLE^|||SUR||||ADM|A0|
```

HL7 v2x messages consist of data elements called *segments*, each with a specific role. Dissecting a message helps you see each segment at work. The **MSH** (Message Header) segment contains message-header information, including the sending application and facility, receiving application and facility, message date/time, message type, HL7 version, and ACK type. All HL7 v2x messages begin with an MSH segment. The first field defines the field delimiter (|) highlighted in red; the second field defines the component (^) and subcomponent (~) delimiters, escape characters (\ ), and repetition delimiter (&) highlighted in green: MSH|^~\&
|SENDAPP|SENDFAC|RECAPP|200812091126||SECURITY|

Each HL7 v2x message has optional fields – those that do not require data – because the information does not exist, is not vital, or is otherwise not required by a specific healthcare entity's messaging protocols (e.g., secondary phone, middle name, apartment number, external patient ID). In the previous MSH-segment example, the optional *Message Type* field is populated with the word SECURITY. The two consecutive delimiters before SECURITY indicate an unpopulated optional field.

The **PID** (Patient Identification) segment contains non-medical patient data:

```
PID|1|PATID1234|PATID1234^5^M11^ADT1^MR^SHM~123456789^^^USSSA^SS||
    FIZZBOTTOM^BURNWELL^V^||19580704|M||C|635 LONG ST^^ENID^OK^73701|
    OC|(580) 621-1234|(580) 621-5678||S||PATID12345001^2^M10^ADT1^AN^A|
    123456789|U1234567^CA|
```

These segments usually end with a carriage return “\r” (with hex spelling 0x0D). Fields in a message segment are delimited by the *pipe* character (**|**). *Composites* are fields with multiple c*omponents* and are usually delimited by a *caret* (^). In the previous example, PID.5 contains the patient’s name (in red), and is a *composite* field, separated into four *component* fields, one of which is empty, as indicated by the consecutive pipe symbols at the end of the segment.

ACK (Acknowledgment)

This is a response to a received message. The *Message Acknowledgment* (MSA) segment has the ACK data: the message was accepted, rejected, or there was some other error condition. The accept-acknowledgement type (segment MSH.15) of the incoming message can be used to determine the type of ACK behavior.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

```
MSH|^~\&|SENDAPP|SENDFAC|RECAPP|RECFAC|20080208163800||
ACK|12345678|P|2.4|||
MSA|AE|ABC098765|Referring provider not found
```

Acknowledgment codes are stored in an ACK's MSA.1 field (previous example, red highlight) and have two modes: 1) *Original* (most common): AA – Application Accept, AE – Application Error, and AR – Application Reject; and 2) *Enhanced*: CA – Commit Accept; CE – Commit Error; CR – Commit Reject.

Batch Processing

This is used to process multiple messages as a group or *batch file*. FHSs (file-header segments) and the BHS (batch-header segment) mark the beginning of the batch file. BTSs (batch-trailer segments) and the FTS (file-trailer segment) mark the end of such a file:

```
FHS|^~\&| SENDAPP|SENDFAC|RECAPP|RECFAC|20090402091523||
    BATCH013.HL7|WEEKLY UPLOAD|00009972
BHS|^~\&|SENDAPP|SENDFAC|RECAPP|RECFAC|20090402091523||||00010223
… (messages) …
BTS|3
FTS|1
```

LLP

The LLP (lower-layer protocol), aka MLLP (minimal lower-layer protocol), is the means by which HL7 v2x messages are transmitted over TCP/IP. This encoding is manifest in special hex characters at the beginning (typically 0x0B) and end (typically 0x1C and 0x0D) of HL7 v2x messages.

The encoding specifies the acknowledgment behavior.

Extending HL7 v2x

The HL7 standard has two mechanisms for extending the standard: *Z-Messages*, which allow custom segment ordering; and *Z-Segments*, which allow custom-composite and data-type ordering. While extensions are allowed, they are not usually recommended because they (especially Z-messages) introduce compatibility issues between systems.

Most Common Message Types

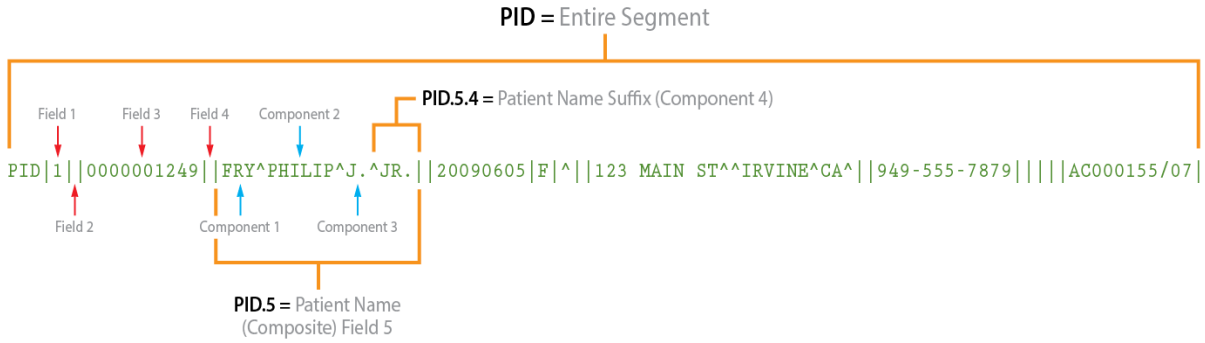
The most widely used types of HL7 v2x messages are:

- ADT (Admission, Discharge, and Transfer) – received over LLP where patient demographic data is extracted and stored on a database
- ORU (Observation Result) – generated from transcription reports, read from a database, and sent to a lab over LLP
- SCH (Schedule) – created from scheduling events and sent to a database to update a calendaring system.

HL7 v2x Notation

Mirth Connect uses dot notation to specify the parts of an HL7 message. **PID** (*Patient Identification*) refers to the entire PID segment, while **PID.5** refers to the *Patient-Name* composite, and **PID.5.4**, the *Patient-Name Suffix* component (JR).

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Line-Ending Characters

HL7 parsers (including MC's parser) expect all lines of an HL7 v2x message to end with a carriage-return character: **\r** (0x0A). Unix ends lines with a new-line character: **\n** (0x0A); Windows ends lines with a carriage-return *and* a new-line character: **\r\n** (0x0D); Mac ends lines with a carriage-return character: **\r** (0x0D). MC automatically replaces new-line characters with carriage returns.

[Back to Top]

HL7 Version 3

HL7 v3, released in 2005, is completely different from and ***not*** compatible with the 2.x iterations. HL7 v3 uses an XML encoding syntax based on a formal methodology and object-oriented principles. The RIM (reference information model) represents the semantic and lexical connections of the message data. This is an example of an HL7 v3 message:

```
xmlns:csi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemalocation="urn::hl7-org:v3 PORI_MT020001.xsd"
    <id root="61d1a24f-41aa-ae95-f8ac302f4eeb"/>
    <code code="93000" codeSystem="2.16.840.1.113883.6.12
        codeSystemName="CPT-4"/>
    <effectiveTime>
        <low value="20021122091000.000" inclusive="true"/>
        <high value="20021122091000.000" inclusive="false"/>
    </effectiveTime>
    <confidentialityCode code="???D_BLINDED???"
        codeSystemName="???CDISC???"/>
    <reasonCode code="???PER_PROTOCOL???" codeSystemName="???CDISC???"/>
```

[Back to Top]

XML (Extensible Markup Language)

This open standard is used to define structured data, encoding documents, and serializing data. XML is platform-independent (can run on platforms other than the one used to implement it) and allows user-defined markup elements. An XML schema, especially DTD (document-type definition), can be used to define a document's validity. This is an example of XML language:

```
<?xml version='1.0' encoding='UTF-8'?>
```


Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

```
<patient id="12345">
    <name>
    <first>John</first>
        <last>Zoidberg</last>
        <middle/>
    </name>
    <dateOfBirth>19640416</dateOfBirth>
    <gender>male</gender>
    <employer>Planet Express</employer>
    <employer>General Hospital</employer>
</patient>
```

XML documents are allowed one root element; however, the declaration that precedes it (previous example, top line)—which defines the XML version, character encoding, and external dependency data—while convenient, is not required. The element content in an XML document is between the *start* and *end* tags (e.g., **<tag>This is the content</tag>**). These case-sensitive elements can have attributes that provide additional data about the element (e.g., **<patient id="12345">**).

Empty Elements

Empty elements are represented as **<element></element>** or **<element />**.

Special Characters

As in HTML, entities are used to represent special characters: **&**, **<** and **>**. Text in a CDATA (character data) block is interpreted as character text, not markup: **<! [CDATA [x < 5 & y > 10]]>**.

Namespaces

These let you avoid element-attribute name conflicts (e.g., patient and physician elements both have the child element: *uniqueID*). References to same-name attributes would be ambiguous without namespaces, so you define a namespace with an *xmlns* attribute at the start of the element (next example, red highlight) or in the root element itself. The value of a namespace attribute is a URI (uniform resource identifier). Add the namespace to the element with a prefix, such as:

```
<message xmlns:pt="http://www.mirthcorp.com/patient" xmlns:dr="http://www.mirthcorp.com/physician">
    <pt:patient>
        <pt:uniqueID>A12345</pt:uniqueID>
    </pt:patient>
    <dr:physician>
        <dr:uniqueID>Z98765</dr:uniqueID>
    <dr:physician>
</message>
```

[Back to Top]

NCPDP (National Council for Prescription Drug Programs)

These messages contain prescription-related transactions. The main components of an NCPDP message are a fixed-length header, one or more segments, and one or more transactions. NCPDP message fields, groups, and segments are delimited by non-printable characters: 0x1C, 0x1D, 0x1E. Two-letter codes define text fields:

```
610066C4B1123456789010445636 63bbbbbbbb2003050198765bbbbb
AM01C419620615C51CAJOSEPHCBSMITHCM123MAINSTREETCNMY
TOWNCOCOCP34567CQ2014658923CZ50Z1231C2AM04C2987654321A
CCJOSEPHCDSMITHAM07EM1D21234567E103D700009011312E71
```


Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

D30D530D61D80DE20030501DF5DJ1DK9ET1C82DT128EAAM03EZ06
DBA12345DRGOITERMANPM20136895722E1DL123456H51014ETROY
AM11D9557{DC100{DX100{H71H801H9150{DQ700{DU807{DN3

[Back to Top]

EDI/X12

EDI and X12 data-type logic is similar enough that Mirth Connect combines them to eliminate any redundancy when converting message data to/from XML.

EDI (Electronic-Data Interchange)

This family of standards is used to transmit data between computer systems or networks. EDI messages consist of a series of segments from headers to user data to trailers. Segments are usually delimited by a single-quote, elements in segments are usually delimited by a plus sign, and sub-elements are usually delimited by a colon, as shown in this example:

UNB+UNOA:1+005435656:1+006415160:1+060515:1434+00000000000778'
UNH+00000000000117+INVOIC:D:97B:UN'BGM+380+342459+9'DTM+3:20060515:102'
RFF+ON:521052'NAD+BY+792820524::16++CUMMINS MID-RANGE ENGINE PLANT'
NAD+SE+005435656::16++GENERAL WIDGET COMPANY'CUX+1:USD'
LIN+1++157870:IN'IMD+F++:::WIDGET'QTY+47:1020:EA'ALI+US'MOA+203:1202.58'
PRI+INV:1.179'LIN+2++157871:IN'IMD+F++:::DIFFERENT WIDGET'QTY+47:20:EA'
ALI+JP'MOA+203:410'PRI+INV:20.5'UNS+S'MOA+39:2137.58'ALC+C+ABG'MOA+8:525'
UNT+23+00000000000117'UNZ+1+00000000000778'

X12

ANSI ASC X12 is a US standard for EDI (electronic-data interchange). The X12 data type consists of a series of segments from headers to trailers. A transaction set defines the segments, which are delimited by a tilde (**~**). Elements in segments are usually delimited by an asterisk and sub-elements by a colon, as shown in this example:

ISA*00* *00* *ZZ*EMEDNYBAT
*ZZ*QCQC*051129*1300*U*00401*333000048*0*P*:~
GS*HP*EMEDNYBAT*QCQC*20051129*13004600*333000048*X*004010X091A1~
ST*835*201400020~BPR*I*0*C*NON**********20051031~
TRN*1*NO PAYMENT*1141797357~REF*EV*QCQC~REF*F2*004010X091A1~
DTM*405*20051031~N1*PR*NYSDOH~
N3*OFFICE OF MEDICAID MGMT*CORNING TOWER, EMPIRE STATE PLAZA~
N4*ALBANY*NY*122370080~
… (truncated) …
SE*28*201400020~
GE*1*333000048~
IEA*1*333000048~

[Back to Top]

DICOM (Digital Imaging and Communications)

This standard is used for storing, printing, and transmitting medical imaging data. It is both a binary file format and a communications protocol over TCP/IP. Besides images, DICOM messages can contain patient data.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

[Back to Top]

Delimited Text

This is a user-defined file format in which records and fields are delimited by a specified character or character sequence (comma, pipe, other). Delimited text fields can be defined by column widths, but the most common delimited format is CSV (comma-separated values):

```
PatientID,LastName,FirstName,MiddleInit,DOB,SSN,Gender
12345,Farnsworth,Hubert,J,19160920,360543210,M
67890,Wong,Amy,19810704,232661963,F
12121,Fry,Philip,J,19730901,558901234,M
```

[Back to Top]

Raw

This data type enables any generic message string to be accepted (e.g., you can use the Raw data type to accept a Base64 message). No conversion to or from XML occurs in filters or transformers.

[Back to Top]

ASTM E1394

The ASTM E1394 extension for MC lets you easily accept, parse, and transform messages following the ASTM E1394 data standard. As with other data types, an incoming message is converted to a simple XML format, after which all the usual transformer steps can be used to transform the message or convert it to/from another data type. Several data-type properties are also provided for specifying the conversion parameters. A standard vocabulary for ASTM E1394 is included to help identify fields in a transformer's message-template trees. (See also About the ASTM E1391 Data Type.)

> The ASTM E1394 data type requires a Platinum-level support subscription. For technical and purchasing information about this data type, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

[Back to Top]

Setting Message Templates

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

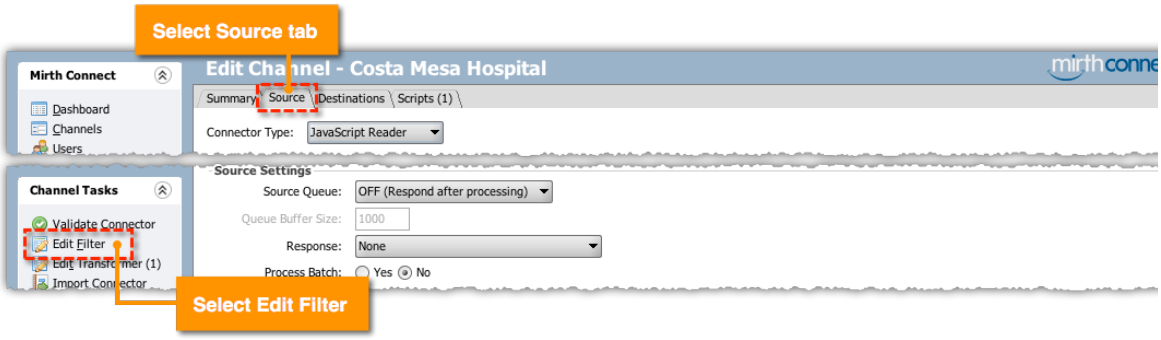
Each channel connector's transformer has inbound- and outbound-message templates.

Setting an Inbound-Message Template

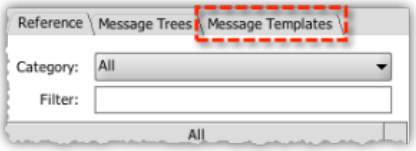
The inbound template is primarily for the convenience of analyzing data from a message and using the drag-drop function to create filter and transformer steps. This procedure instructs you how to set an inbound-message template, which you can access via the **msg** (message) variable.

You can perform this procedure via the **Source** or **Destinations** tab using the **Edit Filter** or **Edit Transformer** channel task. This procedure uses the **Source** tab and the **Edit Filter** task.

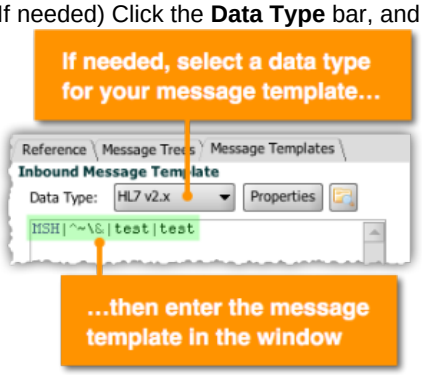
1. Navigate to an **Edit Channel** page, and click the **Source** tab.



2. In the **Channel Tasks** panel (previous graphic), click **Edit Filter**.
3. In the top-right corner of the page, click the **Message Templates** tab.

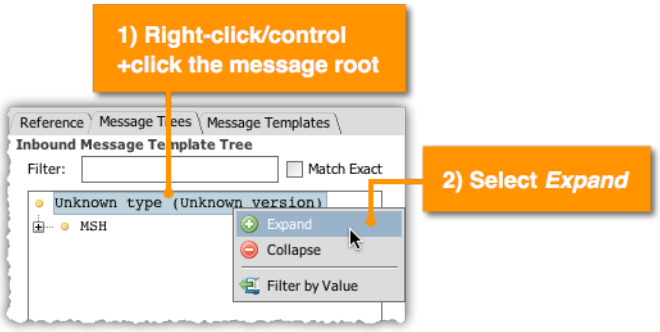


4. (If needed) Click the **Data Type** bar, and select the message's data type.



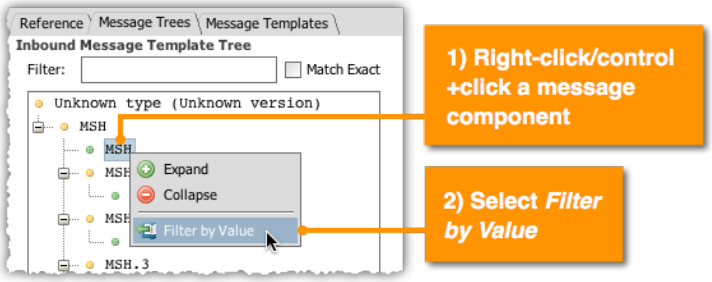
5. In the **Inbound Message Template** window (previous graphic), enter your message template.
6. Click the **Message Trees** tab, then right-click/control+click the message root node, and on the drop-down, select *Expand*.)

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



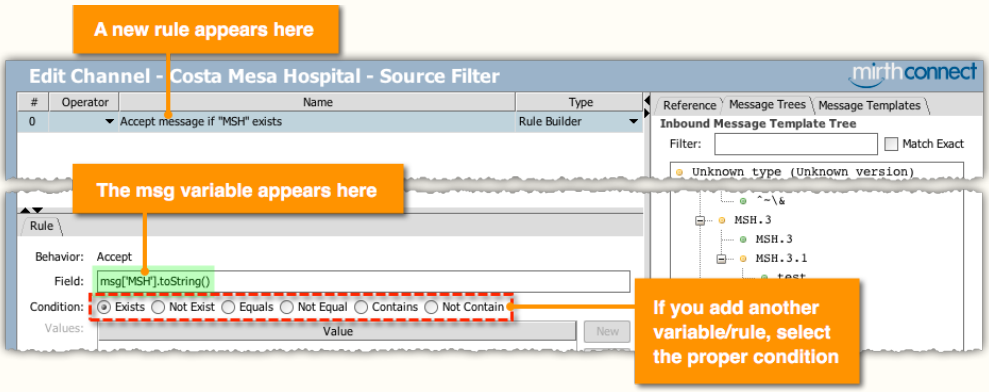
Clicking the Expand/Collapse (+/-) icon next to the top message component expands only that branch of the tree. Depending on the component you want to select in the next step, using this method may require clicking several +/- icons to find the desired component. Using the method from this step conveniently expands the entire tree in one action, eliminating the need to click through various branches to find the desired component.

7. In the expanded message tree, right-click/control+click the desired message component, and select *Filter by Value*.



When you select *Filter by Value*, three things happen simultaneously:

- Various features appear in the space below the Rules window.
- The **msg** (message) variable for the selected message component appears in the **Field** field.
- A new rule appears in the Rules window.



There is no restriction on messages that the Inbound Message Templates can accept. There are, however, certain messages the Outbound Message Templates cannot accept.

Do ***not*** include protected health information (PHI) in message templates! Though Base64-encoded within

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

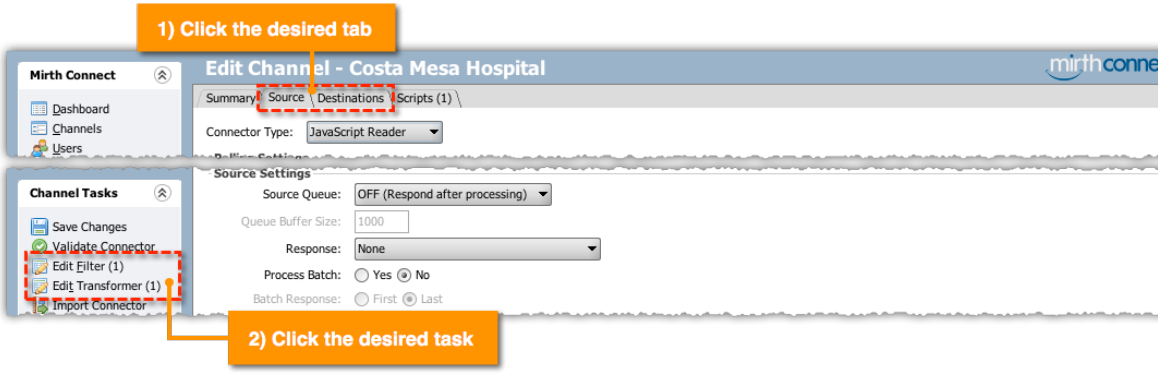
the export file, they are ***not*** encrypted. Storing unencrypted PHI is a HIPAA violation; therefore, you should use only “dummy” messages as templates. You can create dummy messages with the commercial *Message Generating* extension, available for purchase from Mirth Corporation. For details, see About the Message Generator Plugin.

Setting an Outbound-Message Template

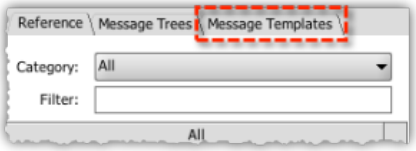
Outbound-message templates define the actual outbound message before it is sent. You should only use these templates to transform a message from one protocol to another or from one message type to another. For this reason, if the outbound template is set, the encoded outbound message is the *outbound* message with any transformations. If the outbound template is ***not*** set, the encoded outbound message is the *inbound* message with any transformations. This means that the data of the message set in the outbound template is sent to the destination. For example, if you set an outbound template and parse out only one message segment, the outbound message contains only that segment.

This procedure instructs you how to set an outbound-message template, which you can access via the **tmp** (template) variable. The procedure uses the **Source** tab and **Edit Transformer** task but could use either the **Source** or **Destinations** tab and the **Edit Filter** or **Edit Transformer** task as needed.

1. Navigate to an **Edit Channel** page, and click the **Source** tab.



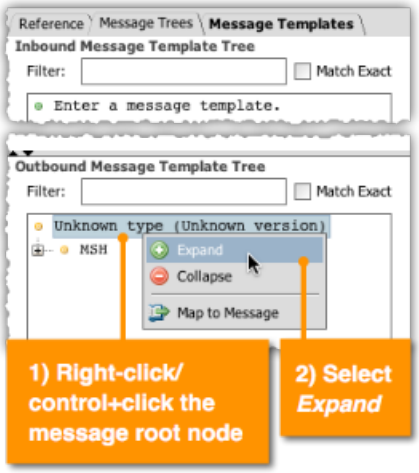
2. In the **Channel Tasks** panel, click **Edit Transformer** (previous graphic).
3. In the top-right corner of the page, click the **Message Templates** tab.



4. (If needed) Click the **Data Type** bar, and select the message's data type.

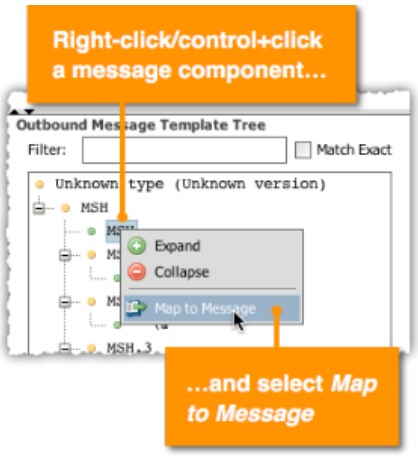
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

5. In the **Outbound Message Template** window (previous graphic), enter your message template.
6. Click the **Message Trees** tab, then right-click/control+click the message root node, and select *Expand*.



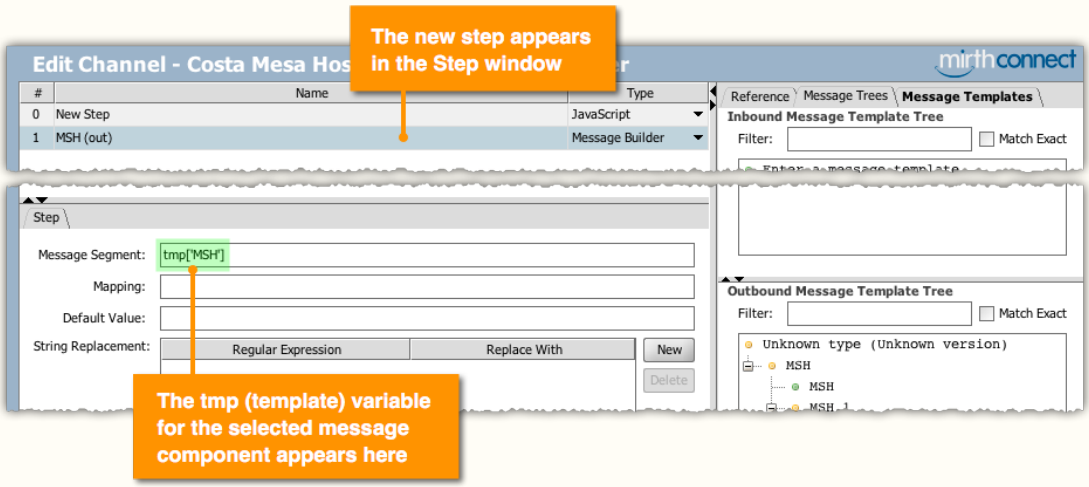
7. In the expanded message tree, right-click/control+click the desired message component, and select *Map to Message*.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



When you select *Map to Message*, three things happen simultaneously:

- Various features appear in the blank space below the Step Window.
- The **tmp** (template) variable for the selected message component appears in the **Message Segment** field.
- A new step appears in the Step Window.



Each destination connector receives a separate copy of the encoded outbound message from the source connector.

Do ***not*** include protected health information (PHI) in message templates! Though Base64-encoded within the export file, they are ***not*** encrypted. Storing unencrypted PHI is a HIPAA violation; therefore, you should use only “dummy” messages as templates. You can create dummy messages with the commercial *Message Generating* extension, available for purchase from Mirth Corporation. For details, see About the Message Generator Plugin.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

About Filters

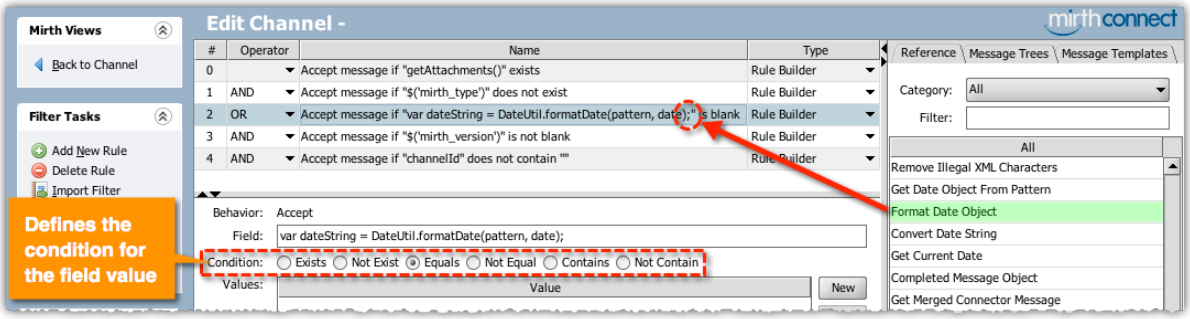
Filters are composed of a series of rules evaluated as Boolean expressions. If these rules are evaluated as *True*, the message is processed by the connector. Each **Source** and **Destination** connector has its own filters, but you can use filters on a **Destination** connector to route messages to specific destinations.

Rules can be created via Rule Builder or written in JavaScript with E4X. JavaScript steps let you define code in the step's editor. An External-Script step points to an external file where code is located. A Rule Builder step is automatically created when you drag from an Inbound Message Template field and drop into the filters list area. List items are ranked by order of precedence. The Rule Builder step **Field** defines the JavaScript/E4X code used to access the desired message field. This field is automatically populated by dragging a segment from the parsed data of the Inbound-Message Template to the **Filter** page window.

Be sure to include **toString()** at the end of the path. This field can contain JavaScript code and function calls, but they do not end the code with a semicolon. Some code templates in the **Reference** tab, however, ***do*** end with a semicolon (next graphic).

Condition (next graphic) defines the condition type to apply to the **Field** value:

- **Exists** – Accept the message if the field exists
- **Not Exist** – Accept the message if the field does not exist
- **Equals** – Accept the message if the field is blank
- **Not Equal** – Accept the message if the field is not blank
- **Contains** – Accept the message if the field contains a certain variable
- **Not Contain** – Accept the message if the field does not contain a certain variable.

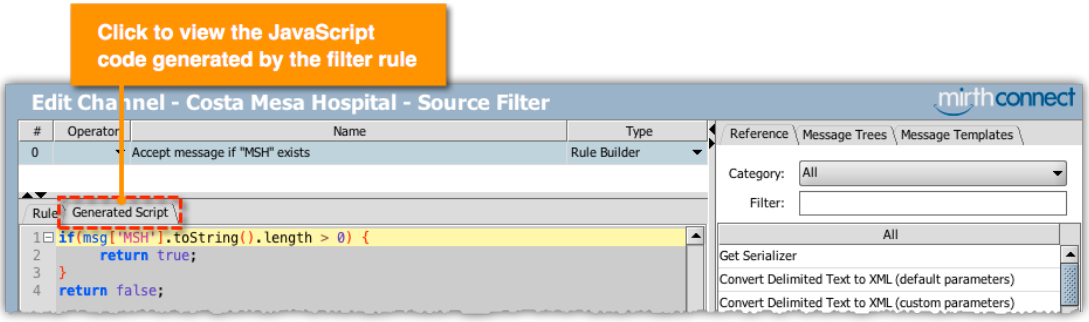


For the **Exists** condition, the field contains any value; for **Not Exist**, the field is empty. All other conditions (**Equals**, **Not Equal**, **Contains**, **Not Contain**) use the **Value** list, which defines one or more values to which to compare the **Field** value. A value can be a string literal or variable. The comparison is case-sensitive, and there is no defined limit to the number of values you can implement.

Accessing Generated Script

You can access the JavaScript code that is generated by a filter rule to simply view the script or copy it into your own custom JavaScript code. On an **Edit Channel** page > **Source**/**Destination** page – **Channel Tasks** panel, click *Edit Filter*, and either select a rule on the list, or in the **Filter Tasks** panel, click *Add New Rule*, then click the **Generated Script** tab below the list to reveal the JavaScript code generated by the filter rule.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



About Transformers

Transformers are composed of steps that are executed in order and that let message data be changed, extracted for later use, or used to create a new message. Each connector has its own set of transformers.

Transformer Maps

These extract field data from a message to a variable in a *map*, a term that refers to the scope of the variable. The available maps are:

- **Connector** – Variables added to this map are only available to later transformer steps in the same connector, including destination-configuration fields if the connector in question is a destination.
- **Channel** – Variables added to this map are available to later transformer steps in the same channel, including all filters, transformers, configuration fields (for connectors after the connector in question), and the postprocessor script.
- **Global Channel** – Variables added to this map are available to all parts of a channel, including its Deploy, Undeploy, Preprocessor, and Postprocessor scripts with the option to keep values persistent across redeploys on the channel's **Summary** page.
- **Global** – Variables added to this map are available across all channels and scripts with the option to have persistent values across redeploys on the **Settings** page.
- **Response** – Variables added to this map can be used as the ACK response for source connectors that support ACK responses. (All destination connectors place their responses into this map automatically.)
- **Source** – Variables are added to this (read-only) map by the source connector; these variables are available in the same area as the channel map variables.

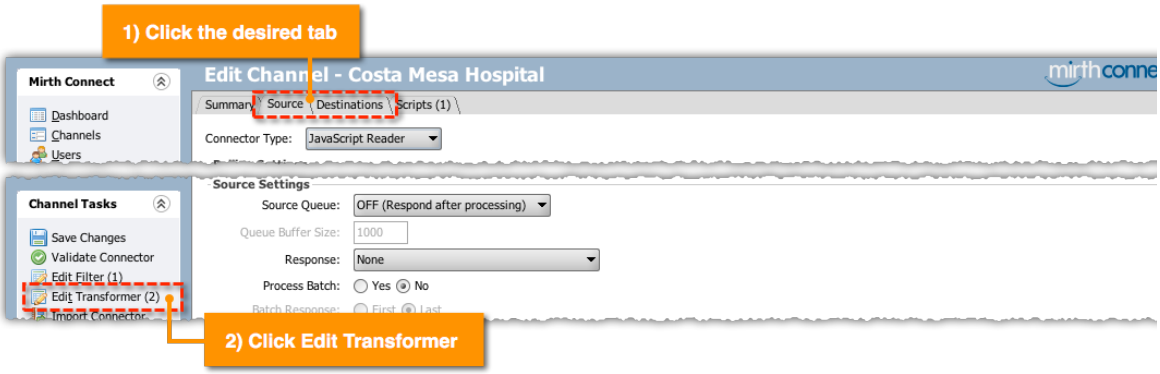
The mapper is created automatically when you drag message data from the parsed data of the inbound-message template and drop it onto the **Transformer** page. The maps define the scope of the variable. When a variable is mapped, it becomes available in the destination mappings for certain destinations. Mapped variables are also available to later filter and transformer steps of the same scope.

Creating a Mapper Step

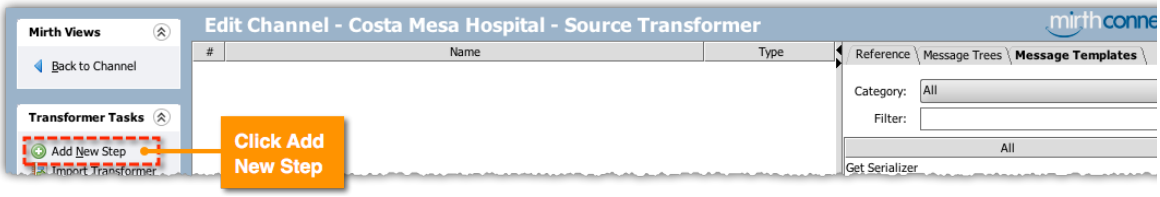
Use a mapper step to add a custom value to a variable map (e.g., a channel map).

1. Navigate to an **Edit Channel** page, and click the **Source** (or **Destinations**) tab.

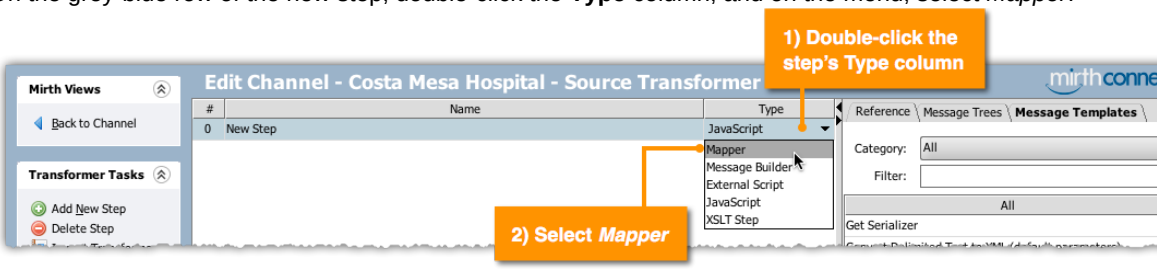
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



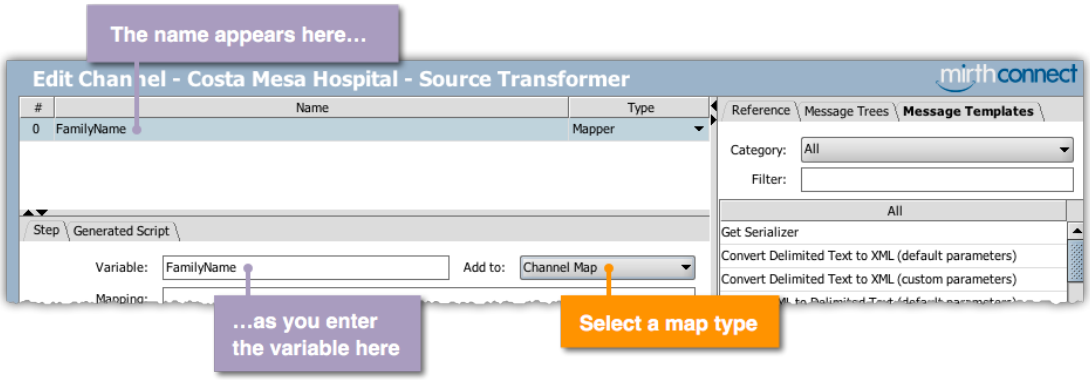
2. In the **Channel Tasks** panel (previous graphic), click **Edit Transformer**.
3. In the **Transformer Tasks** panel, click **Add New Step**.



4. On the grey-blue row of the new step, double-click the **Type** column, and on the menu, select *Mapper*.

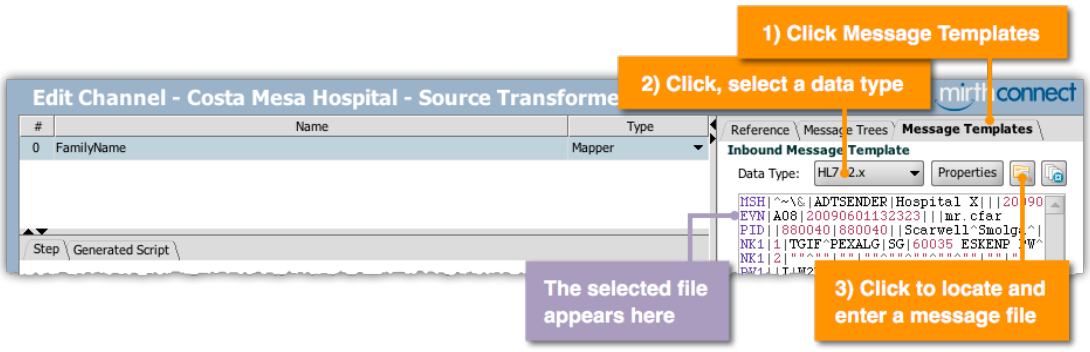


5. In the **Variable** field, enter the variable for the step. (Your entry simultaneously appears in the step's **Name** column).

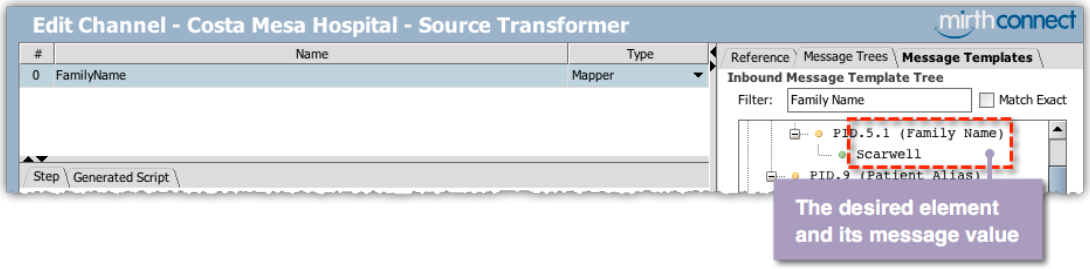


6. Click the **Add to** bar (previous graphic), and select a map type.
7. In the top-right corner of the page, click the **Message Templates** tab.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

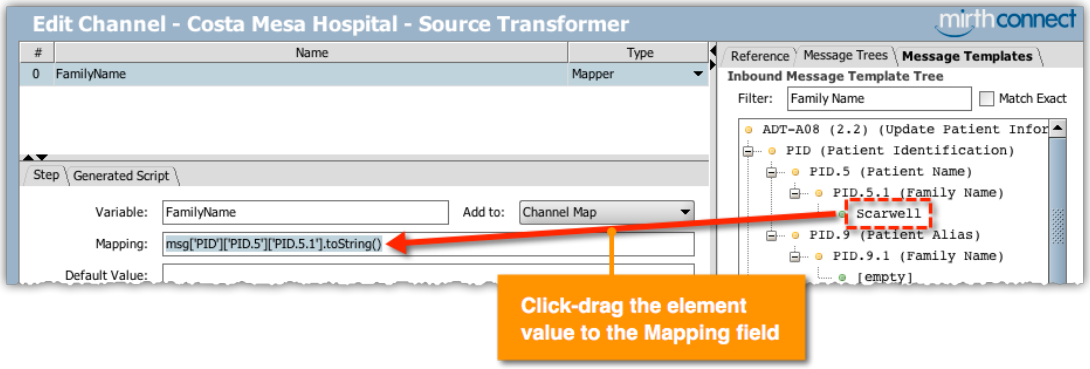


8. Click the **Data Type** bar (previous graphic), and select a data type.
9. Click the **Open File…** icon (previous graphic), navigate to the desired message, and enter it in the **Inbound Message Template** window.
10. Click the **Message Trees** tab, and in the tree, locate the element you entered in the **Variable** field.



> Enter (some or all of) the variable in the **Filter** field to expedite your search. The message tree expands automatically to reveal all matches of your search.

11. Click-drag the desired element to the **Mapping** field, and release the mouse button.



12. Configure other step traits as desired, and in the **Mirth Views** panel on the top-left of the page, click **Back to Channel**.
13. In the **Channel Tasks** panel, click **Save Changes**.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

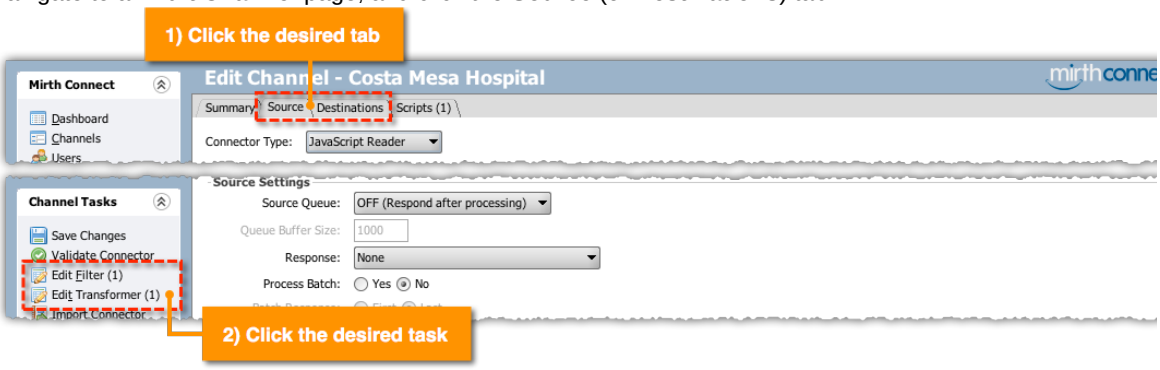
Creating a Message Builder Step

As with the mapper and much of the drop-drag functionality in MC, you can create a message-builder step by dragging-dropping from an inbound-message template field. The message builder copies the data from the **Inbound** field in **msg** to the **Outbound** field in **tmp**. You can also use the message builder to set fields in an inbound-message template to arbitrary values or values from other message fields.

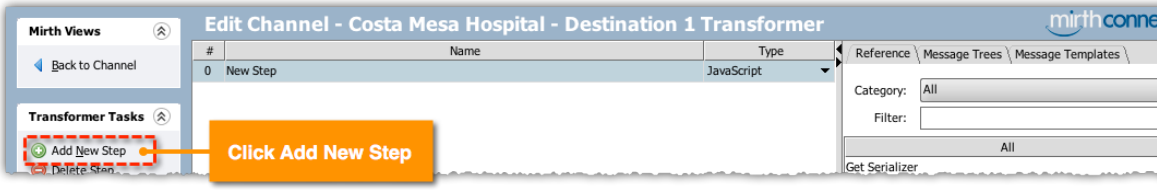
Creating an XSLT Step

XSLTs (extensible stylesheet language transformations) convert XML documents to formats such as HTML, plain text, and other XML types. The XSLT template/stylesheet defines how data is transformed.

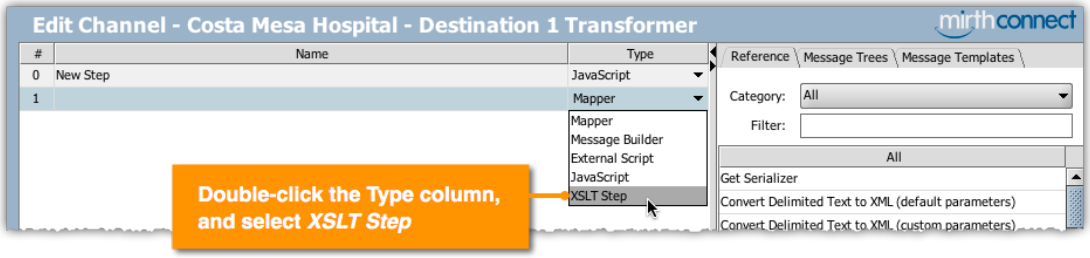
1. Navigate to an **Edit Channel** page, and click the **Source** (or **Destinations**) tab.



2. In the **Channel Tasks** panel, click **Edit Transformer** (previous graphic).
3. In the **Transformer Tasks** panel, click **Add New Step**.

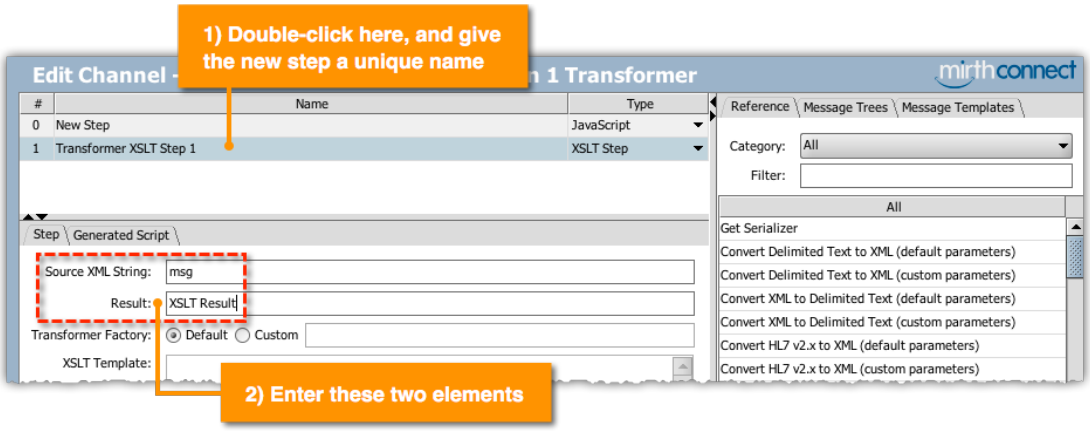


4. On the grey-blue row of the new step, double-click the **Type** column, and on the menu, select *XSLT Step*.

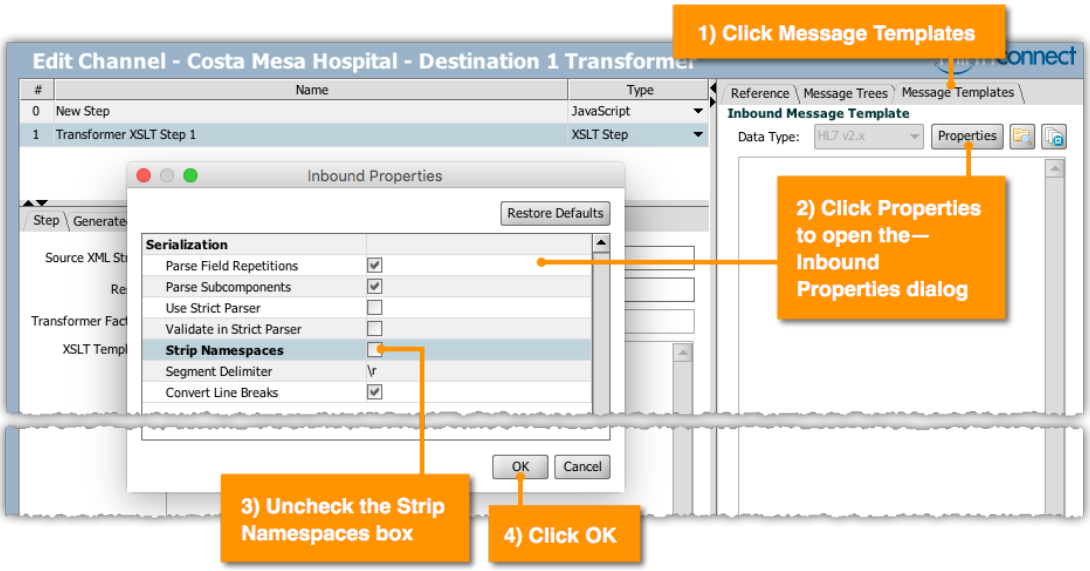


5. On the same row, double-click the **Name** column, and enter a name for the new step.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



6. In the **Source XML String** field (previous graphic), enter the variable that contains XML code (e.g., **msg**) to transform, then in the **Result** field, enter a name for the channel-map variable in which to store the results.
7. Click the **Message Templates** tab > **Properties** button, and uncheck the **Strip Namespaces** box, then click the **OK** button.



JavaScript and External Script

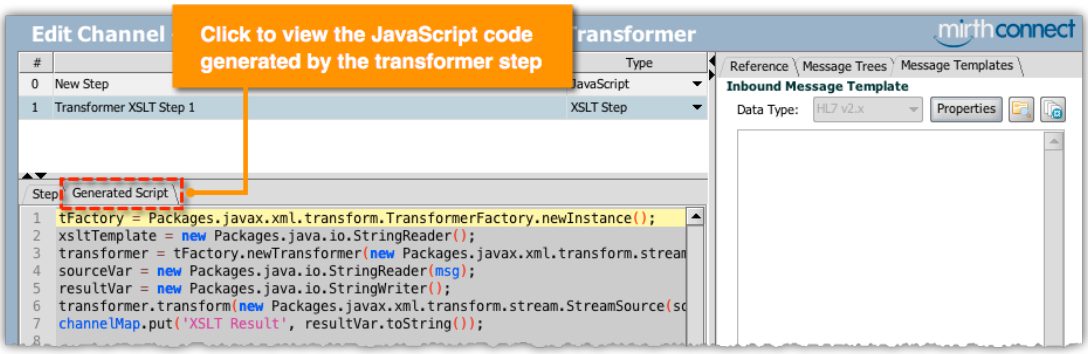
You can execute an arbitrary block of JavaScript to access message content and metadata. (For details on JavaScript, see the Mirth Connect and JavaScript section.) The JavaScript step lets you define the code in the step via its editor. In an External Script step, you can configure paths to external locations where a file that contains

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

JavaScript is kept.

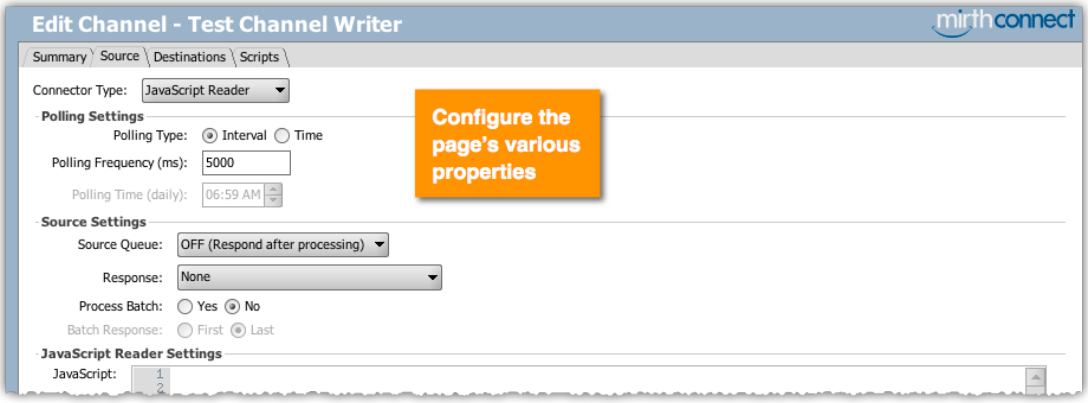
Accessing Generated Script

You can access the JavaScript code that is generated by a transformer step to simply view the script or copy it into your own custom JavaScript code. On an **Edit Channel** page > **Source**/**Destination** page – **Channel Tasks** panel, click *Edit Transformer*, and either select a step on the list, or in the **Transformer Tasks** panel, click *Add New Step*, then click the **Generated Script** tab below the list to reveal the JavaScript code generated by the transformer step.



About Connectors

Mirth Connect links to external systems via one *source* connector and one or more *destination* connectors. Supported protocols include LLP, TCP, HTTP, HTTPS (Silver-/Gold-/Platinum-level support only), JMS, File, and Database. Each connector type has its own properties that you configure via the **Edit Channel** page > **Source** or **Destinations** tab; e.g., under **Connector Type** – **JavaScript Reader**, you can configure Polling (Type, Frequency, Time), Source (Source Queue, Response, Process Batch, Batch Response), and Javascript Reader Settings.



First, MC communicates with the source system via a source connector. Channels define message interfaces to trade data between systems. Source connectors link to a data source and listen/poll for data. Filters accept/reject messages based on programmed rules. Transformers manipulate and extract data from messages and can even create a new outbound message from inbound data. Destination connectors route transformed messages to destination systems and applications.

Source Connectors	Destination Connectors
Channel Reader	Channel Writer
DICOM Listener	DICOM Sender

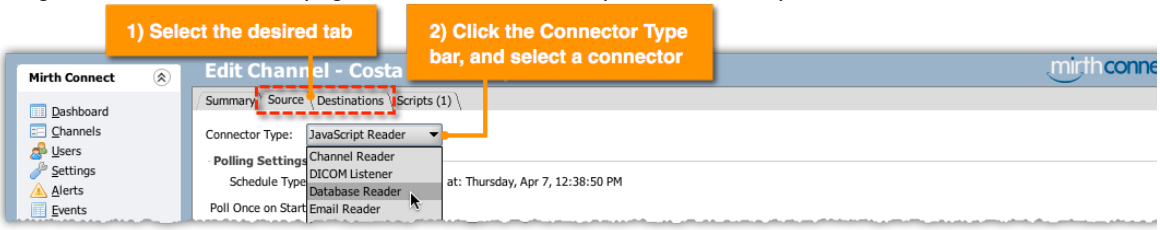
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Database Reader	Database Writer
—	Document Writer
File Reader	File Writer
HTTP Listener	HTTP Sender
JMS Listener	JMS Sender
JavaScript Reader	JavaScript Writer
—	Mirth Results Sender
Email Reader*	SMTP Sender
TCP Listener	TCP Sender
Web-Service Listener	Web Service Sender
Serial Reader^	Serial Sender^
*Commercial plugin (requires a Gold-level support subscription) ^Commercial plugin (requires a Platinum-level support subscription)	

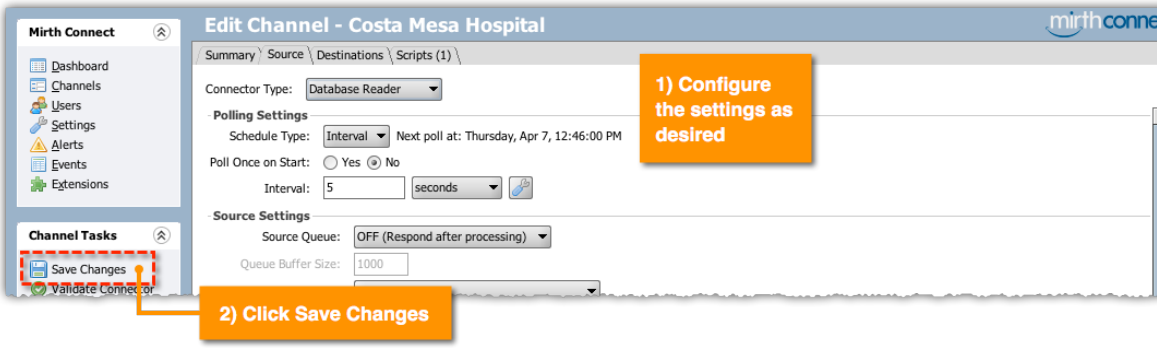
Selecting a Connector Type

Connector-type pages feature parameter options specific to how the connector should function regarding a given message in a given channel. This procedure instructs you how to select a connector type. Make sure you perform the procedure for the source *and* destination connectors.

1. Navigate to an **Edit Channel** page, and click the **Source** (or **Destinations**) tab.



2. Click the **Connector Type** bar (previous graphic), and select a connector.
3. Configure settings as needed, and in the **Channel Tasks** panel, click **Save Changes**.

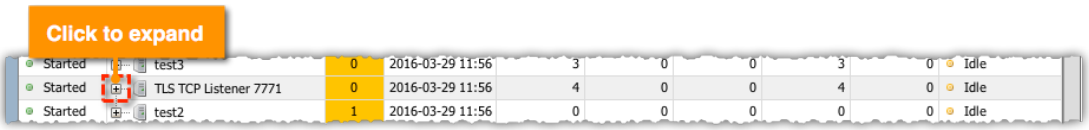


Viewing Connector Statistics

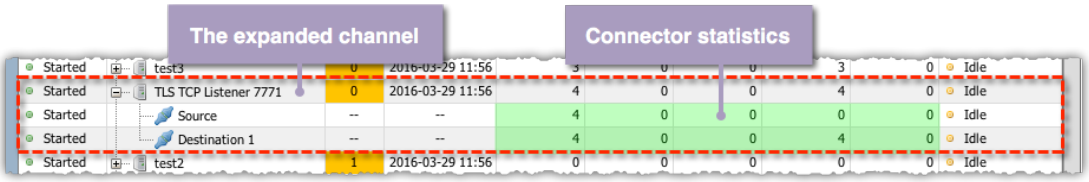
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

On the **Dashboard** page, you can monitor channel activity by viewing channel statistics, specifically, messages *Received/Filtered/Queued/Sent/Errored*. Channel statistics, however, represent the sum of all connector activity in the channel; they do not break down the activity of individual connectors, which you may need to verify on occasion. If, for example, the figure in a channel's **Errored** column is on the rise, only a look at the channel's *connector* statistics would let you determine which connector(s) is/are experiencing the problem.

To view statistics for a channel's individual connectors, click the plus sign in the **Name** column of a channel.



The channel expands to reveal its connectors and their statistics.



About File Connectors

These connectors consist of a *File Reader* (source) and a *File Writer* (destination), which let you read from and write to a file:

- on a local **file** system
- over **ftp** (file-transfor protocol) or **sftp** (secure file-transfer protocol)
- on an **smb** (Samba) network share
- on a **webdav** (web-distributed authoring and versioning) share.

Besides file connectors, this section contains information about how to select a File Reader/Writer "method," and encode/decode message data sent between File Reader/Writer connectors.

File Reader

This source (or *polling*) connector lets Mirth Connect continuously poll (examine) local or network file systems for message files to read. The *File Reader* connector supports HL7 batch-file processing. The file-name filter pattern can consist of wildcard characters (*****, **?**) to match the pattern, multiple patterns in a comma-separated list, or regular expressions. You can set the *File Reader* to poll a directory on a set interval or at a particular time of day.

> You should delete or move files after they are processed; otherwise, the files are read repeatedly, which slows system performance.

You can check a file's age before it is processed. This is particularly important with large files that take more than a few milliseconds to populate a directory. If MC tries to process a file that has not been fully written-to in the target directory, an error results.

File Writer

This destination connector lets MC write files to a local or network file system. A Velocity template is available to dynamically generate files.

> Velocity is a Java-based template engine used in various channel fields as a placeholder for an object of a variable. The syntax is **${objectName}**. Most destination mappings are Velocity template variables, which

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

let you use dynamic variables (instead of static, hard-coded values) when setting source/destination properties. Velocity template variables are easily confused with Javascript global map lookups: **$('objectName')**. Notice the single-quote marks inside the parentheses for the Java version.

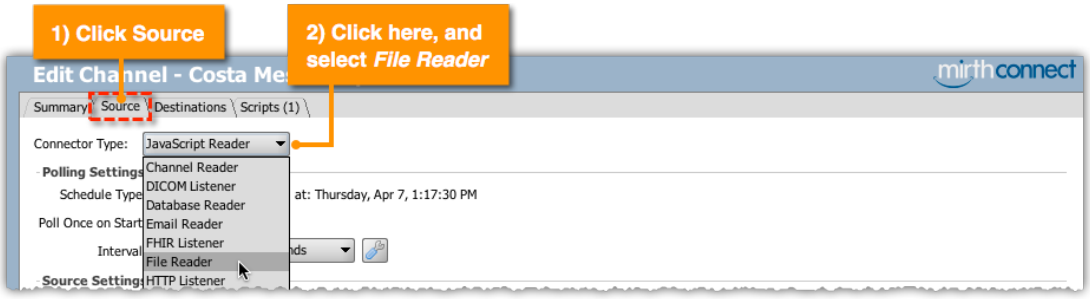
The **Template** text box (**Edit Channel** page > **Destinations** tab > **Connector Type** bar > **File Writer** – **File Writer Settings** section) defines the data that is written to a file, which can be free-form text and can include Map variables.

You can *Append* or *Overwrite* an existing file (**Edit Channel** page > **Destinations** tab > **Connector Type** bar > **File Writer** – **File Writer Settings** section – **File Exists**). *File Writer* automatically creates a Response Map variable (from the message) that shows the name of the file that was written.

Selecting a File Connector Method

Each read/write element appears as a *method* in the **File Reader** (**Source** tab)/**File Writer** (**Destinations** tab) section of an **Edit Channel** page. This procedure instructs you how to select a file-reader method via the **Source** tab. (Selecting a file-*writer* method is the same except: In Step 1, click the **Destinations** tab, and in Step 3, select a *writer* method.)

1. Navigate to an **Edit Channel** page, and click the **Source** tab.



2. Click the **Connector Type** bar (previous graphic), and select **File Reader**.
3. In the **File Reader Settings** section, click the **Method** bar, and select a reader method.

4. Configure other elements as needed, and in the **Channel Tasks** panel, click **Save Changes**.

The **ftp** method supports anonymous/non-anonymous logins, passive/active modes, and connection validation before each operation. File connectors support both ASCII (American standard code for information interchange) and binary files.

Encoding/Decoding Data

Binary data can be added to messages using Base64 encoding, which is a MIME (multi-purpose internet mail extensions) content-transfer encoding scheme that represents the data with alphanumeric characters:

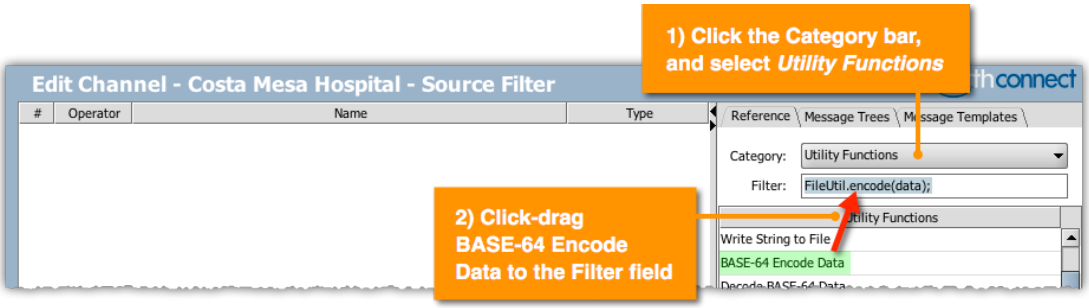
IDU1OCA1NDYgXSANL0JvcmR1ciBbIDAgMCAwIF0gDT4+IA1lbmRvYmoNMTIlIDAgb2JqDTw8IA0v<LF>
RGVzdCAoRzcuMTc0NTkpDS9UeXBlIC0Bbm5vdCANL1NlYnR5cGUgL0xpbmsgDS9SZWN0IFsgNTQg<LF>
NTIwIDU1OCA1MzMgXSANL0JvcmRlciBbIDAgMCAwIF0gDT4+IA1lbmRvYmoNMTI2IDAgb2JqDTw8<LF>
IA0vRGVzdCAoRzcuMTkzNzApDS9UeXBlIC9Bbm5vdCANL1NlYnR5cGUgL0xpbmsgDS9SZWN0IFsg<LF>

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

NTQgNTA3IDU1OCAlMjAgXSANL0JvcmRlciBbIDAgMCAwIF0gDT4+IA1lbmRvYmoNMTI3IDAgb2Jq<LF>
DTw8IA0vRGVzdCAoRzcuMTkzNTUpDS9UeXBlIC9Bbm5vdCANL1NlYnR5cGUgL0xpbmsgDS9SZWN0<LF>
IFsgNTQgNDk0IDU1OCA1MDcgXSANL0JvcmRlciBbIDAgMCAwIF0gDT4+IA1lbmRvYmoNMTI4IDAg<LF>

File Reader and File Writer connectors support binary data files (using the connector configuration option). File Reader automatically converts incoming binary data to Base64, and by default, the message variable is Base64 data. Conversely, File Writer automatically converts outbound Base64 data to binary data. You can use a postprocessor script to encapsulate Base64 data into an XML structure.

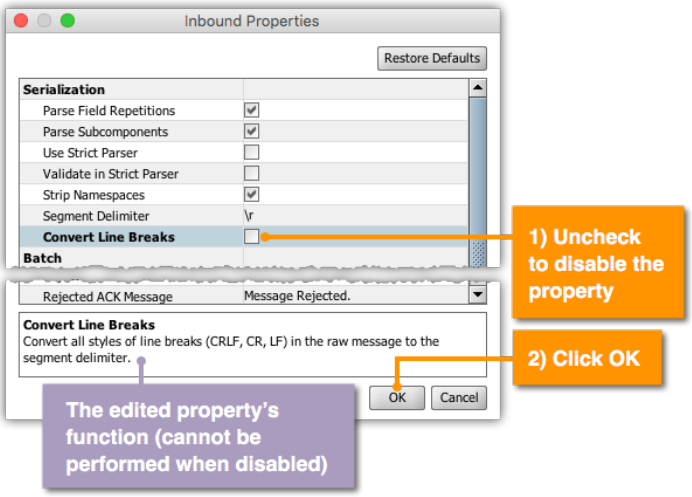
To encode binary data using JavaScript, you would call the **FileUtil.encode()** method, which takes the byte array as a parameter and returns a string. To access this method, navigate to an **Edit Channels** page, and proceed via the **Source** (or **Destination**) tab – **Channel Tasks** panel > **Edit Filter** (or **Transformer**) – **Reference** tab > **Category** drop-down > ***Utility Functions*****,** and click-drag **BASE-64 Encode Data** up into the Filter field.



To decode binary data using JavaScript, you would call the **FileUtil.decode()** method, which takes a string parameter and returns a byte array. Access this method as you would for the **BASE-64 Encode Data** method; **Decode BASE-64 Data** is the next line down on the menu.

When you embed Base64 data into HL7 messages, many systems require line-feed characters to be escaped. As an example, use the string object's **replace()** function to escape the line-feed character \X0A\: Base64data.replace (/ \n/g, \\X0A\\). When you extract Base64 data from inbound HL7 messages, if the linefeed characters are escaped, replace the escape sequences with actual line-feed characters. As an example, use the string object's **replace()** function to replace escape characters with line-feed characters: base64data.replace (/ \\X0A\\ /g, "\n").

If line-feed characters in the inbound message are not escaped, disable the **Inbound Message Template** property *Convert Line Breaks* for all connectors that handle the Base64. Access the **Properties** dialog via **Edit Channel** > **Source** (or **Destinations**) tab – **Channel Tasks** panel > **Edit Filter** (or **Edit Transformer**) > **Message Templates** tab > **Properties** button.



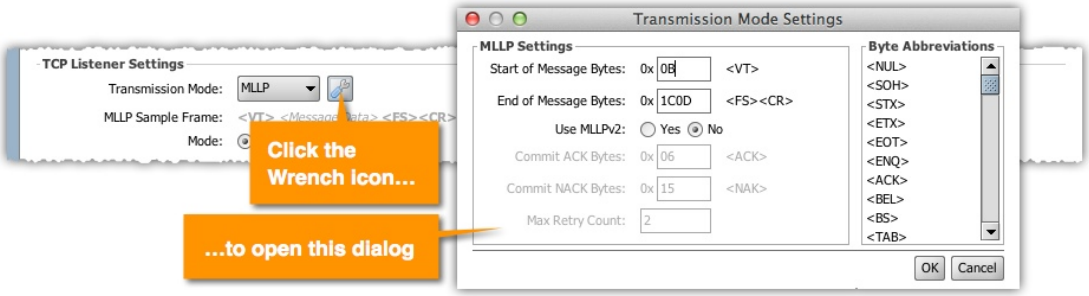
You could also use a preprocessor script to strip out the line-feed characters.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

About TCP Connectors

Transmission control protocol connectors consist of the *TCP Listener* (source) and the *TCP Sender* (destination). These connectors support binary or ASCII data and send/receive data over the TCP protocol. TCP connectors also support multiple transmission modes, including Basic TCP and MLLP.

To configure frame encoding and characters, click the wrench icon next to the **Transmission Mode** bar (**Edit Channel** page > **Source/Destinations** tab > **Connector Type** bar > **TCP Listener/Sender** – **TCP Settings** section).



TCP Listener

This connector lets Mirth Connect listen for inbound messages over a standard TCP connection. A response can be generated by the destination and sent on a new connection. Automatic acknowledgment generation can be enabled by setting the inbound data type to *HL7 v2.x* and selecting an *Auto-generate* option from the **Response Settings** menu. HL7 batch-file processing can also be enabled by setting the inbound data type to *HL7 v2.x* and enabling the *Process Batch* option in the **TCP Listener Settings** section.

The TCP Listener can act as a *server* or a *client*. In Server mode, you listen for connections from other clients; in Client mode, you set up the connector to link to a TCP server.

> In most cases, you use Server mode because the TCP Listener looks for incoming connections.

ACK codes and messages are configurable by editing the inbound HL7 v2.x data-type properties, and ACK behavior can be based on the MSH.15 HL7 segment: AL, ER, NE, SU. You can also send ACKs on a new connection.

TCP Sender

This connector lets MC send messages to a TCP server. The TCP Sender supports connection retries and persistent queues. A response from the TCP Sender can be forwarded to another channel.

The **Template** field (**Edit Channel** page > **Destinations** tab > **Connector Type** bar > **TCP Sender** – **TCP Sender Settings** section) defines data to send over the connection and creates a Response Map variable with data received in the response.

The **Process HL7 ACK** function (**TCP Sender Settings** section) bases message statistics on the ACK code: AA indicates a *Sent* message; AE indicates an *Error* message; AR indicates a *Filtered* message. HL7 ACK validation codes can be configured on the response inbound HL7 v2.x data-type properties. The TCP Sender creates a Response Map variable with ACK messages returned by destination server.

ASTM E1381 Transmission Mode (Option)

This is a commercial extension you can use with TCP connectors to send/receive data via the ASTM E1381

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

lower-layer protocol standard. By default, the official ASTM E1381 standard is strictly followed, but several settings are available to support variants of the standard. The extension supports specifying the bytes used for frame delimiters, the checksum algorithm used, various timeouts, et al.

> ASTM E1381 transmission mode requires a Platinum-level support subscription. For technical and purchasing information about ASTM E1381 transmission mode, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About Database Connectors

These connectors consist of a *Database Reader* (source) and a *Database Writer* (destination) and support any database with a JDBC (Java database connectivity) driver. Mirth Connect supports six database drivers:

- Sun JDBC-ODBC (Java database connectivity-open database connectivity) Bridge
- MySQL (structured query language)
- Oracle
- PostgreSQL
- SQL Server/Sybase
- SQLite.

> Access these options via an **Edit Channel** page > **Source** tab > **Connector Type** bar > **Database Reader** – **Database Reader Settings** section > **Driver** bar – OR – **Edit Channel** page > **Destinations** tab > **Connector Type** bar > **Database Writer** – **Database Writer Settings** section > **Driver** bar.

Database connectors support either straight SQL or JavaScript. An SQL (structured query language) creation feature is included in MC for easy code generation.

You can add databases to MC via JDBC. Put the JDBC driver in a **.jar** file, then create a resource (see About the Resources Page) for your **.jar** file. Add database entries via the MC **installation** folder > **/conf** folder > **dbdrivers.xml** file. The entry needs to specify the driver class, database name, and URL template. You may specify SELECT statement to get column names for specified tables.

> If the tables are not specified, column metadata is retrieved from the TABLE object for all tables, which creates overhead.

Database Reader

This connector lets MC *poll* (examine) any supported JDBC-compatible database for data. Rows are returned as XML, and JavaScript can be used for advanced logic.

When you select Database Reader:

- the Incoming Data property is automatically set to XML
- the Inbound Message Template is automatically generated from selected columns
- one XML message is generated per row returned by query.

Selected columns are available as variables to use with an On-Update SQL statement, which runs for each message after all destinations and postprocessor scripts have run. The On-Update statement even runs when an error occurs in your channel.

Database Writer

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

This connector lets MC write to any JDBC-compatible database with Insert, Update, or JavaScript statements. The Database Writer creates a Response Map variable with the message of *Database write success*.

About Channel Connectors

These connectors consist of a *Channel Reader* (source) and a *Channel Writer* (destination) and are used for internal message routing between channels. During development, channel connectors are commonly used as *null* connectors, essentially making them connector placeholders.

Channel Reader

This connector lets Mirth Connect wait for inbound messages to be forwarded from other channels.

Channel Writer

This connector lets MC select a channel to which messages are sent. The Channel Id field supports velocity replacement to determine dynamically the channel to which each message is sent. The Message Metadata feature lets you add/delete the map variables to be forwarded to the source map of the destination channel's message.

About HTTP Connectors

Hypertext transfer protocol connectors consist of an *HTTP Listener* (source) and an *HTTP Sender* (destination).

HTTP Listener

This connector – which can respond from any destination connector – acts as a web server and lets Mirth Connect listen for inbound HTTP data on one or more interfaces. The message can be body-content-only of any data type, or it can have an XML structure that may contain not only the body content but also request information, headers, and query parameters. For XML messages, the content of multipart requests may also be parsed into XML:

```
<HttpRequest>
    <Content boundary="d8iabj234jfdv90iu3oj2kl" multipart="yes">
        <Part>
            <Headers>
                <Content-Disposition>form-data; name="username"</Content-Disposition>
                <Content-Type>text/plain</Content-Type>
            </Headers>
            <Content multipart="no">jdoe</Content>
        </Part>
        <Part>
            <Headers>
                <Content-Disposition>form-data; name="uploadPicture"; filename="mypicture.jpg"</Content-
                    Disposition>
                <Content-Type>image/jpeg</Content-Type>
            </Headers>
            <Content encoding="Base64" multipart="no">/9j/4AAQSkZJRqABAQEASABIAAD/
2wBDACgcHiMeGSgjISMtKygwPGRBPDc3PHtYXUlkkYCZlo+A&#13;
jIqgtObDoKrarYqMyP/L2u71////m8H////6/+b9//j/2wBDASstLTw1PHZBQXb4pYyl+Pj4+Pj4&#13;
+Pj4+Pj4+Pj4+Pj4+Pj4+Pj4+Pj4+Pj4+Pj4+Pj4+Pj4
+Pj4+Pj/wgARCABgAGQDAREA&#13;
AhEBAxEB/8QAGAAAAwEBAAAAAAAAAAAAAAAAAAQIDAAT/xAAWAQEBAQAAAAAAAAAAAAAAAAAAQL/
    &#13;
```


Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

2gAMAwEAAhADEAAAARmkYYpYwBhRBQCwqkQodNNYIQAJVRAGXEh1olaAI1CCTAiqxEcReq5jKquh

MAQkUFFVzoRSSslACgJGCpgGghQmVQCCwllFaRia0SpltYRIQnSDygzqMMMUsxomsNRYcy4MUCEA
QGOfWQzVjBVolQBCE57lawYwVIRpQEcYhYlgCYaXGCAYpK5//8QAHhAAAglCAwEBAAAAAAAAAAAA
AAEQEQlgMDFBIUL/2gAIAQEAAQUChFiK48RcKGNiZZZZe7H3GPWXYuoemRQkJ0ZfSjHXz2bFZZ7P
+YQ3CZ2ew482ULZSuay4fN4uf//EABQRAQAAAAAAAAAAAAAAAAAAAHD/2gAIAQMBAT8BHP/EABQR&#
13;
AQAAAAAAAAAAAAAAAAAAAD/2gAIAQIBAT8BHP/EABQQAQAAAAAAAAAAAAAAAAAAAHD/2gAIAQEA&
#13;
Bj8CHP/EACAQAAMBAAMAAQUAAAAAAAAAAABERAhMUEgMGFxgZH/2gAIAQEAAT8hxpmnjplGMeoY&
#13;
+OxLpnTg8epoYIJoMu0WqlH1tmEgonlwFwnFKJjPf0JzjkEj2LPSYhlx+il94xCn4OgihjWn4Mn0
U+smVldBZwyhi5RdR2PRijafY9aZTkQn8PuWKiY38EUonjolYT4O/pJIFhCD+ChfghCZ/9oADAMB
AAIAAwAAABBbl4g2n6ZgMIf9fTwMJQ97O9LYDewtz40OJ82R2v8Aje6Vvb+/PuN6GIHNS2ORnM2
k9v6py2CINWSxA//xAAbEQACAwADAAAAAAAAAAAAAAABEQAQMCBAUP/aAAgBAwEBPxDi9DQ1UWLo&
#13;
wYqilNziOwKerjj8H//xAAgBAwEBPxDFbBR6
RrdAw+mdqii7TTn/xAAqEAEAAglCAwEBAQAAAAAAAAABABEhMUFRYXGBEJGx/9oACAEBAAE/ECFn

EHkEwaG/M5biOG5e4IL8T2iiy48y1YiVgjHzdQUbPw2VBz9Op0I9w16jLgfY6AOMxFLVgCrj9xm
0Q5mAdzD5cwymVOY/mlNXzKKHgiJcHzElu+ZIEOsz+/wZBLCz7GNuFKPjWCa+JcAmlu/UogepjwS
x2pavUxjSNXNHxCgYhisMYYkMRjmVVMwAo7i2uZXwwWUmSVbi+;1T3FITLPUFBzOpUAoIJ1NOsR8
tLc8WBJvJ/BjvUwJ9ir/SUjbUI2sHmWMGUJe4NoTIUEqDmXwsdTa34juz8C1Qq1KmLi8mnR1Cx
2mX3KEIvEdx3FiFGADUo5qFvwdirMAdSgMcL6IWHqLceYOYOfwZamtQaJYARtiO6eZp4ZfiGDMGG
4MG2IDdRtg1MqEHcvMdY7jr7KSf/2Q0KLS1kOGIhYmoyMzRqZmR2OTBpdTNvajJrbA==
</Content>
</Part>
</Content>
</HttpRequest>

The content of incoming requests can be encoded into Base64 strings if the Content-Type header matches one of the binary MIME types specified in the connector properties.

Specific incoming GET requests can be automatically tied to static resources. The resource content can come from a file or from a custom user-defined string.

> Mirth Connect supports several types of HTTP authentication to perform for incoming requests. To view these authentication types and explanations of each one, see the About Web Service Connectors section > **Web Service Listener** > **HTTP Authentication Types.**

HTTP Sender

This connector, which lets MC send messages to an HTTP server, supports:

- using a proxy server to handle requests
- GET, POST, PUT, and DELETE methods
- Multipart content type for binary data, which can be used with the POST method.
- optional BASIC or DIGEST authentication
- persistent queues
- header variables
- query parameters.

There are **Content Type** and **Content** fields for the request body.

The ACK response can be forwarded to another channel via an **Edit Channel** page > **Destinations** tab > **Connector Type** bar > **HTTP Sender**. The ACK response is stored in the Response Map or forwarded to another channel. The response can include *Headers and Body as XML* or simply response *Body Only* content (**Edit Channel** page > **Destinations** tab > **Connector Type** bar > **HTTP Sender** – **HTTP Sender Settings** section – **Response Content**). For XML responses, the body is enclosed in a CDATA block, as with this example:

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

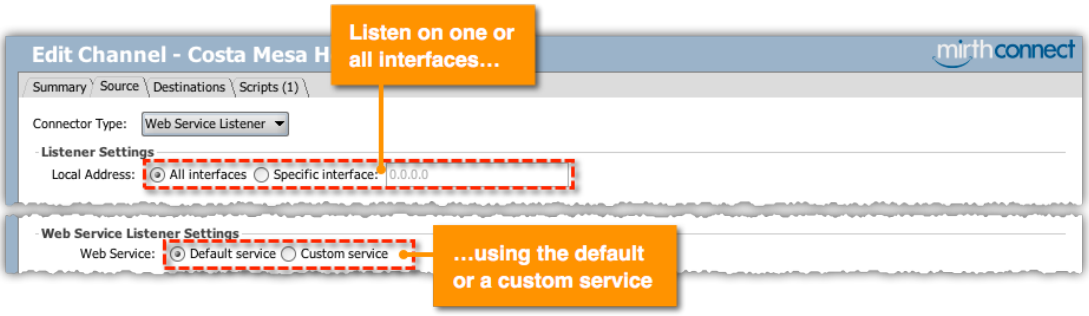
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<HttpResponse>
    <Status>HTTP/1.1 200 OK</Status>
    <Header>
        <Field>
            <Name>Date</Name>
            <Value>Fri, 15 Oct 2011 01:13:04 GMT</Value>
        </Field>
        <Field>
            <Name>Server</Name>
            <Value>Jetty/5.1.15 (Windows Vista/6.0
x86 java/1.6.0_20</Value>

        </Field>
        …
    </Header>
    <Body><![CDATA[This is the HTTP respons
body/content]]></Body>
</HttpResponse>
```

About Web Service Connectors

Web services provide an API (application programming interface) to execute services hosted on a remote system, using HTTP to transmit the data. Mirth Connect uses SOAP (simple object access protocol) to define execution methods, parameters, etc for messages. The available web service is defined by a WSDL (web services description language). These connectors consist of a **Web Service Listener** (source) and **Web Service Sender** (destination).

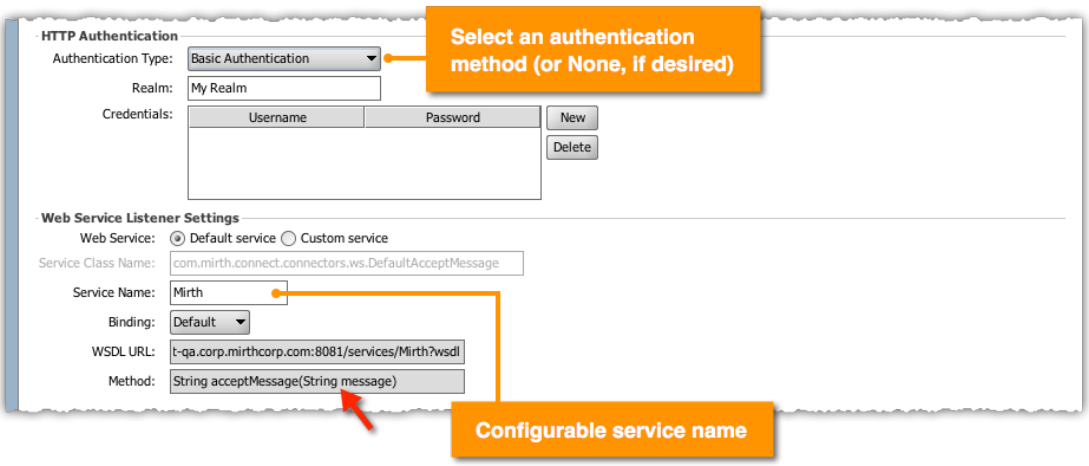
Web Service Listener

This connector lets MC listen on an HTTP port for inbound web-service calls and provides a WSDL (web-service definition language) for SOAP clients to use. You can set the **Web Service Listener** to listen on *All interfaces* or on a *Specific interface* using the.



The default service receives messages via the acceptMessage() method (red arrow, next graphic). The web **Service Name** is configurable, and a response can be generated by destination. The **Web Service Listener** supports five types of HTTP authentication (or you can select no authentication).

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



You can build your own web service, put it in your Mirth Connect installation directory, and point to it instead of the **DefaultAcceptMessage** class. The easiest way to implement your own web-service class is to take the **DefaultAcceptMessage.java** file (from the MC source-code repository), and modify it to your needs. To do this, you need to keep extending **acceptMessage** as part of the code. You can create multiple web-service methods, but each must contain the **processData()** method so it can be recognized by the Web Service connector. For example:

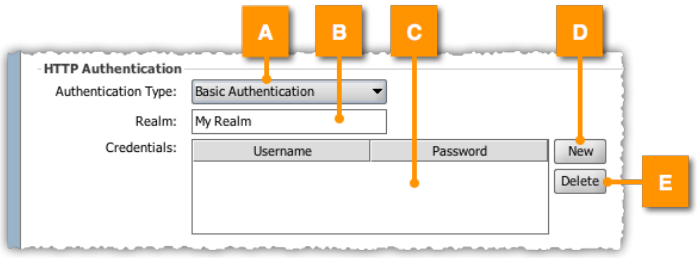
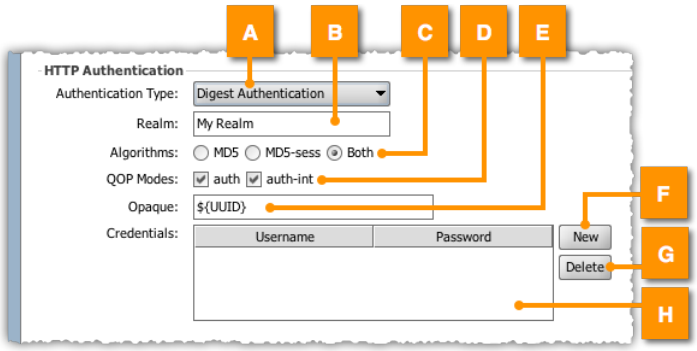
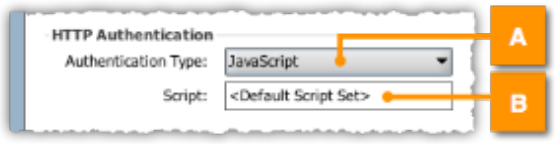
```
public String acceptMessage(String message) {
    return
webServiceMessageReceiver.processData(message);
}
```

HTTP Authentication Types

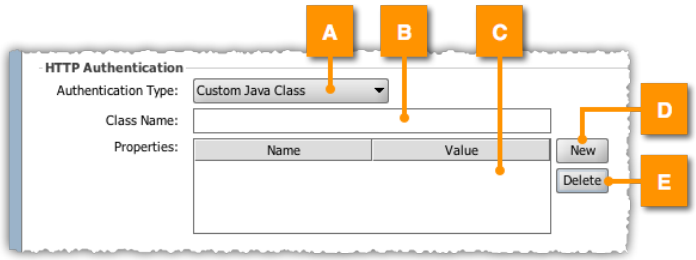
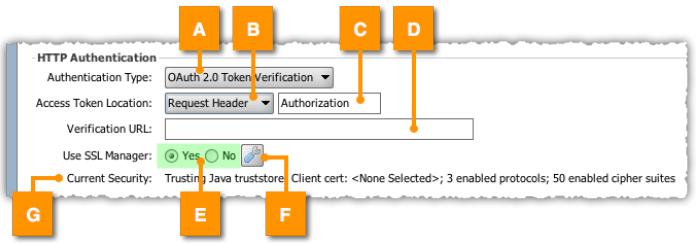
This table lists and describes the various HTTP authentication types.

Authentication Type	Description of Elements
None 	This selection means there will be no authentication; thus, there are no configurable elements.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Basic Authentication 	A – Authentication Type bar: selects the HTTP authentication type B – Realm field: enter the realm name C – Credentials window: enter authentication username and password D – New: click to add a set of authentication credentials E – Delete: click to delete the selected authentication credentials
Digest Authentication 	A – Authentication Type bar: selects the HTTP authentication type B – Realm field: enter the realm name C – Algorithms: specifies the digest algorithms supported by this server; MD5: The security data A1 will contain the username, realm, and password; MD5-sess: The security data A1 will also contain the server and client nonces; Both: Combines the elements of both algorithm options D – QOP Modes: The quality-of-protection modes to support (check one or both); auth: Regular auth with client nonce and count in the digest; auth-int: Same as auth but with message-integrity protection enabled E – Opaque: A string of data that should be returned by the client unchanged F – New: click to add a set of authentication credentials G – Delete: click to delete the selected authentication credentials H – Credentials: enter authentication username and password
JavaScript 	A – Authentication Type bar: selects the HTTP authentication type B – Script field: click the field to open the HTTP Authentication Script dialog; the return value of this script is used to accept/reject requests

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

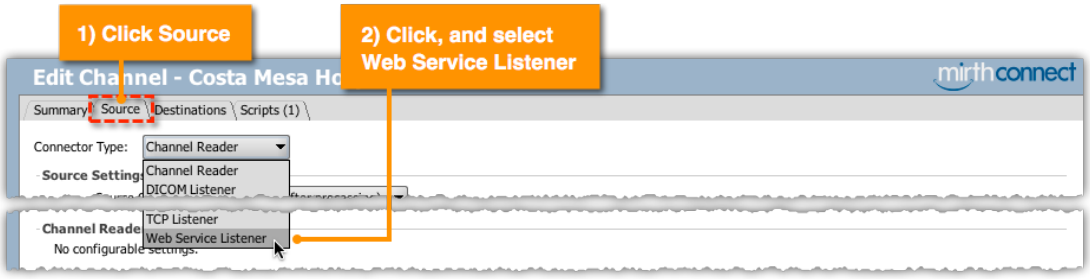
Custom Java Class 	A – Authentication Type bar: selects the HTTP authentication type B – Class Name: The fully qualified Java-class name of the authenticator class to use C – Properties: (Optional) properties to pass into the authenticator class when it is substantiated D – New: Click to add a property E – Delete: Click to delete the selected property
OAuth 2.0 Token Verification 	A – Authentication Type bar: selects the HTTP authentication type B – Determines where (in the Request Header or Query Parameter) the access token is located in client requests C – The header or query parameter to pass along with the verification request D – The HTTP URL to which to perform a GET request for access-token verification; if the response code is 400, the server rejects the authentication attempt E – If Yes, messages are dispatched using SSL via the server and client certificates specified in the advanced settings F – Click to open the SSL Settings dialog on which you can configure Destination SSL Settings G – Current security settings based on your Destination SSL Settings

Creating a Custom Web Service

To create and use a custom web service, you first need to create a Java class that extends the MC class **com.mirth.connect.connectors.ws.AcceptMessage**. Build a **.jar** file containing the class, then create a resource (see About the Resources Page) containing your .jar file. Restart the MC server, and in the **Mirth Connect** panel, click **Channels**, select a channel, and proceed via **Channel Tasks** panel > **Edit Channel** > **Source** tab – **Connector Type** menu > **Web Service Listener**. In the Web Service Listener attributes, select *Custom service* to enable the **Service Class Name** field, in which you specify the name of the class that extends acceptMessage().

1. Restart the Mirth Connect server, and navigate to an **Edit Channel** page.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



2. (Previous graphic) Click the **Source** tab > **Connector Type** bar > **Web Service Listener**.
3. In the **Web Service Listener Settings** section – **Web Service**, select **Custom service** to enable the **Service Class Name** field.

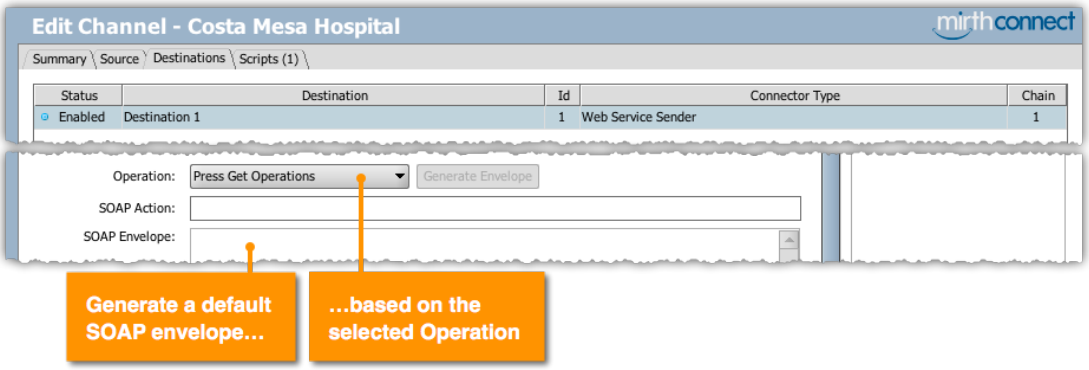
4. Delete the default service-class name, and enter a name for your custom web service.

> Mirth Connect populates the **Service Class Name** field with the default service-class name merely to illustrate the proper naming convention. This name is *not* for actual use and will not lead to the creation of your custom web service.

5. In the **Channel Tasks** panel, click **Save Changes**.

Web Service Sender

The *Web Service Sender* (**Edit Channel** > **Destinations** tab) can auto-populate fields and available operations based on a given WSDL (Web Service Definition Language). If there are multiple services/ports defined in the WSDL, they can be selected with a drop-down menu. *Web Service Sender* creates the default **SOAP Envelope** based on the selected **Operation**. (You can edit the envelope to provide required parameters, including using destination mappings.)



Invocation can be two-way (default) or one-way. WB Sender supports web-service authentication, persistent queuing, and binary attachments using MTOM (message-transmission-optimization mechanism). As with other connectors, the response can be forwarded to another channel, and the *Web Service Sender* creates a response-map variable that contains the response envelope from the destination web service. The default **Service** is **com.mirth.connect.connectors.ws.DefaultAcceptMessage**.

Using a web-service client (e.g., the *Web Service Sender* in MC), you should be able to invoke any method listed within the class you wrote that contains the **processData()** method.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

About JavaScript Connectors

For details on these connectors, see JavaScript Connectors in the *Mirth Connect and JavaScript* section of this guide.

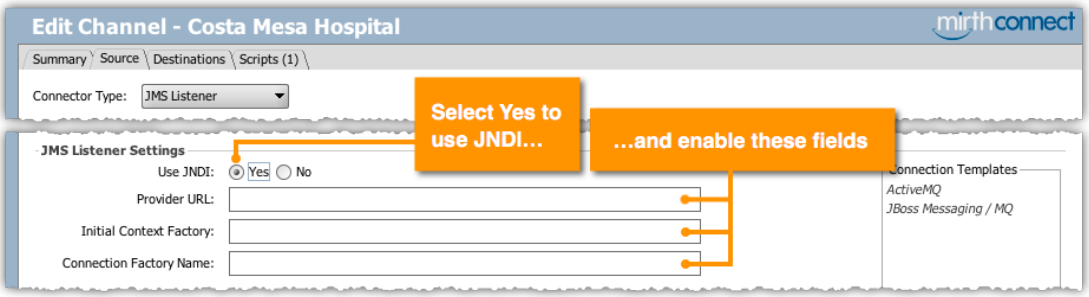
About the Mirth Results Connector

This is a commercial plugin available via subscription from Mirth Corporation. For more information about the Mirth Results connector, see About the Mirth Results Connector Plugin in the Commercial Extensions section of this guide.

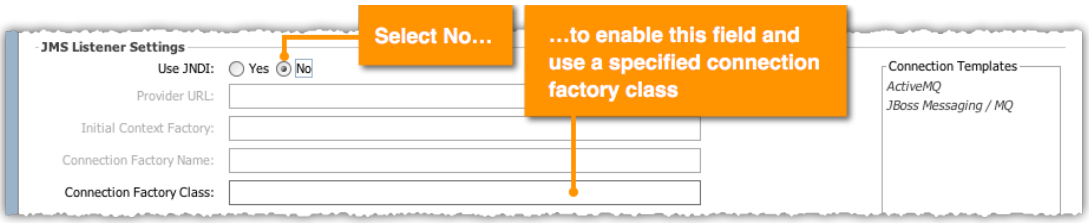
About JMS Connectors

Java messaging service connectors consist of a *JMS Listener* (source) and a *JMS Sender* (destination). JMS is a Java API (application programming interface) that sends messages between a provider and one or more clients, which API you use to listen or send to/from a JMS queue or topic. **Queue** messages are consumed by one client, **Topic** messages by multiple clients.

The JMS connector can use JNDI (Java naming and directory interface) or a specified connection factory class to make connections via an **Edit Channel** page > **Source/Destinations** tab > **Connector Type** bar > **JMS Reader/Writer** – **JMS Listener/Sender Settings** section:



or



JMS Listener

This connector lets MC read messages from a JMS queue. You can use an optional **Selector** expression to get a subset of available messages (**Edit Channel** page > **Source** tab > **Connector Type** bar > **JMS Listener** – **JMS Sender Settings** section).

JMS Sender

This connector lets MC write messages to a JMS queue. The **Template** field defines data to be placed in the JMS queue as the message (**Edit Channel** page > **Destinations** tab > **Connector Type** bar > **JMS Sender** – **JMS Sender Settings** section). The *JMS Sender* creates a response-map variable with the message: **Jms message sent** .

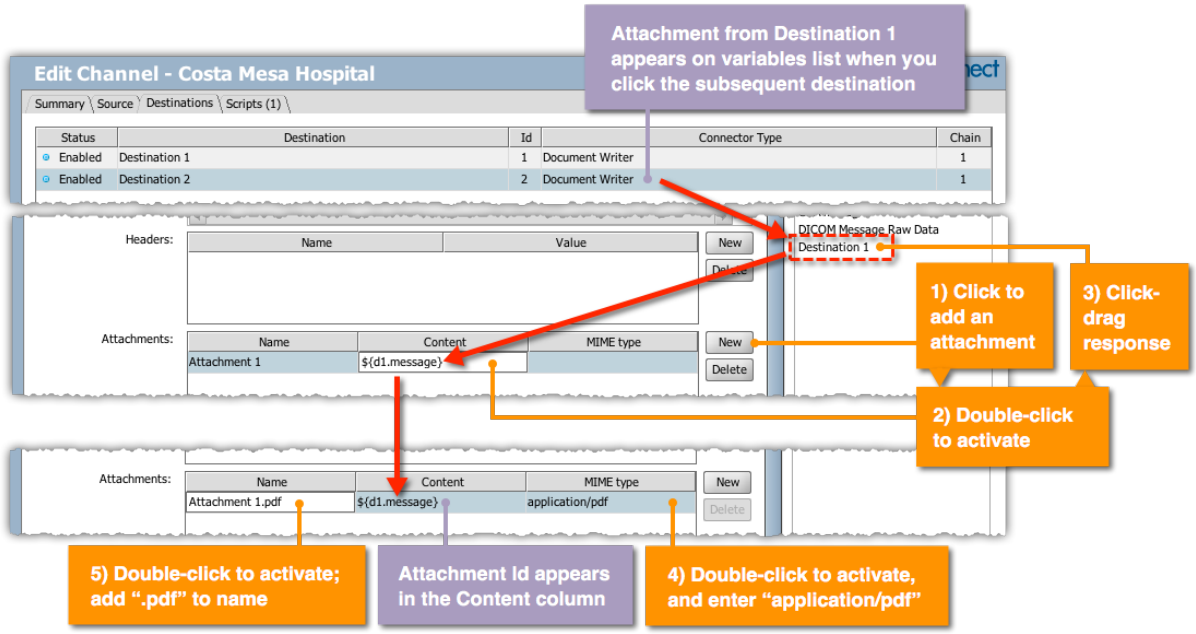
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

About the Document Writer Connector

This connector lets Mirth Connect create an RTF (rich-text format) or a PDF (portable-document format) file based on an HTML template. (PDF files can be password protected.) *Document Writer* is a destination-only connector that creates a response-map variable that shows the name of the written file.

You can output the document as an attachment on the channel. The attachment ID is returned as the response of the Document Writer destination, which you can copy to an attachment on a subsequent destination to be converted to PDF format and sent as an email attachment.

On an **Edit Channel** page – **Channel Tasks** panel, click *New Destination*, then click the **Connector Type** bar, and select *Document Writer*. In the **Document Writer Settings** section – **Output**, click *Attachment*, then *PDF*. In the **Channel Tasks** panel, click *New Destination*, then click the **Connector Type** bar, and select, for instance, *SMTP Sender*. The destination's response appears on the list of variables on the right-hand side of the page.



Next to the **Attachments** table at the bottom of the page (previous graphic), click the **New** button to display a new attachment in the table. Double-click the Name row, and enter a name, being sure to conclude the name with ".pdf." Double-click the attachment's Content column to activate the box, then click-drag the DW's response/attachment ID from the variables list into the box. (A green circle with a "+" appears when the pointer is over the box.) Release the pointer to reveal the attachment ID in the attachment row. Double-click the MIME Type row, and enter "application/pdf." Now you can send the generated PDF as an email attachment.

About Email Connectors

Email Reader

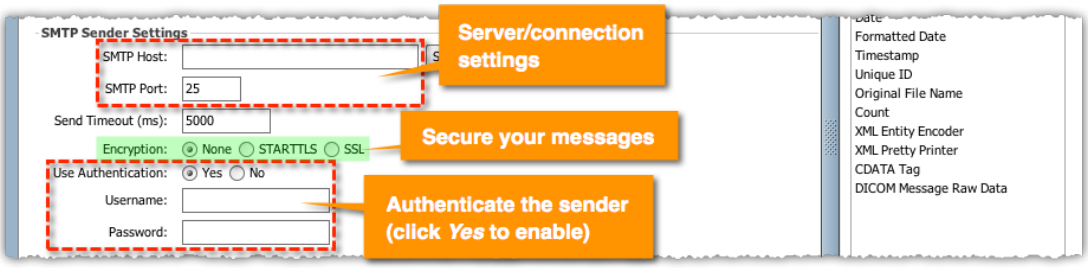
Mirth Connect supports reading emails as incoming messages via this connector. (For details, see About the Email Reader Plugin in the *Commercial Extensions* section.)

SMTP Sender

This connector lets MC send email messages via SMTP (simple mail-transfer protocol); a template is available to dynamically create messages. *SMTP Sender* is a destination-only connector. It requires an SMTP server and connection settings that you supply in the **SMTP Host** and **Port** fields (**Edit Channel** page > **Destinations** tab >

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Connector Type bar > **SMTP Sender** – **SMTP Sender Settings** section).



SMTP Sender lets you secure the connection to send files privately via **TLS/STARTTLS** (transport layer security) or **SSL** (secure sockets layer) and authenticate the sender. (See the previous graphic; click **Yes** in **Use Authentication** to enable the **Username** and **Password** fields.)

SMTP Sender sends to one email address, the body of which can be HTML or plain text.

About DICOM Connectors

Digital Imaging and Communications in Medicine connectors consist of a *DICOM Listener* (source) and *DICOM Sender* (destination). DICOM connectors receive/send DICOM messages over the DICOM protocol, all attributes of which are supported by MC. DICOM header information is available in the Inbound/Outbound Message Templates as XML and is automatically generated when a DICOM message is opened as a message template.

Attaching an Image to a DICOM Message

1. On the **Dashboard** page, select a DICOM channel.

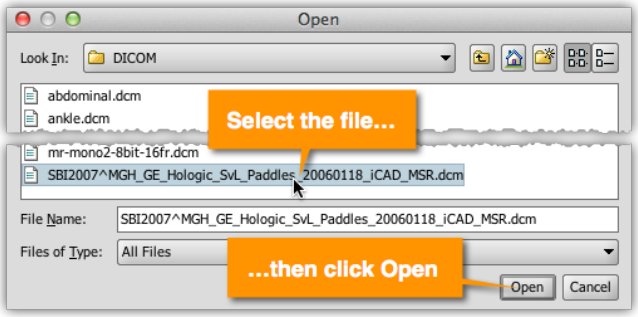
> To be a DICOM channel, the channel's inbound/outbound properties have to be set to DICOM data types.

2. In the **Dashboard Tasks** panel, click **View Messages** (previous graphic).
3. In the **Message Tasks** panel, click **Send Message**.

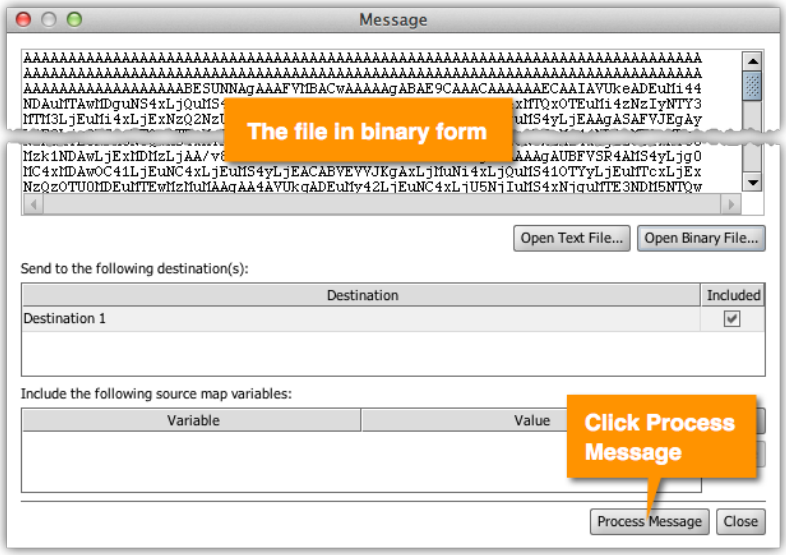
4. In the **Message** dialog, click the **Open Binary File…** button.

5. In the **Open** dialog, navigate to a folder that contains a DICOM-image file, select the file, and click the **Open** button.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

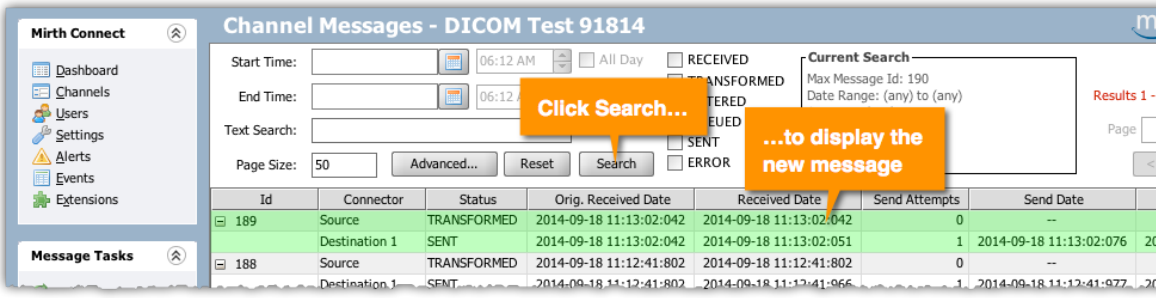


6. (The file appears in the Message window in binary form). Click the **Process Message** button.



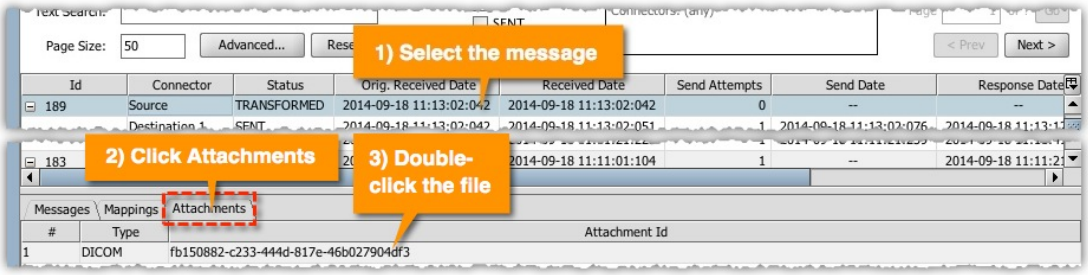
> The **Message** dialog closes, but the new message does not appear on the Message List. Go to the next step.

7. Click the **Search** button to display the new message on the Message List.



8. Select the new message, and click the **Attachments** tab that appears in the lower portion of the page.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



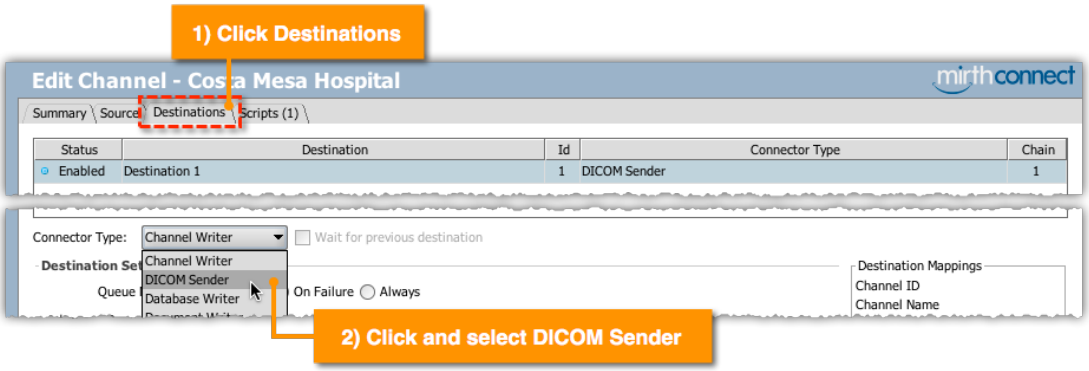
If you did not choose a DICOM message, the **Attachments** tab will not appear.

9. (Note that the message has one attachment) Double-click the attachment (previous graphic) to view the image.

Writing DICOM Data

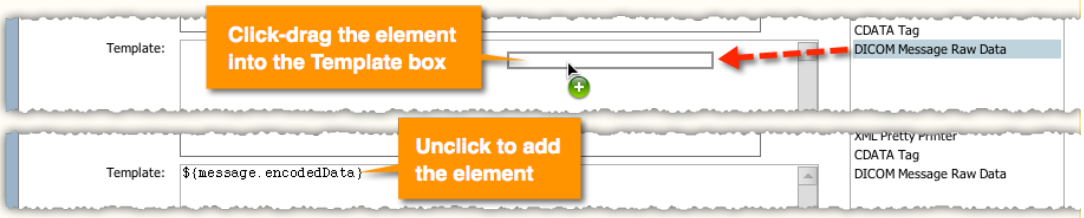
The raw DICOM message is a Base64 encoding of the binary image data. To write DICOM data, use the DICOM-message raw-data mapping in the **Template** field.

1. Navigate to an **Edit Channel** page, and click the **Destinations** tab > **Connector Type** bar > *DICOM Sender*.



Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

The **Destination Mapping** element, *DICOM Message Raw Data*, appears in the **Template** text box. (If not, go to the **Destination Mappings** window, and click/drag *DICOM Message Raw Data* to the **Template** text box. When the element is inside the text box, a green circle with a "**+**" inside appears. Release the mouse button to reveal the destination mapping element in the text box.)



2. In the **Channel Tasks** panel, click **Save Changes**.

About Serial Connectors

This is a commercial plugin available via subscription from Mirth Corporation. For more information about serial connectors, see About the Serial Connector Plugin in the Commercial Extensions section of this guide.

Messaging Tools

HL7 Inspector

This useful, open-source tool lets you create sending/receiving points in front of and behind Mirth Connect so you can essentially test inbound/outbound messages.

> You can download HL7 Inspector via http://sourceforge.net/projects/hl7inspector/.

OsiriX

The OsiriX open-source DICOM viewer is useful in testing DICOM channels in MC.

> You can download OsiriX (OS X only) via http://www.osirix-viewer.com/Downloads.html.

Mirth Corporation Tutorials

Mirth Corporation has produced a suite of tutorial videos for your convenience.

> You can view the Mirth Connect tutorial videos via http://www.mirthcorp.com/services/support/online-tutorials.

Advanced Messaging Topics

Click a link to learn about an advance messaging topic:

- About Message Attachments

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

- About Advanced Message Routing.

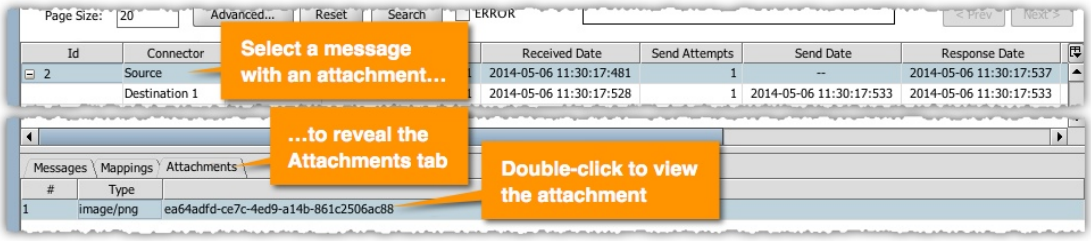
About Message Attachments

This section provides a general description of message attachments and the attachment handlers available in Mirth Connect.

Attachments in General

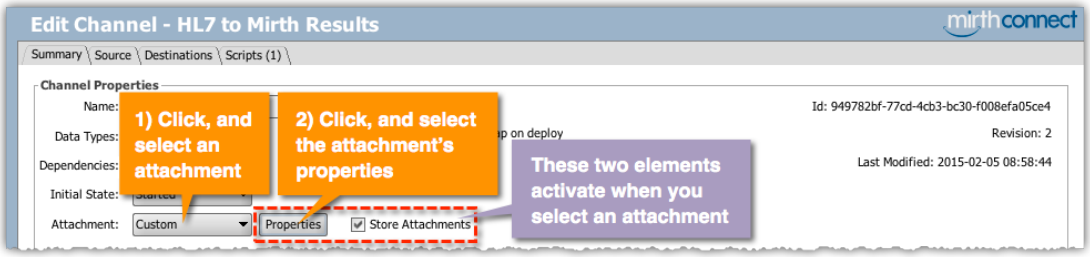
You can temporarily remove Base64-encoded data from messages and “attach” it to message data (e.g., documents/images from HL7 messages, image data from DICOM files) in the internal Mirth Connect database, primarily to trim large messages for faster processing.

If a selected message has an attachment, the **Attachments** tab appears on the lower portion of the **Channel Messages** page. Double-click the attachment to view its image.



> Until you select a message that has an attachment, or if you select a message that has no attachment, the **Attachments** tab does not appear.
>
> MC comes with built-in viewers for PDF, RTF, and DICOM images. With DICOM messages, the image is automatically attached to message data.

Attachments are extracted from messages before the preprocessor step.



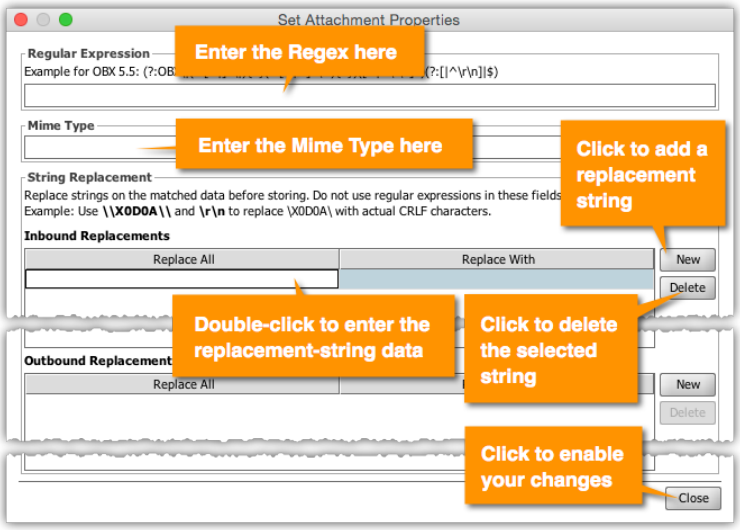
About Attachment Handlers

MC supports four attachment handlers: *Regex*, *DICOM*, *JavaScript*, and *Custom*. This section consists of descriptions of each handler, including the dialog that appears when you click the **Properties** button (previous graphic) on an **Edit Channel** page.

Regex Attachment Handler

This handler lets you use regular expressions to extract attachment data. Click the *Regex* handler's **Properties** button to open the **Set Attachment Properties** dialog.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



In the **Regular Expression** field, enter the regex that will be used to match and extract attachments. (If capturing groups are used, only the last group is extracted). In the **Mime Type** field, enter the MIME (multipurpose internet mail extension) type of the extracted data. In the **String Replacement** section, click the **New** button next to the **Inbound**/**Outbound Replacements** windows, and double-click the blue bar in the **Replace All** column, then enter the string that will be replaced. Next, double-click the blue bar in the **Replace With** column, and enter the string that will replace all instances of the string in the **Replace All** column. The **Delete** button lets you remove replacement strings from the list (but is inactive if replacement strings are absent). Click the **Close** button to enable your changes.

DICOM Attachment Handler

This handler extracts image data from DICOM messages and stores them as attachments. The DICOM handler does not have properties. Use this handler if you are using a DICOM data type.

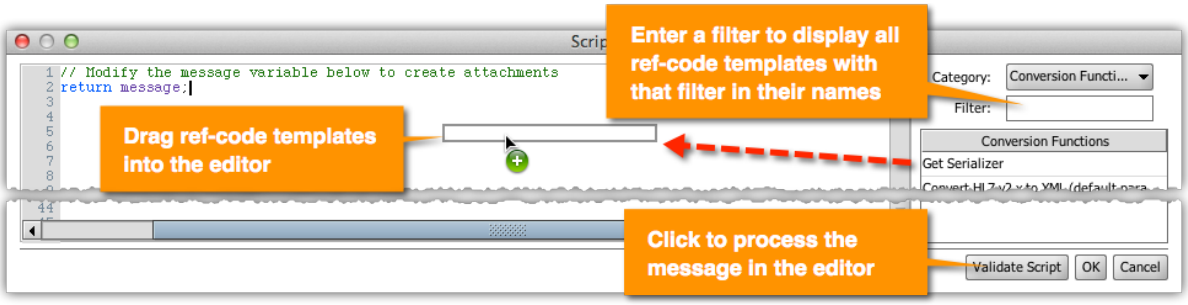
> If you change your data type to DICOM via an **Edit Channel** page > **Set Data Types** bar, the **Attachment** bar selection automatically changes to DICOM.

JavaScript Attachment Handler

To add an attachment to any message via this handler, call the **addAttachment()** function, which takes Base64 and MIME type as parameters and returns an internal attachment-class representation. To return an attachment to a message, call the **getAttachments()** function, which returns an **ArrayList** of attachment objects. The **getContent()** method returns the attachment's binary data.

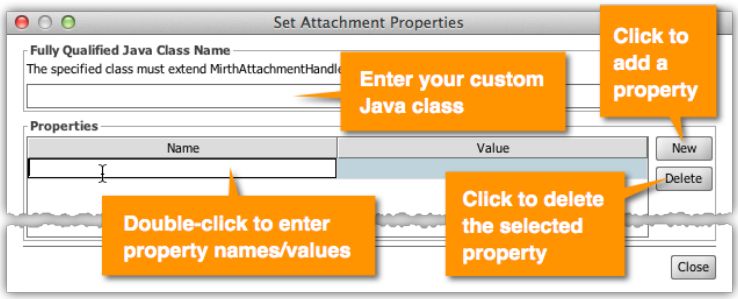
To configure your scripts, click the **Category** bar, and select a reference-code-template category. (To speed the search for the desired template(s), enter a filter in the **Filter** field to return all ref-code templates that match your filter.) Once the desired ref-code templates appear on the list, drag the desired template(s) into the Editor, and click the **Validate Script** button. (An **Information** dialog appears, informing you that the validation was successful.) Click the *Javascript* handler's **Properties** button to open the **Script** dialog.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Custom Attachment Handler

This handler lets you create your own attachment handler. First, create a Java class that extends the Mirth attachment handler (com.mirth.connect.server.util.MirthAttachmentHandler), then package that class into a .jar file, and create a resource (see About the Resources Page) containing your .jar file. Restart the MC server, and on the channel's edit page – **Summary** page – **Channel Properties** section, click the **Attachment** bar, and select **Custom**. Click the **Properties** button, and on the **Set Attachment Properties** dialog, enter your custom Java class in the **Fully Qualified Java Class Name** field, then add properties as needed.



About Advanced Message Routing

This section explains some advanced ways to handle messages, and how/where you can route them.

Common Routing Patterns

You can use several routing modes in Mirth Connect:

- **Broadcast** – An incoming message is filtered or transformed in the source connector, then routed to any number of destinations.
- **Router** – An incoming message is filtered and transformed at the destinations.
- **Application Integration** – An incoming message is routed to an application, and a response is sent back.
- **Chaining** – One (or more) destination connector routes messages to another channel that has its own filter and transformer.

Internal Routing

This refers to message routing between channels. Messages can be routed internally via the Channel Reader/Writer or the VMRouter. A Channel *Reader* lets a channel listen only for new messages from other channels and can

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

NEXTGEN HEALTHCARE

respond as can any other channel. A Channel *Writer* lets messages pass between channels. You can access the VMRouter from any JavaScript context in MC and can route data to any channel:

```
router.routeMessage (channelName, rawMessage);

router.routeMessage (channelName, message);

router.routeMessageByChannelId (channelId, rawMessage);

router.routeMessageByChannelId (channelId, message);
```

The **router.routeMessage()** method returns a Response object. Message data must be in the format expected by the source connector of the channel to which you are sending it.

> For details about the above four methods, click: http://javadocs.mirthcorp.com/connect/3.1.0/user-api/com/mirth/connect/server/userutil/VMRouter.html.
>
> For either the Channel Writer or VMRouter, the destination channel does not need a Channel Reader. You can route to any channel, regardless of its source-connector type.

Dynamic Configuration

Many channel properties can be determined dynamically on a per-message basis. Channel-map variables can be derived from message values and used as properties on destination connectors. You can route with a single connector and create global variables in Deploy Script for use in source/destination connectors.

To add a placeholder such as a Velocity template variable, use **${ }** for the value: ${ipaddress}.

Mirth Connect and JavaScript

JavaScript is an object-oriented, case-sensitive scripting language that is dynamic, easy to use, and shares many Java properties including naming conventions and syntax. Click a link to learn about these JavaScript topics:

- About JavaScript
- Using JavaScript in Mirth Connect
- JavaScript Transformers
- JavaScript Filters
- JavaScript Connectors
- Global and Channel Scripts
- About the User and Client APIs.

About JavaScript

This section consists of various aspects of JavaScript that provide a basic understanding of how the language works:

- Variables
- Comments
- Arrays
- Operators
- Conditional Statements
- Functions
- Loops and Iterations
- Exception Handling.

Variables

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Unassigned variables are undefined; string literals can use single or double quotes; braces { } create a block of statements that can be used for loops, conditionals, and statements. These are some examples of variable declarations:

```
var x;

var y = r;

z = "abc"
```

[Top]

Comments

You can start a single-line comment with two forward slashes: // This is a comment – or – you can enclose bullet-line comments with /* and */ (e.g., /* This is a comment., This is another line in the same comment. */).

[Top]

Arrays

Arrays are native objects indexed with bracket notation that can contain data types. Arrays can be initialed using Java-like constructors or bracket notation and do not need to be sized upon construction:

```
var myArray1 = new Array (10);

var myArray2 = new Array (5, "some string", new Array ());

var myArray3 = [ ];

var myArray4 = [ "one", "two", "three" ];
```

Uninitialized elements in arrays are undefined, so use the length property (var size = myArray.length;) to get a size. Use the **delete** operator to remove the index value, which sets the element as *undefined*. There are several built-in methods for arrays: concat(), reverse(), replace(), sort(), indexOf(). Arrays in JavaScript are zero-based indexed, so use myArray[0] to access the first element. Other important native objects in JavaScript include:

- String
- Date
- Boolean
- Regular Expressions
- Math
- XML.

[Top]

Operators

Mirth Connect uses four JavaScript operators:

- Arithmetic
- Assignment
- Comparison
- Logical.

Arithmetic Operators

These operators are used to perform arithmetic between variables and/or values. In this table, the **y**-variable has a

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

value of 5 to explain the JavaScript Arithmetic operators:

Operator	Description	Example	Result of x	Result of y
+	Addition	x=y+2 (7=5+2)	7	5
-	Subtraction	x=y-2 (3=5-2)	3	5
*	Multiplication	x=y*2 (10=5*2)	10	5
/	Division	x=y/2 (2.5=5/2)	2.5	5
%	Modulus (division remainder)	x=y%2 (remainder is 1 for the equation 5/2)	1	5
++	Increment	x=++y (6=++6)	6 (a)	6 (b)
		x=y++ (5=6++)	5 (c)	6 (b)
--	Decrement	x=--y (4=--4)	4 (d)	4 (e)
		x=y-- (5=4--)	5 (f)	4 (e)

(a) The increment occurs *before* the variable, so **x** is the incremented value of **y**, which is 6; (b) The value is 6 because **y**, which is 5, is increased by 1 for this operator; (c) The increment occurs *after* the variable, so **x** is the original value of **y**, which is 5; (d) The decrement occurs *before* the variable, so **x** is the decremented value of **y**, which is 4; (e) The value is 4 because **y**, which is 5, is decreased by 1 for this operator; (f) The decrement occurs *after* the variable, so **x** is the original value of **y**, which is 5.

Assignment Operators

These operators are used to assign values to JavaScript variables. In this table, the **x**-variable has a value of 10, and the **y**-variable has a value of 5 to explain the JavaScript Assignment operators:

Operator	Description	Example	Same As	Result
=	assign	x=y	x=y (5=5)	x=5
+=	add and assign	x+=y	x=x+y (15=10+5)	x=15
-=	subtract and assign	x-=y	x=x-y (5=10-5)	x=5
*=	multiply and assign	x*=y	x=x*y (50=10*5)	x=50
/=	divide and assign	x/=y	x=x/y (2=10/5)	x=2
%=	modulus (division remainder) and assign	x%=y	x=x%y (remainder is 0 for the equation 10/2)	x=0

Comparison Operators

Comparison operators are used in logical statements to determine equality or difference between variables or values. In this table, the **x**-variable has a value of 5 to explain the JavaScript Comparison operators:

Operator	Description	Comparing	Returns

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

==	is equal to	x==8	false
		x==5	true
!=	is not equal to	x!=8	true
>	is greater than	x>8	false
<	is less than	x<8	true
>=	is greater than or equal to	x>=8	false
<=	is less than or equal to	x<=8	true

Logical Operators

Logical operators are used to determine the logic between variables or values. In this table, the **x**-variable has a value of 6, and the **y**-variable has a value of 3 to explain the JavaScript Logical operators:

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
||	or	(x==5 || y==5) is false
!	not	!(x==y) is true

[Top]

Conditional Statements

This example shows the basic syntax structure for conditional statements in JavaScript:

```
if (condition1)
{
    // Code to execute
}
else if (condition2)
{
    // Code to execute
}
else
{
    // Code to execute
}
```

[Top]

Functions

This example shows the basic syntax structure for the creation of functions in JavaScript:

```
function functionName (p1, p2, …, pN)
{
    // Code to execute
    return someValue;
}
```


Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

There are various ways to call a function [e.g., var x = myFunction ("ABC", 100, myVar);]. Parameters are optional and unlimited. The return statement is also optional.

[Top]

Iterations and Loops

An *iteration* is the execution of a process with a specific goal/result in mind. (Each repetition of the process is also an iteration). The results of one iteration are used to start the next iteration. A *loop* is a type of programming-language statement that lets code be executed repeatedly; that is, a loop is a series of iterations. In programming language, there are four types of loops:

- for
- for each…in
- while
- do…while.

Loops can be unconditionally exited with a *break* statement: **break;.** The *continue* statement: **continue;** unconditionally skips to the next iteration of the loop.

for loops

These loops are often distinguished by an explicit loop *counter* (variable), which lets the body of the ***for*** loop (the code that is being repeatedly executed) know about the sequencing of each iteration. ***for*** loops are typically used when the number of iterations is known before the loop is entered.

Syntax (*for* loops)	Example

for (index=startValue; endCondition; incIndex)
{
// Code to execute
}
	for (var i=0; i<10; i++)
	{
	logger.info ("i = " + i);
	}

for each…in loops

These loops are usually used instead of a standard ***for*** statement. Unlike other ***for*** loop constructs, however, ***for each…in*** loops usually have no explicit counter; they essentially say "do this to everything in this set" rather than "do this X times." This avoids possible off-by-one errors and makes code easier to read. These loops are used to iterate through elements in an array (or collection) or in the property values of an object.

Syntax (*for each...in* loops)	Example

for each (var in object)
{
// Code to execute
}
	var sources = new Array ();
	sources [0] = "Customer 1";
	sources [1] = "Customer 2";
	sources [2] = "Customer 3";
	for each (src in sources)
	{
	logger.info(src);
	}

while loops

These loops are control-flow statements that let code execute repeatedly based on a given Boolean (true/false)

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

condition. A ***while*** loop can be thought of as a repeating "if" statement and consists of a block of code and a condition. Upon evaluation, if the condition is *true*, the code in the block is executed, repeating until the condition becomes *false*. A ***while*** loop checks the condition *before* the block is executed, in contrast to a ***do…while*** loop, which tests the condition *after* the block is executed.

Syntax (*while* loops)	Example

while (condition)
{
// Code to execute
}
	var index = 0;
	var found = false;
	while (!found)
	{
	if (myArray[index++] == "ABC")
	{
	found = true;
	}
	}

do…while loops

These loops let code execute *once* based on a Boolean (true/false) condition. A ***do...while*** loop consists of a process symbol and a condition. The code in the block executes, and the condition is evaluated. If the condition is *true*, the code in the block is executed again, repeating until the condition becomes *false*. A ***do…while*** loop checks the condition *after* the block is executed, in contrast to the ***while*** loop, which tests the condition *before* the block is executed. It is possible—and sometimes desirable—for the condition to always evaluate as *true*, which creates an infinite loop. When such a loop is created purposely, there is usually another control structure, such as a *break* statement, that terminates the loop.

Syntax (do...*while* loops)	Example

do
{
// Code to execute
} while (conditional);
	var index = -1;
	do
	{
	var val = getValue(++index);
	} while (val != "ABC");

[Top]

Exception Handling

The variable in Exception Handling in a *catch* statement is of the ERROR type or one of its subclasses; the two properties are *name* and *message*. You can raise exceptions with a *throw* statement:

```
throw "This is my exception message!";

throw new RangeError("Var x is not between 1 and 100");
```

Uncaught exceptions inside a connector result in an error status for the processed message.

Syntax (Exception Handling)	Example

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

```
try
{
// Code to execute
}
catch (exception)
{
// Exception handling
// code to execute
}
finally
{
// Code to always execute
}
```

```
try
{
var x = undefinedVarName;
}
catch (e)
{
logger.error("Error: "+e);
}
```

[Top]

Using JavaScript in Mirth Connect

This section shows you how to use JavaScript to perform various messaging operations. Click on a link to go to the operation:

- About E4X
- Accessing Message Data with E4X
- Adding Segments to a Message
- Deleting a Segment
- Iterating Over Message Segments
- Iterating Over Repeating Fields
- Adding a New Repeating Field
- Message Variables
- Built-In Code Templates
- Using Java Classes
- Regular Expressions
- Logging with JavaScript.

About E4X

E4X (ECMAScript for XML), introduced in JavaScript 1.6, is a JavaScript extension that provides native XML support to *ECMAScript*:

```
var person1 = new XML("<person></person>");
var person2 = <person></person>;
```

E4X supplies a simpler alternative to DOM (document object model) interfaces for accessing XML documents. E4X also offers a new way to make XML visible. Prior to E4X, XML had to be accessed at an *object* level. E4X regards XML as *primitive* level, which suggests quicker access, improved support, and acknowledgment as a component (data structure) of a program. These are several useful XML object methods:

- appendChild()—appends a child element
- name()—gets the name of an element
- attribute()—gets an attribute of an element
- children()—gets a list of all of an element's child elements
- length()—gets the count of an element's child elements.

Use DOM-like syntax to access XML elements and use @ for element attributes:

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

```
<person>
    <name>
        <first/>
        <last/>
    </name>
<address type="home"/>
</person>
person.name.first = "Joe"
person['name']['first'] = "Joe"
…
person.address.@type = "work";
person['address']['@type'] = "work";
```

You can access additional E4X resources via these links:

- https://wso2.org/project/mashup/0.2/docs/e4xquickstart.html
- http://www.devx.com/webdev/Article/33393/1954?pf=true
- https://developer.mozilla.org/En/Core_JavaScript_1.5_Guide/Processing_XML_with_E4X

[Top]

Accessing Message Data with E4X

```
<?xml version="1.0" encoding="UTF-8"?>
<HL7Message>
    <MSH>
        <MSH.1>|</MSH.1>
        <MSH.2>|^~\&amp;</MSH.2>
        <MSH.3>
            <MSH.3.1>SENDAPP</MSH.3.1>
        </MSH.3>
        <MSH.4>
            <MSH.4.1>General Hospital</MSH.4.1>
        </MSH.4>
        <MSH.5>
            <MSH.5.1>RECAPP</MSH.5.1>
        </MSH.5>
        <MSH.6/>
…
    </MSH>
…
</HL7Message>
```

The XML variables ***msg*** and ***tmp*** represent root-level elements. Use JavaScript bracket notation for each element level in the document below the root (this works with any message data, not just HL7):

```
var sendingFacility =
    msg ['MSH']['MSH.4'][MSH.4.1'].toString();
```

> The transformed version of a message is always written in XML.

Most values in an HL7 message go three levels deep. The first level is a *segment;* the second level is a *field* within the segment; the third level is a *component* within a field of the segment. If a field or component does not exist, it is created automatically. Examples for repeating segments and fields include using bracket notation to index:

```
var obx5 = msg ["OBX"][4];var valueType =
```


Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

```
obx5 ["OBX.2"]["OBX.2.1"].toString();var providerId1 =
    msg["ORC"]["ORC.12"][0]["ORC.12.1"].toString();
```

[Top]

Adding Segments to a Message

To add a segment to a message, create a new XML object with a segment code, then add it after the segment it should follow:

```
var seg = new XML("<ZZZ><ZZZ.1><ZZZ.1.1>My
Value</ZZZ.1.1></ZZZ.1></ZZZ>");
msg["QRF"] += seg;
```

Insert the += operator into a message after a particular segment or at the end of the message. Mirth Connect has four global functions by which you can create segments for messages:

Function	Function Description	Example

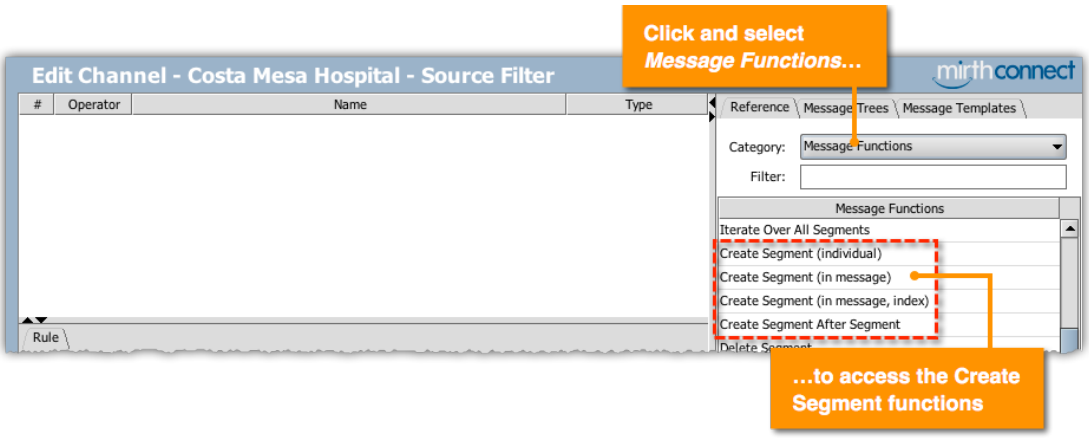
Create Segment (individual)	Creates an XML object for the segment that has not been inserted into a message	var newSeg = createSegment('ZYX');
		newSeg ['ZYX']['ZYX.1']['ZYX.1.1'] = "My Value";
		msg += newSeg;

Create Segment (in message)	Creates an XML object for the segment at the end of a specified message	createSegment ('ZYX', msg); msg ['ZYX']['ZYX.1']['ZYX.1.1'] = "My Value";
Create Segment (in message, index)	Creates an XML object for the segment in a specified message (msg or tmp) in a specified index and is issued for repeating segments; if a segment is already in the index, the new segment overwrites it	createSegment ('OBX', msg, 4); msg ['OBX'][4]['OBX.3']['OBX.3.2'] = "Glucose";

Create Segment After Segment	Creates an XML object for the segment and adds it after the target segment	createSegmentAfter("ZZZ", msg ["QRF"]);
		msg ["ZZZ"]["ZZZ.1"]["ZZZ.1.1") = "My Value";
		which function call is equivalent to:
		msg ["QRF"] += createSegment("ZZZ");

To access these functions, navigate to an **Edit Channel** page – **Channel Tasks** panel > **Edit Filter** or **Edit Transformer**. On the **Reference** tab, click the **Category** bar > *Message Functions*, among which you will find the *Create Segment* functions.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



[Top]

Deleting a Segment

To delete a segment, use the JavaScript delete keyword: delete msg ["ZZZ"];

In repeating segments, the above code deletes ***all*** instances of that segment. When you delete a segment, all subsequent segments move up.

[Top]

Iterating Over Message Segments

To iterate over all segments, follow this example:

```
for each (segment in msg.children())
{
    if (segment.name().toString() == "ORC")
    {
        // Do something…
    }
}
```

To iterate specifically named segments, use this formula:

```
for each (segment in msg..OBX)
{
    // Do something…
}
```

[Top]

Iterating Over Repeating Fields

To iterate over repeating fields, check the **Handle Repetitions** property. First, use the XML object's **length()** method

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

to get the number of repetitions to iterate over: var reps = msg ['PV1']['PV1.7'].length();

Then, use a ***for each*** loop to iterate over repetitions:

```
for each (attendingDr in msg ['PV1']['PV1.7'])
{
    lastName = attendingDr ['PV1.7.2'].toString();
}
```

[Top]

Adding a New Repeating Field

To add a new repeating field, you need to create an XML object at the segment's field level. Populate the fields, then link to the message at the repeating field level:

```
var newDr = new XML ("<PV1.7/>");
newDr ["PV1.7.1"] = "C3333";
newDr ["PV1.7.2"] = "Jones";
msg ['PV1']['PV1.7'] += newDr;
```

[Top]

Message Variables

Message variables available in JavaScript contexts include:

- **message –** a string that represents a raw inbound message in native format
- **msg –** an XML object that represents a transformed version of the inbound message
- **tmp –** an XML object that represents an outbound-message template; available only if an outbound template is defined
- **connectorMessage –** an instance of connectorMessage Java class; an internal representation of the message and its attributes in all its states.

[Top]

Built-In Code Templates

These templates are used for *function calls* and *code snipes* for common JavaScript tasks. The templates are available in all JavaScript contexts on the **Reference** tab via drag-and-drop function and can be extended with custom code templates.

[Top]

Using Java Classes

You can access any Java class in any JavaScript context:

```
var list = Packages.java.util.ArrayList;

importPackage (java.io);
```

For classes in custom or non-standard libraries, create a resource (see About the Resources Page) containing your .jar file.

[Top]

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Regular Expressions

These are used to search, match, or manipulate text strings based on patterns. In Mirth Connect, regular expressions are used for:

- String replacement in the Mapper and Message Builder transformer steps
- File-reader filename filter patterns
- Error-condition matching in Alerts
- String methods in JavaScript contexts.

Regular expressions have their own set of rules.

Character	Character Name	Usage	Example
|	Vertical bar/pipe	Separates alternatives	this|that – matches “this” or “that"
( )	Parentheses	Groups characters together	(A|a)bc – matches “Abc” or “abc”
^	Caret	Matches characters only at the start position	^ab – matches “abc” but not "lab"
$	Dollar sign	Matches characters only at the end position	Ab$ - matches “lab” but not “abc”
?	Question mark	Indicates 0 instances or 1 instance of the previous character	Ab?cd – matches “acd” or “abcd”
*	Asterisk/star	Indicates 0 instances or multiple instances of the previous character	ab*c – matches “ac,” “abc,” “abbc,” “abbbc,” etc.
+	Plus	Indicates 1 instance or multiple instances of the previous character	ab+c – matches “abc,” “abbc,” “abbbc,” etc.
[ ]	Brackets	Denotes a set of characters that match	[abc] – matches “a,” “b,” or “c” [^abc] – matches any characters except “a,” “b,” or “c” [a-d] – matches “a,” b,” “c,” or “d”

When you use regular expressions in JavaScript, define them with the regular-expression object or with this syntax: **/pattern/attributes**. Regular expression objects take two strings: *pattern* and *attributes*.

Attributes include “g” (global) and “i” (case insensitive): var exp = new RegExp ("abc", "gi");

The **test()** method matches an expression to a given string: found = exp.test ("I know my abc's");

JavaScript string object methods that use regular expressions include: match (), replace (), search (), split ()

Example:

```
var myString = "Line1\nLine2\nLine3\n";
myString.replace (/ \n/g, "\r");
```

[Top]

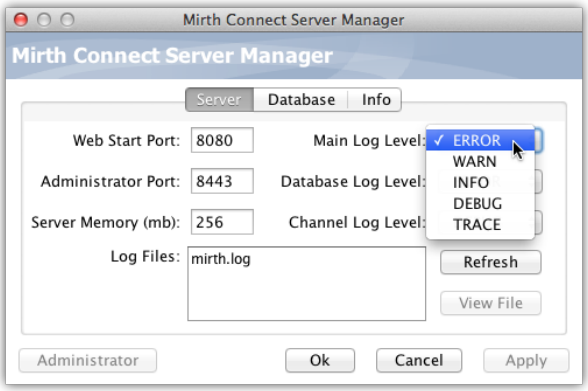
Logging with JavaScript

Use **logger.info()** and **logger.error()** to log information in JavaScript contexts: logger.info ("The value of x =" + x);

You can view the output of this method in the Mirth Connect Dashboard's Server Log. The default logging level, **ERROR**, is the quickest, having the least amount of overhead. **INFO** and **DEBUG** levels give more details but have more overhead, so are somewhat slower. You can change the logging level via the Mirth Connect **Server Manager**

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

for users who run the latest version of Mirth Connect on a Mac (**Applications** > **Mirth Connect** > **mcmanager**) or PC:



> Mirth Appliance users can change the logging level from the Mirth Connect **Settings** page via **Applications** > **Mirth Connect** > **Manage from the Appliance UI**.

[Top]

Using the JavaScript Editor

The JavaScript Editor lets you customize the appearance of script in the Editor via the Context menu and add various elements to the script via the Auto-Completion popup, both of which are described in procedures in this section. The JavaScript Editor appears on any page where JavaScript is used (e.g., **Edit Transformer** pages if JavaScript is the chosen step type, **Edit Global Scripts** page, **Edit Code Template** page).

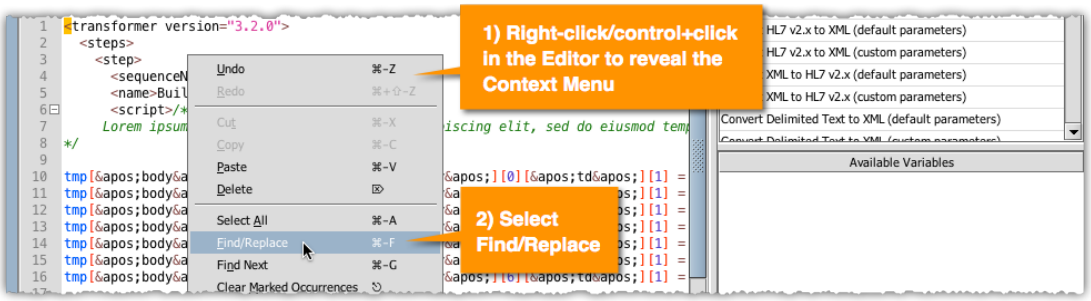
Using the Context Menu in the JavaScript Editor

This menu contains items that let you Undo/Redo actions; Select/Copy/Cut/Paste/Delete code; Find/Replace code; collapse/expand sections of code; and characterizes tabs/whitespace/line endings.

Finding/Replacing Code in the JavaScript Editor

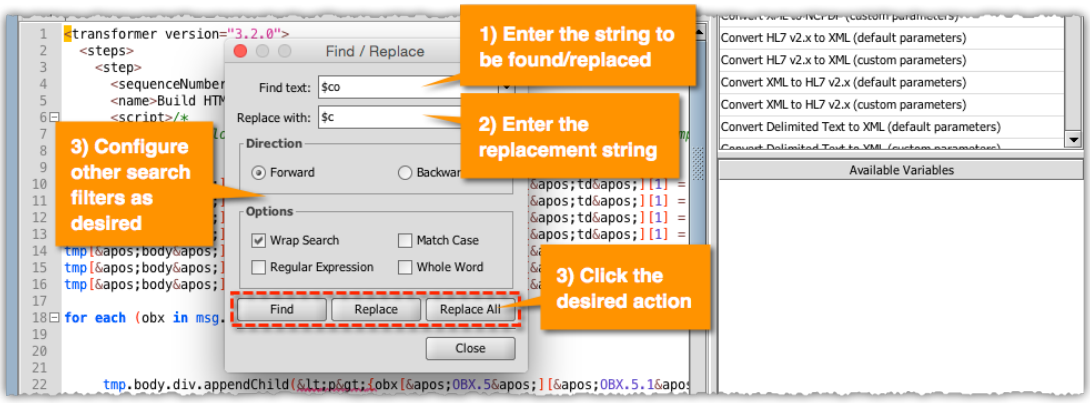
You can use the Context menu to find code, to find and replace certain of the found code, or to find and replace all of the found code.

1. Navigate to the page that has the desired JavaScript Editor, and right-click/control+click in the Editor.



2. On the Context menu (previous graphic), select *Find/Replace*.
3. On the **Find/Replace** dialog > **Find text** field, enter the code string you want to replace (in this case, **$co**).

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



If you are merely searching for all incidents of "$co" in the code, click the **Find/Replace** dialog's **Find** button > **Close** button. In the JS Editor, all incidents of the entered string are highlighted.

4. In the **Replace with** field (previous graphic), enter your replacement string (in this case, **$c**).
5. (Previous graphic) Configure other search filters as desired, and, depending on the situation, click the **Replace** button (to replace only the first incident of the string) or the **Replace All** button (to replace all incidents of the string).

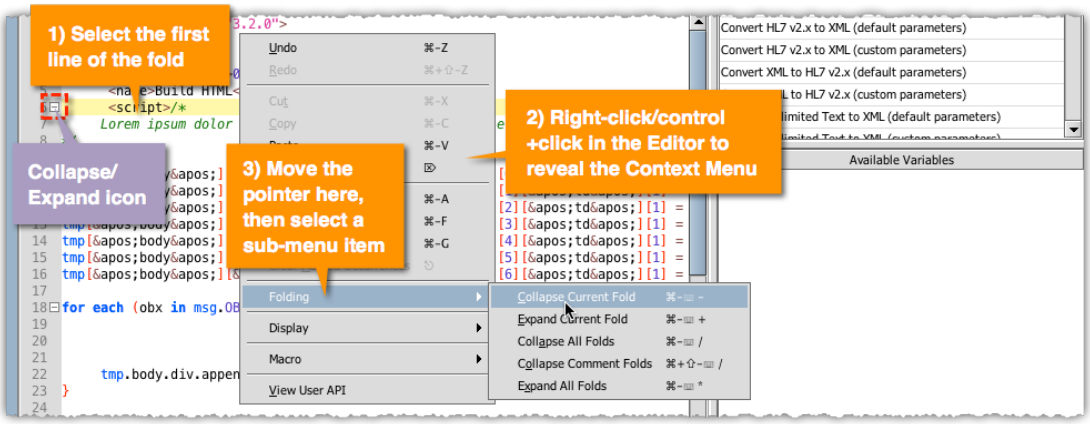
MC performs the selected action and highlights the strings in the Editor.

6. Click the **Close** button.

Folding in the JavaScript Editor

Folding, which refers to collapsing/expanding portions of code in the JavaScript Editor, can be accomplished manually by clicking the **+/-** icon in the left margin of the JS Editor or via the Editor's Context menu (which offers more ways to collapse/expand folds faster than you can do manually). This procedure instructs you how to collapse/expand a fold via the Context menu. (The procedures to perform the other folding actions are similar.)

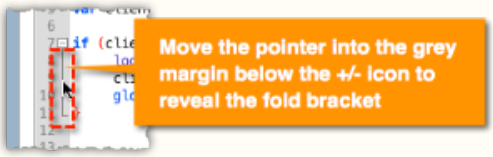
1. Navigate to the page that has the desired JavaScript Editor, and click the first line of the desired fold.



Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

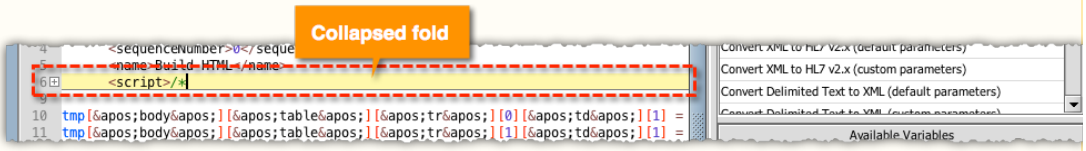
You need to select the first line in the desired fold. which lines are distinguished by a **+/-** icon in the grey margin of the JS Editor (previous graphic). If you do not select one of these lines, the Editor does not collapse/expand the fold.

You can see how many lines of code are in a fold by moving the pointer into the grey margin below a **+/-** icon, which action reveals a bracket that extends from the icon down to the last line of the fold.



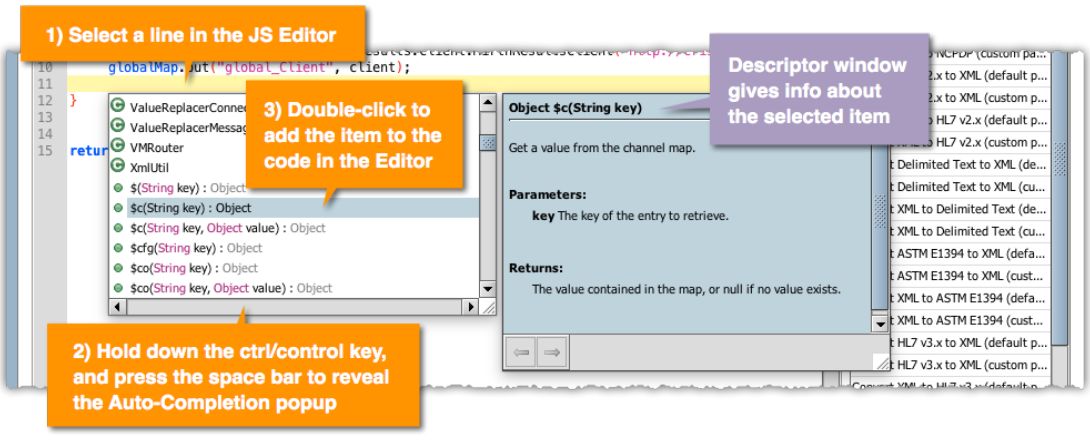
2. Right-click/control+click in the Editor (previous graphic).
3. On the Context Menu (previous graphic), move the pointer over *Folding*, and select an option on the sub-menu (in this case, *Collapse Current Fold*).

The *Collapse Current Fold* action is performed.



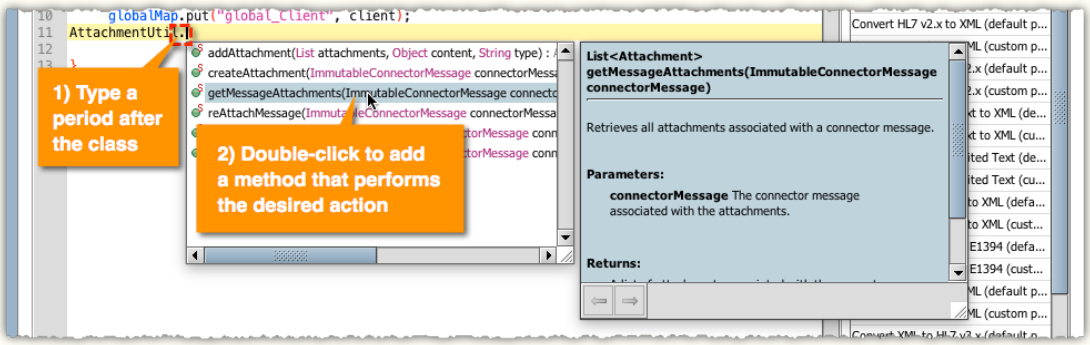
Using the Auto-Completion Popup in the JavaScript Editor

The Auto-Completion popup simplifies channel coding, eliminating the need to go back and forth between the JS Editor and the User API or even the Message Template list on the same page as the Editor by putting all coding variables (e.g., code templates, classes, String keys, createSegments, variable-map functions) in one popup within the Editor itself. Accompanying the AC popup is a descriptor window that displays, depending on the selected item, its name and various traits. To display the AC popup, click on the beginning/end of a line in the JavaScript Editor, then hold down the ctrl/control button, and press the space bar. On the list, double-click the desired item to add it to the code in the Editor.



When adding classes in the JavaScript Editor, double-click the desired class on the Auto-Completion popup, and type a period after the class to reveal a list of its methods (that perform specific actions), then double-click the desired method to add it to the Editor.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



JavaScript Transformers

JavaScript transformer steps let you execute any JavaScript code. (An editor for writing your code is available.) There are two types of JavaScript steps:

- **JavaScript**—JavaScript code contained in Mirth Connect's JavaScript transformer context
- **External Script**—code contained in an external file.

As you write your code, if you want your JavaScript to access a message field, drag that field from the message tree, and drop it into the JavaScript editor. In MC, you can access maps via JavaScript:

- connectorMap.put (key, value);
- value = connectorMap.get (key);
- channelMap.put (key, value);
- value = channelMap.put (key);
- globalChannelMap.put (key, value);
- value = globalChannelMap.get (key);
- globalMap.put (key, value);
- value = globalMap.get (key);

There is a special-case use for response maps:

```
responseMap.put (connector,
    ResponseFactory.getSuccessResponse (message));
responseMap.put (connector
    ResponseFactory.getFailureResponse (message));
Value = responseMap.get (connector);
```

To look up a value in all maps: Value = $ ('key');

You can also access maps via convenient shortcuts:

Map	Get	Put
Configuration	$cfg('key')	—
Global	$g('key')	$g('key', value)
Global Channel	$gc('key')	$gc('key', value)

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Source	$s('key')	—
Channel	$c('key')	$c('key', value)
Connector	$co('key')	$co('key', value)
Response	$r('key')	$r('key', value)

JavaScript Filters

JavaScript filter logic can be much more complex than what the rule-builder steps support, which is why this guide removes JavaScript filters from the standard Filters section. JavaScript filter rules let you execute any JavaScript code to evaluate data. You can filter fields in repeating or optional segments, and you can write a JavaScript filter to perform database lookups, table lookups, etc.

There are two types of JavaScript steps: *JavaScript* (code within the filter) and *External* script (code in an external file).

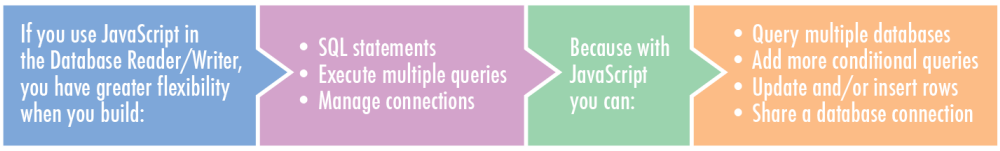
If you want your JavaScript to access a specific message field, drag that field from the message tree, and drop it in the JavaScript editor. The expression returns *true* if the message should be processed or *false* if it should be rejected.

> If there is no return statement in your code, the default return value is *false*.

JavaScript Connectors

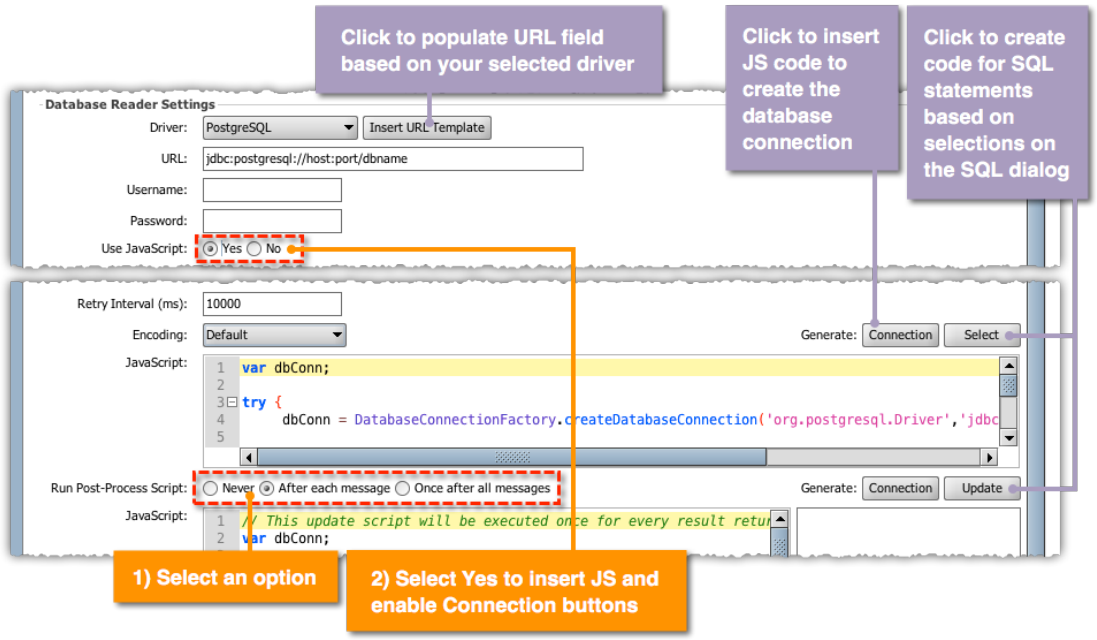
Two types of connectors use JavaScript: the *Database Reader/Writer* and the *JavaScript Reader/Writer*.

Database Reader/Writer



On an **Edit Channel** – **Source** page – **Database Reader Settings** section, the **Connection** button inserts JavaScript code to create the database connection. The **Select/Update/Insert** buttons generate code for SQL statements based on selections in the **SQL Creation** dialog. Enable the **Run Post-Process Script** function by selecting **After each message** or **Once after all messages** and on the **Use JavaScript** option, selecting **Yes**.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



The **dbConn** object returned by **calltocreateDatabaseConnection()** is a custom MC Database Connection class that encapsulates **java.sql.Connection**, **java.sql.Statement**, etc.

Simple methods to execute queries include **executeUpdate()** and **executeCachedQuery()**, which returns **javax.sql.rowset.CachedRowSet** (a subset of **java.sql.ResultSet**).

Useful **ResultSet** methods include:

- ResultSet.next()
- getDouble()
- getInt()
- getObject()
- getString()
- getTimestamp().

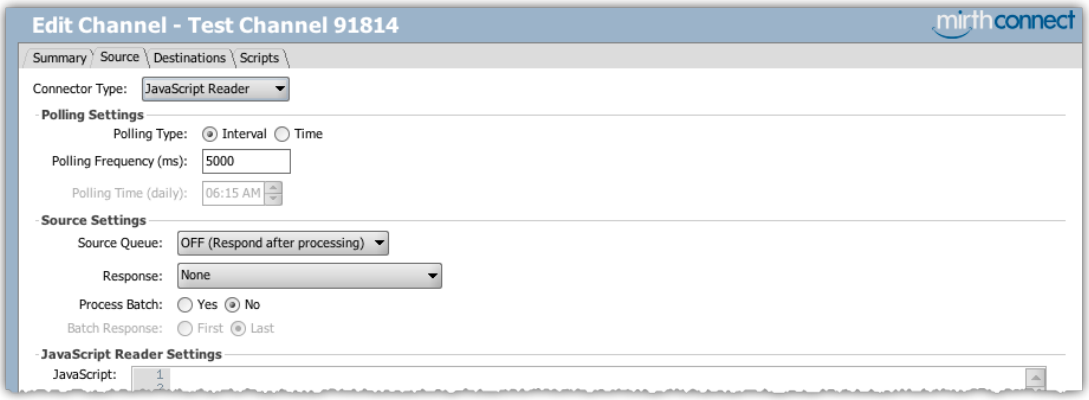
executeUpdate() returns a count of rows updated. Inserts/updates can be parameterized. In the Database Reader, the ResultSet returned by a query becomes a message for each row in the set. You can also return any object that implements the **java.util.List** interface (e.g., **ArrayList**, **LinkedList**). Each map in this list becomes a message. In this case, each key/value pair is similar to a column/value pair in a result set.

The Channel Map is not available in the JavaScript context of the Database Reader, but the Global Channel Map is available in both JavaScript contexts of the Database Reader; the Response Map is available in the On-Update statement code.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

JavaScript Reader/Writer

The JavaScript Reader/Writer executes an arbitrary block of JavaScript to process and send messages, and can be used to interface with system-using methods not supported by installed connectors (polling an HTTP server, directly communicating with a Java application, issuing system commands, etc.).

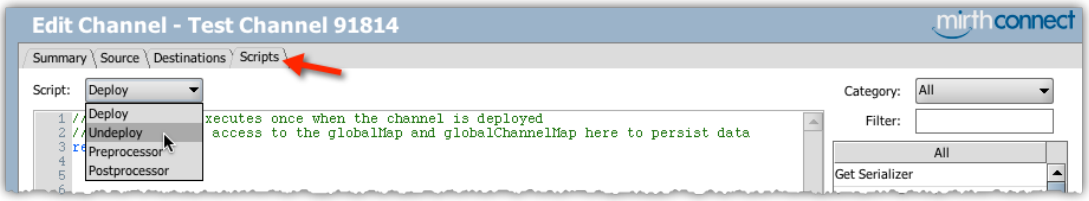


The return value for the JavaScript Reader can be a string object that represents an individual message or any object that implements the **java.util.List** interface (e.g., **ArrayList**) that contains a list of message strings. The Channel Map is not available in the JavaScript context for the JavaScript Reader but is available for the JavaScript filters and transformers. The Global Channel Map is available in the JavaScript context for the JavaScript Reader.

The JavaScript Writer creates a Response Map variable with the message: *Script execution successful*. You can use the JavaScript Reader as a background process to be run at specified intervals, for example, to periodically delete old temporary data from a database table used by another channel or to Return an empty instance of a **java.util.List** -derived object (such as an **ArrayList**) if you want the script to run but not process messages.

Global and Channel Scripts

Four scripts (*Deploy/Undeploy/Preprocessor/Postprocessor*) are available for arbitrary use at certain points as messages are processed. Each script is available at global and channel levels. Global scripts are executed for all channels; channel scripts are executed for their own channels. All are written in JavaScript. Scripts are selected on an **Edit Channel** page > **Scripts** tab > **Script** bar.



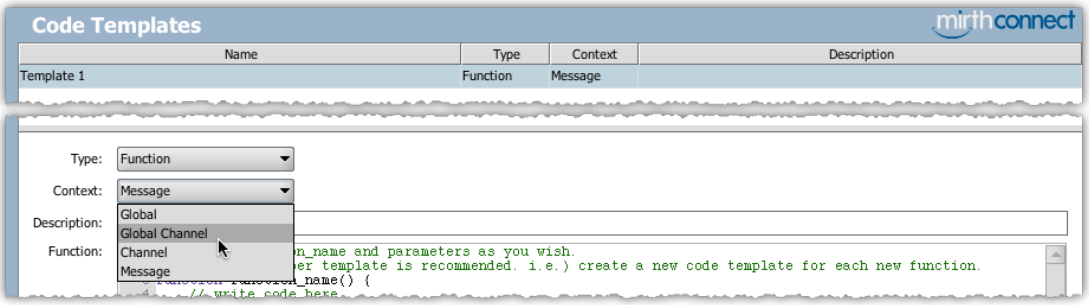
Script	Description
Deploy	Executes when channels are deployed, including when the MC service starts. The *global* deploy script executes (once per deployment) when a channel is deployed and before a *channel* deploy script, which executes when a channel is deployed from an undeployed state. Deploy scripts are generally used to perform required channel setup to define global variables, create shared connections, or load lookup data from a database or file.
Undeploy	Executes when channels are shut down (including for the MC server shutdown). The *global* undeploy script executes when a channel is undeployed and after all *channel* undeploy scripts execute. (Before a deployed channel is redeployed, it is undeployed.) Undeploy scripts are generally used to perform required cleanup (e.g., closing open connections).

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Preprocessor	Executes before the incoming message is sent to the encoder. The *global* preprocessor script executes when a channel receives a message and before the *channel* preprocessor script, which executes when the channel receives a message. Preprocessor scripts are often used to remove invalid characters, fix invalid fields, or create messages from unknown data types. For high-volume pass-through channels, making small, simple changes in the preprocessor can be much more efficient than using transformer steps. The incoming message (in its native format) is available in the JavaScript variable **message**. The return value is a string – the *fixed* version of the message – and should be in the format specified by the channel's inbound data type on the **Summary** page. The Preprocessor script returns the **message** variable by default.
Postprocessor	Executes after all destinations of a channel are processed. The *global* postprocessor script executes when *any* channel processes a message but *after* the *channel* postprocessor script, which executes when that channel processes a message. Postprocessor scripts only execute when a channel is synchronized and even when no destinations are used (e.g., if a message is filtered on its way to a destination). Postprocessor scripts are often used to handle enhanced acknowledgments and custom responses. Each destination puts its response in the response map, so you can use a postprocessor script to build custom response messages and put them in the response map. To create a custom ACK (acknowledgment) response, use the **generateAckResponse()** method of the built-in ACKGenerator Java class in MC in the com.mirth.connect.connectors.mllp. An example of a custom ACK response in action is a pass-in of the inbound message, ACK code, and text message. The return value is generated in an ACK-message string, and the ACK message is put in the response map. You would then set the LLP Listener to respond from the ACK message variable in the response map.

Code Templates

These let global functions be compiled, and commonly used code blocks and variables be stored in one location. You can add custom code templates and functions to the list of built-in code templates and filter them with user-defined functions. View the **Code Templates** page via the **Channels** page – **Channel Tasks** panel > **Edit Code Templates** – **Code Template Tasks** panel > **New Code Template**.



The code template **Context** property determines the scope of the available template or function, whether **Global**, **Global Channel**, **Channel**, or **Message**.

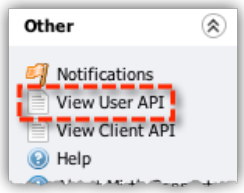
About the User and Client APIs

An API (application programming interface) specifies how certain software components should interact. Two APIs are available in Mirth Connect: one for users and the other for users' Java applications (or clients, so-called because the user-written applications use Mirth Connect's resources much like human clients use a corporation's resources).

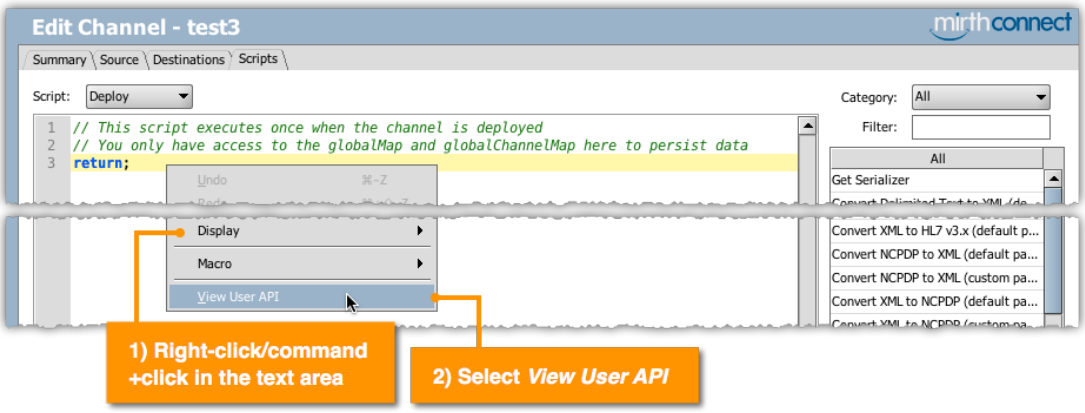
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

About the User API

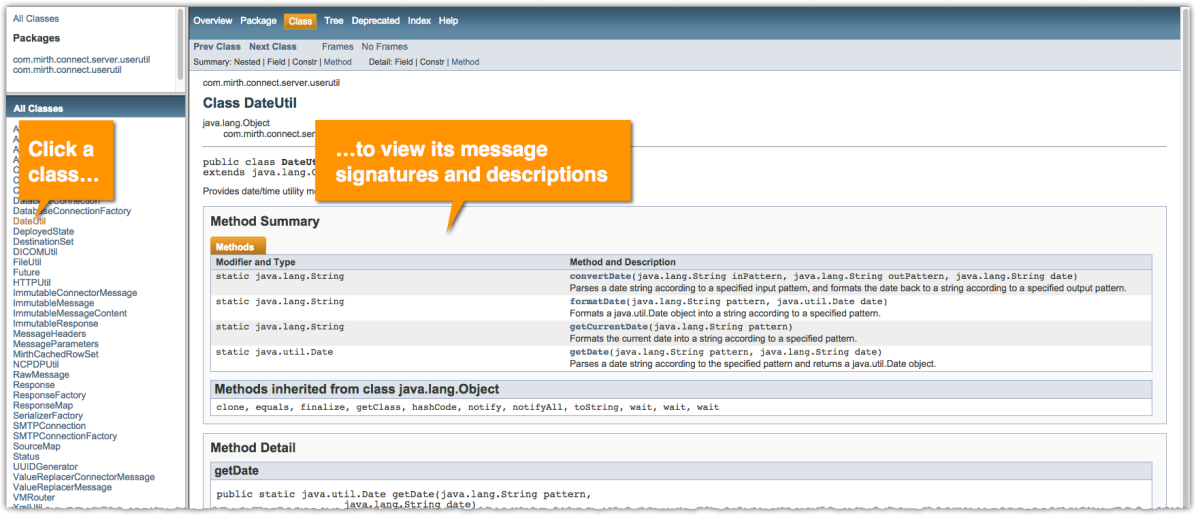
The user API (application programming interface) is a collection of Java classes and methods that helps you interact with channels and the message that is being processed. It also provides helper methods for common tasks such as date formatting. Wherever you are editing JavaScript code in Mirth Connect, you can view the API by clicking the **View User API** function in the **Other** panel, which is available on almost all Mirth Connect pages.



(Or) You can right-click/command+click in the text area of an **Edit Channel** > **Scripts** page, and on the drop-down menu, select *View User API*.



The javadoc appears, on which you can select classes to view their method signatures and descriptions.

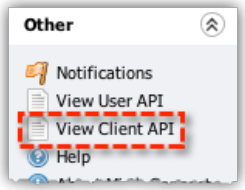


You can also view the API via this link: http://javadocs.mirthcorp.com/connect/3.1.0/user-api/com/mirth/connect/server/userutil/VMRouter.html.

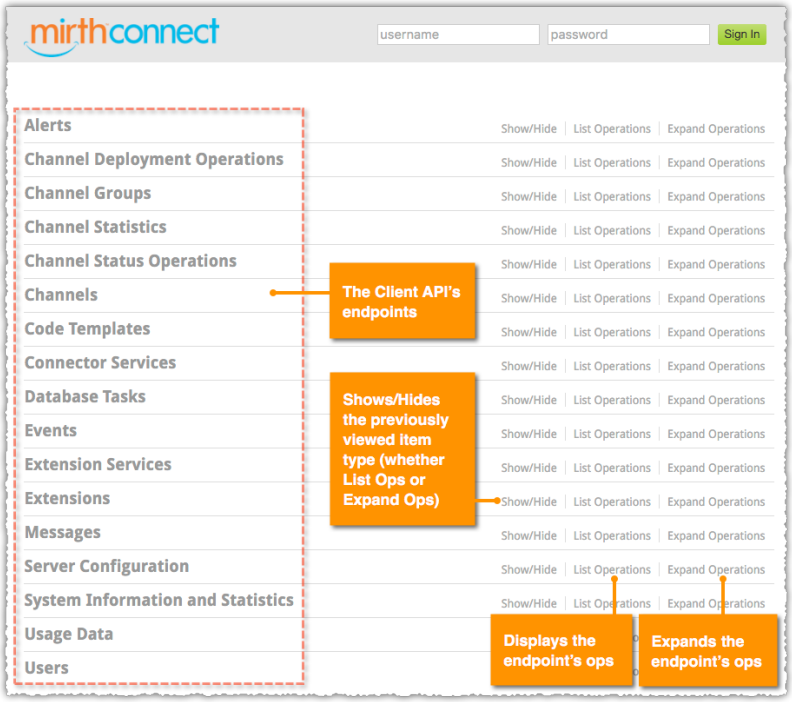
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

About the Client API

Any Java application can interface with the Mirth Connect server via the API described in the **com.mirth.connect.client.core** client class in the **mirth-client-core.jar** file. The same class is used by the Mirth Connect Administrator and CLI (command-line interface). The client class communicates with the Mirth Connect server via a **REST** architecture that is extensible to other platforms such as **.NET**.

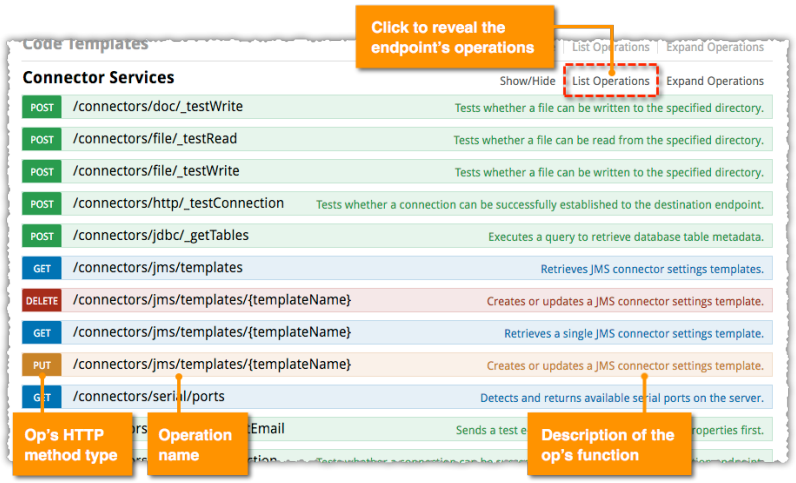


Click the **View Client API** action to open a page that contains all of Mirth Connect's client API endpoints.



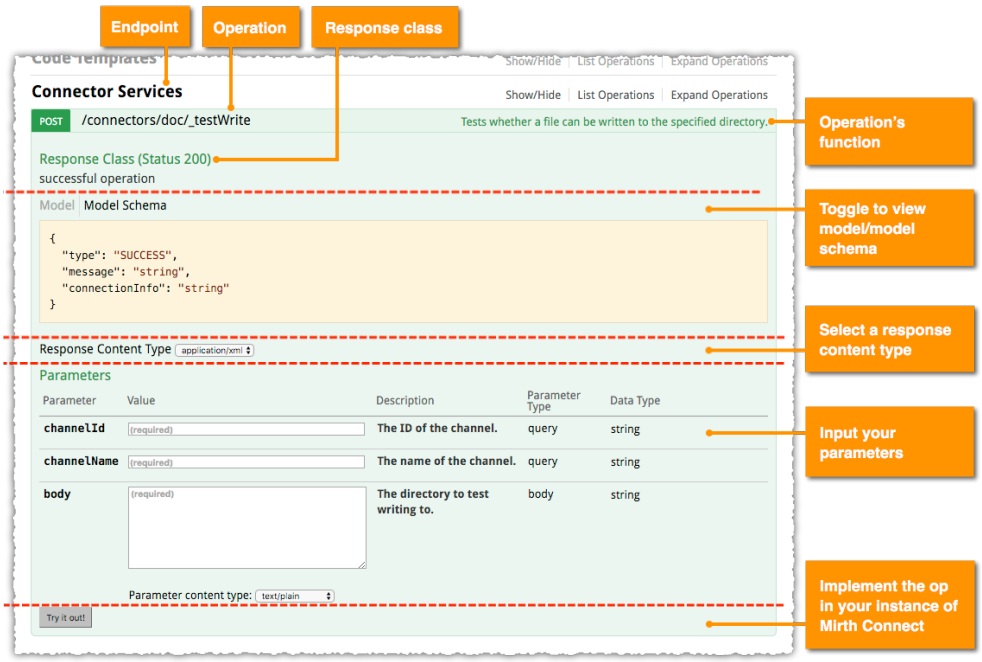
Click an endpoint's **List Operations** action to view its operations. This list shows the operation's HTTP method type, name, and description of function.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Click an endpoint's **Expand Operations** action to view the details of all its operations. (If desired, you can click a specific operation to view only that op's details.) Each operation's details include response class, model and model schema, response content type, parameters fields (channelId, channelName, body), and **Try it out!** button so you can try the op in your own instance.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Mirth Connect Operations

This section consists of various elements of MC operations. Click a link to learn about that element:

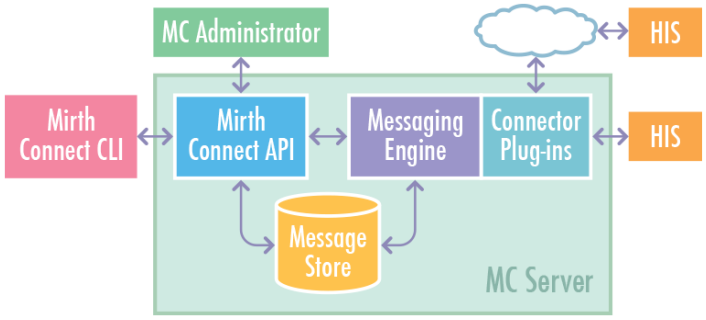
- About Mirth Connect Architecture
- About Server/Connection Logs and Global Maps
- About Settings (Server/Administrator/DataPruner)
- About Alerts
- About Events

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

- Managing Users
- About Import/Export Tasks
- Managing Extensions
- About the Command-Line Interface.

About Mirth Connect Architecture

MC is built on an internal messaging engine, which is a customized routing engine for broadcast/router modes. For strict parsing, MC uses the HAPI (HL7 application programming interface) encoder/decoder; for non-strict parsing, MC uses its own parser (for X12 and EDI connectors) and the Mozilla Rhino JavaScript Engine (for message transformation). Custom code extensions are loaded via new JAR (Java ARchive) packages.



MC stores messages, channels, users, events, configurations, etc in a back-end database. The database available with the downloadable version of MC is **Apache Derby**, which is limited, and performance suffers considerably as message storage reaches 100,000. Derby, therefore, is not recommended for production. It is included to make software testing as fast and easy as possible.

> You can configure MC to use other back-end databases, four of which are officially supported: **PostgreSQL**, **Oracle**, **MySQL**, and **SQL Server**. For production, Mirth Corporation recommends PostgreSQL – an open-source database used on all Mirth appliances – as your back-end database.

The MC server automatically creates a database schema if there is none. Before you change the database, back up the server configuration, then select the database and the set attributes via the **MC Server Manager**. You may also directly edit **mirth.properties** in the **conf** folder of your MC application.

> If you change databases, the data in the previous database remains intact.

About Server/Connection Logs and Global Maps

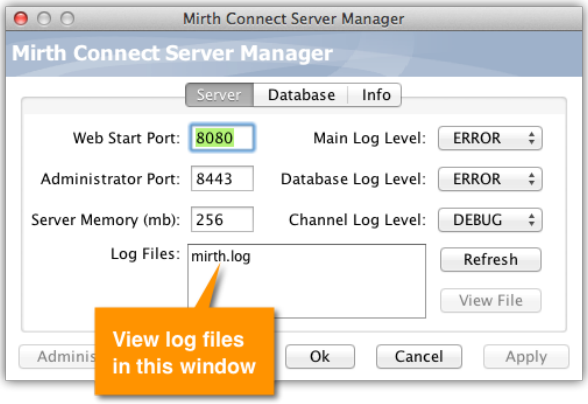
The Server Log

Information about MC is logged to the **Mirth Connect** folder > **logs** folder > **mirth.log** file. Logging is done via the **log4j** utility and is automatically rotated. You can view log files and set the logging level on the **Server** page of the **MC Server Manager**.

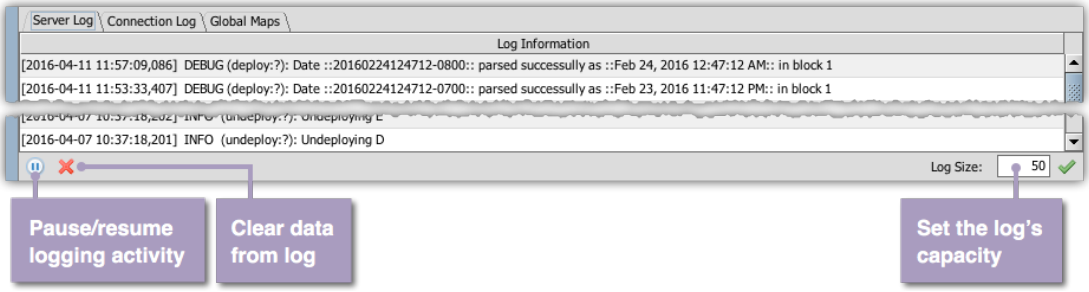
> You can access the **MC Server Manager** via: (Mac) **Finder** > **Applications** > **Mirth Connect** > **mcmanager**; (PC) **Computer** > **Local Disk** (C drive) > **Program Files** > **Mirth Connect** > **mcmanager.exe** . These are the application's default locations. If you installed the **MC Server Manager** elsewhere, follow the corresponding path to the application.
>
> The Server Manager graphic in this section is from a Mac. The Mac and PC configurations of the **MC Server**

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Manager dialog vary somewhat, but the essential elements and log procedures are the same.



Logging data (which you can pause/resume/clear) appears in the **Server** Log, which you can access via **Dashboard** – **Server Log** tab.



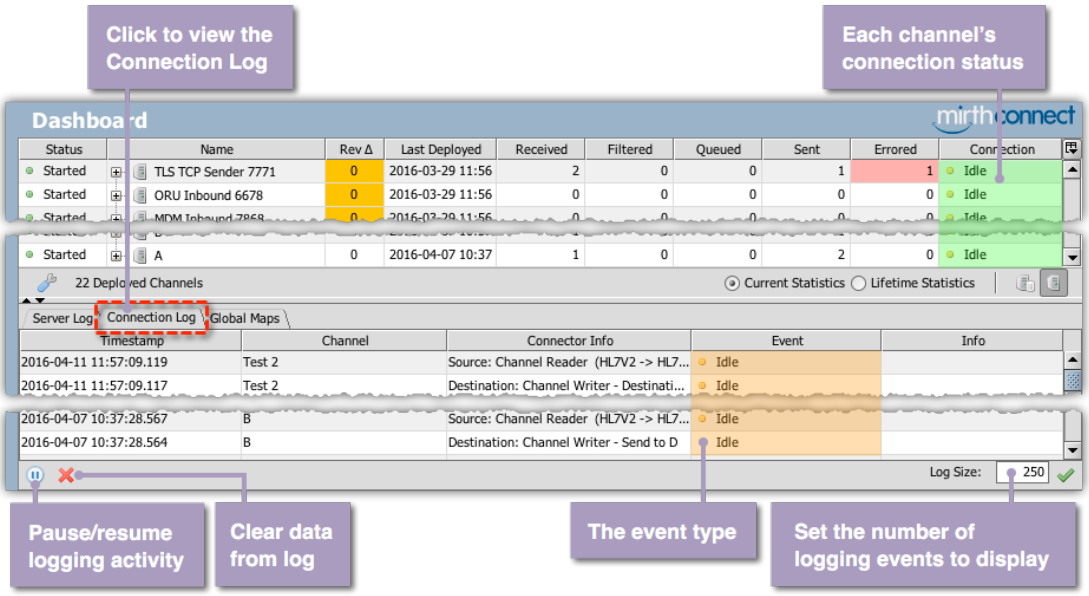
In the **Log Size** field in the bottom-right corner of the Server Log, you can specify how many (1-99) logging events appear in the log. When that amount is exceeded, the log is automatically cleared.

For greater logging control (e.g., over class-specific levels or output format), use the **log4j.properties** file in the **conf** subfolder of the **Mirth Connect** folder. (For details on how to configure **log4j**, go to: http://logging.apache.org/log4j/1.2/manual.html.) You can also use JavaScript to log messages to the **Server Log** via **logger.info()**, **logger.debug()**, **logger.warn()**, and **logger.error()**.

The Connection Log

This function displays connection statuses to other systems on the **Dashboard** page. The connection status of each channel: *Idle*, *Waiting*, *Connected*, *Receiving*, etc., appears in the **Connection** column.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

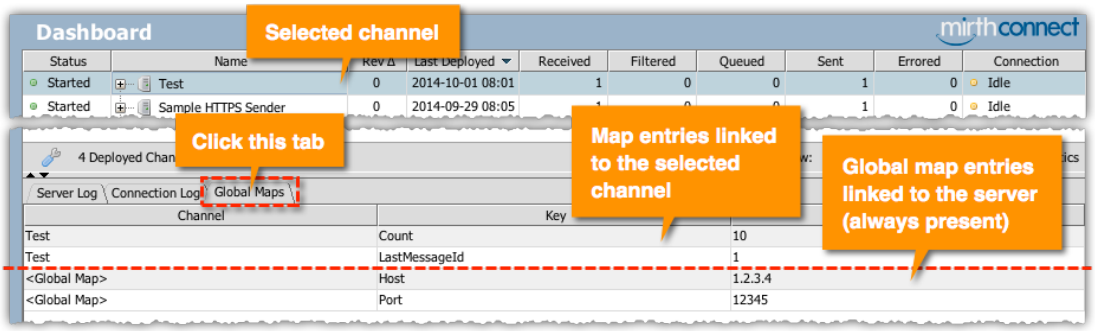


Connection data (which you can pause/resume/clear) is displayed in the **Connection Log**, which you can access via the **Dashboard** > **Connection Log** tab (previous graphic). This log shows each event (*Idle*, *Reading*, *Writing*, etc.) for each deployed channel. In the **Log Size** field (**Dashboard** page, bottom-right corner), you can specify the most events that can appear in the log before it is automatically cleared.

Global Maps

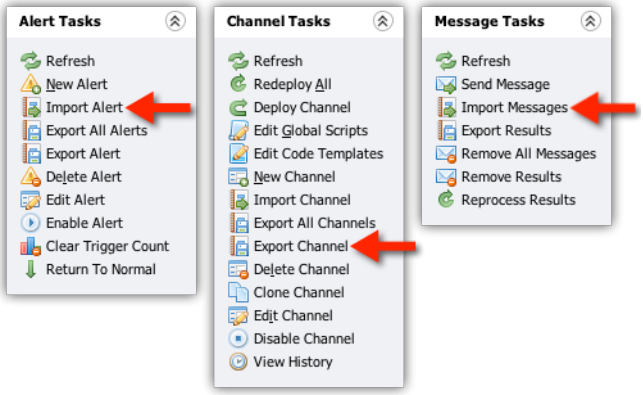
On the **Dashboard** page, you can select a channel and click the **Global Maps** tab to view the current variables in the global map(s) and any global channel maps associated with the selected channel. Because global-map values are associated with the server and not a channel, they always appear in the **Global Maps** table no matter which channel you select. Global channel maps appear only when you select the channel with which they are associated. (You can select multiple channels if desired.)

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



About Import/Export Tasks

Mirth Connect lets you import/export a number of elements, including: channels, global scripts, connectors, code templates, transformers, alerts, filters, and messages. These functions appear in a particular element's **Tasks** panel. These are some examples:



All exports are saved as XML files.

About the Command-Line Interface

An alternate CLI (command-line interface) is available in Mirth Connect. With interactive and scripted modes, the CLI lets you:

- import/export channels
- modify users
- check channel statistics and statuses
- stop/start/pause/deploy channels.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

> You cannot edit channels via the CLI.

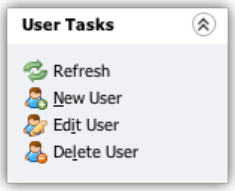
To run the CLI on Windows, execute the **Mirth Connect CLI.exe** file in the MC application folder; to run the CLI on Mac OS X/Linux, use **mirth-cli-launcher.jar** in the MC application folder.

> Set the login properties (address, username, password, etc.) for executing the CLI via the **Mirth Connect** application folder > **conf** folder > **mirth-cli-config.properties** file. If this file's credentials are incompatible with your server, CLI does not launch, and an error condition results.

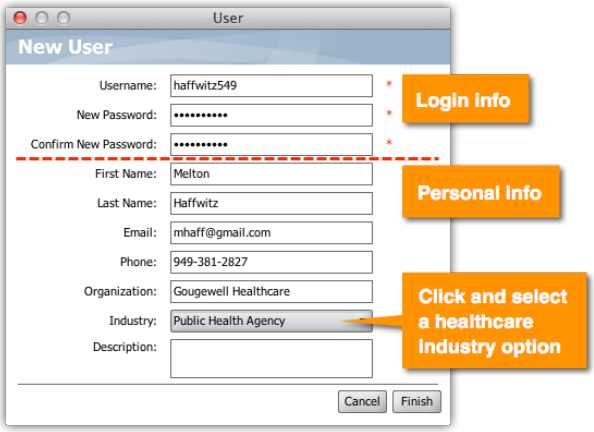
Script mode lets you execute a batch of commands and automatically disconnects when all commands are complete. To specify scripted mode and script file, add **–s *scriptfile*** to a command line such as: **"Mirth Connect CLI.exe" –s *scriptfile***, then add **script=scriptfileproperty** to the **conf/mirth-cli-config.properties** file.

Managing Users

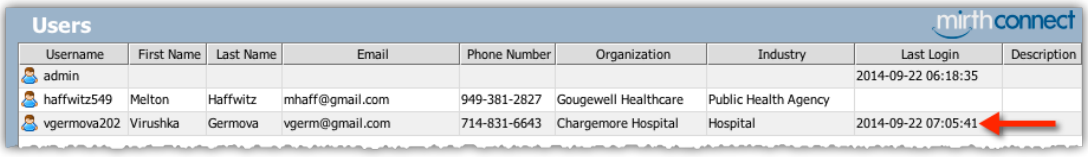
This table describes the actions available in the **User Tasks** panel.

Panel	Function	Description
	Refresh	Updates data on the User List
	New User	Add a user to the list
	Edit User	Modify information regarding the selected user
	Delete User	Remove the user from the list

MC allows unlimited user accounts. Add new users via the **Mirth Connect** panel > **Users** – **User Tasks** panel > **New User**. Complete the fields in the **User** – **New User** dialog, then click the **Finish** button.



The new user's personal information, including their most current login data, appears on the **Users** page – Users List.

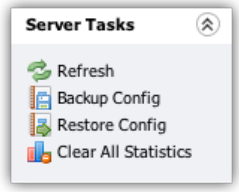


Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

To edit user information: In the **Mirth Connect** panel, click **Users**, then double-click the desired user to open the **User** – **Edit User** dialog. Change any information as needed, and click the **Finish** button.

About Settings

This table describes the actions available in the **Server Tasks** panel.

Panel	Function	Description
	Refresh	Updates data on the selected Settings page (see below)
	Backup Config	Back up your server configuration to an XML file (incl. channels, alerts, code templates, server properties, global scripts, plugin properties)
	Restore Config	Restore your server configuration from a server-configuration XML file (this action removes channels, alerts, code templates, server properties, global scripts, plugin properties)
	Clear All Statistics	Resets current and lifetime statistics for all channels

Access client settings via the **Mirth Connect** panel > **Settings** function. This section consists of descriptions of these **Settings** sub-pages:

- About the **Server** page
- About the **Administrator** page
- About the **Configuration Map** page
- About the **Database** page
- About the **Resources** page
- About the **Data Pruner** page.

Settings made on these pages are global for ***all*** users.

About the Server Page

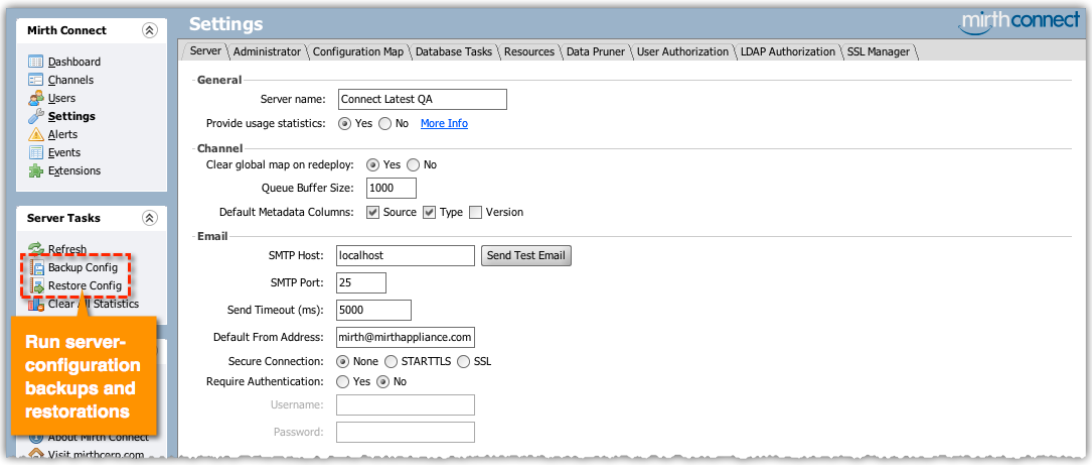
Server Page

The **SMTP Host** settings (used to relay email alerts) and **Email** sender connector settings are maintained on the **Server** page.

You can back up and restore the server configuration via the **Mirth Connect** panel > **Settings** > **Source** tab – **Server Tasks** panel > **Backup/Restore Config**. The configuration is saved to an XML file and includes channels, alerts, code templates, global scripts, and properties on the **Settings** page but does ***not*** include user accounts or messages.

When restoring a server configuration, any enabled channels in the configuration will be deployed; any channels not in the configuration will be deleted. A server-configuration backup only saves channels, alerts, and server properties; it does ***not*** back up message data.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



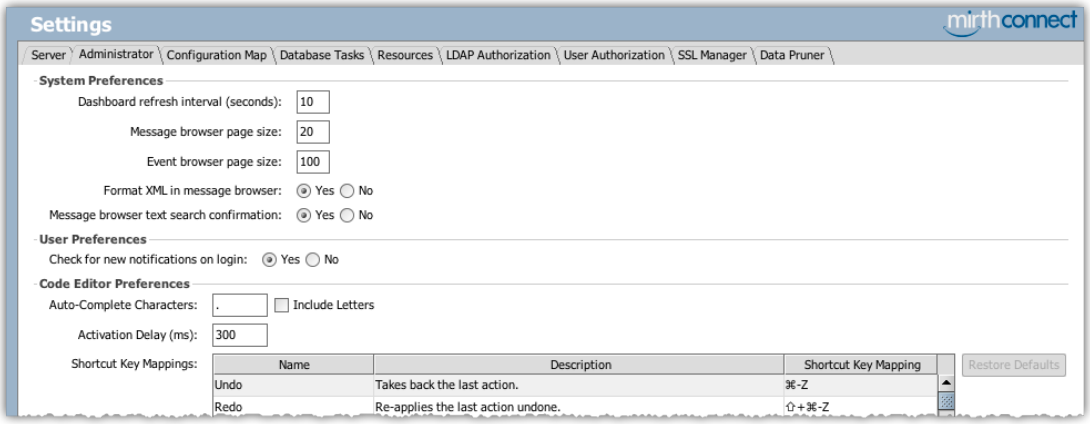
Feature	Description
General	**Server Name** – The name you give the server; appears in the title of the MC Administrator, the dock (Mac)/taskbar (PC), and the desktop shortcut; **Provide usage statistics** – *Yes*: usage statistics are sent to Mirth (which statistics do not contain any Personal Health Info and help Mirth determine which connectors or areas of Mirth Connect are most widely used); *No*: stats are not sent; More Info: Click this link to view privacy info
Channel	**Clear global map on redeploy** – *Yes*: clears the global map when redeploying all channels; *No*: the global map is not cleared when redeploying all channels; **Queue Buffer Size** – The default user-defined message capacity of the queue buffer for new channels; once this capacity is exceeded, the buffer is not cleared, but subsequent messages are stored only in the database; the buffer capacity is configurable (default capacity: 1000); you can also configure the queue buffer size via any source connector's Source Settings > Queue Buffer Size and any destination connector's Advanced Queue Settings > Queue Buffer Size; **Default Metadata Columns** – If checked, the **Source**/**Type**/**Version** metadata column is added automatically when you create a channel (you can remove the column via the channel's **Summary** tab)
Email	**SMTP Host** – The SMTP host used for global SMTP settings; **Send Test Email** – Performs a test send of an email message (e.g., to validate your connection settings); **SMTP Port** – The SMTP port used for global SMTP settings; **Send Timeout (ms)** – SMTP socket connection timeout (in milliseconds) used for global SMTP settings; **Default From Address** – The default "From" address used for global SMTP settings; **Secure Connection** – *None*: Neither a STARTTLS nor an SSL connection is used for global SMTP settings; *STARTTLS*: this connection is used; *SSL*: this connection is used; **Require Authentication** – *Yes*: activates the **Username**/**Password** fields for user authentication to use global SMTP settings; *No*: user authentication is not required (**Username** and **Password** fields are disabled)

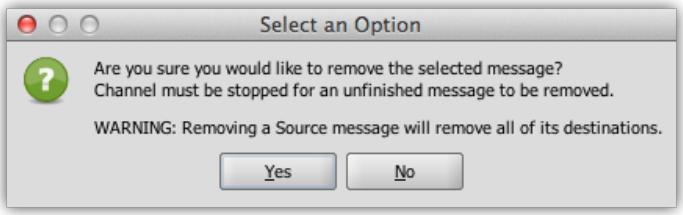
About the Administrator Page

Administrator Page

On this page, you can set various MC Administrator preferences.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Preference	Description
System Preferences	**Dashboard refresh interval (seconds)** – The timespan (in seconds) between each Dashboard refresh; decrease the amount for faster updates, and increase the amount for slower servers with more channels; **Message browser page size** – Sets the default page size (the most messages that will appear on each page) for the message browser; **Event browser page size** – Sets the default page size (the most events that will appear on each page) for the event browser; **Format XML in message browser** – (**Channel Messages** pages – **Messages** tab) If *Yes*, XML messages are broken into easily readable fragments; **Message browser text search confirmation** – If *Yes*, a confirmation dialog appears (when you click the **Search** button on a **Channel Messages** page), warning you that the search may take a long time depending on the number of messages being searched 
User Preferences	**Check for new notifications on login** – If *Yes*, checks for Mirth announcements, available updates, etc. relevant to this version of Mirth Connect whenever you log in
Code Editor Preferences	**Auto-Complete Characters** – The Auto-Completion popup is triggered when you type any of these; **Include Letters** – If checked, auto-completion is triggered when you type any letter; **Activation Delay (ms)** – How much time will pass after you type an activation character before the Auto-Completion popup menu opens; **Shortcut Key Mappings** – A table listing the names, descriptions, and keystrokes of keyboard shortcuts; to edit a shortcut: In the **Shortcut Key Mapping** column, double-click a shortcut, and replace it with your preferred keystroke shortcut; when you change a shortcut, the **Restore Defaults** button appears, which you would press to restore all shortcuts that were changed from their original keystrokes

About the Configuration Map Page

Configuration Map

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

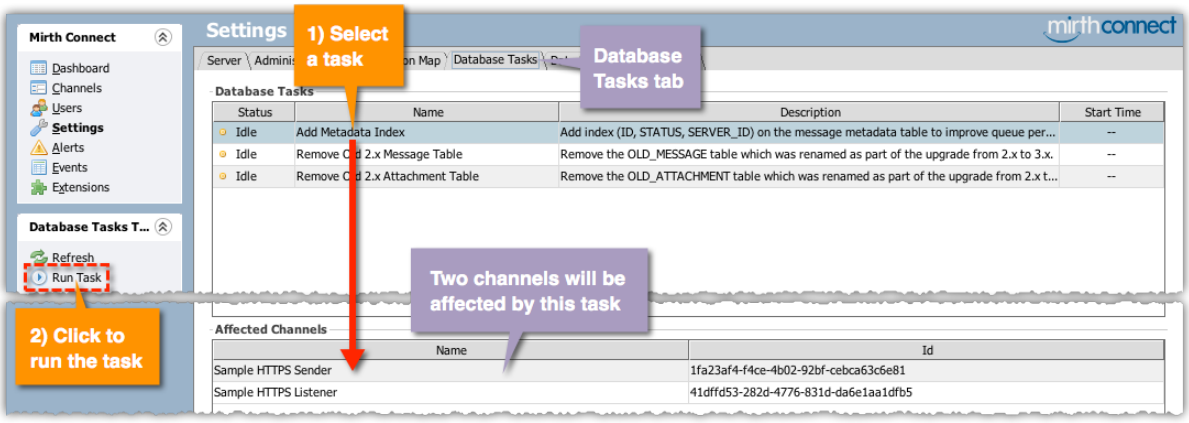
This is a read-only map that can be configured through the Administrator and used wherever global maps can be used.

Feature	Description
Add/Remove	Adds/removes properties from the configuration map
Import/Export Map (Configuration Map Tasks panel)	Import: inserts a properties file into the configuration map (which action removes and replaces any existing map values); Export: exports the configuration map to a properties file

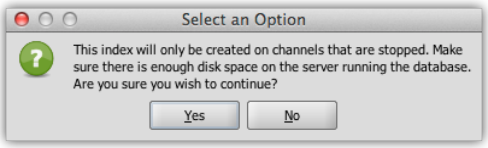
About the Database Tasks Page

Database Tasks

These are generated by the server when it detects "loose ends" (e.g., an unused table should be deleted or an index added to a table to improve performance) that, when resolved, serve to optimize the system. Some database tasks can take considerable time or have negative effects (e.g., deletion of old data). The **Database Tasks** feature gives users a way to perform such tasks at their discretion (i.e., during periods of low-volume message processing).



A task may effect certain channels, which appear in the **Affected Channels** list (previous graphic) when you select a task. When you then click **Run Task** (**Database Tasks Tasks** panel), a **Select an Option** dialog appears that has important information relative to the selected task and asks ***Are you sure you wish to continue?*** You choose the appropriate option to continue/cancel the job.



About the Resources Page

The features on this page let you configure shared objects or libraries that can be used across the entire server or on specific channels and connectors. You do not need to restart your server to include new libraries, and you do not need to redeploy channels to manually reload resources.

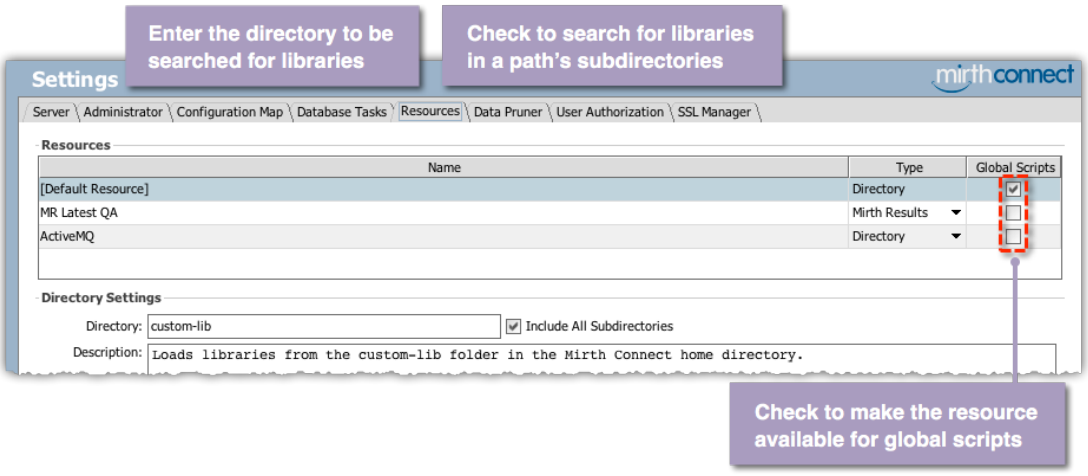
- (**Resources** section) You can check a resource's **Global Scripts** box to make that resource available for global deploy/undeploy/preprocessor/postprocessor scripts. (To prevent misconfiguration, only 1000 files per resource are scanned.)
- (**Directory Settings** section – **Directory** field) You can specify the directory to be searched for libraries. If

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

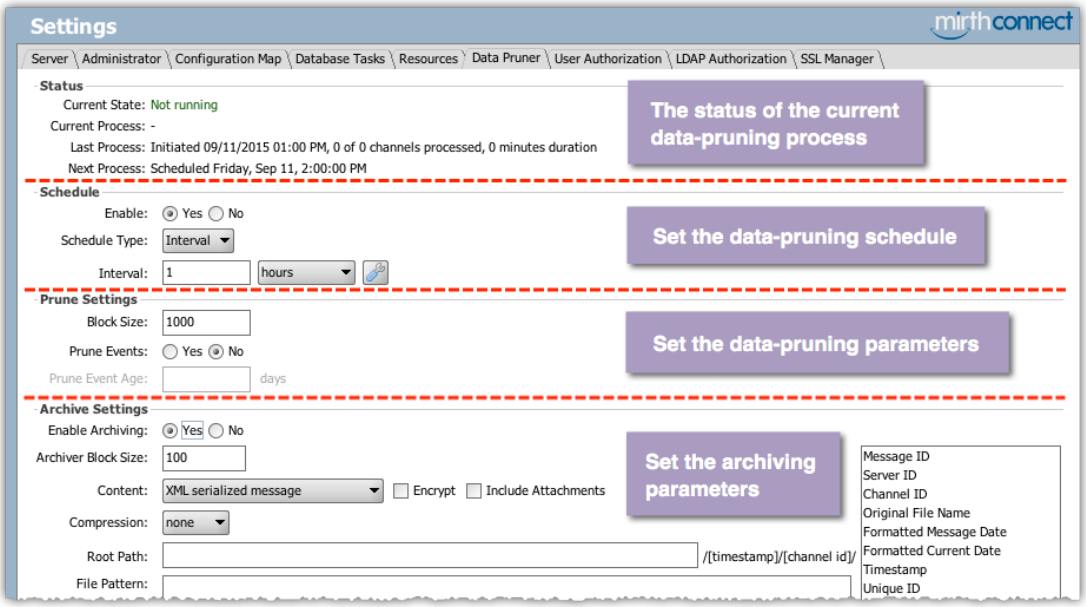
the directory does *not* begin with a forward slash (/), the path is relative to the Mirth Connect directory; if the directory *does* begin with a forward slash, the path is absolute.

- (**Directory Settings** section – **Directory** field) If the **Include All Subdirectories** box is checked, MC also searches for libraries in the path's subdirectories.

> Use this feature carefully, especially when using absolute paths, as the greater the number of files searched, the greater the amount of memory and time used to conduct the search, and the greater the risk of a server crash.

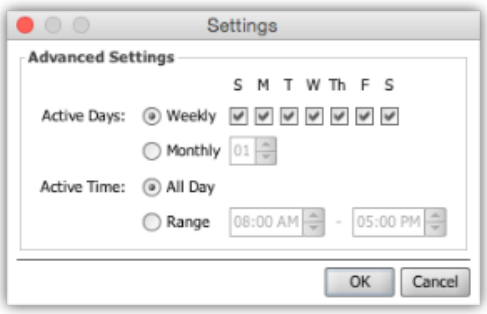


About the Data Pruner Page



Section Name	Description
Status	Displays the status of the current data-pruning process

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Schedule	Set the data-pruning schedule. **Schedule Type**: select either *Interval* (polls each "n" units of time), *Time* (polls once per day at the specified time), or *Cron* (polls at the specified cron expression(s); **Interval** field: Specify the repeating time interval in *hours/minutes/seconds/milliseconds* (for milliseconds, units must be less than 24 hours of time); Wrench icon: click to open a **Settings** dialog by which you can set active polling days and active polling time: 
Prune Settings	**Block Size** field: Set the most messages that can be deleted in a transaction. (If the **Block Size** is zero, there is no limit to the number of messages that can be deleted in one transaction.) To optimize performance and memory use, the pruner can delete fewer messages than the block size in a transaction; **Prune Event Age** field: Enter a value (in days) to prune messages older than that amount of days.
Archive Settings	**Enable Archiving**: Click the **Yes** button and configure archiving parameters as needed. This archives messages before pruning, which preserves message content in its pre-pruned state. The archiving settings work the same as the export-message settings. (See Importing/Exporting a Message > **Exporting a Message**); **Encrypt** checkbox: Check to encrypt archived messages; **Include Attachments** checkbox: check to archive a message's attachment(s) with the message.

You do not need to redeploy a channel when you change its prune-data value.

About Alerts

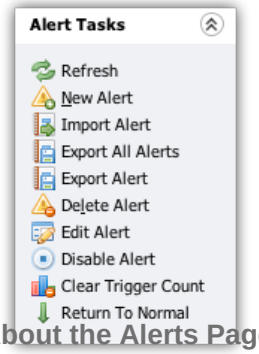
You can configure alerts to send emails when an exception is thrown as messages are processed. You can apply each alert to multiple channels and set alerts to run on an error condition based on a regex (regular expression). This section consists of an overview of the elements on the **Alerts** page and procedures that instruct you how to add and edit an alert.

About the Alert Tasks Panel

This table describes the actions available in the **Alert Tasks** panel.

Panel	Function	Description
	Refresh	Updates data on the Alerts List
	New Alert	Add an alert to the list
	Import Alert	Import an alert from an XML file
	Export All Alerts	Export all alerts to an XML file
	Export Alert	Export the selected alert to an XML file

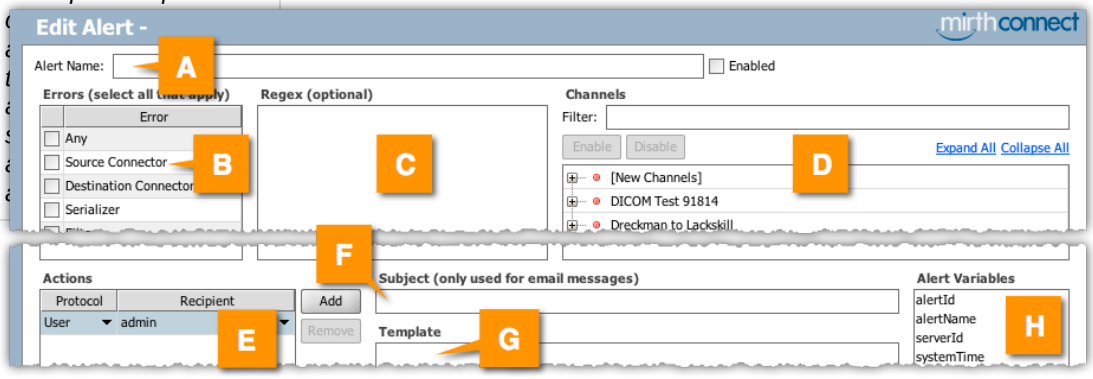
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Delete Alert	Remove the selected alert from the list
Edit Alert	Modify the selected alert
Disable Alert	Disable the selected alert
Clear Trigger Count	Resets the selected alert's trigger count to zero
Return To Normal	Resets the selected alert's level to Normal

About the Alerts Page

This section explains the various elements of the **Alerts** – **New Alert**/**Edit Alert** pages.



Alerts Page Elements

Element	Name	Description
A	Alert Name	The name you give the alert; *Enabled* checkbox: if checked, the alert is active
B	Errors	Check all boxes of the error types that will trigger the alert
C	Regex (optional)	(Regular expression) Specify a regex (e.g., "timeout"), if desired, against which the error message will be tested; only errors that match the regex in the window will trigger
D	Channels	**Filter**: Enter alphanumeric text to automatically display channels (and sources/destinations) that include the entered text; **Enable**/**Disable**: In the Channels window, select a channel/source/destination, and click a button to enable/disable the selection [for consecutive selections: select an option, hold down the Shift key, and select the final option in the group – the original two options and all options in between are selected; for non-consecutive selections: select an option, hold down the Ctrl key (PC) or command key (Mac), and make the other desired selections]; **Expand All**: Click to reveal each channel's source/destination(s); **Collapse All**: Click to show only the channel titles; **Channels Window**: Shows all channels that match the entry in the **Filter** field (if this field is blank, all channels appear in the window)

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

E	Actions	**Protocol**: (After adding an action or for an existing action) Click the menu arrow, and select *Email*, *Channel*, *User*, or *Role (the Role protocol only appears if the User Roles commercial plugin is installed)*; **Recipient**: Add an action, or select an existing action, then (for *Email*) click this column, and enter the email address; (for *Channel/User/Role*) click the menu arrow, and select a channel, user, or role; **Add**: Click to add an action; **Remove**: Click to delete the selected action
F	Subject	(Email messages only) Enter the email subject in this field
G	Template	Drag Alert Variables, as desired, into this box so when the email is sent, the body contains the replaced variables
H	Alert Variables	List of variables available for replacement in the **Subject** and **Template** fields (see the next table)

Alert Variables

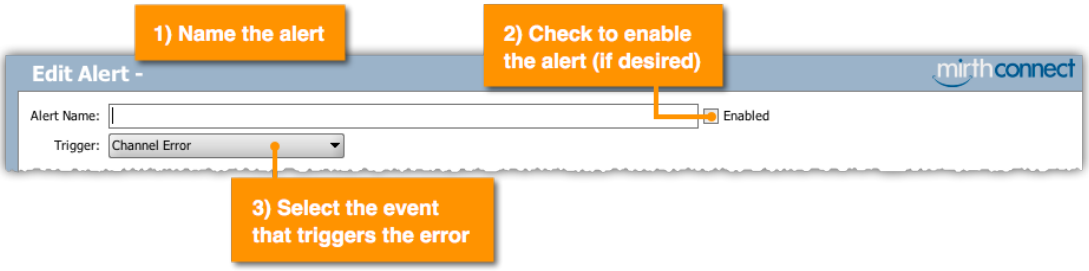
Variable	Output Function
alertId	The alert's unique Id
alertName	The alert's name
serverId	The server's unique Id
systemTime	System time in numeric form
date	The formatted system date/time
globalMapVariable	The value of a user-defined variable in the global map
error	Full error text including exception message and full stack trace
errorMessage	Error messages only
errorType	The type of error that triggered the alert
channelId	The channel's unique Id
channelName	The channel's name
connectorName	The connector's name
connectorType	The connector's type

Adding an Alert

This procedure instructs you how to add an alert. Be sure to configure an SMTP server on the **Settings** page before adding the alert so the alert can be sent. (See the note in Step 4.)

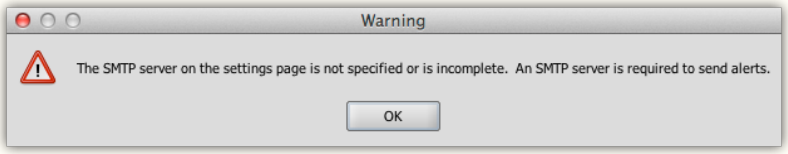
1. In the **Mirth Connect** panel, click **Alerts**.
2. In the **Alert Tasks** panel, click **New Alert**.
3. In the **Alert Name** field, enter a name for the alert, then check the **Enabled** box (if desired) to activate the alert.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



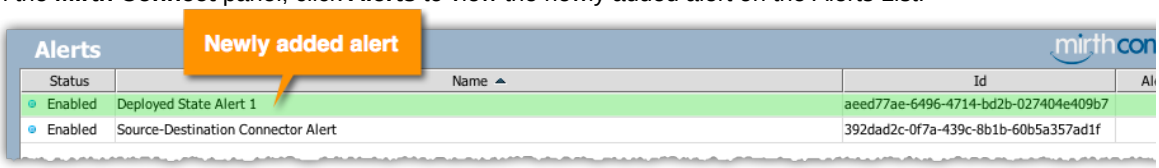
4. Configure other elements as needed, and in the **Alert Edit Tasks** panel, click **Save Alert**.

If you have not configured an SMTP host server via the **Mirth Connect** panel > **Settings** page, a **Warning** dialog appears informing you that you need to do so.



Go to the **Settings** page, and configure the SMTP server.

5. In the **Mirth Connect** panel, click **Alerts** to view the newly added alert on the Alerts List.

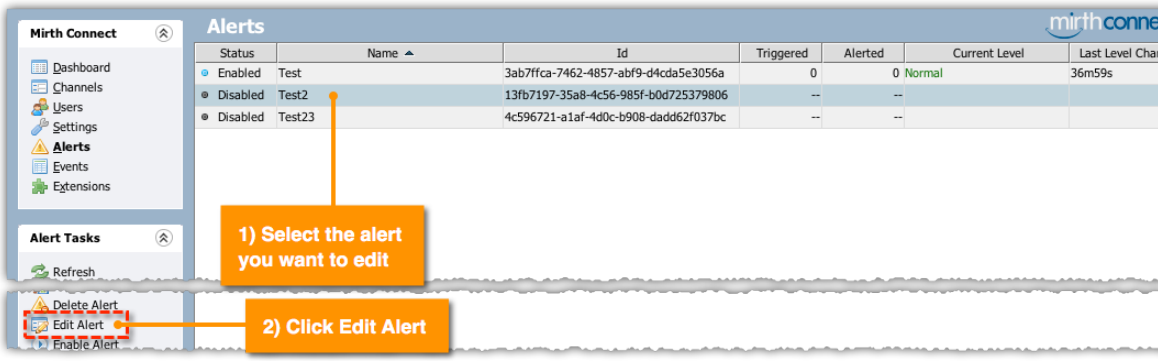


Editing an Alert

This procedure instructs you how to edit an alert.

1. In the **Mirth Connect** panel, click **Alerts**.
2. On the Alerts List, select the alert you want to edit, and in the **Alert Tasks** panel, click **Edit Alert**.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

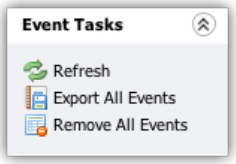


3. Make your desired changes, and in the **Alert Tasks** panel, click **Save Alert**.

> If you try to save the alert after making a change that requires corresponding changes – which you did not make – a **Warning** dialog appears informing you of each pending required change, and the respective fields on the **Alerts** page automatically highlight. Click the dialog's **OK** button, and make the changes, then save the alert.

About Events

This table describes the actions available in the **Alert Tasks** panel.

Panel	Function	Description
	Refresh	Updates data on the Event List
	Export All Events	Export all events to a file on the server
	Remove All Events	Remove all events (with the option of exporting them to a file on the server)

In MC, system events (e.g., logins, errors, channel updates) log to an internal database that you can view via the **Mirth Connect** panel > **Events**. The log includes:

- the date/time of the event
- a descriptive name for the event (e.g., *Get channel* means the user created a new channel; *Resume channel* means the user restarted a paused channel)
- the entity (*user* or *system*) that caused the event
- the event's outcome
- the event's IP address.

To search for an event, in the **Name** field, enter some of all of the event name, and click the **Search** button. The events that share the **Name**-field entry appear in the event log (e.g., if you enter *Get*, all events appear in the log that have *Get* in their names).

To restore all events to the log, delete the event name from the **Name** field, and click the **Search** button (or in the

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

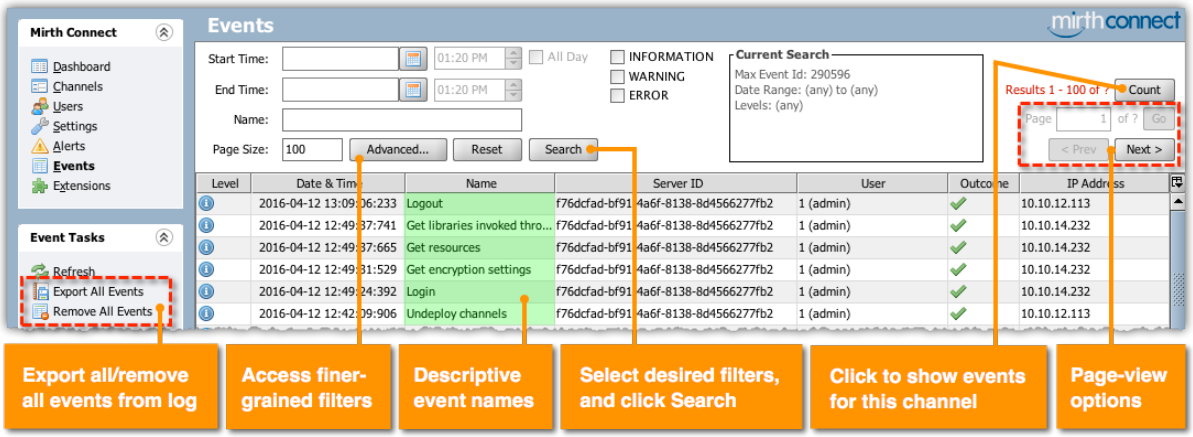
Event Tasks panel, click **Refresh**.)

In the **Page Size** field below the **Name** field, you can enter a value that represents the most events that can appear on every log page.

> When the number of events exceeds the value in the **Page Size** field, the **< Prev**/**Next >** buttons below the **Page** field activate, and you can click them to go back or advance one page.

You can check one or more event-type filter boxes (*INFORMATION/WARNING/ERROR*) to restrict your search to events of any of those types.

To display the total number of events for the selected channel, click the **Count** button in the top-right corner of the page. In the **Page** field, you can enter a page number to view the events on that page.

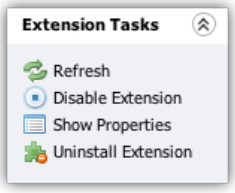


You may export all events in the Event Log via the **Event Tasks** panel > **Export All Events**. The contents of the log are moved to an export file in the server's export directory. To clear the Event Log without moving its contents to an external file, in the **Event Tasks** panel, click **Remove All Events**.

Managing Extensions

About the Alert Tasks Panel

This table describes the actions available in the **Alert Tasks** panel.

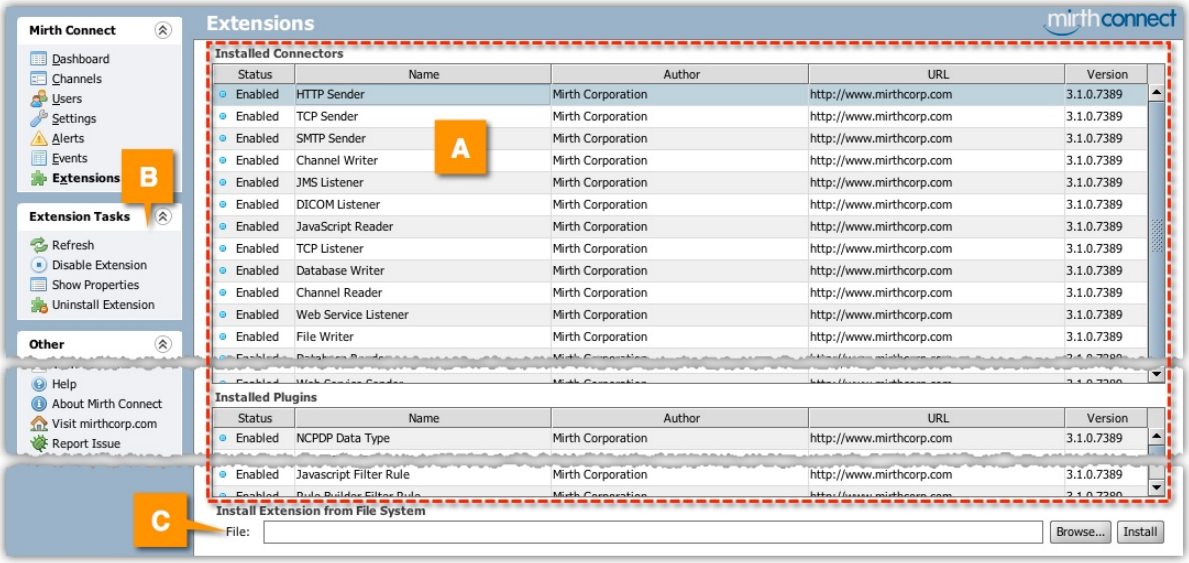
Panel	Function	Description
	Refresh	Updates data on the Extensions List
	Disable Extension	Disables the selected extension
	Show Properties	Displays the selected extension's properties (name, type, priority, author, et al.
	Uninstall Extension	Uninstall the selected extension

In the **Mirth Connect** panel, click **Extensions** to reveal the Extensions Manager, where you can:

- Manage installed connectors and plugins (from the **Installed Connectors** and **Installed Plugins** lists – A)
- Disable/enable/uninstall extensions (**Extension Tasks** panel – B)

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

- Install extensions from the file system (**Install Extension from File System** field – C).



Commercial Extensions

Mirth Connect allows commercial extensions, many of which come standard with the MC download. Certain other extensions, however, are only available via customer-support subscriptions, of which there are three levels: *Silver*, *Gold*, *Platinum*. The table shows the commercial extensions (aka *Enterprise* extensions) available for MC and the customer-support level(s) required to obtain them. Descriptions of each extension appear in this section. Click a link in the table to learn more about that extension:

Extension	Support-Level Availability
Advanced Alerting	Platinum
Advanced Clustering	Platinum
ASTM E1381 Transmission Mode	Platinum
ASTM E1394 Data Type	Platinum
Channel History	Silver, Gold, Platinum
Email Reader	Gold, Platinum
LDAP Authorization	Gold, Platinum
Message Generator	Silver, Gold, Platinum
Mirth Results Service	Platinum
SSL Manager	Silver, Gold, Platinum
Serial Connector	Platinum
User Roles	Gold, Silver

You can view which extensions are installed in your system via the **Mirth Connect** panel > **Extensions** – **Installed Connectors**/**Installed Plugins** lists.

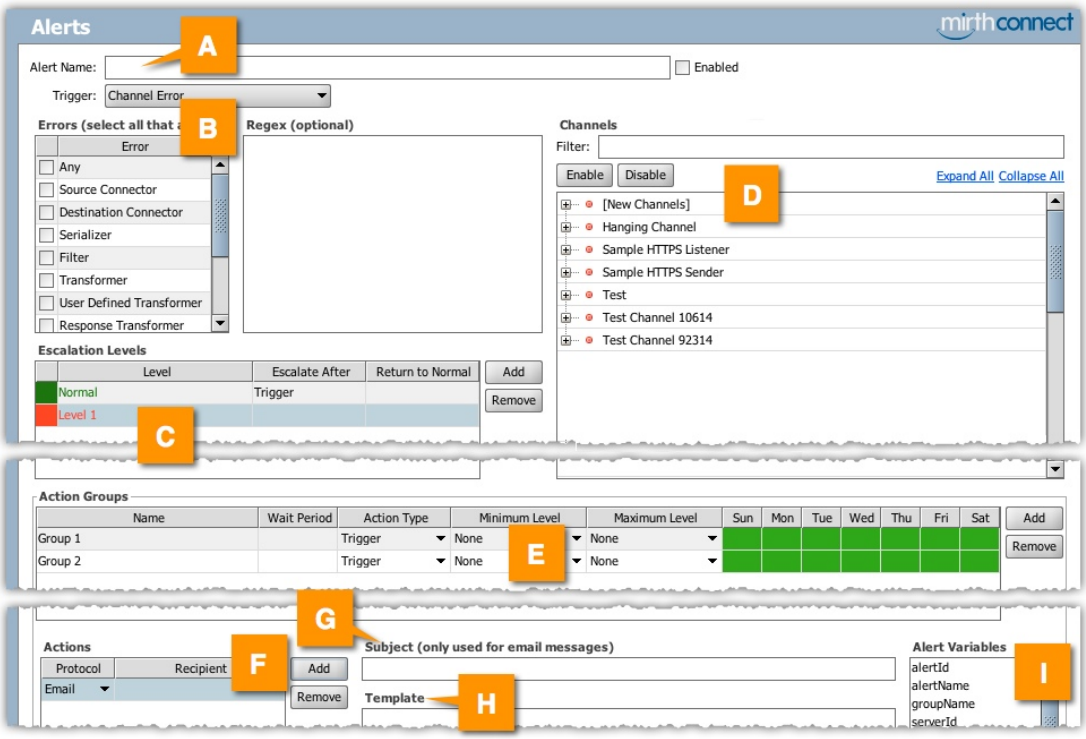
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

For details about support-level subscriptions for Enterprise extensions or for technical/purchasing information about the extensions themselves, visit the Mirth Corporation website: http://www.mirthcorp.com/ > **Products** > **Mirth Connect** > **Extensions** (and) **Pricing**, or contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About the Advanced Alerting Plugin

This extension provides metric-, state-, and error-based monitoring of channels and connectors. Additional features include automatic escalation/de-escalation, scheduling, and notification throttling. Advanced alerts can dynamically send different alert messages to different user groups based on the current escalation level, time, and day. The new alert dashboard provides a view of all alert statistics and logs.

This section consists of an overview of the elements on the **Alerts** page when the Advanced Alerting plugin is installed. (For directions on adding/editing alerts, see About Alerts.)



The previous graphic shows the *Channel Error* trigger. Other triggers are, from left, *No. of messages (sliding window)* ; *No. of messages (threshold)*; *Deployed State*:

Advanced Alerts Page Elements

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Element	Name	Description
A	Alert Name	The name you give the alert; *Enabled* checkbox: if checked, the alert is active
B	Trigger	The factor that initiates the alert: **No. of messages (sliding window)** – **Status**: The message status (*Received/Filtered/Sent/Queued/Errored*) pertaining to the alert; **Condition**: The condition to check against the specified status count (>=/<=); **Limit**: The least/most messages (depending on the Condition) that must/can be set to the specified status in the window length before this alert triggers; **Window Length**: The message-monitoring timespan (e.g., 1d6h30m); **Aggregate By**: *Connector* – applies the specified trigger condition to an individual enabled connector; *Channel* – applies the specified trigger condition to all of a channel's enabled connectors; *Global* – applies the specified trigger condition to all enabled connectors in all channels **No. of messages (threshold)** – **Status**: The message status (*Queued/Errored*) pertaining to the alert; **Threshold**: The alert triggers when the number of messages with the specified status is the value in this field; **Longer Than**: The alert triggers when the number of messages with the specified status has been the threshold value for longer than the specified amount of time (e.g., 1d6h30m) – if blank, the alert triggers immediately; **Re-trigger Interval**: After the first trigger, the alert re-triggers periodically (e.g., 1d6h30m) until the status count is below the threshold; **Aggregate By**: *Connector* – applies the specified trigger condition to an individual enabled connector; *Channel* – applies the specified trigger condition to all of a channel's enabled connectors; *Global* – applies the specified trigger condition to all enabled connectors in all channels **Channel Error** – **Errors**: Check all boxes for the error types that will trigger the alert; **Regex** (Regular Expression): You can specify a regex (e.g., "timeout") against which the error message will be tested; only the errors that match against this regex will trigger **Deployed State** – Check all boxes for the deployment states (*Starting/Started/Stopping/Stopped*) that will trigger the alert; **Longer Than**: The alert triggers after a *Channel/Connector* (selected via the **Type** bar) has been in a selected state longer than the given timespan (e.g., 1d6h30m)
C	Escalation Levels	**Level**: The name of the level; **Escalate After**: The condition that escalates this alert to the next level (*Normal*: escalation occurs when the alert triggers; all others: escalation occurs after the given timespan); **Return to Normal**: The alert returns to normal level if no triggers occur in the given timespan (e.g., 1d6h30m); **Add**: Click to add a level; **Remove**: Click to delete the selected level (you cannot remove the alert's original level)
D	Channels	**Filter**: Enter alphanumeric text to automatically display channels (and sources/destinations) that include the entered text; **Enable**/**Disable**: In the Channels window, select a channel/source/destination, and click the corresponding button to enable/disable the selection [for consecutive selections: select an option, hold down the Shift key, and select the final option in the group – all options in between are selected; to select non-consecutive options: select an option, hold down the Ctrl key (PC) or command key (Mac), and make the other desired selections]; **Expand All**: Click to reveal each channel's source/destination(s); **Collapse All**: Click to show only the channel titles; **Channels Window**: Shows all channels that match the entry in the **Filter** field (if this field is blank, all channels appear in the window)

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

E	Action Groups	**Name**: Double-click the row in this column, and enter a name for the action group; **Wait Period**: The period (e.g., 1d6h30m) between alert messages sent by the action group (triggers and level changes that occur during this period are ignored); **Action Type**: The condition (*Trigger/Level Change*) that causes this action group to send an alert message (click the menu arrow, and select an action type); **Minimum/Maximum Level**: *Trigger* action – the min/max level to which the alert can fall/rise, prompting the group to send an alert message; *Level Change* action – applies to the level being entered (menu options – via the menu arrow – reflect the levels available in the Escalation Levels window); **Days of the Week**: The daily/hourly periods of the action group's activity (click any cell to edit the schedule); **Add**: Click to add an action group; **Remove**: Click to delete the selected group
F	Actions	**Protocol**: (After adding an action or for an existing action) Click the menu arrow, and select either *Email* or *Channel*; **Recipient**: Add an action or select an existing action, then (for *Email*) click this column, and enter the email address, or (for *Channel*) click the menu arrow, and select a channel; **Add**: Click to add an action; **Remove**: Click to delete the selected action
G	Subject	(Email messages only) Enter the email subject in this field
H	Template	Drag Alert Variables as desired into this box so when the email is sent, the body contains the replaced variables
I	Alert Variables	List of variables available for replacement in the Subject and Template fields

Advanced Alert Variables

Many variables are available for Alert messages (not all variables are available for all Alert triggers):

Variable	Output Function
alertId	The alert's unique Id
alertName	The alert's name
groupName	The action group's name
serverId	The server's unique Id
systemTime	System time in numeric form
date	The formatted system date/time
globalMapVariable	The value of a user-defined variable in the global map
currentLevel	The alert's current escalation level
error	Full error text including exception message and full stack trace
errorMessage	Error messages only
errorType	The type of error that triggered the alert
channelId	The channel's unique Id
channelName	The channel's name

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

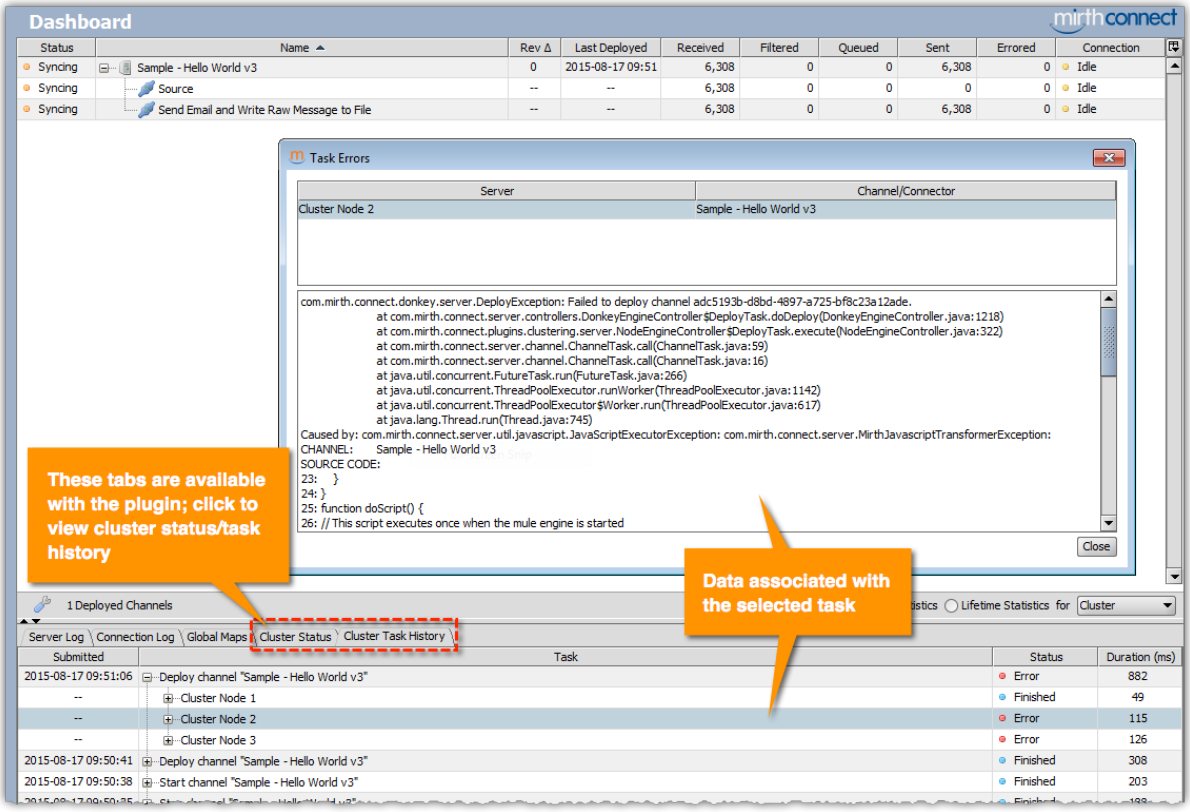
connectorName	The connector's name
connectorType	The connector type

You can also trigger alerts via JavaScript (except in the **Database Reader** and **Database Writer** connectors): Alerts.sendAlert ("Alert message").

The Advanced Alerting extension requires a Platinum-level support subscription. For technical and purchasing information about Advanced Alerting, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About the Advanced Clustering Plugin

This plugin improves the availability of your Mirth Connect cluster with automatic message takeover. If one server in the cluster fails, another automatically resumes processing of any queued or unfinished messages from the failed server. Managing your cluster becomes easier as you can monitor each server's status, deploy/start/stop/pause channels across all servers, and view message statistics for the whole cluster or for a particular server via the Mirth Connect dashboard. Use this plugin with standalone Mirth Connect instances using your own load-balancing solution and seamlessly integrate with the load-balancing and fail-over services provided on Mirth Appliances.



For details, including installing, setting up, and operating advance clustering, see the *Advanced Clustering User Guide*.

The Advanced Alerting extension requires a Platinum-level support subscription. For technical and

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

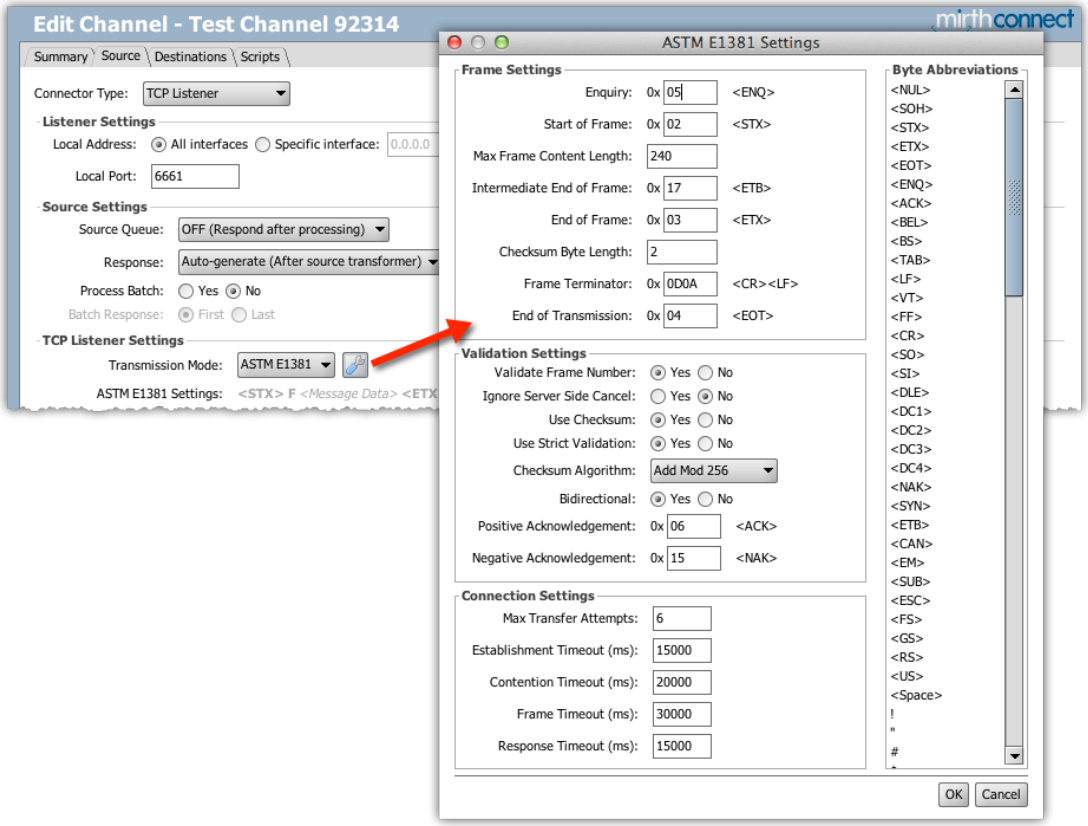
purchasing information about Advanced Alerting, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About the ASTM E1381 Transmission Mode Plugin

The ASTM E1381 extension for Mirth Connect lets you send/receive data using the ASTM E1381 lower-layer protocol standard. It can be used with the TCP Connector or the Serial Connector. By default, the official ASTM E1381 standard is strictly followed, but several settings are available to support variations from the standard. The extension supports specifying the bytes used for frame delimiters, the checksum algorithm used, various timeouts, and more.

Accessing the ASTM E1381 Settings

On an **Edit Channel** page – **Source** tab – click the **Connector Type** bar, and select *TCP Listener*. In the **TCP Listener Settings** section, click the **Transmission Mode** bar, and select *ASTM E1381*, then click the Wrench icon to open the **ASTM E1381 Settings** dialog.

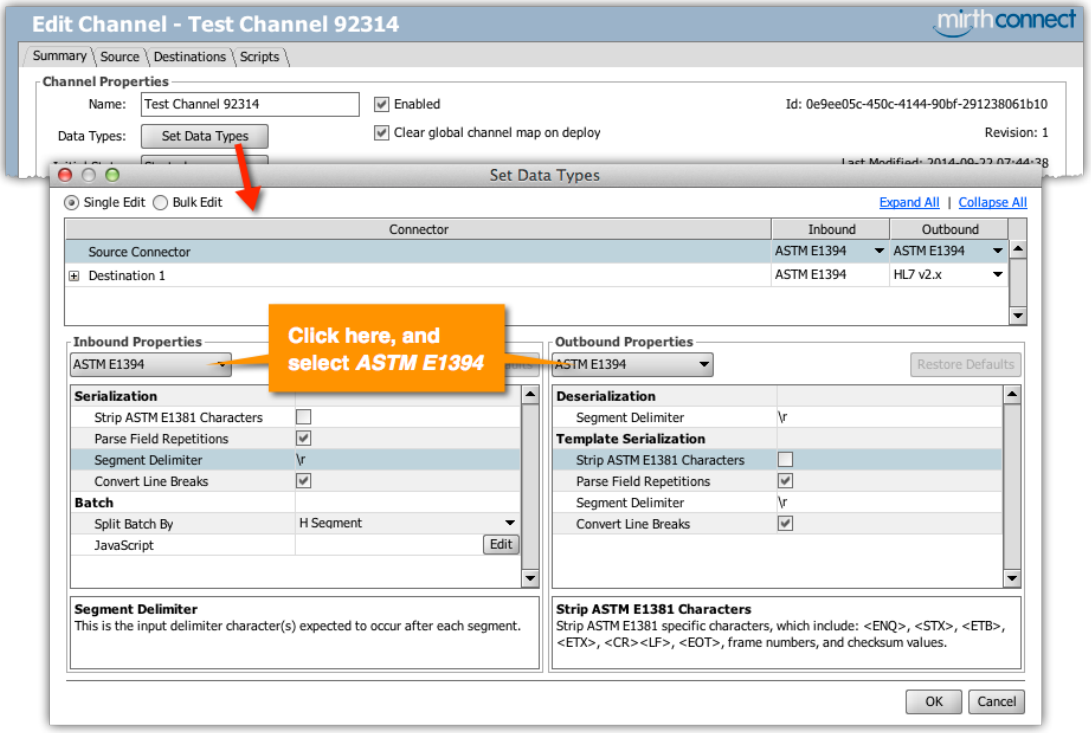


ASTM E1381 Transmission Mode requires a Platinum-level support subscription. For technical and purchasing information about ASTM E1381 Transmission Mode, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

About the ASTM E1394 Data Type Plugin

With this extension you can easily accept, parse, and transform messages following the ASTM E1394 data standard. As with other data types, incoming messages are converted to XML so all the usual transformer steps can be used to transform the message or convert it to/from a different data type. Several data-type properties are also provided for specifying how messages should be converted to/from XML. A standard vocabulary for the ASTM E1394 data type is also included to help identify fields in a transformer's message-template trees.



> The ASTM E1394 Data Type requires a Platinum-level support subscription. For technical and purchasing information about the ASTM E1394 data type, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About the Channel History Plugin

To access the **Channel History** dialog: In the **Mirth Connect** panel, click **Channels**, and select a channel with two or more revisions in the **Rev** column, then in the **Channel Tasks** panel, click **View History**.

> This plugin also lets you track the revision histories of code templates and code-template libraries on the **Code Templates** page (**Channels** – **Channel Tasks** > **Edit Code Templates**). Because the **Code Template History** and **Code Template Library History** dialogs contain the same functions as the **Channel History** dialog, only one set of descriptions of those functions is included here.

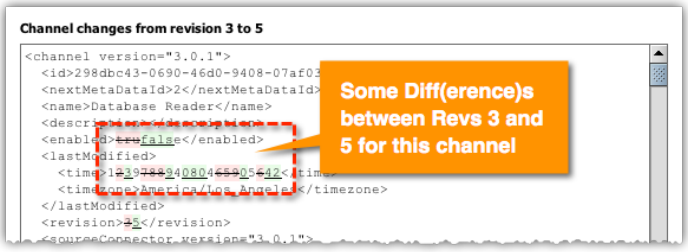
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

Feature	Name	Description
A	Diff(erence)	Check two boxes for comparison of the two channel revisions; you cannot check more than two boxes
B	Rev(ision)	Identifies the channel revision
C	Date & Time	The year/month/day and hour/min that the revision occurred
D	User	Identifies the user who revised the channel
E	Rev Tally	The number of revisions for this channel since its last deployment
F	Prune	Reveals a **Select an Option** dialog, which asks if you are sure you want to remove all revisions prior to the selected one (click **Yes** to remove them or **No** to cancel the action)
G	Channel Viewer	(With the **Diff** boxes of two revisions checked and the **Compare** button clicked) An XML version of the channel's properties, with the differences between revisions highlighted, appears here
H	Revert	(Select a Rev to enable) Reveals a **Select an Option** dialog, which asks if you are sure you want to revert to the selected version of the channel (click **Yes** to revert to the selected version – eliminating all later versions of the channel – or **No** to cancel the action)
I	Compare	(With the **Diff** boxes of two revisions checked) Reveals, in the Channel Viewer, an XML version of the channel's properties with the differences between revisions highlighted
J	Close	Closes the **Channel History** dialog

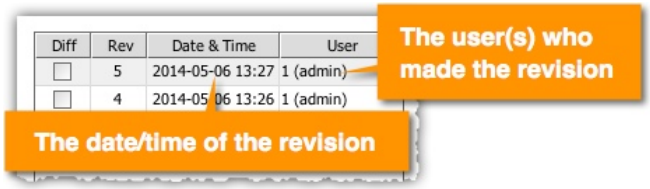
This extension lets you manage your channels' revision histories via a series of useful methods:

You can track a channel's evolution, comparing current and past traits side by side. The pink, strike-through portion is the earlier revision; the green, underlined portion is the current revision.

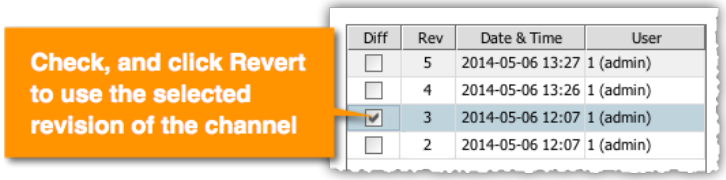
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



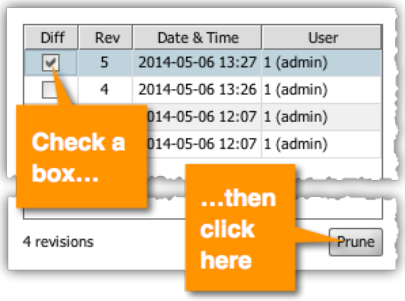
You can see when changes were made to the channel and who made them.



You can revert to a past version of the channel.



You can prune unneeded revisions of the channel by selecting the earliest revision you want to keep and clicking the **Prune** button. A **Select an Option** dialog appears, asking if you are sure you want to remove all revisions prior to the selected one. Click **Yes** to eliminate those revisions. You can only prune revisions previous to your selection; you cannot prune revisions later than your selection. You cannot prune all revisions or non-consecutive revisions, nor can you prune the selected revision.



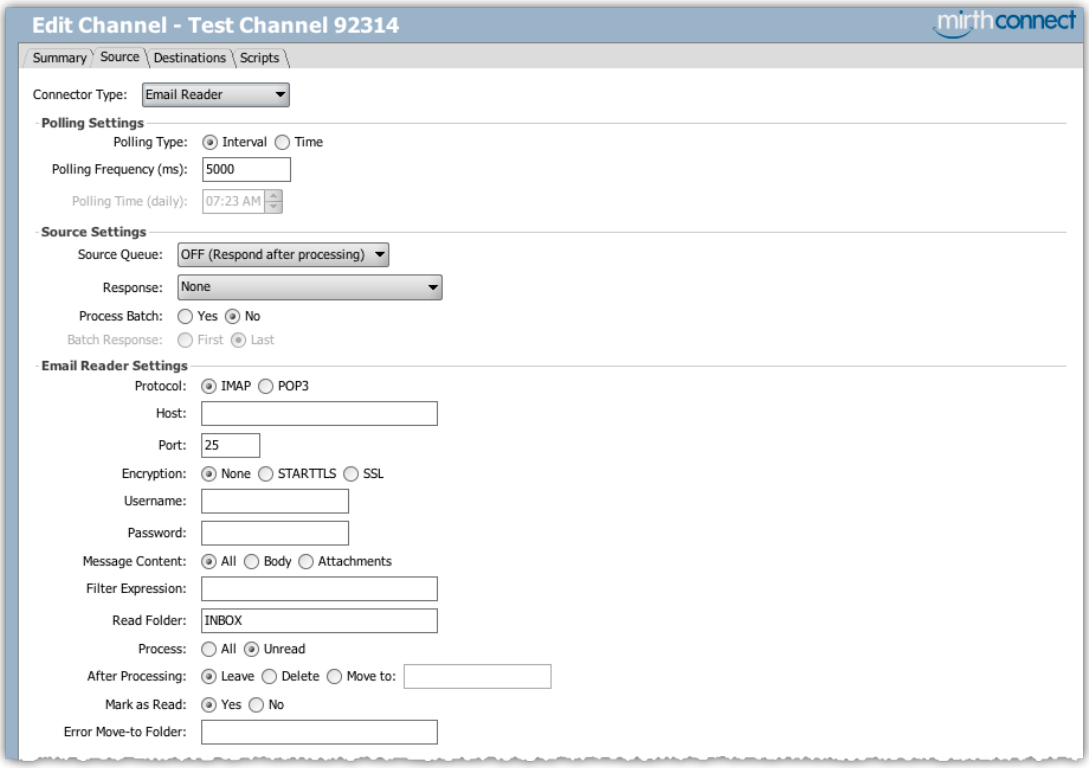
> The Channel History plugin requires a Silver-, Gold-, or Platinum-level support subscription. For technical and purchasing information about the Channel History plugin, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About the Email Reader Plugin

This extension lets you securely connect to a POP3 (post office protocol v3) email server or IMAP (internet message

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

access protocol) email server and download email messages for processing in a channel. Via the Message Content parameter, you can specify whether or not an email message should be read as XML, including 1) the metadata and the body, 2) just the body, or 3) a set of attachments. Many options are available for specifying behavior once a message is read.

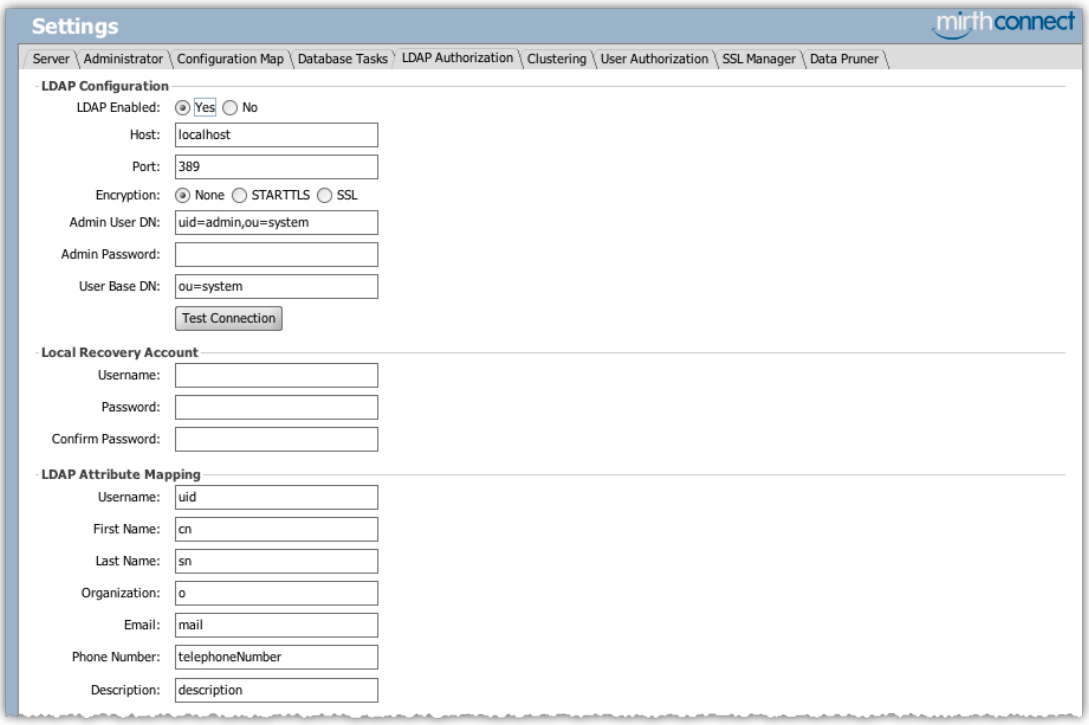


> The Email Reader requires a Gold- or Platinum-level support subscription. For technical and purchasing information about the Email Reader, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About the LDAP Authorization Plugin

This extension replaces the existing authentication mechanism and authenticates against an LDAP (lightweight directory access protocol) server so user accounts can be managed on a centralized LDAP server. Any user in the specified User Base DN on the LDAP server can log in to Mirth Connect. When a user logs in, their attributes are copied from the LDAP server into MC. The connection to the LDAP server can use SSL or STARTTLS encryption.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

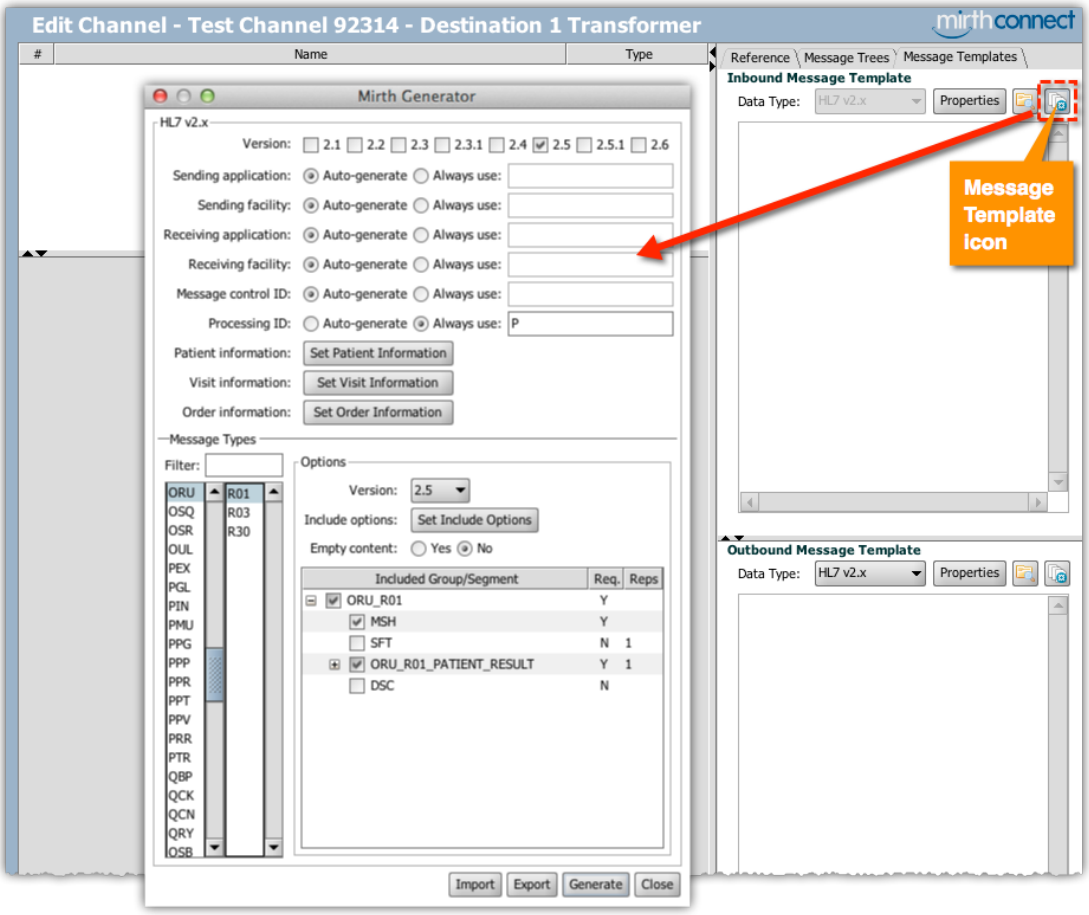


LDAP Authorization requires a Gold- or Platinum-level support subscription. For technical and purchasing information about LDAP Authorization, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About the Message Generator Plugin

This extension lets you quickly and easily generate HL7 v2.x messages for: 1) a transformer's inbound/outbound template, 2) sending to a channel, or 3) testing. You can create messages of any type/trigger and any version (HL7 v2.1 - 2.6), with specific options for which segments/fields/components to include. Where applicable, pseudo-data (dates, names, free text, and even random values from the HL7 specification tables) is generated, though there are several options for overriding parts of the message with your own data.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



The Message Generator plugin requires a Silver-, Gold-, or Platinum-level support subscription. For technical and purchasing information about the Message Generator plugin, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About the SSL Manager Plugin

This extension lets you quickly enable/configure certificate-based SSL (Secure Socket Layer) connectivity for the socket-based connectors HTTP Listener/Sender, Web Service Listener/Sender, and File Reader/Writer (in FTP mode). Keystore-based certificate management lets you store your certs (certificates) in one location and apply them to your existing connectors.

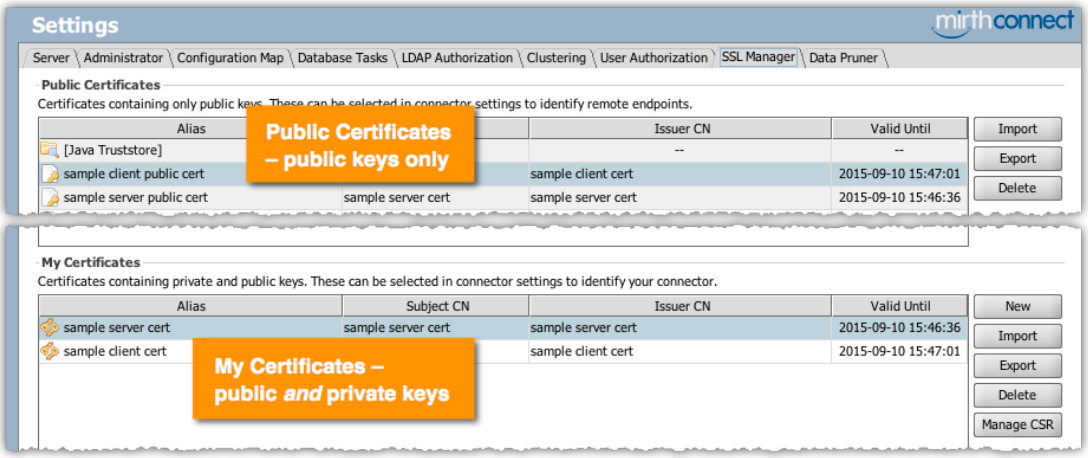
The SSL Manager plugin requires a Silver-, Gold-, or Platinum-level support subscription. For technical and purchasing information about the SSL Connector plugin, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

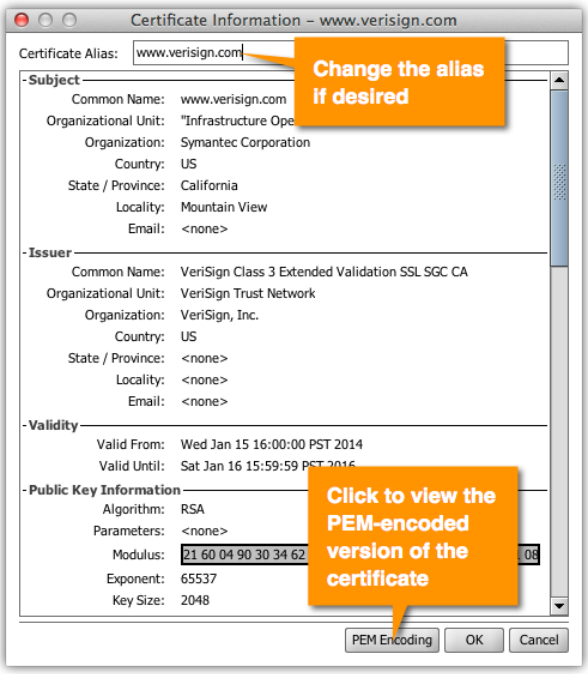
Managing Certificates

You can manage certificates and key pairs on the **Settings** page > **SSL Manager** page.

The **Public Certificates** table (top) displays certs (public keys only) that you trust. These could be server certificates returned to a destination connector or client certs returned to a source connector. The **My Certificates** table (bottom) displays certs (public and private keypairs) that identify your server directly. These could be server certs used by a source connector or client certs used by a destination connector.



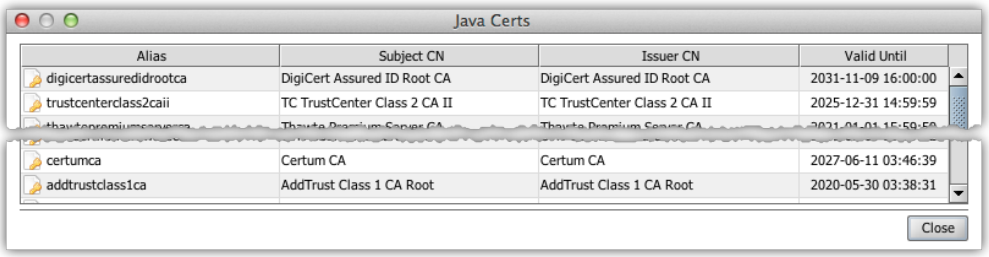
Double-click any row in a certs list to open the **Certificate Information** dialog, on which you can view details about the selected cert.



On this dialog, you can change the alias being used, and you can click the **PEM Encoding** button to view the PEM-encoded version of the cert.

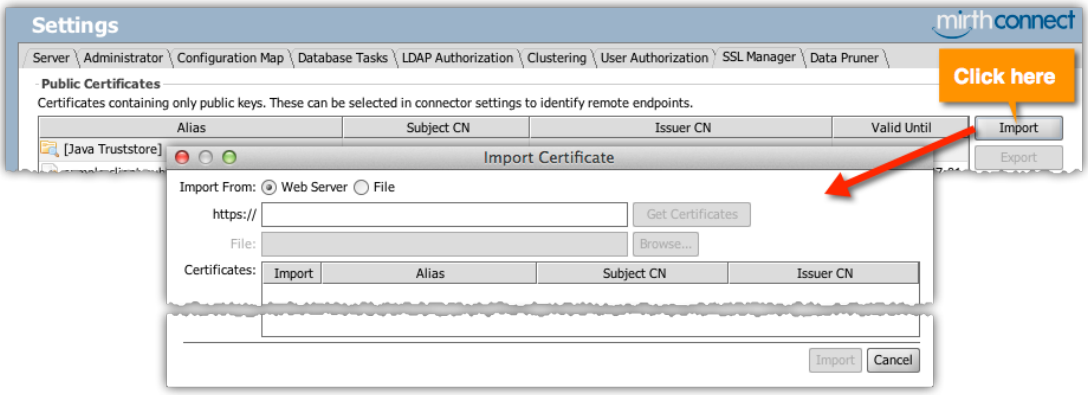
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

On the **Public Certificates** list, click the **Java Truststore** row to open the **Java Certs** dialog on which you can view a read-only, default Java list of *trusted* certificates (aka *cacerts*).



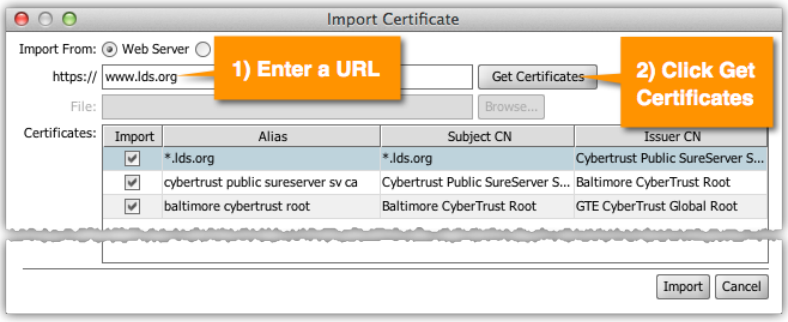
Importing a Public Certificate

On the right-hand side of the **Public Certificates** table, click the **Import** button to open the **Import Certificate** dialog.

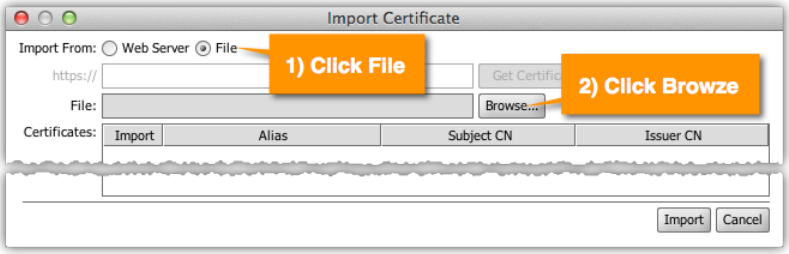


In the **https://** field, enter a URL, and click the **Get Certificates** button to pull certs from an external web server…

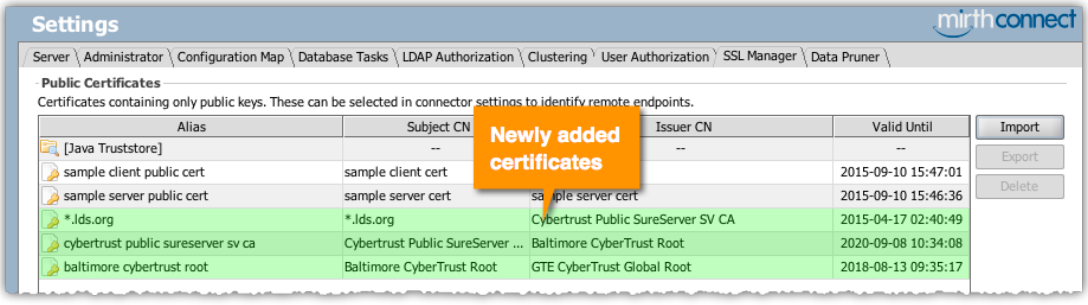
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



…or click the **File** radio button > **Browse…** button, and choose a certs to import. Once you do either, the cert chain is added to the **Certificates** list.



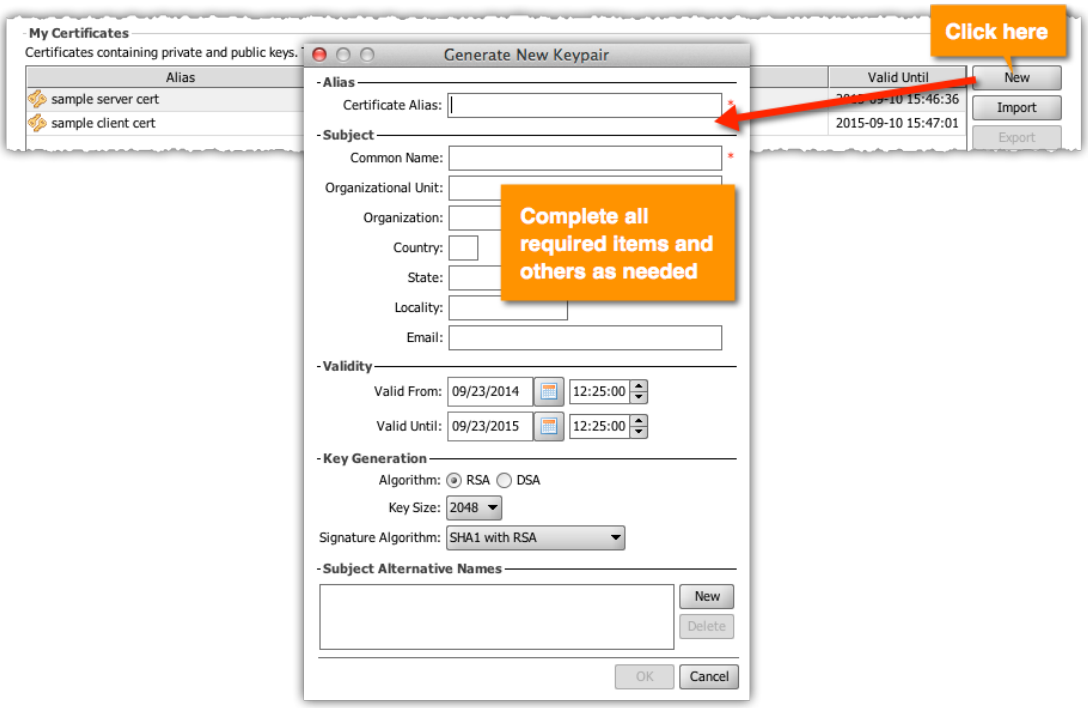
Double-click any of the newly added rows to view details about the alias to edit the alias. In the **Import** column (**Import Certificate** dialog – **Web Server**), check the boxes of the certs you want to import, then click the **Import** button to add the selected certs to the **Public Certificates** table.



Creating a New Public/Private Key Pair

Click the **New** button next to the **My Certificates** table to open the **Generate New Keypair** dialog.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Complete the required* fields and others as needed, and click the **OK** button. The new cert appears on the **My Certificates** list.

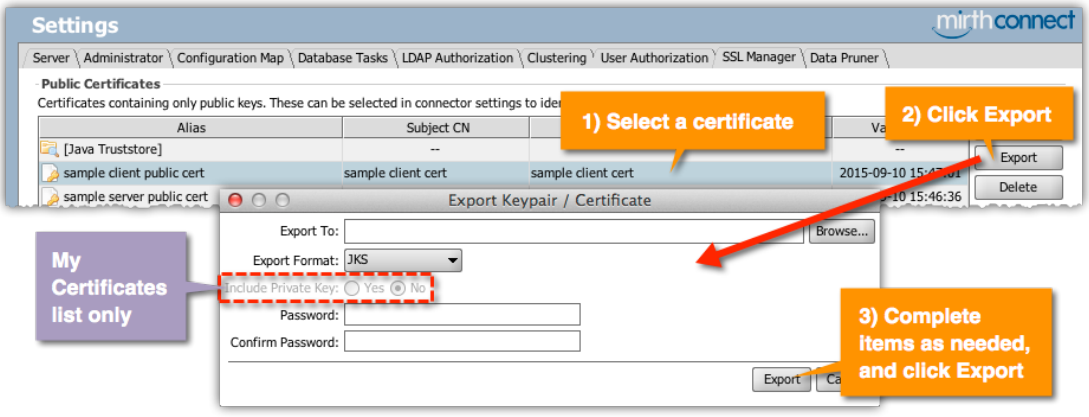
Importing a Public/Private Keypair

This procedure is essentially the same as importing a public cert, except that you can only import from a file. (See Importing a Public Certificate).

Exporting Certificates

In either Certificates table, select a cert, and click the table's **Export** button to open the **Export Keypair/Certificate** dialog.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



Select a file name and an **Export Format**. If you are exporting a public/private key pair from the **My Certificates** list, you can include the private key, and provide a password for the selected file.

Deleting Certificates

To delete a cert from either Certificates table: Select the cert, and click the list's **Delete** button.

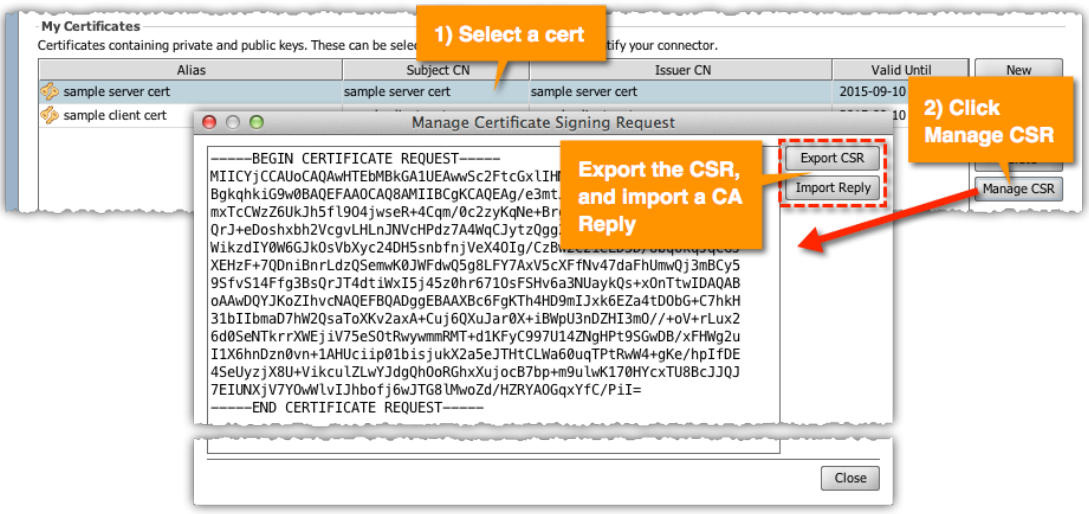
Saving Changes

After editing items on the **Settings** – **SSL Manager** page, save your changes before leaving the page by clicking the **Save** function in the **SSL Manager Tasks** panel. If you try to leave the **SSL Manager** page before saving your changes, a dialog appears that asks **Would you like to save the SSL Manager settings changes?** Click *Yes* to save your changes and go to the selected page, *No* to forfeit the changes and go to the selected page, or *Cancel* to stay on the **SSL Manager** page with your changes intact though unsaved.

Managing Certificate Signing Request (CSRs)

To generate a CSR for a self-signed key pair, select a cert from the **My Certificates** table, and click the table's **Manager CSR** button to open the **Manage Certificate Signing Request** dialog with the PEM-encoded version of the CSR. You can export it to a file if desired, in which case, you would need to send the CSR to an external certificate authority such as VeriSign or Entrust. The selected cert authority will return a CA (Certificate Authority) reply file, which you can add to the dialog by clicking the dialog's **Import Reply** button.

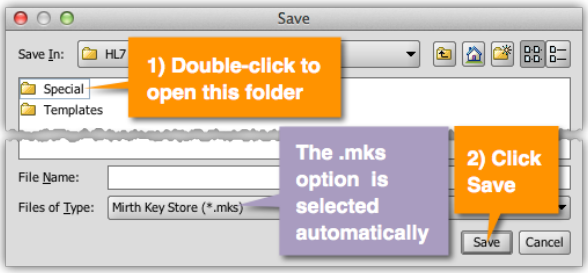
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



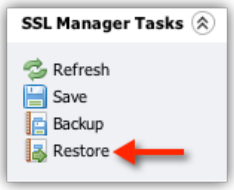
Backing Up/Restoring an SSL Manager Configuration

To back up your **Public Certificates** and **My Certificates** lists, in the **SSL Manager Tasks** panel, click **Backup**.to open the **Save** dialog.

The backup should be saved as a special .mks (Mirth Key Store) format; consequently, on the **Save** dialog, the *Mirth Key Store (*.mks)* option automatically appears on the **Files of Type** bar. Navigate to the **HL7** folder, and double-click to open the **Special** folder, then click the **Save** button.



After backing up your configuration, you can restore it onto another Mirth Connect server with the SSL Manager installed by clicking **Restore** in the **SSL Manager Tasks** panel.

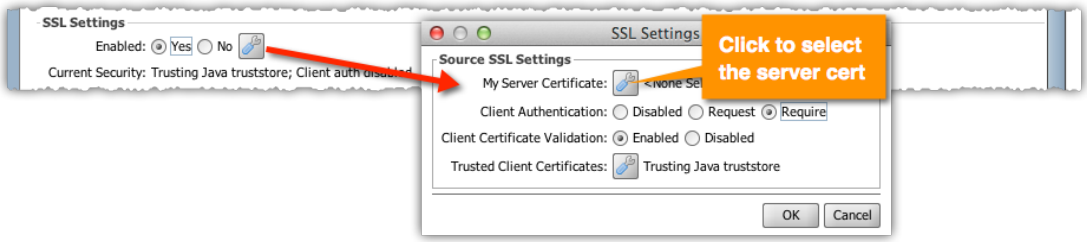


Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

On the **Open** dialog that appears, navigate to your backed-up file, and click the **Open** button.

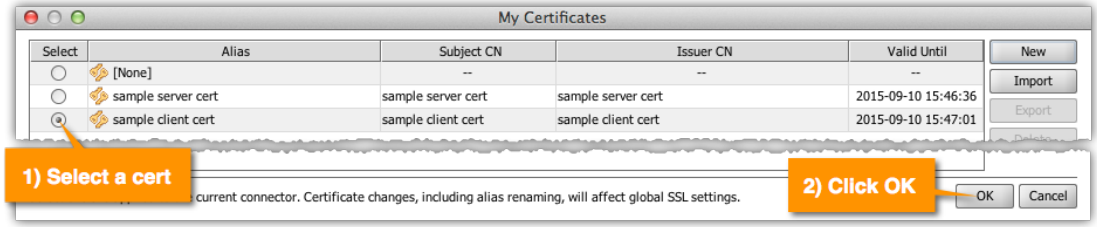
Applying Source Connector SSL Settings

On an **Edit Channel** > **Source** page, select a **Connector Type** that supports the SSL Manager (HTTP Listener, Web Service Listener, File Reader in FTP mode). In the **SSL Settings** section – **Enable** element, select *Yes*, then click the Wrench icon to open the **SSL Settings** dialog.

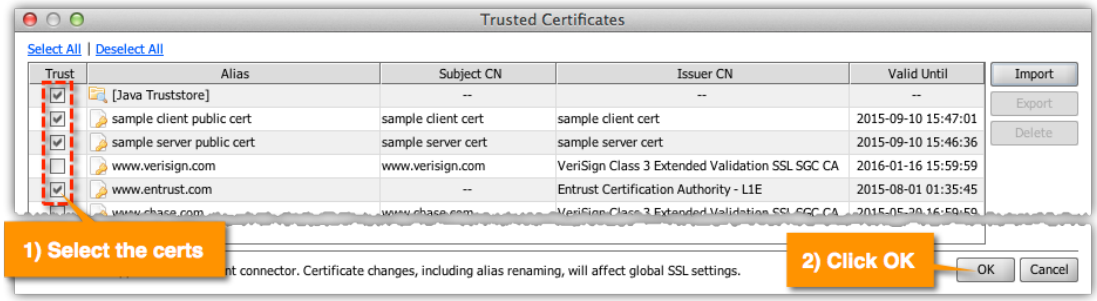


On the **SSL Settings** dialog, you select the required server cert by clicking the wrench next to **My Server Certificate**. (Optional settings include requesting/requiring client authentication, setting client cert validation, and setting the client cert trusted list.

A dialog appears that is identical in task options and appearance to the **My Certificates** table on the **Settings** page – **SSL Manager** page, except for the **Select** column (left end of the table) by which you select the desired server cert.



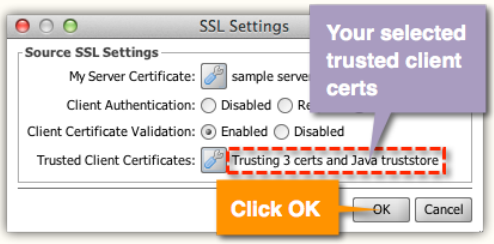
Refer to the **SSL Settings** dialog at the start of this section: To enable the **Client Authentication** option, click the **Request** or the **Require** button. Options are then enabled by which you can choose whether or not to validate client certs and select trusted client certs. To do the latter, click the Wrench icon next to **Trusted Client Certificates** to open the **Trusted Certificates** dialog, which is identical in task options and appearance to the **Public Certificates** table on the **Settings** – **SSL Manager** page, except for the **Trust** column (left end of the table) by which you select the desired server cert.



Check the boxes of the desired certs (or click the **Select All**/**Deselect All** links above the list), then click the **OK** button. If you check **[Java Truststore]**, all cacerts in the default Java Truststore will be trusted.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

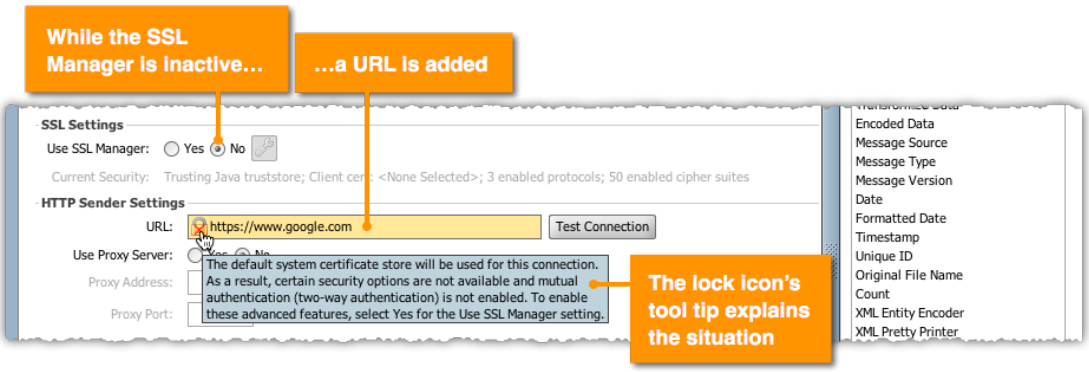
To view the default Java truststore contents, double-click the **[Java Truststore]** row. When you click the **OK** button, the **Trusted Certificates** dialog closes, and the **Trusted Client Certificates** section of the **SSL Settings** dialog reflects your selections.



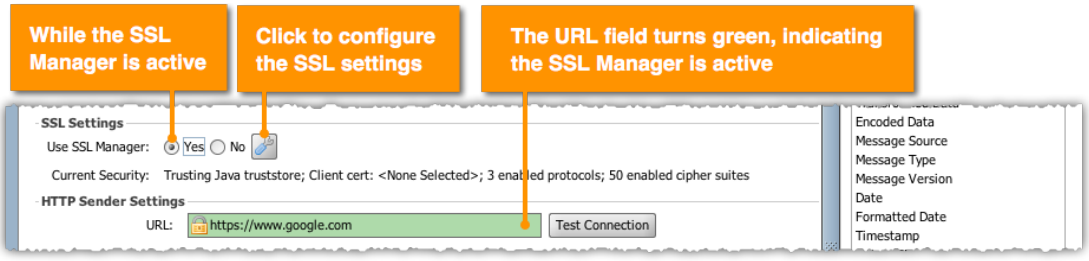
Save and deploy your channel. The channel's source connector will automatically use the server cert and will trust the client certs you set. You do not need to restart the Mirth Connect server.

Applying Destination Connector SSL Settings

On an **Edit Channel** > **Destinations** page, select a **Connector Type** that supports the SSL Manager (*HTTP Listener* , *Web Service Listener*, *File Reader* in FTP mode). In the **SSL Settings** section, the **SSL Manager** is inactive. If you populate the **URL** field in the **HTTP Settings** section, the field turns yellow with a Lock icon next to the URL. If you move the pointer over the Lock icon, a tool tip appears that explains the reason for the situation. (Clicking the Lock icon opens a dialog with the same content as the tool tip.)

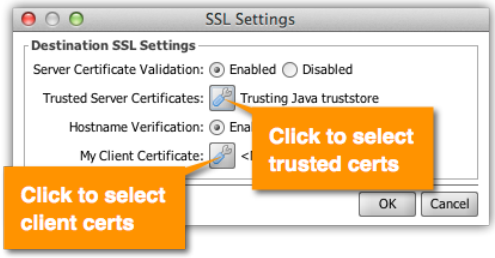


In the **SSL Settings** section – **Use SSL Manager** element, click *Yes* to nullify the locked condition and enable all security options and two-way authentication. (The **URL** field turns green to reflect the change.)

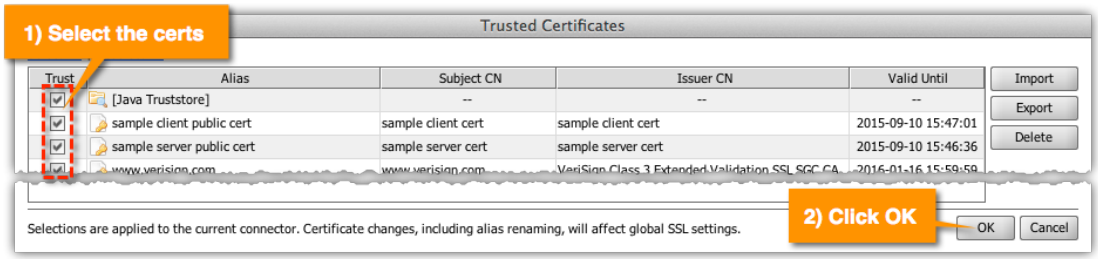


Click the Wrench icon in the **Use SSL Manager** element to open the **SSL Settings** dialog, on which you can configure the connector's SSL settings (Server cert validation, select trusted server certs, Hostname verification, and select My Client certs).

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

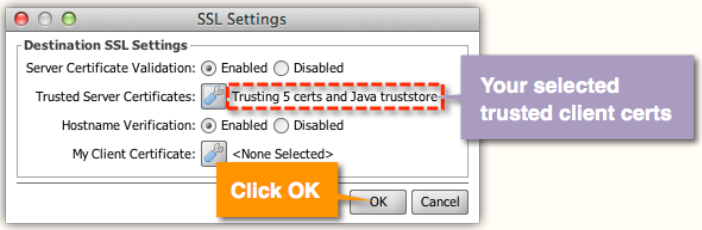


If **Server Certificate Validation** is disabled, the destination connector assumes that any cert it receives is automatically trusted. When Validation is enabled (previous graphic), you can select specific trusted certs by clicking the Wrench icon next to **Trusted Server Certificates**. A **Trusted Certificates** dialog appears that is identical in task options and appearance to the **Public Certificates** table on the **Settings** – **SSL Manager** page, except for the **Trust** column (left end of the table) by which you select the desired server cert.



Check the boxes of the desired certs (or click the **Select All**/**Deselect All** links above the list), then click the **OK** button. If you check **[Java Truststore]**, all cacerts in the default Java Truststore will be trusted.

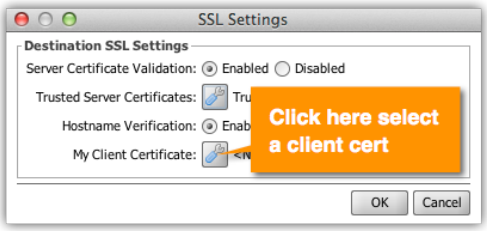
To view truststore contents, double-click the **[Java Truststore]** row (previous graphic). When you click the **OK** button, the **Trusted Certificates** dialog closes, and the **Trusted Server Certificates** section of the **SSL Settings** dialog reflects your selections.



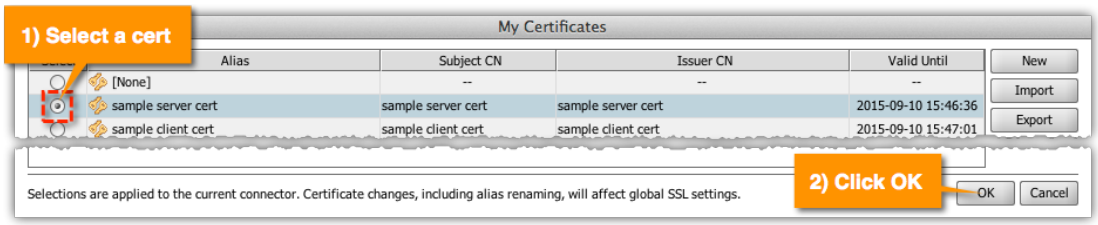
On the **SSL Settings** dialog, you can also enable/disable **Hostname Verification** (previous graphic.) When enabled, the requested hostname (in the request URL) is checked against the common name (CN) of the server cert (or any subject alternative names (SANs). If they do not match, messages will not send.

If the server to which you are connecting requires client (two-way) authentication, click the Wrench icon next to **My Client Certificate.**

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



A **My Certificate** dialog appears that is identical in task options and appearance to the **My Certificates** table on the **Settings** – **SSL Manager** page, except for the **Select** column (left end of the table) by which you select the desired client cert.



Select the desired cert, and click the **OK** button, and if you have finished configuring the destination's SSL settings, click the **OK** button on the **SSL Settings** dialog. In the **Channel Tasks** panel, click **Save Changes**, then deploy the channel. The channel's source connector automatically uses the client cert. You do not need to restart the Mirth Connect server.

Testing SSL Connections

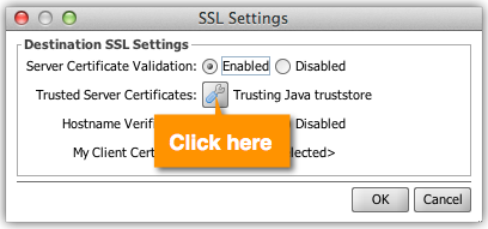
On the *HTTP Sender*, *Web Service Sender*, and *File Reader/Writer* (FTP mode), you can test SSL connectivity using the appropriate connector service. For example, for the *HTTP Sender* connector, on an **Edit Channel** – **Destinations** page – **HTTP Sender Settings** section, click the **Test Connection** button.

Several possible error messages could appear when you test the connection. First, if the server presented certificates that are not trusted by the connector, a **Warning** dialog appears.

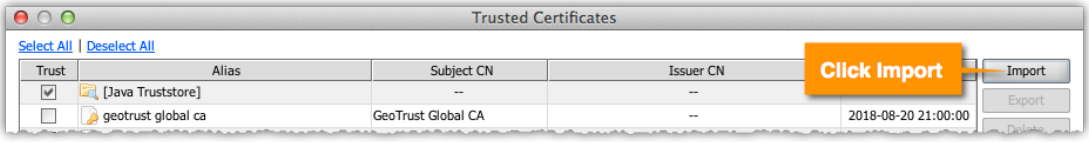
This **Warning** dialog instructs you that, to proceed, you need to add the cert in question to the list of trusted certs. To do this, on the **Edit Channel** page – **SSL Settings** section, click the Wrench icon.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

On the **SSL Settings** dialog – **Trusted Server Certificates**, click the Wrench icon.

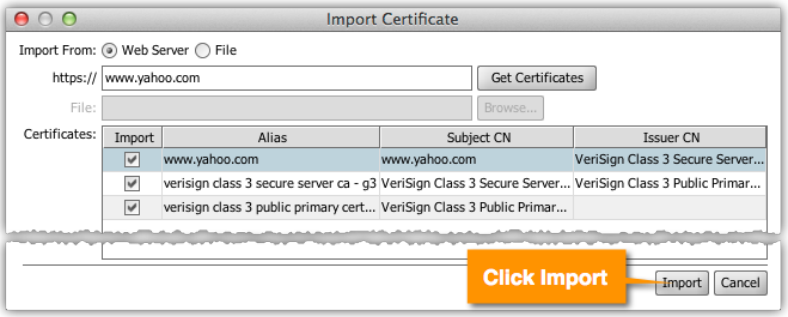


On the **Trusted Certificates** dialog, click the **Import** button.

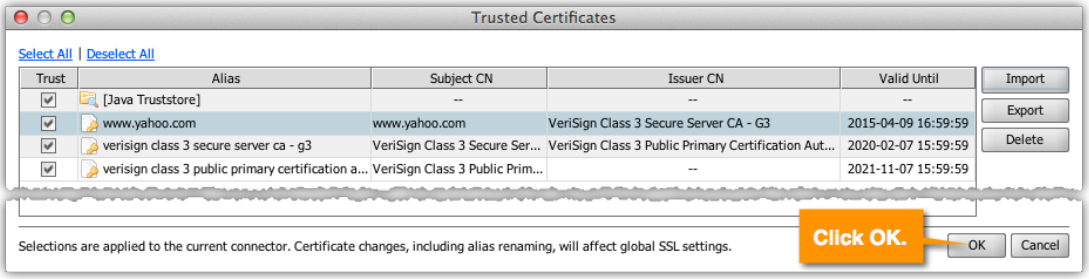


The **https://** field is pre-populated with the appropriate URL. Now start the import certs process as usual by clicking the **Get Certificates** button.

The certs appear in the **Certificates** list. Click the **Import** button to open the **Trusted Certificates** dialog.

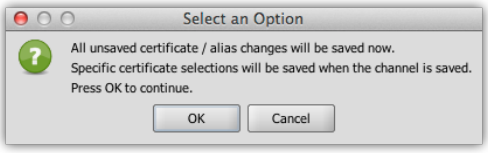


On the Trusted Certificates dialog, click the **OK** button.

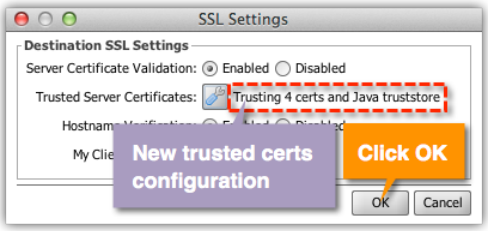


Because you have edited the configuration, a **Select an Option** dialog appears, informing you of certain information concerning your changes. Click the **OK** button.

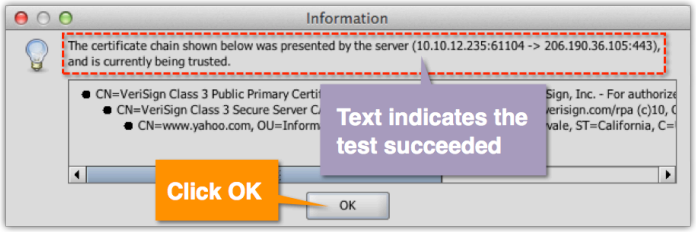
Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.



The **Trusted Certificates** and **Import Certificate** dialogs close to reveal the **SSL Settings** dialog with the newly trusted values for **Trusted Server Certificates**. Click the **OK** button.

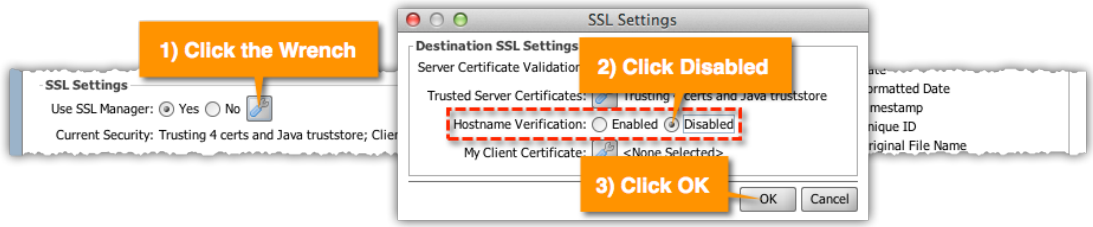


On the **Edit Channel** page – **Destination** page – **HTTP Sender Settings** section, click the **Test Connection** button. Information on the dialog indicates that the test succeeded. Click the **OK** button.



A different **Warning** dialog appears if the SSL connection test fails due to an invalid **hostname**.

Resolve this situation by correcting your **hostname** – or by contacting the server's manager and having them present a server certificate that includes the hostname you are using as a CN or SAN – or – by disabling hostname verification on the **SSL Settings** dialog.



Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

> Enabling hostname verification gives your SSL connection a higher level of security, which prevents certain types of "man-in-the-middle" attacks.

FTPS Connectors

File transfer protocol with SSL (FTPS) connectors are configured on an **Edit Channel** – **Source**/**Destination** page. For File Readers/Writers using the SSL Manager in FTP mode: Besides the regular SSL settings, the **Implicit FTPS** option is available. When you enable **Implicit FTPS**, the connector immediately initiates a secure SSL connection; otherwise, an unsecure connection is established first, which is then upgraded to a secure connection (e.g., via the AUTH command). Many FTP servers support both modes, but you should contact the manager of the FTP server to determine which mode you should use.

About the Serial Connector Plugin

This extension lets Mirth Connect send/receive data over serial communication ports (e.g., those compliant with the RS-232 standards). Any installed transmission mode (raw, MLLP, ASTM E1381, etc) can be used with the connectors. You can select the connection port, baud rate, parity, et al, to customize communications.

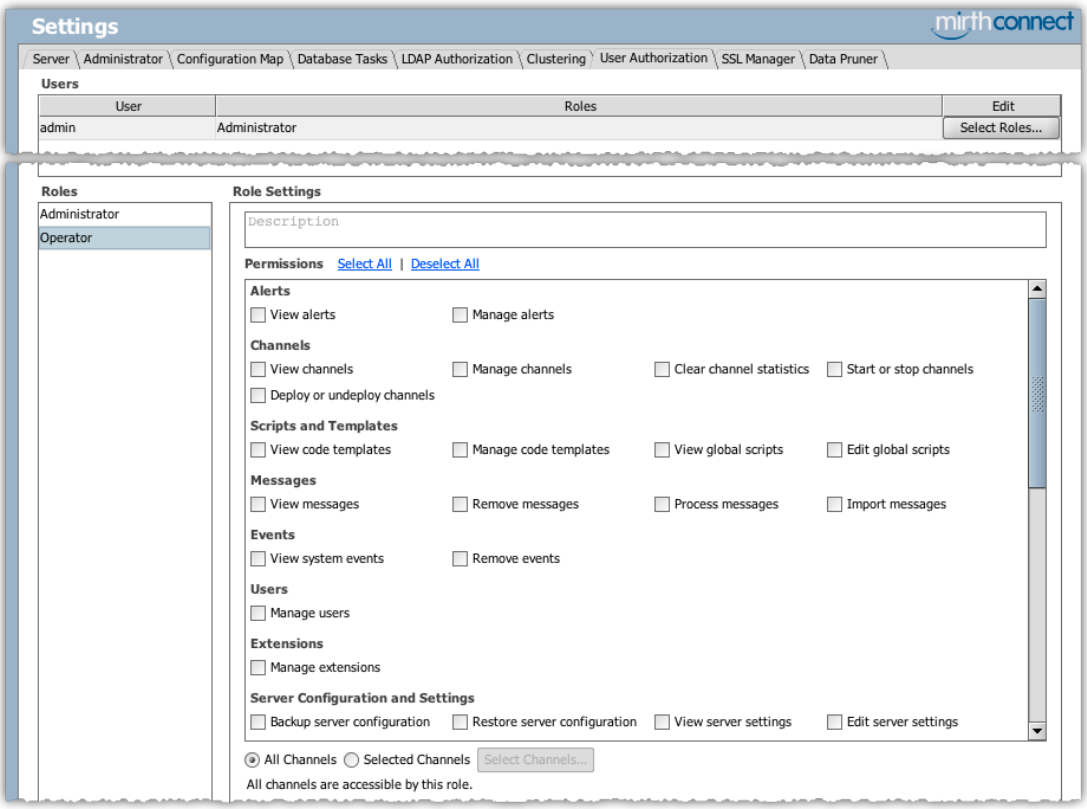
> The Serial Connector requires a Platinum-level support subscription. For technical and purchasing

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

information about the Serial Connector, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About the User Roles Plugin

This extension lets you manage user access to any aspect of the MC Administrator; that is, you can create a role and, under any of the many available categories, assign specific permissions relative to the category. For instance, under **Alerts**, you may grant a user permission to *View alerts* and/or *Manage alerts*. Under **Scripts and Templates**, you may grant a user permission to *View code templates*, *Edit global scripts*, et al.



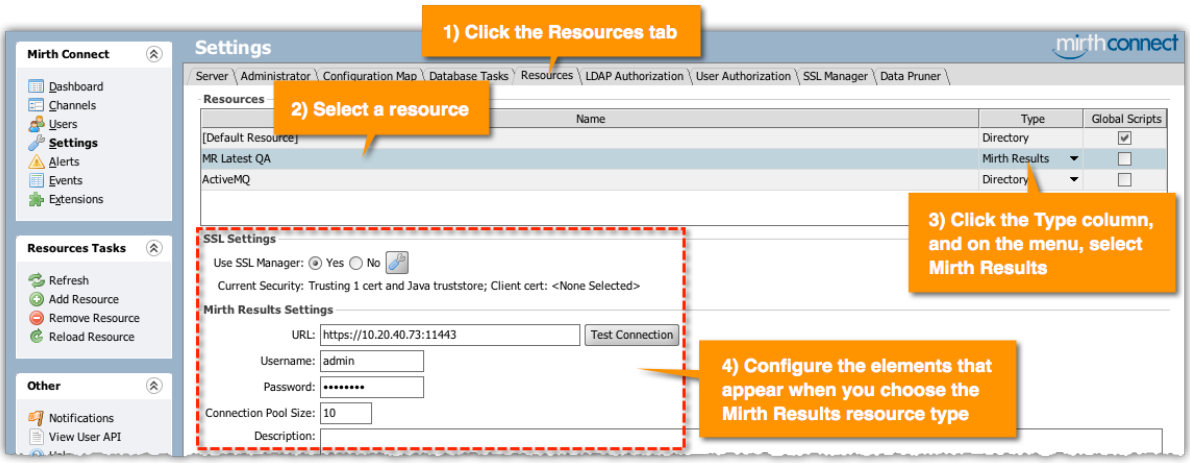
The User Roles plugin requires a Platinum-level support subscription. For technical and purchasing information about the User Roles plugin, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

About the Mirth Results Connector Plugin

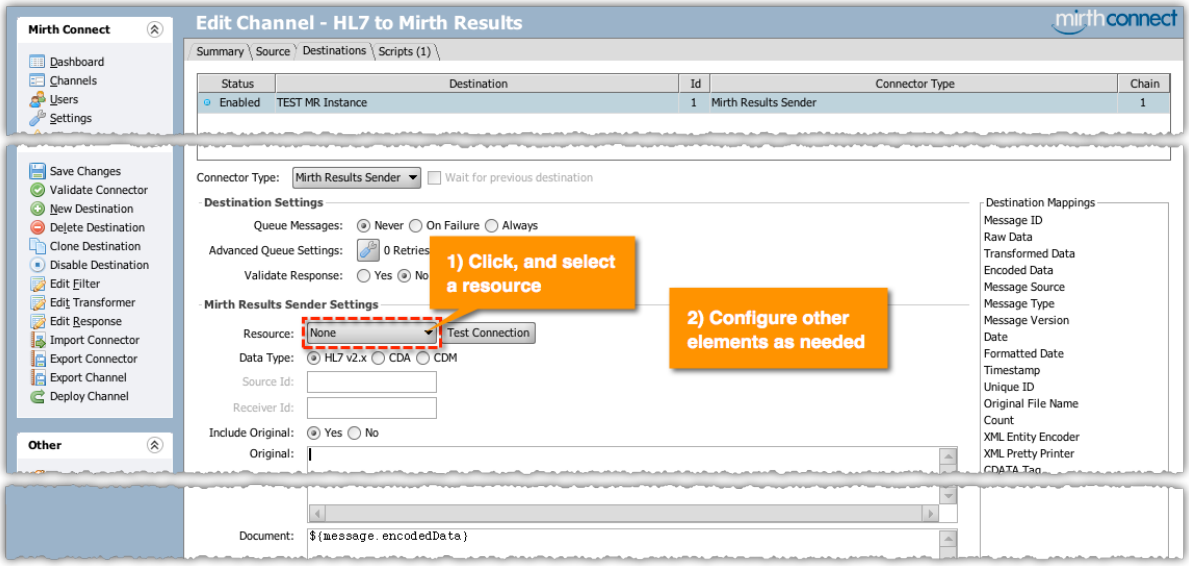
The Mirth Results connector plugin requires a Platinum-level support subscription. For technical and purchasing information about this plugin, contact the Mirth Corporation Sales Dept: (855) 289-6478 or sales@mirth.com.

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

This commercial plugin helps you manage Mirth Results resources and endpoints. With it, you can post clinical documents to a Mirth Results server or use the Mirth Results client (or libraries like CDAPI/HL7API) in JavaScript without having to manage connection pooling or libraries. Also, you do not need to restart the Mirth Connect server after adding a Mirth Results resource.



The Mirth Results resource type lets you configure a Mirth Results endpoint to use in the Mirth Connect server. You can post messages to that endpoint, and use its associated libraries in any channel/connector (ties in with the **Set Libraries** button on the channel's **Edit Channel** page > **Summary** tab).
The SSL Settings are basically the same as you would see for any other SSL-enabled connector. They are only visible/configurable when the SSL Manager is also installed.
The extension also provides the Mirth Results Sender destination connector. (Navigate to an **Edit Channel** page > **Destinations** page – **Connector Type** > *Mirth Results Sender*.)



In the **Mirth Results Sender Settings** section, the **Resource** drop-down menu is populated with the Mirth Results resources you configured on the **Settings** page > **Resources** tab. Click the **Resource** bar, and select a resource. Messages will be posted to that endpoint.

Mirth Connect FAQs

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

- What is Mirth Connect?
- Is Mirth Connect the same as "Mirth?"
- Who develops Mirth Connect?
- What is the Mirth Connect license, and how much does it cost?
- How can Mirth Connect be free and open source?
- Is there a difference between the free, open-source Mirth Connect download and the supported version of Mirth Connect?
- How does Mirth Connect compare to commercial integration engines?
- How many production installations of Mirth Connect exist?
- What can I expect next from Mirth Connect?
- Is there data that shows how fast Mirth Connect operates?
- Is Mirth Connect hard to install and configure?
- Do I need Mirth Appliance to run Mirth Connect?
- As a member of the Mirth community, how can I get more help?
- How do I become a Mirth Connect expert?
- What are the system requirements for Mirth Connect?
- Which databases does Mirth Connect support for its data store?
- Does Mirth Connect use the Mule ESB?
- Do I need an application server to run Mirth Connect?
- Can Mirth Connect send data to _____ or transform data from _____ to _____?
- What message standards does Mirth Connect support?
- What transfer protocols does Mirth Connect support?
- How do I transform a data segment?

What is Mirth Connect?

MC is an open-source, standards-based healthcare integration engine that speeds message routing, filtering, and transformation between health-info systems over various messaging protocols (e.g., HL7, X12, EDI, DICOM, XML).

Top

Is Mirth Connect the same as "Mirth?"

Yes. Mirth 1.0 was released in 2006 by WebReach, Inc. Based on the success of the Mirth application, WebReach, Inc. was renamed "Mirth Corporation" in 2009. To avoid confusion between the new company name and its products, the Mirth application was renamed *Mirth Connect*.

As of early 2016, Mirth Corporation, as a business entity, no longer exists, having been absorbed into the Quality Systems, Inc. family of products in Sept. 2013. QSI has since become NextGen Healthcare, and Mirth Connect lives on with other former Mirth Corporation products as part of the "Mirth Solutions" suite.

Top

Who develops Mirth Connect?

NextGen Healthcare develops and fully sponsors MC with the unofficial assistance of its users, who report bugs and contribute source-code patches, feature requests, and online support.

Top

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

What is the Mirth Connect license, and how much does it cost?

MC is released under the Open Source Initiative (OSI) approved MPL 1.1 (see Mozilla Public License 1.1.), and it costs you absolutely nothing! You can download MC free via the Mirth download page.

Top

How can Mirth Connect be free and open-source?

NextGen Healthcare, the primary developer of MC, backs it with commercial support, services, training, and appliances. Also, the contributions of thousands of users help keep MC a leading HIT (health-information technology) integration engine.

Top

Is there a difference between the free, open-source Mirth Connect download and the supported version of MC?

No; however, support subscriptions make available to you various commercial plug-ins and connectors as well as day-to-day improvements that are unavailable to non-support users until the next software release.

Top

How does Mirth Connect compare to commercial integration engines?

Based on user feedback, MC's features compare favorably to commercial integration engines and are usually superior in head-to-head comparisons. Despite being free and open-source, MC has NextGen Healthcare's full backing with support, training, and consulting services similar to those of commercially vended software. We are proud of our agile development process and frequent release cycle, which are possible because MC is open-source and community-driven.

Top

How many production installations of Mirth Connect are there?

NextGen Healthcare formally supports hundreds of specific production installations of MC, but because it is free and open-source, the overall number of production installations is unknown. We do know, however, that MC has been downloaded over 100,000 times, and the active online MC community has 30,000 members, so MC installations in production worldwide are undoubtedly in the thousands.

Top

What can I expect next from Mirth Connect?

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

MC is continually evolving, and each new version includes various enhancements and new features. For upgrade details, see the Mirth Connect Roadmap.

Top

How fast does Mirth Connect operate?

Performance varies widely depending on your hardware setup and channel configuration. Benchmarks are available for the hardware appliances that Mirth Corporation offers.

Top

Is Mirth Connect hard to install and configure?

No. In fact, we've heard from users that they installed MC and minutes later were processing messages. There are several installation methods, but the easiest is the cross-platform GUI (graphical user interface) installer that guides you through the process. If you prefer a more hands-on method, however, we offer a **zip/tar.gz** distribution.

Top

Do I need Mirth Appliance to run Mirth Connect?

No, but running MC on the Appliance platform provides such advantages as easy updates, added security and reliability through clustering, and a platform that can host the entire Mirth Solutions suite. To view an extensive list of features, see the Mirth Appliance page.

Top

As a member of the Mirth Solutions community, how can I get more help?

NextGen provides numerous support options if you have a specific problem or would like support for running MC for production. Sign up for support to access online training videos and monthly Q&As with MC developers.

Top

How do I become a Mirth Connect expert?

NextGen Healthcare provides extensive Mirth Connect training and a certification program.

Top

What are the system requirements for Mirth Connect?

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

You can run MC on any system that supports Java and requires the Sun/Oracle JRE (Java Runtime Environment) 7 or newer.

Top

Which databases does Mirth Connect support for its data store?

Apache Derby (default), PostgreSQL, MySQL, Oracle, and Microsoft SQL Server, but the Database Reader/Writer connectors can support any type of database if you add the right client libraries to the **custom** folder in the MC installation directory.

Top

Does Mirth Connect use the Mule ESB?

Not anymore. Prior to the release of Mirth Connect 3.0, Mule was eliminated from the application.

Top

Do I need an application server to run Mirth Connect?

No. MC runs as a stand-alone executable or service in its own JVM (Java virtual machine).

Top

Can Mirth Connect send data to _____ or transform data from _____ to _____?

Whichever variables fill these blanks, the answer is generally a resounding Yes! Even if a message standard (protocol) is foreign to MC, advanced Java and JavaScript capabilities are so flexible that almost any data type can be transformed and transferred.

Top

What message standards does Mirth Connect support?

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.

- HL7 v2.x
- HL7 v3
- XML
- NCPDP
- EDI/X12
- DICOM
- Delmited Text
- Raw
- ASTM E1394 (commercial)

Top

What transfer protocols does Mirth Connect support?

- MLLP
- TCP/IP
- HTTP
- Files
- Database
- S/FTP
- Email
- JMS
- Web Services
- PDF/RTF Documents
- Custom Java and JavaScript

Top

How do I transform a data segment?

Example: You have the data segment **20080624175854-0700**, and you want to eliminate the hyphen and the four digits after it. The easiest way to do this is to create a new **STEP** in the transformer with some JavaScript such as:

```
var originalValue =msg['PID']['PID.18']['PID.18.1'].toString(); //, which represents: 20080624175854-0700

var splittedValueArray = originalValue.split("-"); msg['PID']['PID.18']['PID.18.1'] = splittedValueArray[0]; // splittedValueArray[0], which represents: 20080624175854

// splittedValueArray[1], which represents: 0700
```

Use of the split function returns an array of the values split by the delimiter.

Top

Copyright ©2016 QSI Management, LLC. All Rights Reserved.
Private and Confidential - Do No Distribute.