Compliant Kubernetes Service documentation has moved
Please note: You are not reading Kubernetes documentation. If you're looking for Compliant Kubernetes Service documentation, it has moved. Read more here.

Navigating Mirth

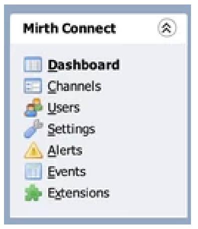
Rick Wattras - June 24, 2020 15:25

This section will guide you through how to navigate through the Mirth application user interface in order to provide the groundwork for building out your integration resources.

Overview of User Interface

General Navigation

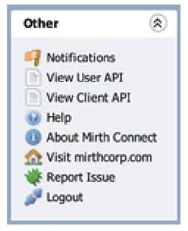
The "Mirth Connect" pane in the top left of the Mirth application is a static navigation list of available sections:



Clicking any of those items will take you to that particular section of the UI, for each of which we will be discussing in detail in the following sections.

The middle pane on the left is reserved for section-specific tasks and options, which we will also cover shortly.

The bottommost "Other" pane on the left is also static and provides links to additional resources:



- 'Notifications' will provide a popup window with the latest updates from the Mirthconnect developer team.
- 'View User API' will launch out to https://<Mirth IP>:8443/javadocs/user-api/ in a local browser and will open the documentation for the Mirth source code, including all built-in libraries and classes. This will be useful if you plan to do any custom development on top of Mirth, or just want to take advantage of some of the built-in libraries. Navigating through the docs will be familiar if you've ever gone through Java class documentation, et al.
- 'View Client API' will also launch out to a local browser, this time opening https://<Mirth IP>:8443/api/. Here you will have access to an overview of the Mirth API, which is a collection of HTTP-based calls that you can make external requests against for a variety of uses, as well as some built-in demo functionality for testing said calls. These can be useful for developing additional functionality on top of the Mirth UI, some examples of which are: running reports against channel

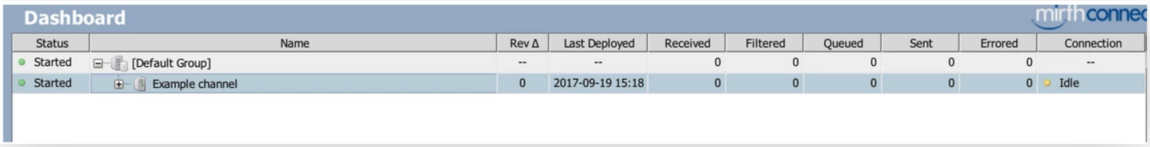
data, automating data exports, and deploying/undeploying channels remotely. For more information on using the API, see the "Using the built-in Mirth API" section of Advanced Mirth Functionality

- 'Help' will launch the Mirth community wiki here: http://www.mirthcorp.com/community/wiki/display/mirth/Home
- 'About Mirth Connect' will give you a popup with the version information of your Mirth client.
- 'Visit mirthcorp.com' is pretty self-explanatory as it launches www.mirth.com
- 'Report Issue' launches the http://www.mirthcorp.com/community/issues/ JIRA page for Mirth development and allows you to login (you'll need to make an account if you don't already have one) and report an issue or submit a bug fix request.
- 'Logout' will log you out of your Mirth session

Dashboard Section

Once you've launched and logged into Mirth, the first part of the interface that you'll be exposed to is the Dashboard. At first launch the main window on the page should be blank as you will have no active interfaces running, but once you're at a point where you're actively monitoring message traffic this will be the main hub of interface/message activity.

The main page will list the deployed channels and will show some info about it including if there are any pending (undeployed) changes ("Rev delta") and when the channel was last deployed. It also include the overall number of messages received, filtered, queued, sent, and errored for that channel:

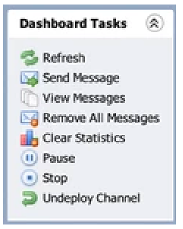


Dashboard

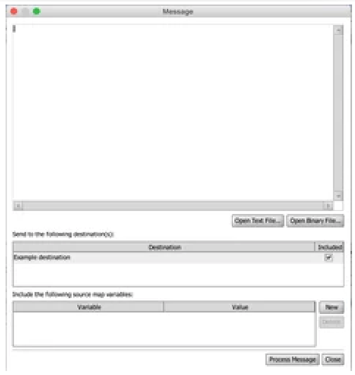
Status	Name	Rev Δ	Last Deployed	Received	Filtered	Queued	Sent	Errored	Connection
Started	[Default Group]	--	--	0	0	0	0	0	--
Started	Example channel	0	2017-09-19 15:18	0	0	0	0	0	Idle

The "Connection" column at the far right can also be useful in that it will show as a green "Connected" if there are active connections to that interface.

While on the main dashboard page you can select a specific channel and a set of options will appear in the task list on the left:



These are functions that can be performed on the selected interface via the options above:

- 'Refresh' will just refresh the entire pane, not just the selected channel. This can be useful if a deployed channel hasn't showed up yet or the message counters in the view need to be manually refreshed. This doesn't affect anything beyond the view of the dashboard, similar to refreshing a website
- 'Send Message' will give you a popup that will allow you to manually send a message to the selected channel. The popup window will look like this:
- 

Send to the following destination(s):

Destination	Included
Example destination	☑

Include the following source map variables:

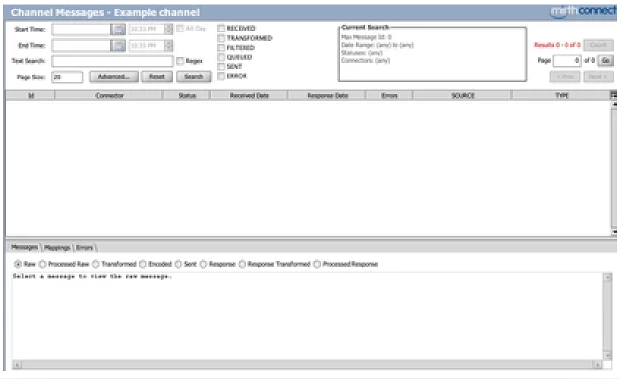
Variable	Value

  - The main top pane is where you can paste in or manually compose a message to send to the channel. Make sure it's in the format the channel is expected to consume (ex: valid HL7)

    - The second section shows the enabled destinations. More on this later but basically this is where you select what paths you'd like the message to take
    - The third section is reserved for manually populating "source variables" which are basically metadata that can be included in an incoming message when coming from another system vs. being manually sent. One example includes headers and parameters in an HTTP request

- 'View Messages' will take you down into the channel-specific dashboard, which we'll get to shortly
- 'Remove All Messages' is pretty self-explanatory as it will remove all messages for a specific channel. Be careful with this! At least if you do inadvertently click it, you will get a warning popup asking if you are sure you want to perform that action. Another safeguard is that by default you can only delete messages from a stopped channel, unless you enable a check box to delete messages from a running channel in the warning popup.
- 'Clear statistics' will allow you to reset all of the channel messages statistics (received, sent, etc) to 0. This prompts you with a second popup allowing you to select single columns to clear if necessary. This will not remove messages, just the counters.
- 'Pause' will pause a channel, which will cause it to stop processing or receiving new messages although it will continue sending any messages that it is currently in the processing of sending, as well as any messages queued up on the destinations
- 'Stop' will stop a channel, which will cause it to stop processing or receiving messages - although it won't fully stop until any currently in-process message is complete. Unlike 'Pause', it will not continue to process messages in the queue
    - Sometimes you need to forcibly stop a channel in the middle of processing. If you click Stop a second time, it will change to a 'Halt' button, which if you press will forcibly stop a channel from processing. This is useful if a message causes an infinite loop or if you just need to stop it before it finishes a certain message. Once the channel is fully stopped, you can remove the offending message (if necessary) in order for it to not begin processing it again once it is started back up
- 'Undeploy Channel' will undeploy a channel, so that it will no longer be active for receiving or processing messages

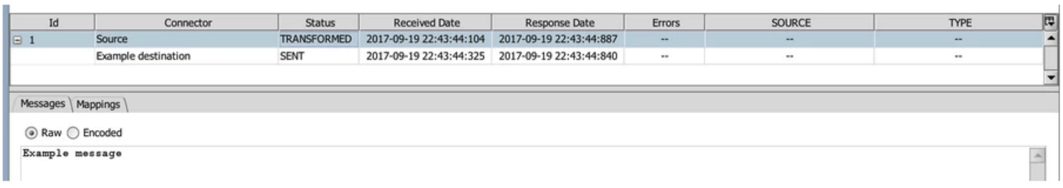
If you click 'View Messages' while a channel is selected - or if you just double-click a channel - you'll be taken to the Channel Messages view where all of the messages sent to that channel can be viewed.



Current Search
Max Message Id: 0
Date Range: (any) to (any)
Statuses: (any)
Connectors: (any)

Select a message to view the raw message.

There is a lot going on here, but we'll break it down by each piece of functionality available on the page:

- The large middle pane is where messages will be listed when there are some that have been sent to the channel
- At the bottom is the message-specific details. When a message is selected, you'll see information about that message populated in the three tabs and their respective info panes
    - 

Id	Connector	Status	Received Date	Response Date	Errors	SOURCE	TYPE
⊟ 1	Source	TRANSFORMED	2017-09-19 22:43:44:104	2017-09-19 22:43:44:887	--	--	--
	Example destination	SENT	2017-09-19 22:43:44:325	2017-09-19 22:43:44:840	--	--	--

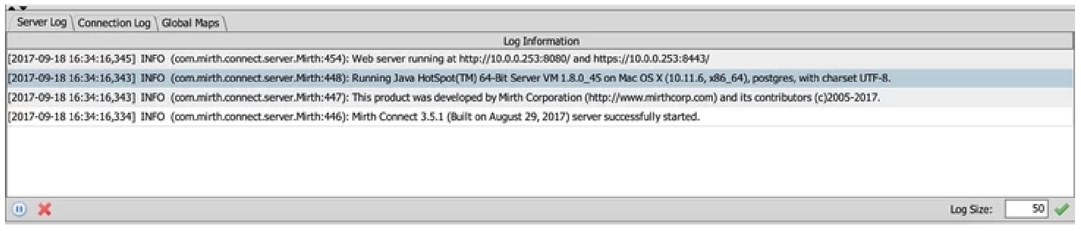
```
Example message
```

    - Here we see a received message is in the list and is selected. In the main pane you can also see the dates/times for when it was received and processed. The lower pane shows its raw content by default, but clicking the Mappings tab will show metadata as well as any mapped variables that were used during processing - more on that later. If an error would have occurred in processing, a third "Errors" tab would appear and detail the nature of the failure.

- The top left is where all of the search functionality exists to allow you to search through all messages sent to the channel. You can search by date/time range, by text content of a message, and by message status ("SENT", "ERROR", etc). There also exists some "Advanced" search functionality, which is covered later in the "Advanced message search" section of Advanced Mirth Functionality
    - Note that this can take a long time, and even time out sometimes, if you attempt to search over an extremely large range of messages. Limiting the parameters of the search may allow it to run faster. It's basically performing a SQL query against the backend database behind the scenes, so use your judgment if you're familiar with how long some broad queries can take to run.
- The message counter in the top right will allow you to receive a count of all messages in the current search; by default it will count all messages sent to the channel.
    - Note that similar to above that this has the potential to take a really long time, for the same reasons.

Log Information

At the bottom of the dashboard is a partial view into the Mirth log:



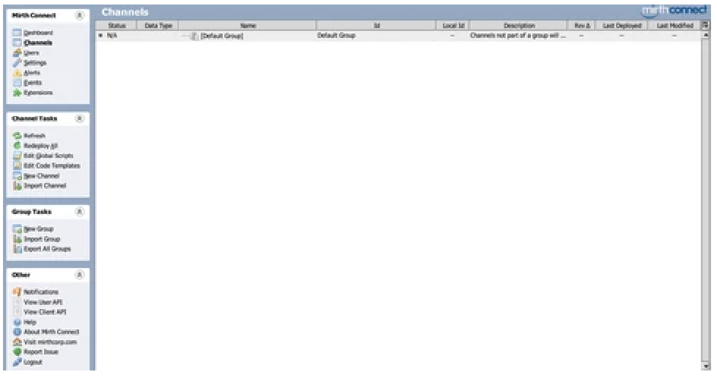
```
[2017-09-18 16:34:16,345] INFO  (com.mirth.connect.server.Mirth:454): Web server running at http://10.0.0.253:8080/ and https://10.0.0.253:8443/
[2017-09-18 16:34:16,343] INFO  (com.mirth.connect.server.Mirth:448): Running Java HotSpot(TM) 64-Bit Server VM 1.8.0_45 on Mac OS X (10.11.6, x86_64), postgres, with charset UTF-8.
[2017-09-18 16:34:16,343] INFO  (com.mirth.connect.server.Mirth:447): This product was developed by Mirth Corporation (http://www.mirthcorp.com) and its contributors (c)2005-2017.
[2017-09-18 16:34:16,334] INFO  (com.mirth.connect.server.Mirth:446): Mirth Connect 3.5.1 (Built on August 29, 2017) server successfully started.
```

Here all events are logged in a dashboard view, where double-clicking an entry allows you to pull up the entirety of the entry (say, if the text is too long to fit the window). This log will only show the last 50-200 (the value in the Log Size box in the bottom right corner can be changed, but is limited to 200) entries, with the full log being stored on the Mirth host. If you need access to the full log, please reach out to Datica for assistance or request that they connect your Mirth logs to your Kibana logging instance.

The first tab is the Server Log, which list entries as explained above. The second tab is the "Connection Log", which - when a particular channel is selected - will show you the connection history for that channel, which can be useful when determining if there were any connectivity issues for a connected entity. The transmission of messages is also logged in the Connection Log. The third tab is for displaying Global Map variables, which are covered in more detail in later chapters.

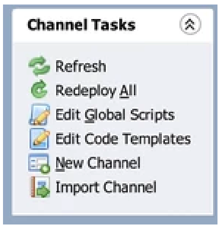
Channel Section

The second link in the top left Navigation pane will take you to the Channels section, which is where all of the interface creation and configuration happens. Similar to the Dashboard page, here you'll find a list of all channels - active or inactive, deployed or undeployed - for your Mirth instance.



Status	Data Type	Name	Id	Local Id	Description	Rev Δ	Last Deployed	Last Modified
N/A		[Default Group]	Default Group	--	Channels not part of a group will ...	--	--	--

At first you should just see a blank screen like the one above, but eventually the large pane on the right will host the channel list where you'll create, delete, and modify channels. There are now two middle panes on the left, one for Channel Tasks and one for Group Tasks:



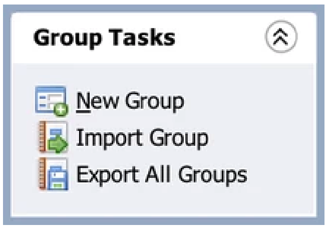
- "Refresh" = similar to on the Dashboard, this simply refreshes the screen, similar to refreshing a webpage
- "Redeploy All" = this will deploy (redeploy if they're already deployed) all non-diabled channels
- "Edit Global Scripts" = this will take you to the Global Scripts configuration screen where you can create scripts that will run for all channels when a certain action (Deploy, Undeploy, Preprocessor, and Postprocessor) occurs for any channel. This is useful if you need some functionality to be performed across all channels and corresponding to either each time the channel is Deployed/Undeployed, or before (Preprocessor) or after (Postprocessor) any messages are processed. More info on using Global Scripts is in the section on "Scripts" in Mirth Channel Overview
- "Edit Code Templates" = similar to Global Scripts, this will take you to a configuration section where you can create code libraries files with code templates that act as functions that can be called by any channel with access to the library. These work like *functions* do when programming - they're useful if you find yourself calling the same code several times in the same program and want to make it more efficient and easier to manage. More info on using these can be found in Utilizing Datica Open Source Resources
- "New Channel" = this will create a new channel, and if you currently have a group selected it will be assigned to that group (otherwise it will be put in the "Default Group" channel group). Clicking this immediately takes you to the channel configuration screen - this is where you will configure and build an interface for its desired use. We'll go into great detail on all of the channel build steps in a separate section of the guide: Mirth Channel Overview
- "Import Channel" = clicking this will give you a popup that will allow you to browse to any local XML files containing exported Mirth channels that have already been built elsewhere. This allows you to import a fully-functional channel right out of the box! If you're going to start out by using the Datica repository of Mirth templates, you'll use this a lot. There are more details on how to do this in another section: Utilizing Datica Open Source Resources
  - One note about this is that there is some version clash with importing channels: you cannot import a channel that was built on and exported from a newer version of Mirth than what you're currently using. You *can* however import channels that were built on and exported an older version of Mirth - you will likely just be prompted whether you'd like for Mirth to automatically optimize the channel for the current version you are running on (selecting Yes is the only option to proceed with importing, however the optimization usually doesn't affect any of the content of the channel itself).
  - Another note is that if the exported channel you're attempting to import used certain code templates that were included upon export (see the "Export Channel" item below), you will get a popup asking if you'd like to import the code templates as well as the channel. Usually the channel is dependent upon these templates to run, however that's not always the case. Use your judgment when importing Mirth channels without knowing much about the associated code templates. For the Datica OCI repository, none of our main template channels include code templates so as to not cause confusion, but we do include exported code templates separately with some useful functionality if you'd like to take advantage of those! The templates are available here: https://github.com/daticahealth/Mirth-Transforms/tree/master/channel_templates. The popup will allow you to say "No" and just import the channel, or to select only certain templates in the list to import. Simply check/uncheck the boxes to choose which ones you'd like.

When a channel is selected a couple more options appear, such as:

- "Deploy Channel" = this will deploy the currently-selected channel, but only if it's not disabled. If it is disabled, you'll get a popup saying that the channel cannot be deployed
- "Delete Channel" = delete the currently selected channel; you will get a warning popup before it is deleted, so don't worry if you click this by accident! Head's up that once a channel is deleted it cannot be recovered, so if you want to be safe it's a good idea to export the channel to somewhere safe beforehand just in case
- "Export Channel" = this will export the selected channel into an XML file that can be imported later into a separate (or the same) Mirth instance. If there are code templates in use by the channel (or if the channel is attached to a code template library, whether it's using the code templates within or not) you will be prompted whether or not you'd like to include the associated code template library within the exported channel. This is up to you: if the templates are crucial to the channel, it's a good idea to include them in the export.

Also when a channel is selected, another setting shows up in the Group Tasks:

- "Assign To Group" = assign the currently selected channel to a group. A popup list of available groups will allow you to select any group that's already been created.



- "New Group" = This creates a new channel Group that can allow you to organize channels in a way that's easier than thumbing through a large list
- "Import Group" = pretty self-explanatory: you can import exported groups similar to how the Channel import/export works
- "Export All Groups" = export all channels groups to individual group XML files

Similar to the channel options, when a Group is selected a couple more options become available:

- "Export Group" = export the currently selected group to an XML file
- "Delete Group" = delete the current group. This will *not* actually prompt you if you are sure, but it also doesn't remove the channels inside the group so there's not much harm done if you do it by accident. The channels will appear in the Default Group and can be assigned back from there if necessary.

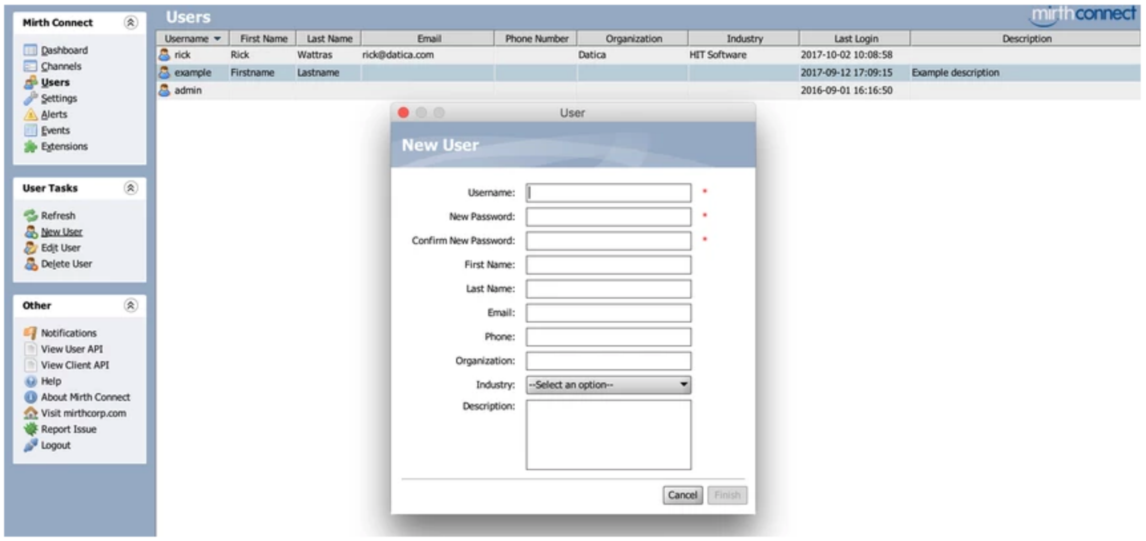
One note on Groups: any time you make a change to a group (such as by creating it, re-naming it, or assigning channels to it) you must first save the group (the "Save Group Changes" button will appear if there are unsaved changes) before you move onto any other task in Mirth. Attempting to perform another task (such as going to the dashboard or editing a channel) will cause a popup to prompt you to save the group first.

When viewing the channel list, you can select multiple channels at once by holding the Shift key while clicking. Once multiples are selected, you can deploy them all at once or enable/disable/delete them as necessary.

Users Section

The third section linked from the top left pane of Mirth is the Users section, and it's here that you can create/remove/modify user accounts that have access to the current Mirth instance. Everything here is a bit straightforward, as far as the options available and what's displayed in the main window pane. The main window is the list of user accounts with some additional (optional) identifying information such as full name, email, and phone number. The "Last Login" column can also be useful as well. However, when creating a new user account the only required information is the username and password that the user will login to Mirth with.

With the Mirth OCI process, if this is the first time you're launching Mirth you will be given credentials to first login with and will be prompted to change your password (you can also change your username as well) and are advised to do so. If you create an account for someone else, they will be similarly prompted when they first login as well. Be advised that if you create an account for someone, they will still need to go through the VPN connection process in order to have enough access to launch Mirth. So they will still need to be added to the VPN access configuration and will have to go through the steps to connect from VPN client on their local machine.



Username	First Name	Last Name	Email	Phone Number	Organization	Industry	Last Login	Description
rick	Rick	Wattras	rick@datica.com		Datica	HIT Software	2017-10-02 10:08:58	
example	Firstname	Lastname					2017-09-12 17:09:15	Example description
admin							2016-09-01 16:16:50	

Username: *
New Password: *
Confirm New Password: *
First Name:
Last Name:
Email:
Phone:
Organization:
Industry: --Select an option--
Description:

Also, you will see that there is an *admin* user listed in the user list, even upon your first launch of Mirth. Please do not alter or remove this account as this will allow Datica technical support to access Mirth in the case that it is requested or required for maintenance purposes. In cases where Datica is not involved in your day-to-day build and integration work, Datica will not access your Mirth instance or make any changes unless specifically requested, or - in the case of maintenance - you are given notice ahead of time.

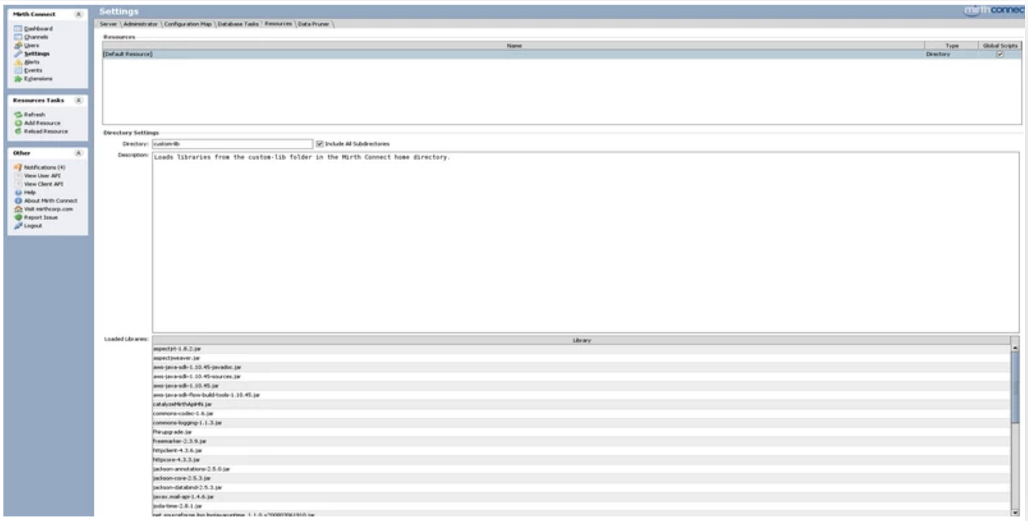
The one downside to the free version of Mirthconnect that we utilize for the OCI functionality is that there are no user-level permissions that would allow you to set limitations on what certain users can do within Mirth (for example, read-only access so that they can't make any changes). So be aware that if a user has an account and access to Mirth, they have unlimited access to all messages on all channels and can modify/deploy/create/delete channels. You can monitor user access and related activity in the Events section (more info below), but that won't track details around specific changes implemented by users. Another option could be to have a separate development Mirth instance that you allow developers to work on and have unlimited access to, but then not give them accounts on the Production instance. There are pros and cons to the various options, and there is probably development work that can be done to implement some amount of user-level permissions but nothing out of the box. If you have concerns around this or would like to discuss potential options, please reach out to Datica.

Settings Section

The next section that we'll cover is the Settings section, which can provide you with a lot of customization if you want it but typically the default settings are good enough that you shouldn't need to do much in here. We'll cover a couple of the most useful options but for more details information, seek out the Mirthconnect User Guide, or hover your mouse over each of the options for a description of the setting.

- "Server" tab:
  - Server Name = Feel free to set this to assign a name to this Mirth instance. This doesn't actually change the name of the host server that Mirth is running on, but does populate the window header and allows you (and anyone else who launches the Mirth instance) to know which instance you're currently using - which is useful if you use multiple instances
  - Global map settings = This one is only useful if you utilize global map variables (variables that are available to all channels within the Mirth instance). If you do use these, you might consider allowing these to persist beyond reboot so that if you're saving a value it's still available after the instance has been restarted
  - Email settings = If you intend to use the built-in email functionality (used for things like error alerting, etc), and also have an SMTP host available, then here is where you can setup your SMTP connection information. If you don't have an SMTP server, don't worry - we have built an error alerting solution that we cover in a separate chapter here: Error Alerting and Troubleshooting
- "Resources" tab:
  - Here you can view all of the Java libraries currently loaded into Mirth. By default, Mirth contains several of the base Java and Apache libraries, among others. If you intend to use a custom Jaca library, it must be loaded onto the Mirth hosts into the *custom-lib* directory. Just contact Datica and provide them with the JAR you wish you use and they can drop it onto the host. Once it is in place, simply click the "Reload Resource" button and you should see your .jar in the list.

- You can also create additional 'Resources' that look at different directories on the host - if you'd like to separate them out beyond the default directory or for organizing different libraries. Contact Datica to have them create the desired directories, then use "Add Resource" and enter the directory name in the appropriate box



Resources

Name	Type	Global Scripts
[Default Resource]	Directory	☑

Directory Settings

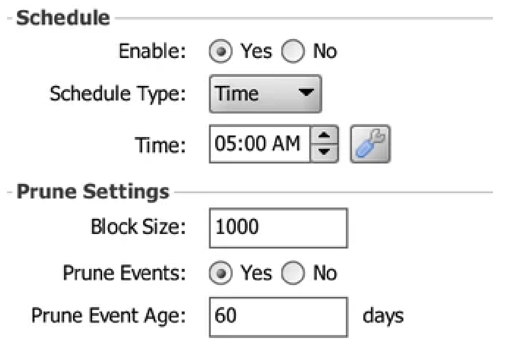
Directory: custom-lib ☑ Include All Subdirectories

Description: Loads libraries from the custom-lib folder in the Mirth Connect home directory.

Loaded Libraries:

Library
aspectjrt-1.8.2.jar
aspectjweaver.jar
aws-java-sdk-1.10.45-javadoc.jar
aws-java-sdk-1.10.45-sources.jar
aws-java-sdk-1.10.45.jar
aws-java-sdk-flow-build-tools-1.10.45.jar
catalyzeMirthApiHl.jar
commons-codec-1.6.jar
commons-logging-1.1.3.jar
fhirupgrade.jar
freemarker-2.3.9.jar
httpclient-4.3.6.jar
httpcore-4.3.3.jar
jackson-annotations-2.5.0.jar
jackson-core-2.5.3.jar
jackson-databind-2.5.3.jar
javax.mail-api-1.4.6.jar
joda-time-2.8.1.jar
net.sourceforge.lpg.lpgjavaruntime_1.1.0.v200803061910.jar

- "Data Pruner" tab:
  - This configuration is where you can enable and configure the pruning functionality that removes messages from the database after a certain amount of time. The actual settings for how long to keep messages around for (and whether to prune them at all) are set in the channels themselves, but here is where you actually turn on pruning and set the schedule. Pruning basically runs a form of a postgresql DELETE query against messages that fit a date range criteria (ie. "Older than 14 days). It's not a typically CPU-intensive process, but it's usually recommended to set the schedule to run daily and in the middle of the night. Remember that the Mirth instance may be running on server time (UTC) so check the time at the bottom of the Mirth GUI and set your pruning time appropriately! It's also useful to prune events after a certain amount of time, but that's up to you on how long you'd like to keep those around (they don't take up that much space so it can be pretty long and not be an issue).



Schedule

Enable: ● Yes ○ No

Schedule Type: Time

Time: 05:00 AM

Prune Settings

Block Size: 1000

Prune Events: ● Yes ○ No

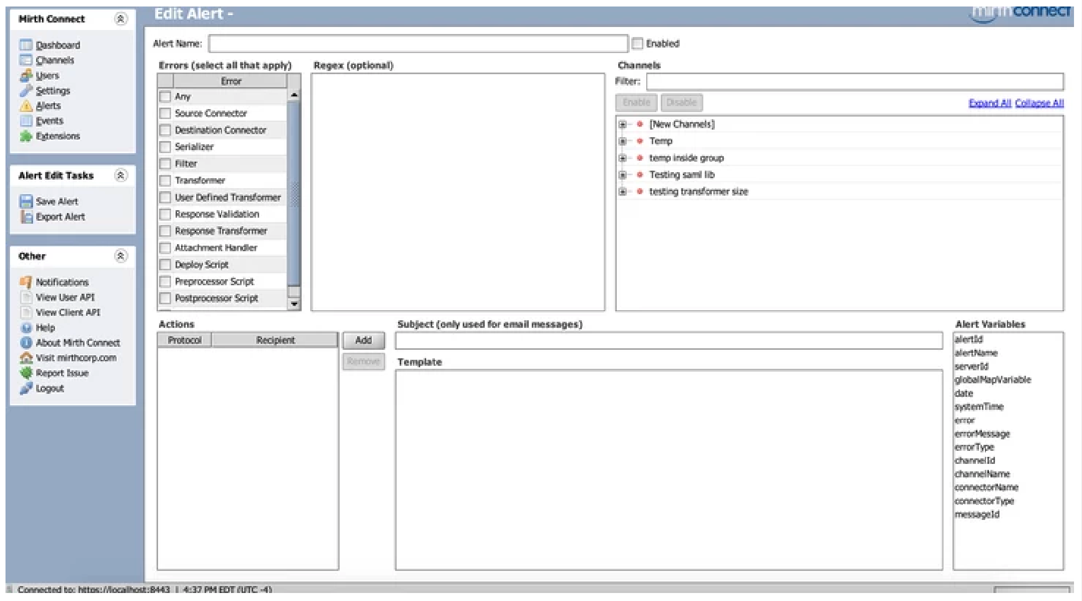
Prune Event Age: 60 days

  - The default Block Size of 1000 is usually enough, but if you're noticing that the pruning is taking a very long time to run then you can increase it (max value is 10000). Be aware that this will increase the amount of memory being used in exchange for the faster run time.

Alerts Section

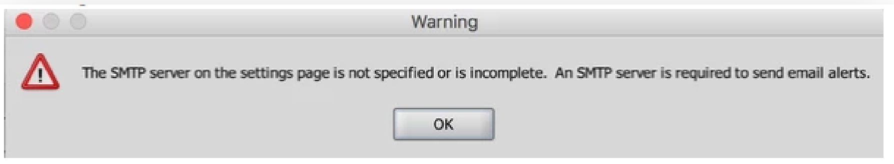
The Alerts section is where you can setup various error alerting functionality to notify you or a dedicated service when an error occurs. You can create as many Alerts as you want, and organize them in whichever way is most effective for you as each Alert can monitor several channels at once or just one channel at a time. If you decide to build one general Alert that covers all of your channels, then you can easily enable/disable alerts at either the base level (all) or per channel. However you're limited in the functionality that sends out the notification as they'd all share the same trigger. If you break them up to have a narrower monitoring focus, you can get more granular around notification functionality, but it can be more difficult to track which Alerts are watching which channels.

At first use, the main pane of the section will be blank as no alerts have been built yet. To Build a new Alert, click the button on the left and you'll be taken to the Edit Alert page, where you can configure what kind of errors to track, what channels to watch, and what action to take:



- "Alert Name" = Name the alert so you can recognize it in the Alerts list on the main section page. It's especially important to be specific if you have several alerts.
- "Errors" pane = Here you can select which types of errors you'd like to alert on. Typically it's easiest to just select "All" to cover the bases, but you do have the option to be granular about certain types if you should so choose
- "Regex" pane = If you only care about errors that match a regular expression, you can configure that here. Again, it's typical to just leave this blank and allow them all to come through so you don't miss anything
- "Channels pane = Enable/Disable which channels you want the Alert to monitor. You can even click the + next to the channel name and enable/disable at the source and destination levels.
- "Actions" pane = Click the "Add" button and you'll see an item added to the list. By default it will default to a Protocol of "Email", and if you did the SMTP setup earlier you can type an email address into the "Recipient" box to have them receive an email notification whenever an error occurs. If you don't have the SMTP configuration setup, you can press the down error next to Email" and choose either "Channel" or "User". "User" will still attempt to send an email by using the email address attached the the selected user's Mirth account, but Channel will send the notification content (set in the Template section, which we'll describe next) to a separate Mirth channel where you could build a custom handler (more on that here: Error Alerting and Troubleshooting).
- "Template" pane = Here you set the content of the notification message. It can be template-ized by utilizing some built-in variables that act as quasi-"environment variables" that will be replaced with the actual relevant content when the Alert processes an error. These variables are listed in the "Alert Variables" pane to the right, and can be dragged and dropped into the Template box for building your template message there.
  - An example template might be configured something like:
    - Template = There is Mirth error on channel ${channelName} at date/time ${date}. Error content: ${error}
      - Translation after processing = "There is a Mirth error on channel HealthSystem1 ADT at date/time Oct 3, 2017 1:53:45 PM. Error content: Read Timed Out

When you're finished configuring the Alert, just click the "Enabled" checkbox at the top and you're all set! If you don't have the SMTP configuration setup in the Settings, then you might see the following error:



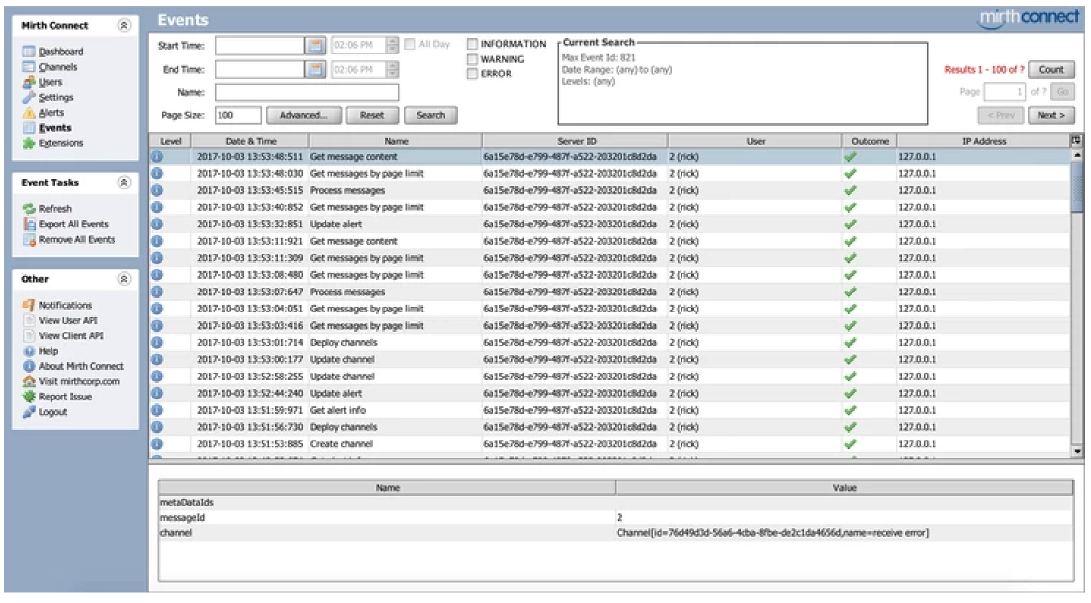
The SMTP server on the settings page is not specified or is incomplete. An SMTP server is required to send email alerts.

However, you can just click "OK" and the Alert will still be saved appropriately.

Once you've built an Alert you'll see it in the list in the Alerts section. On that main page, you can interact with the alerts similar to how you would a channel: you can create/modify/delete and even import/export Alerts. You can also Disable/Enable alerts right from the main page of the section.

Events Section

All of the user actions and related logging are stored in the Events section of Mirth, which contains a searchable list of events similar to a standard Event Log. Events are sorted by time with the most recent at the top, and some of the useful information available for each event includes the time, name of the action/event (ex. "Deploy channels"), and the user who performed the event. Similar to searching through messages, the event search functionality allows you to set a start/stop time window as well as to search for a specific action.



Start Time: 02:06 PM □ All Day
End Time: 02:06 PM
Name:
Page Size: 100

□ INFORMATION
□ WARNING
□ ERROR

Current Search
Max Event Id: 821
Date Range: (any) to (any)
Levels: (any)

Results 1 - 100 of ?
Page 1 of ?

Level	Date & Time	Name	Server ID	User	Outcome	IP Address
ℹ	2017-10-03 13:53:48:511	Get message content	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:48:030	Get messages by page limit	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:45:515	Process messages	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:40:852	Get messages by page limit	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:32:851	Update alert	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:11:921	Get message content	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:11:309	Get messages by page limit	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:08:480	Get messages by page limit	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:07:647	Process messages	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:04:051	Get messages by page limit	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:03:416	Get messages by page limit	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:01:714	Deploy channels	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:53:00:177	Update channel	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:52:58:255	Update channel	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:52:44:240	Update alert	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:51:59:971	Get alert info	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:51:56:730	Deploy channels	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1
ℹ	2017-10-03 13:51:53:885	Create channel	6a15e78d-e799-487f-a522-203201c8d2da	2 (rick)	✓	127.0.0.1

Name	Value
metaDataIds	
messageId	2
channel	Channel[id=76d49d3d-56a6-4cba-8fbe-de2c1da4656d,name=receive error]

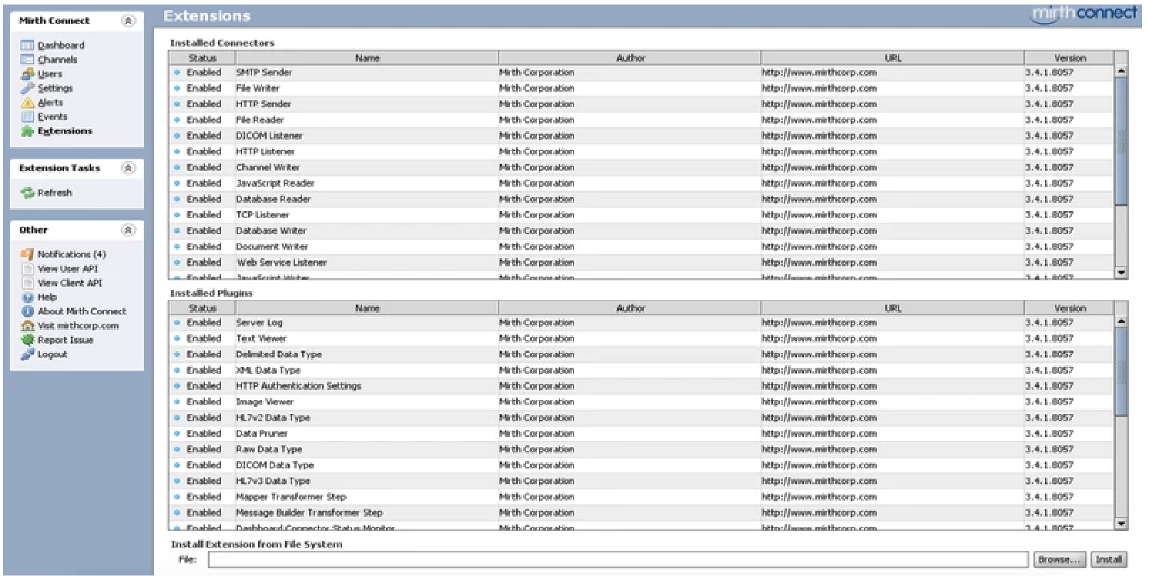
Selecting an entry will give you some additional relevant details, such as the channel ID and name if the event is related to a channel, or information about the filters if a user performed a message search.

The one thing that the Events won't tell you are what specific changes were made to a channel if the event is "Update channel" or "Deploy channel". So in that regard it won't act as a versioning log, or tell you what changed about a channel configuration, but it will tell you what user performed the action to what channel and at what time. That information alone might be enough if you're just looking to audit a particular change. Unfortunately the free version of Mirthconnect doesn't come with channel versioning and change control tracking, so if you are looking for that kind of functionality you'll have to research the licensed version of Mirth or create some external versioning workflow - one example might be exporting channels to a git repository.

That's really all there is to the Events section, as far as relevant functionality goes! Feel free to check out the official Mirthconnect user guide or click around yourself for any additional information.

Extensions Section

The last section is actually one that rarely needs to be touched, but does allow you to take advantage of additional functionality not available out of the box. The Extensions section is where all of the available Connectors (ex. "HTTP Sender") and Plugins (ex. "Image Viewer") are listed and can be disabled/uninstalled or installed if you have one on your local file system that you'd like to install.



Installed Connectors

Status	Name	Author	URL	Version
Enabled	SMTP Sender	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	File Writer	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	HTTP Sender	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	File Reader	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	DICOM Listener	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	HTTP Listener	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Channel Writer	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	JavaScript Reader	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Database Reader	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	TCP Listener	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Database Writer	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Document Writer	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Web Service Listener	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	JavaScript Writer	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057

Installed Plugins

Status	Name	Author	URL	Version
Enabled	Server Log	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Text Viewer	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Delimited Data Type	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	XML Data Type	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	HTTP Authentication Settings	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Image Viewer	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	HL7v2 Data Type	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Data Pruner	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Raw Data Type	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	DICOM Data Type	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	HL7v3 Data Type	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Mapper Transformer Step	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Message Builder Transformer Step	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057
Enabled	Dashboard Connector Status Monitor	Mirth Corporation	http://www.mirthcorp.com	3.4.1.8057

Install Extension from File System

File:

The latter ability, where you can install a new Connector option or Plugin is really where you could theoretically build your own, or download someone else's already-build custom extension and then inject that into Mirth for use within your custom workflow. Any new Connectors would become available in a channel's Source/Destination Connector dropdown list as a new option, and Plugins would make their functionality useable as well. How to actually build these custom extensions yourself is unfortunately beyond the scope of this guide.

Navigation Tips

- Hover over configuration text boxes to see a brief explanation of the purpose
- Double-click a channel in the dashboard to view messages; double-click a channel in the channel list to edit it
- While viewing a specific message in the dashboard, you can double-click it to bring up the "Message" console pre-populated with the selected message's content. This allows you to modify the content if you want then re-send the message through while skipping the need to copy/paste.
- Clicking the "mirthconnect" icon in the top right of the Mirth application will take you to the Mirth homepage of https://www.nextgen.com/Interoperability/Mirth-Solutions/Connect-Overview
    -

- At the bottom of the application window, the time and timezone offset are shown so that you know what time the application is currently using
    -
| 2:37 PM EDT (UTC -4)

Was this article helpful? 3 out of 3 found this helpful