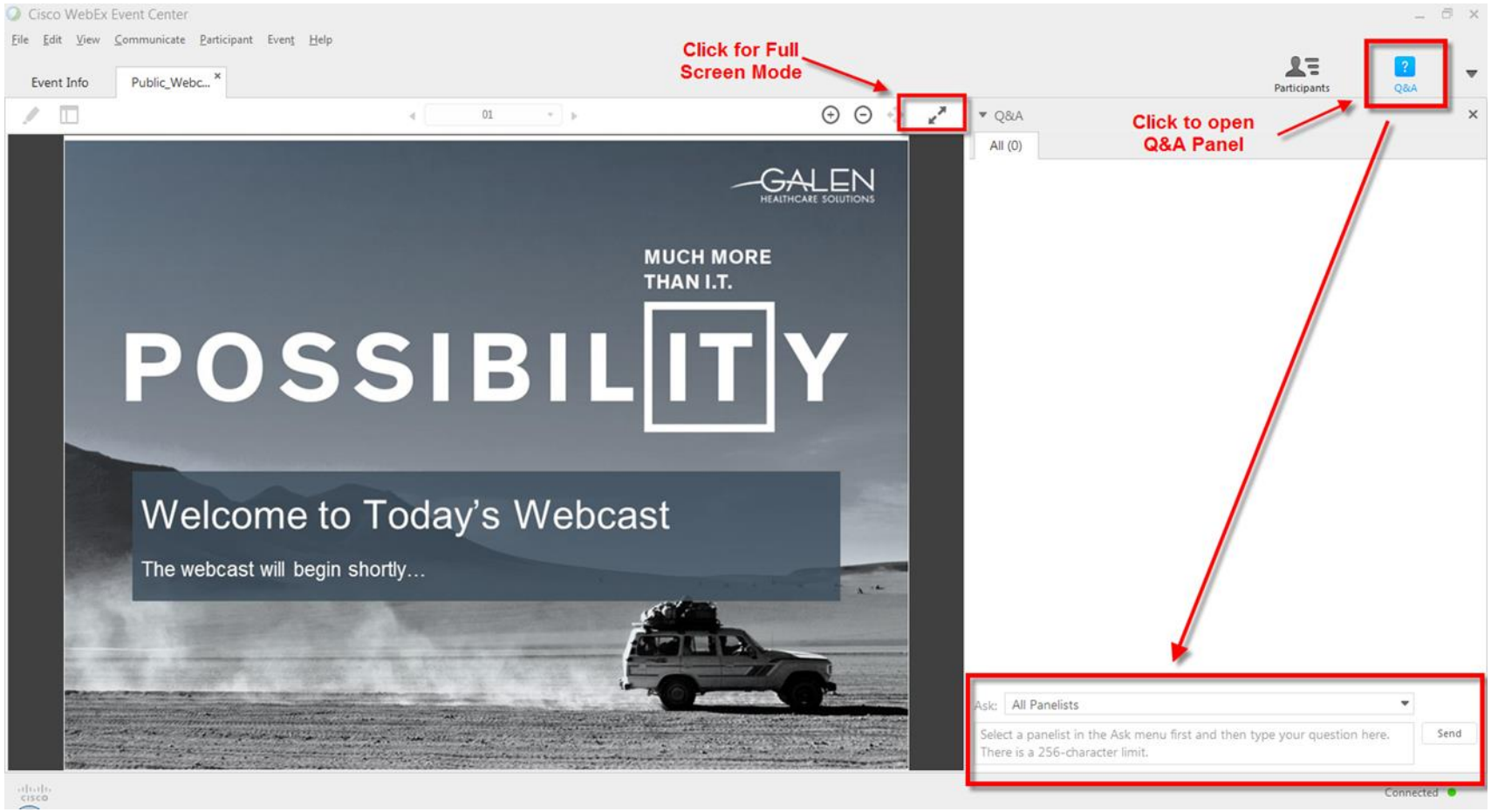
Advanced Topics on the Mirth Connect Interface Engine

July 6, 2016

You have been automatically muted. Please use the Q&A panel to submit questions during the presentation



PRESENTER

Nate Bessa
Technical Consultant

- Based in Boston, MA
- Specializing in custom programming projects, HIE connectivity, data migrations, and HL7 interface development
- 2+ years experience with Mirth Connect
- 5+ years in healthcare IT

Nate.Bessa@galenhealthcare.com

POLL #1

Topics

GALEN HEALTHCARE SOLUTIONS

I. Performance tweaks
II. Security improvements
III. Programming suggestions
IV. Maintenance strategies
V. Q & A

GALEN
HEALTHCARE SOLUTIONS

PERFORMANCE TWEAKS

Channel Metadata

GALEN HEALTHCARE SOLUTIONS

Edit Channel - ADT A04 - Patient Registration - Source Transformer

#	Name	Type
1	Add important data to channel map variables for use as message metadata	JavaScript

Step

```
1 var patientId = msg['PID']['PID.3']['PID.3.1'].toString();
2 var attendingPhysicianLastName = msg['PV1']['PV1.7']['PV1.7.1'].toString();
3 var attendingPhysicianFirstName = msg['PV1']['PV1.7']['PV1.7.2'].toString();
4 var insuranceProvider = msg['IN1']['IN1.4']['IN1.4.1'].toString()
5
6 // Save to channel map variables the data that we want to serve as message metadata
7 channelMap.put('PatientID', patientId);
8 channelMap.put('AttendingPhysician', attendingPhysicianFirstName + " " + attendingPhysicianLastName);
9 channelMap.put('InsuranceProvider', insuranceProvider);
```

Custom Metadata

Column Name	Type	Variable Mapping
PATIENT_ID	STRING	PatientID
ATTENDING_PHYSICIAN	STRING	AttendingPhysician
INSURANCE	STRING	InsuranceProvider

Add | Delete | Revert

Add clarity to your message logs and minimize data storage needs

Channel Messages - ADT A04 - Patient Registration

Start Time: 09:34 AM All Day
End Time: 09:34 AM
Text Search: Regex
Page Size: 20 Advanced... Reset Search

RECEIVED
TRANSFORMED
FILTERED
QUEUED
SENT
ERROR

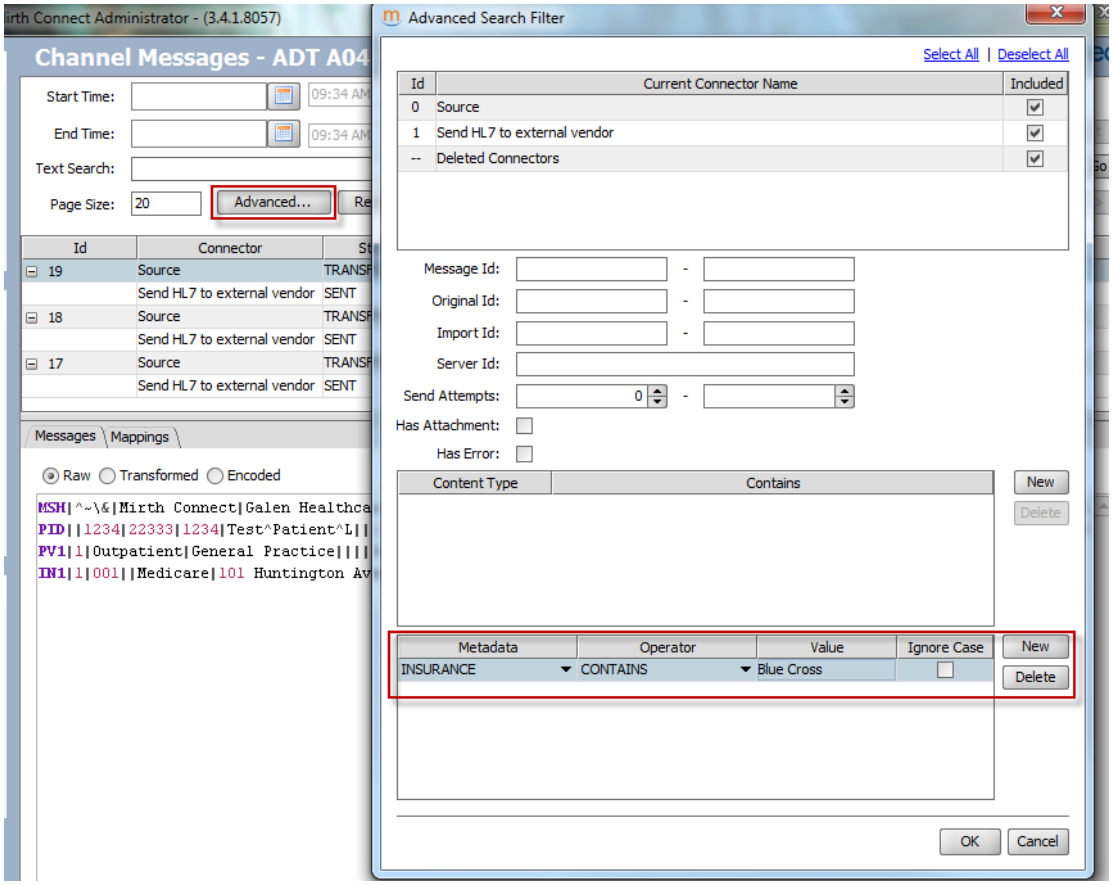
Current Search: Max Message Id: 19 Date Range: (any) to (any) Statuses: (any)

Results 1 - 3 of 3 Page 1 of 1 < Prev Next >

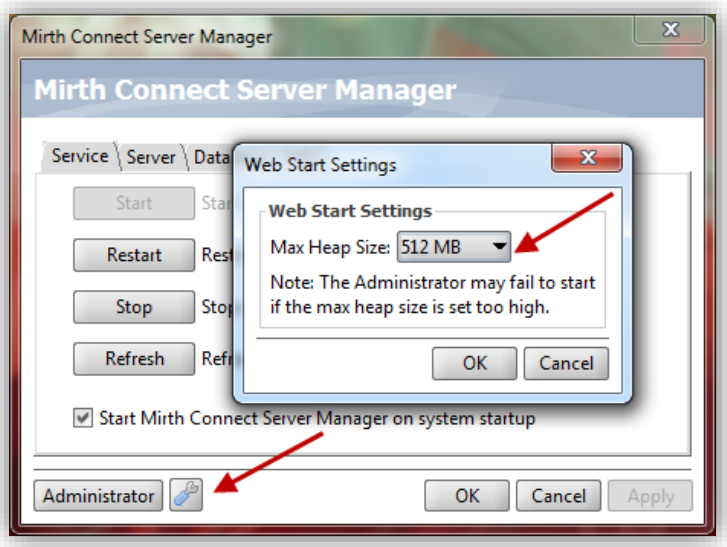
Id	Connector	Status	Received Date	PATIENT_ID	ATTENDING_PHYSICIAN	INSURANCE
19	Source	TRANSFORMED	2016-06-24 13:37:51:387	22333	John Doctor	Medicare
	Send HL7 to external vendor	SENT	2016-06-24 13:37:51:393	22333	John Doctor	Medicare
18	Source	TRANSFORMED	2016-06-24 13:37:24:727	31231	Nurse Test	Blue Cross Blue Shield
	Send HL7 to external vendor	SENT	2016-06-24 13:37:24:737	31231	Nurse Test	Blue Cross Blue Shield
17	Source	TRANSFORMED	2016-06-24 13:36:44:177	1234	John Doctor	Blue Cross Blue Shield
	Send HL7 to external vendor	SENT	2016-06-24 13:36:44:183	1234	John Doctor	Blue Cross Blue Shield

Channel Metadata

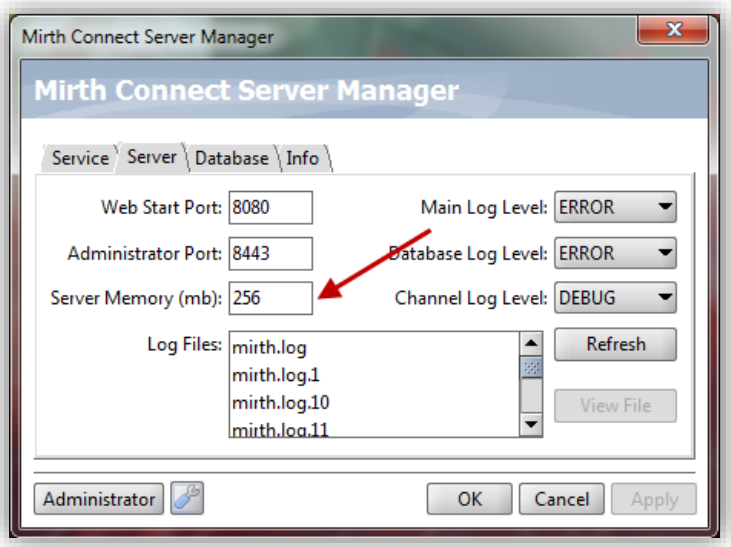
But most of all, speed up your message searches big time.



Increasing Heap Memory



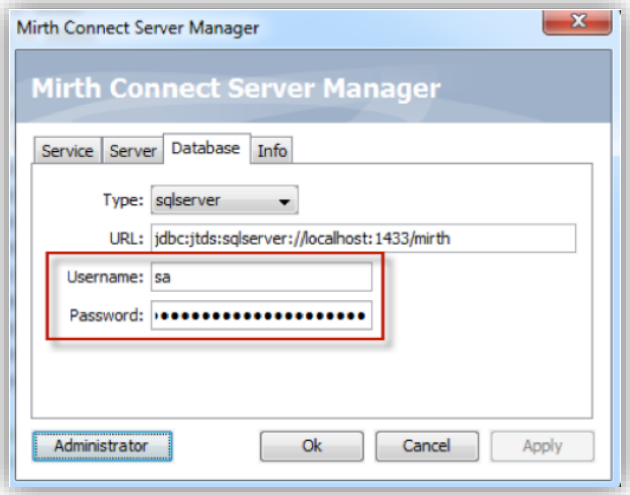
The Heap Size is the amount of memory made available for your Mirth Connect front end application. This plays a role especially when searching for messages in channels with lots of traffic.



The Server Memory setting dictates the amount of memory that Mirth Connect as an engine will be able to use in the background. This allows Mirth to process large messages and handle heavy traffic.

SECURITY IMPROVEMENTS

Encrypt Plain-Text Config Passwords



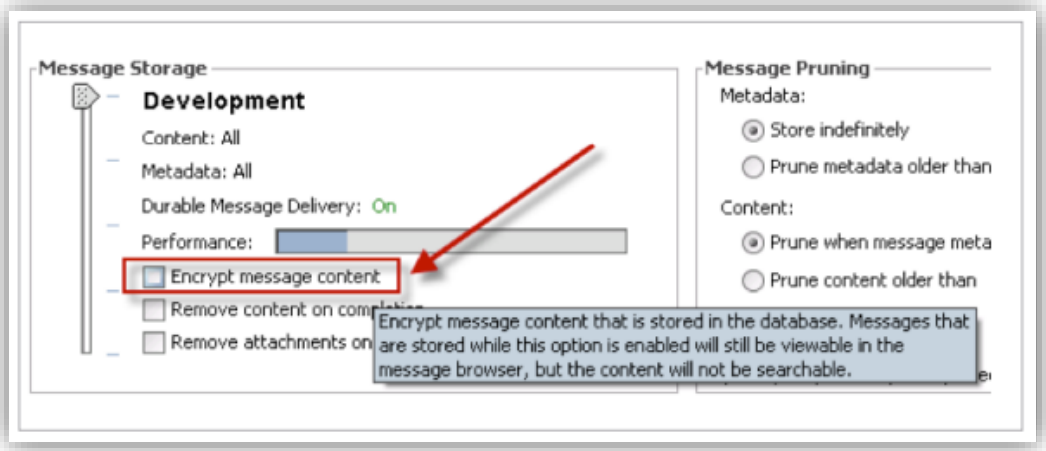

```
65 # examples:
66 #   Derby       jdbc:derby:${dir.appdata}/mirthdb:create=true
67 #   PostgreSQL  jdbc:postgresql://localhost:5432/mirthdb
68 #   MySQL       jdbc:mysql://localhost:3306/mirthdb
69 #   Oracle      jdbc:oracle:thin:@localhost:1521:DB
70 #   SQLServer   jdbc:jtds:sqlserver://localhost:1433/mirthdb
71 database.url = jdbc:jtds:sqlserver://localhost:1433/mirth
72
73 # if using a custom driver, specify it here
74 #database.driver =
75
76 # maximum number of connections allowed for the connection pool
77 database.max-connections = 20
78
79 # database credentials
80 database.username =sa
81 database.password =Im_a_Plaintext_Password
```

```
32
33 # Enabled encryption of properties with passwords.
34 encryption.properties = 1
35
```

Add this to your mirth.properties file and restart the Mirth Connect service.

```
81
82 # database credentials
83 database.username =sa
84 database.password ={enc}O8HCUSpgBBesW6gwE+nTa2Q8AUNLoYwChecqkjyWpsM=\r\n
85
```

Encrypt Message Storage

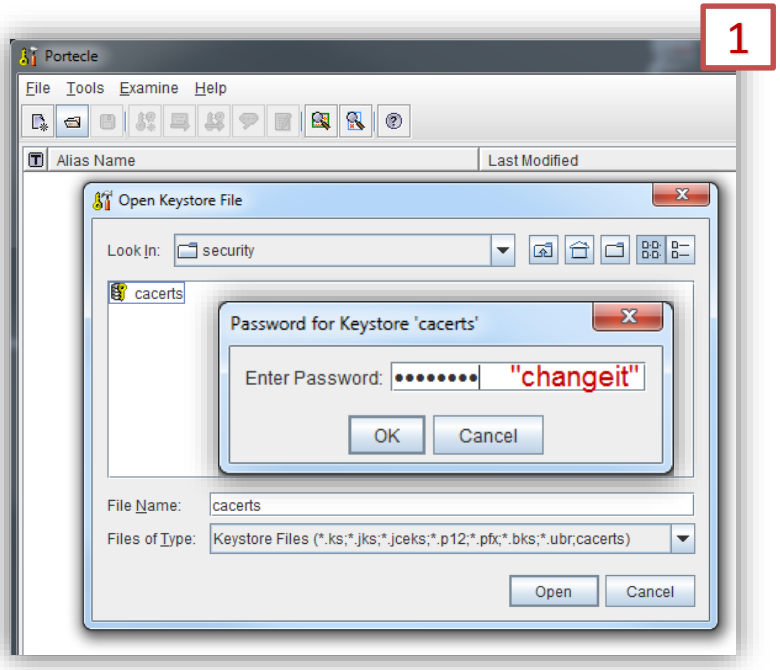


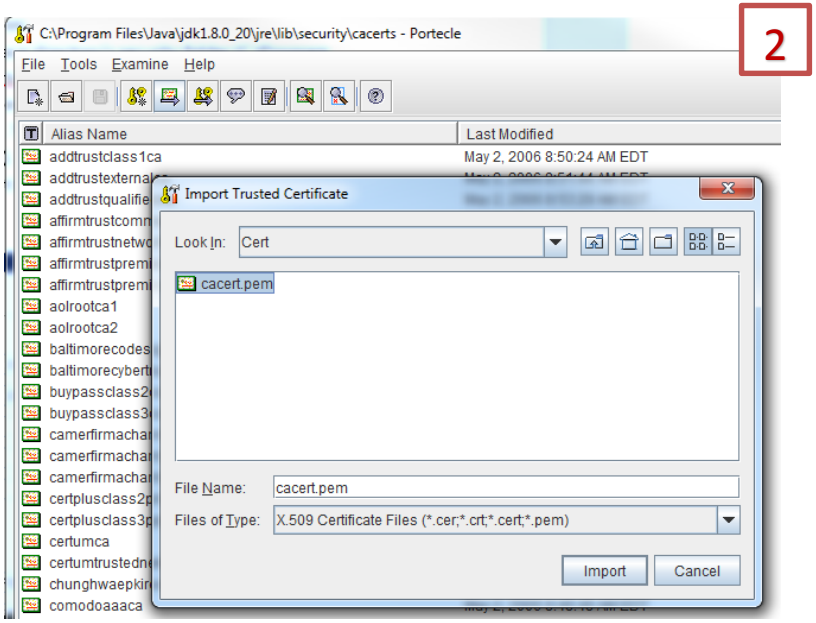
For added security, you can encrypt message content stored in the database.

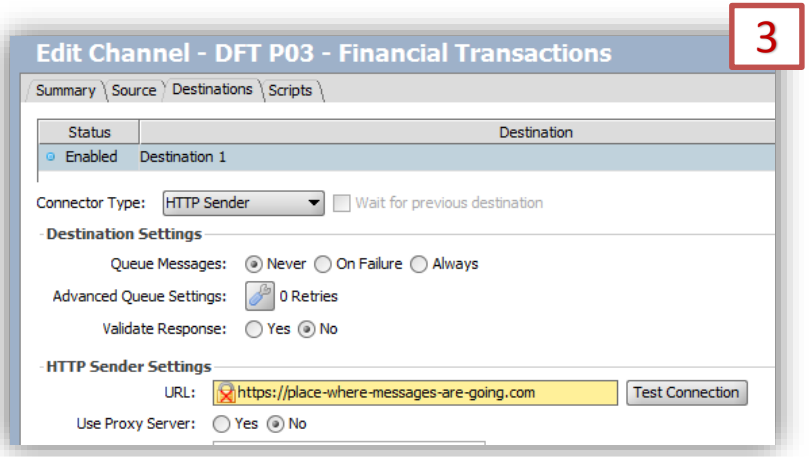
If you choose to use this feature, add these settings to your mirth.properties file and restart the Mirth service.

```
23
24  # The algorithm to use for encryption (DES, AES, etc.)
25  encryption.algorithm = AES
26
27  # The key length
28  encryption.keylength = 256
29
30  # Enables encryption/decryption of export/import through the Administrator.
31  encryption.export = 1
32
33  # The algorithm to use for one-way hashing (MD5, SHA, etc.)
34  digest.algorithm = MD5
35
36  # The security provider to use for all encryption and hashing.
37  security.provider = org.bouncycastle.jce.provider.BouncyCastleProvider
38
```

Importing a Self-Signed SSL Certificate







PROGRAMMING SUGGESTIONS

POLL #2

Move Functions to Code Templates

GALEN HEALTHCARE SOLUTIONS

Code Templates

Name

Miscellaneous

Format Date For HL7 — Converts

1 Library, 1 Code Template

Library: Miscellaneous

Type: Function

Code:

```
/**
     Converts a date in the YYYY-MM-DD HH:MM:SS format to the HL7 standard (YYYYMMDDHHMMSS)

     @param {String} date - string representing date in YYYY-MM-DD HH:MM:SS format
     @return {String} return string represting date in YYYYMMDDHHMMSS format (HL7 standard)
*/
function formatDateForHl7(datetime)
{
     // If the value given is an empty string return an empty string
     if (datetime.length <= 0)
          return "";

     // Check that the given timestamp is at least 19 characters long (YYYY-MM-DD HH:MM:SS)
     if (datetime.length < 19)
          throw ("Code Template formatDateToHl7 - Error: Given datetime of " + datetime + " does not " +
               "have at least 19 characters. Format should be YYYY-MM-DD HH:MM:SS[.000]");

     var year = datetime.substr(0,4);
     var month = datetime.substr(5,2);
     var day = datetime.substr(8,2);
     var hours = datetime.substr(11,2);
     var minutes = datetime.substr(14,2);
     var seconds = datetime.substr(17,2);

     var formattedDate = year + month + day;

     // Only include the time if there was one provided
     if (hours != "00" || minutes != "00" || seconds != "00")
          formattedDate += hours + minutes + seconds;

     return formattedDate;
}
```

Allows you to edit a function in one location instead of having to make changes in every channel where you are using.

Edit Channel - DFT P03 - Financial Transactions - Source Transfor

#	Name
0	Format patient date of birth in HL7 standard

Step

```
// Copy the incoming HL7 message to a variable we can manipulate
tmp = msg;

// Change patient date of birth from YYYY-MM-DD HH:MM:SS format to YYYYMMDDHHMMSS
var patientDateOfBirth = msg['PID']['PID.7']['PID.7.1'].toString();
var patientDateOfBirthInHl7Format = formatDateForHl7(patientDateOfBirth);

// Place newly formatted date of birth in the outgoing HL7 message
tmp['PID']['PID.7']['PID.7.1'] = patientDateOfBirthInHl7Format;
```

Edit Channel - ADT A04 - Patient Registration - Source Transformer

#	Name
0	Format insurance expiration date in HL7 standard

Step

```
// Copy the incoming HL7 message to a variable we can manipulate
tmp = msg;

// Change insurance expiration date from YYYY-MM-DD HH:MM:SS format to YYYYMMDDHHMMSS
var insuranceExpirationDate = msg['IN1']['IN1.13']['IN1.13.1'].toString();
var insuranceExpirationDateInHl7Format = formatDateForHl7(insuranceExpirationDate);

// Place newly formatted insurance expiration date in the outgoing HL7 message
tmp['IN1']['IN1.13']['IN1.13.1'] = insuranceExpirationDateInHl7Format;
```

Also lets you keep your channels simple by moving long and complicated code out of them and into the code templates screen.

Modify an HL7 Message with Database Data

Edit Channel - ADT A04 - Patient Registration - Source Trans

#	Name
0	Find the patient's MRN from our database and add it to the HL7 message

Step

```
// Open a database connection
var dbConn = GetDatabaseConnection();

// Pull the patient ID in the incoming HL7 message
var patientId = msg['PID']['PID.3']['PID.3.1'].toString();

// Build a query to find this patient's MRN in our table
var sqlQuery =
     "SELECT PatientMrn " +
     "FROM dbo.PatientRegistration " +
     "WHERE id = '" + patientId + "'";

// Execute the query
var queryResults = dbConn.executeCachedQuery(sqlQuery);

// Check if a row was found containing the patient
if (queryResults.next())
{
     // Pull the MRN (a string) from the query results
     var patientMrn = queryResults.getString('PatientMrn');

     // Save the incoming HL7 message to the transformation variable
     tmp = msg;

     // Add the MRN to PID 4.1
     tmp['PID']['PID.4']['PID.4.1'] = patientMrn;
}
```

Source Transformer Javascript Writer

```
SELECT Id, PatientMrn FROM dbo.PatientRegistration
```

Results | Messages

	Id	PatientMrn
1	15159465-B820-4D5F-8BE2-07A1AE24956C	1234

The row in our table we want to query

Code Templates

Name	
Get Database Connection	Returns a

1 Library, 2 Code Templates

Library: Miscellaneous

Type: Function

Code:

```
/**
     Returns a database connection object used for querying data

     @return {DatabaseConnection} returns a DatabaseConnection object
*/
function GetDatabaseConnection() {
     // These variables are defined in Mirth's (Deploy) Global Script
     return DatabaseConnectionFactory.createDatabaseConnection(
          $('sqlServerDriver'),
          $('databaseUrl'),
          $('databaseUsername'),
          $('databasePassword'));
}
```

Using a code template for database connections

```
MSH|^~\&|Mirth Connect|Galen Healthcare Solutions|Your Hospital|General Pr
PID|||15159465-B820-4D5F-8BE2-07A1AE24956C||Test^Patient^L||19911006|M|||7
PV1|1|Outpatient|General Practice||||Doctor^John|||||||||598545||||||||
IN1|1|001||Blue Cross Blue Shield|101 Huntington Avenue^^Boston^MA^02199||
```

Incoming Hl7 Message (note PID 4 is blank)

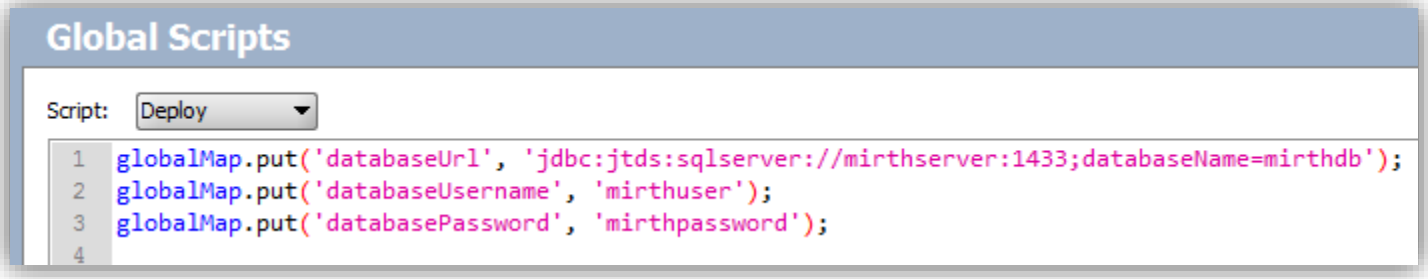
```
MSH|^~\&|Mirth Connect|Galen Healthcare Solutions|Your Hospital|General Pr
PID|||15159465-B820-4D5F-8BE2-07A1AE24956C|1234|Test^Patient^L||19911006|M
PV1|1|Outpatient|General Practice||||Doctor^John|||||||||598545||||||||
IN1|1|001||Blue Cross Blue Shield|101 Huntington Avenue^^Boston^MA^02199||
```

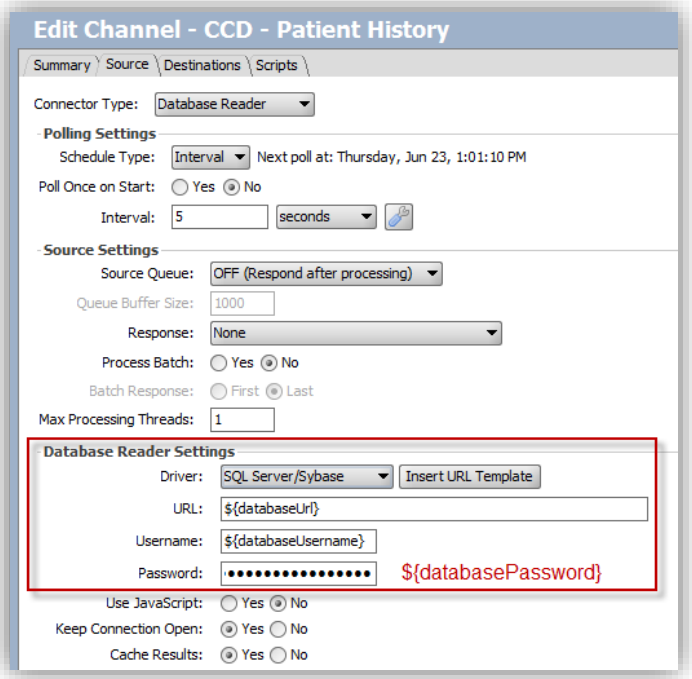
Transformed Hl7 Message (PID 4 now has the PatientMrn)

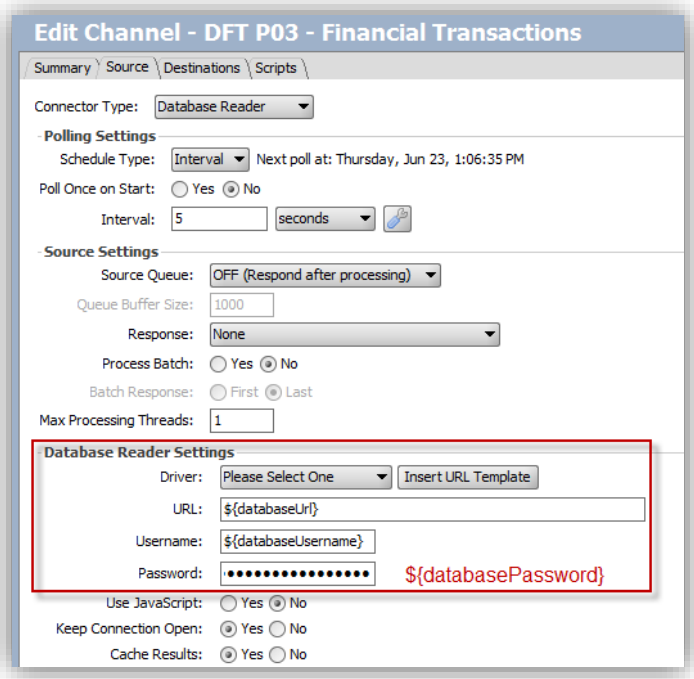
Programming Suggestions

Common Channel Settings

GALEN HEALTHCARE SOLUTIONS







Environment Strategies

GALEN HEALTHCARE SOLUTIONS

Global Scripts

Script: Deploy

```
1  // Set the names for the production and test Mirth environments
2  var productionMirthServerName = 'PRODSERVER';
3  var testMirthServerName = 'TESTSERVER';
4
5  // Get the name of this server running Mirth
6  var serverName = Packages.java.net.InetAddress.getLocalHost().getHostName();
7
8  // General database connection information
9  globalMap.put('sqlServerDriver', 'net.sourceforge.jtds.jdbc.Driver');
10
11 // Set server specific variables
12 if (serverName == productionMirthServerName)
13 {
14     globalMap.put('systemEnvironment', 'PROD');
15     globalMap.put('databaseUrl', 'jdbc:jtds:sqlserver://prodserver:1433;dat
16     globalMap.put('databaseUsername', 'mirthuser');
17     globalMap.put('databasePassword', 'mirthpassword');
18 }
19 else if (serverName == testMirthServerName)
20 {
21     globalMap.put('systemEnvironment', 'TEST');
22     globalMap.put('databaseUrl', 'jdbc:jtds:sqlserver://testserver:1433;dat
23     globalMap.put('databaseUsername', 'mirthuser');
24     globalMap.put('databasePassword', '1Ac#jkas710z');
25 }
26 else
27 {
28     throw "FATAL ERROR: The server launching Mirth is unrecognized. Look at
29 }
30
```

Edit Channel - Run Nightly Processes

#	Operator
0	Filter Destination in TEST Environment

Rule

```
1  if ($('systemEnvironment') == 'TEST')
2      return false;
3  else
4      return true;
```

- Programmatically dictate desired behavior based on server names rather than manually making changes to channels between development and production environments.
- This eliminates potential for human error when exporting channels from development environment to production.

Routing A Message to Other Channels

Channels

Status	Data Type	Name	Id
Enabled		Message Routing Example	70c0a01e-7d26-4c3a-b748-3e48cd24ecbd
Enabled	HL7 v2.x	Update Patient	92b11d53-f9d5-455a-b614-da38be986405
Enabled	HL7 v2.x	Get Patient Status	2593837b-02f7-46dd-b621-7f255c43baa9
Enabled	HL7 v2.x	Create Patient	c6eef31d-0006-4108-9105-3a49beb65c9c

Edit Channel - Get Patient Status

Summary | Source | Destinations | Scripts

Status	Destination
Enabled	Get Patient Status

Connector Type: Web Service Sender — Wait for previous destination

Destination Settings

Queue Messages: Never / On Failure / Always
Advanced Queue Settings: 0 Retries
Validate Response: Yes / No

Web Service Sender Settings

WSDL URL: {some http address here}
Service: {some service address here}
Port / Endpoint: {some port here}
Location URI: {some URI here}
Socket Timeout (ms): 30000
Authentication: Yes / No
Username:
Password:
Invocation Type: Two-Way / One-Way
Operation: Press Get Operations — Generate Envelope
SOAP Action: GetPatientStatus
SOAP Envelope:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
   <soapenv:Header/>
   <soapenv:Body>
      <ns:SearchClient_Input>
         <mes:MessageContextInput ProgramID=""/>
```

Edit Channel - Get Patient Status - Get Patient Status Response Transformer

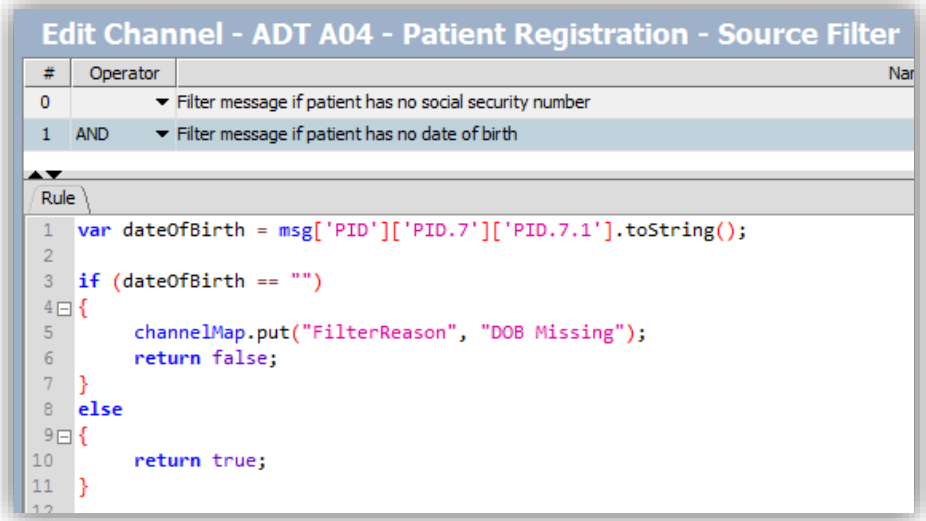
#	Name	Type
0	Route message to correct channel based on response	JavaScript

Step

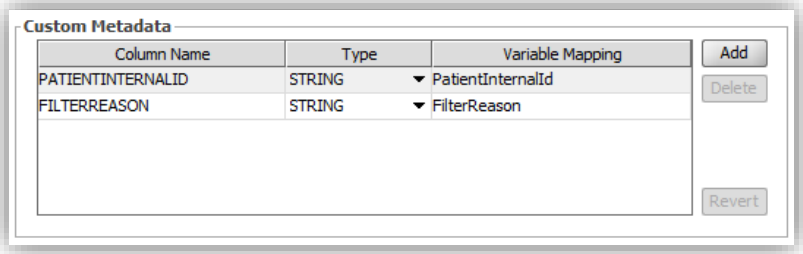
```
1  // Check Web Service response for indication if patient exists (this is a hypothetical example)
2  if (msg['ResponseMessage'] == "Patient Already Exists")
3  {
4      // Since patient exists, route message to the Update Patient channel
5      router.routeMessageByChannelId('92b11d53-f9d5-455a-b614-da38be986405', connectorMessage.getRawData());
6  }
7  else
8  {
9      // Patient does not already exist, so route message to Create Patient channel
10     router.routeMessageByChannelId('c6eef31d-0006-4108-9105-3a49beb65c9c', connectorMessage.getRawData());
11 }
```

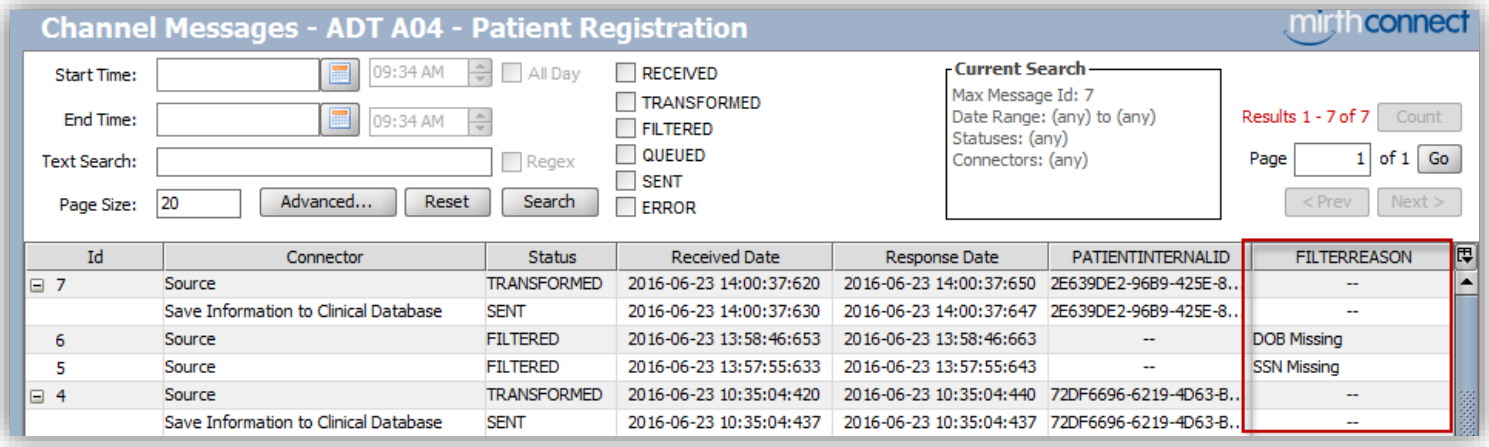
Programming Suggestions

Know the Reason for Filtering

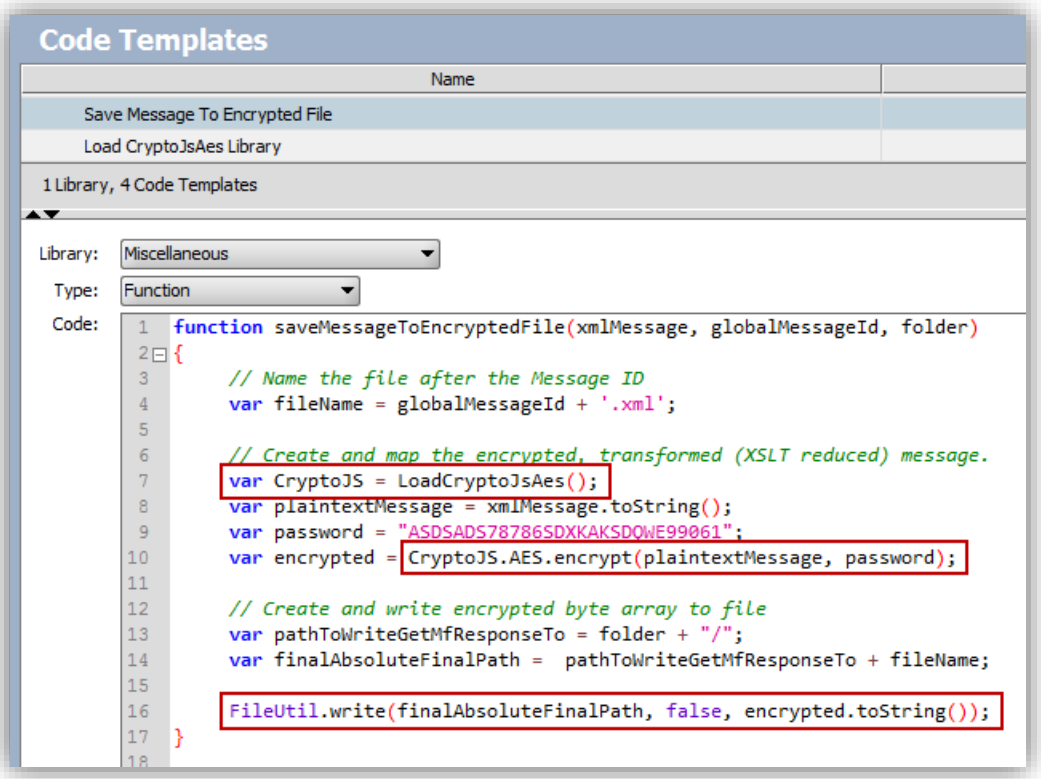


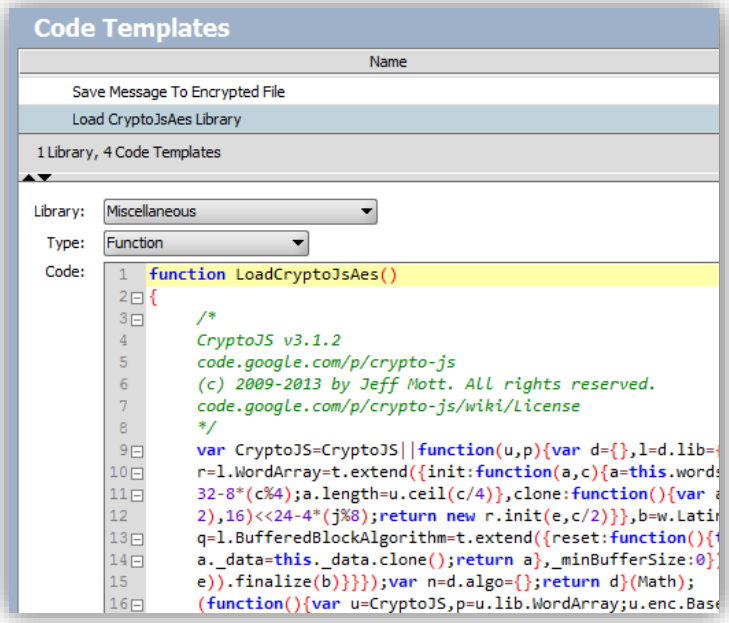
If you use a lot of filters, Mirth Connect currently does not tell you which filter was triggered. If that information is useful, here is a custom way of showing that.





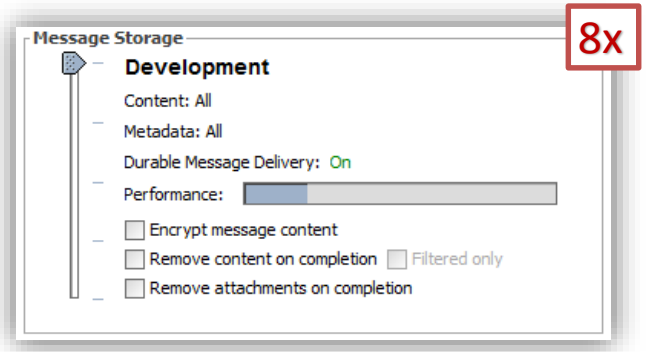
Using a JavaScript Library To Encrypt Files



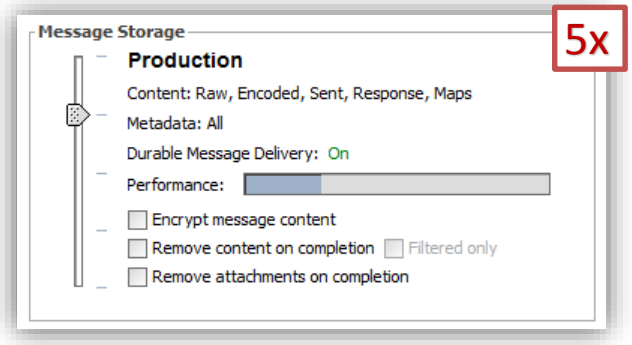


MAINTENANCE STRATEGIES

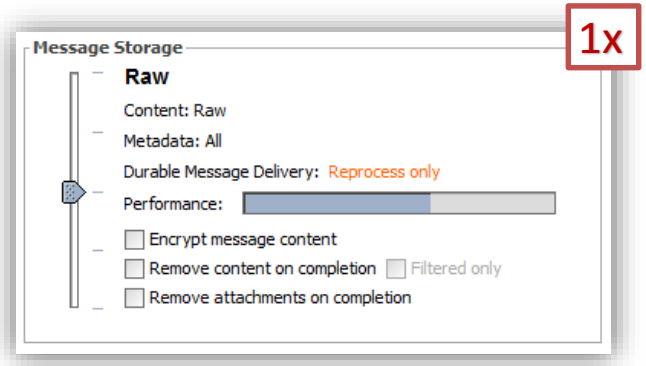
Message Storage Impact



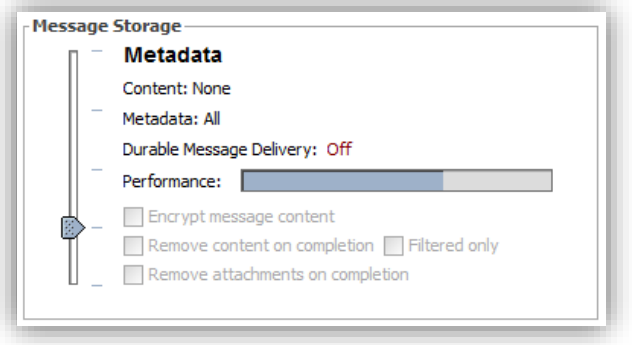
Uses the most disk space which also slows processing time.



Preserves essential troubleshooting data. Pair with pruning options to keep disk usage in check.

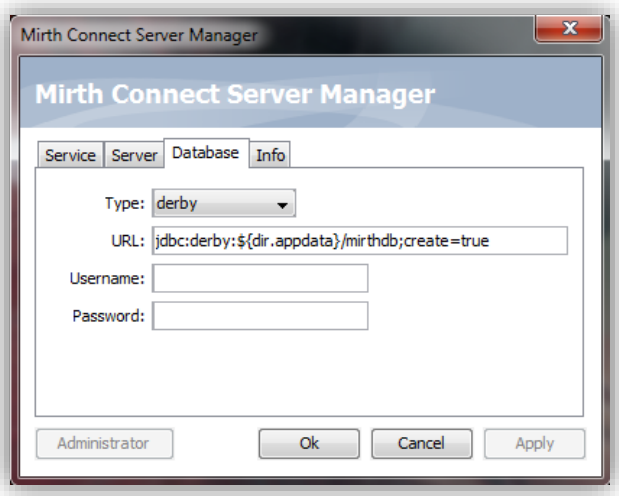


Stores just the original source message. No mapped variables. Reprocess a message when you need to troubleshoot one.

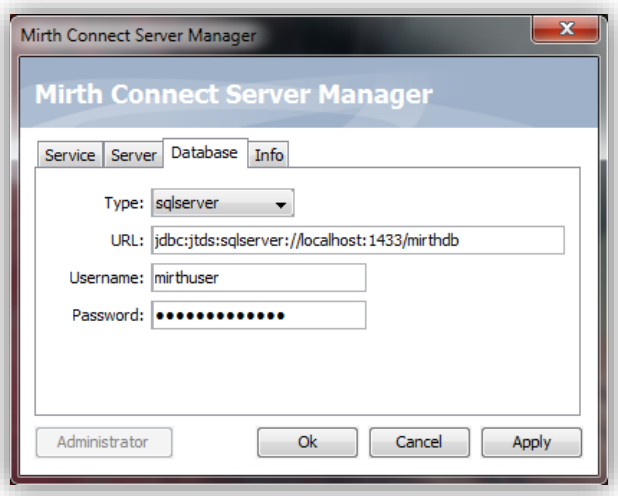


Deletes everything except what you see on the channel message log. *Cannot reprocess a message.*

Database Backend



Default Setting



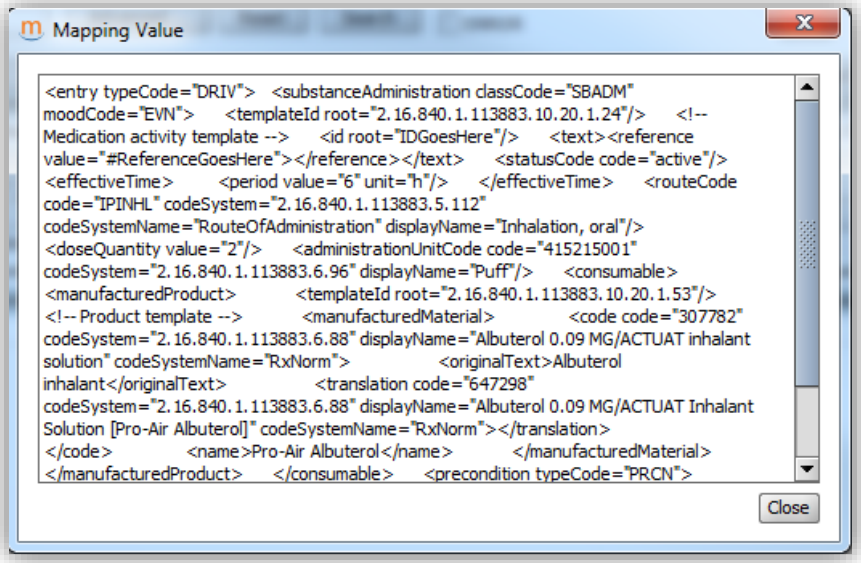
Preferred Configuration

- The default Apache Derby option is for quick deployment and testing.
- Mirth Corp recommends PostgreSQL, MySQL, Oracle, or SQL Server.
- These offer better performance, disk space efficiency, backup options, security, etc.

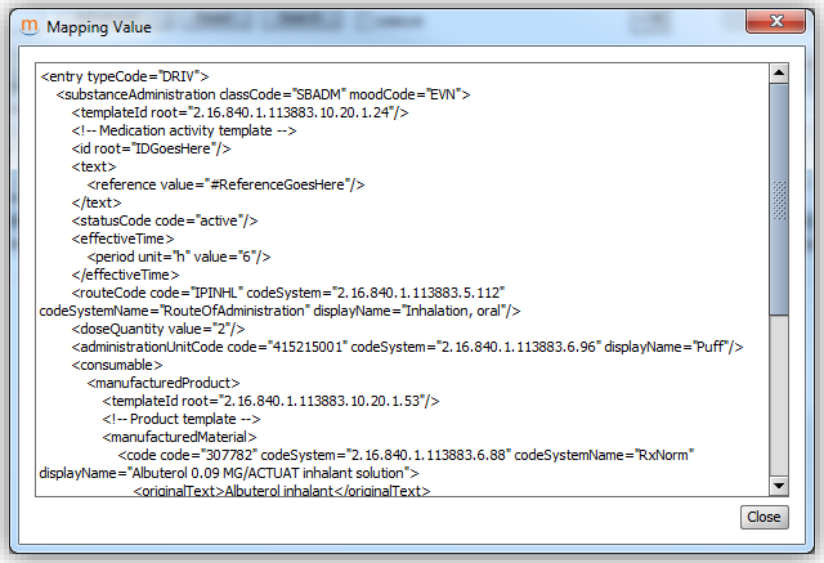
PRETTYPRINTXML

GALEN HEALTHCARE SOLUTIONS

```
1  var xmlData = connectorMessage.getRawData();
2  var xmlDataWithPrettyPrintFormatting = XmlUtil.prettyPrint(xmlData);
3
4  channelMap.put('SourceXmlUnformatted', xmlData);
5  channelMap.put('SourceXmlPrettyPrintFormat', xmlDataWithPrettyPrintFormatting);
```



SourceXmlUnformatted

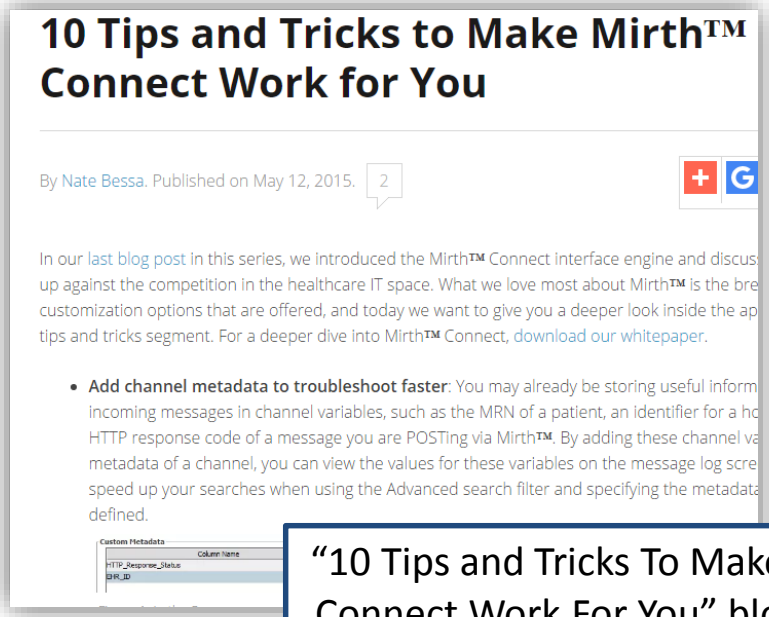


SourceXmlPrettyPrintFormat

POLL #3

Wrapping Up

Additional Resources


10 Tips and Tricks to Make Mirth™ Connect Work for You

By Nate Bessa. Published on May 12, 2015. 2

In our last blog post in this series, we introduced the Mirth™ Connect interface engine and discus up against the competition in the healthcare IT space. What we love most about Mirth™ is the bre customization options that are offered, and today we want to give you a deeper look inside the ap tips and tricks segment. For a deeper dive into Mirth™ Connect, download our whitepaper.

- **Add channel metadata to troubleshoot faster**: You may already be storing useful inform incoming messages in channel variables, such as the MRN of a patient, an identifier for a hc HTTP response code of a message you are POSTing via Mirth™. By adding these channel va metadata of a channel, you can view the values for these variables on the message log scre speed up your searches when using the Advanced search filter and specifying the metadata defined.

Custom Metadata

Column Name
HTTP_Response_Status
QHR_ID

"10 Tips and Tricks To Make Mirth Connect Work For You" blog post

Google "Mirth Connect Tips and Tricks Galen"

"Best Practices and Vulnerabilities in Mirth Connect" white paper

Google "Mirth Connect Security White Paper Galen"

"Mirth Connect - Introduction" webcast

Search "Mirth Connect Galen" on Youtube

Thank you for joining us today.
To access the slides from today's presentation, please visit:

http://wiki.galenhealthcare.com/Category:Webcasts

For additional assistance or to request information about our many products and services, including help with Mirth Connect, please contact us through our website:

www.galenhealthcare.com