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.
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
I. Performance tweaks
II. Security improvements
III. Programming suggestions
IV. Maintenance strategies
V. Q & A
PERFORMANCE TWEAKS
Add clarity to your message logs and minimize data storage needs
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

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

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

```plaintext
# The algorithm to use for encryption (DES, AES, etc.)
extraction.algorithm = AES

# The key length
extraction.keylength = 256

# Enables encryption/decryption of export/import through the Administrator.
extraction.export = 1

# The algorithm to use for one-way hashing (MD5, SHA, etc.)
digest.algorithm = MD5

# The security provider to use for all encryption and hashing.
security.provider = org.bouncycastle.jce.provider.BouncyCastleProvider
```
IMPORTING A SELF-SIGNED SSL CERTIFICATE

1. Open Keystore File
2. Import Trusted Certificate
3. Edit Channel - DFT P03 - Financial Transactions

Password for Keystore 'cacerts':

Enter Password: ********

"changeit"

Status | Destination 1
--- | ---
Enabled |

Connector Type: HTTP Sender

Queue Messages:
- Never
- On Failure
- Always

Advanced Queue Settings:
- 0 Retries

Validate Response:
- Yes
- No

HTTP Sender Settings:
- URL: https://place-where-messages-are-going.com
- Test Connection

SECURITY IMPROVEMENTS
PROGRAMMING SUGGESTIONS
Poll #2
Move Functions to Code Templates

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

Allows you to edit a function in one location instead of having to make changes in every channel where you are using.
MODIFY AN HL7 MESSAGE WITH DATABASE DATA

Source Transformer Javascript Writer

Incoming HL7 Message (note PID 4 is blank)

Transformed HL7 Message (PID 4 now has the PatientMrn)

Using a code template for database connections

The row in our table we want to query

PROGRAMMING SUGGESTIONS
COMMON CHANNEL SETTINGS

Global Scripts

```java
1. globalMap.put('databaseUrl', 'jdbc:postgresql://mirthserver:1433;databaseName=mirthdb');
2. globalMap.put('databaseUsername', 'mirthuser');
3. globalMap.put('databasePassword', 'mirthpassword');
```

Edit Channel - CCD - Patient History

```
Connector Type: Database Reader
Polling Settings:
- Schedule Type: Interval
- Poll Once on Start: Yes
- Interval: 5 seconds
Source Settings:
- Source Queue: OFF (Respond after processing)
- Queue Buffer Size: 1000
- Response: None
- Process Batch: Yes
- Batch Response: First
- Max Processing Threads: 1
```

Database Reader Settings:
```
Driver: SQL Server/Sybase
URL: $(databaseUrl)
Username: $(databaseUsername)
Password: $(databasePassword)
```

Edit Channel - DFT P03 - Financial Transactions

```
Connector Type: Database Reader
Polling Settings:
- Schedule Type: Interval
- Poll Once on Start: Yes
- Interval: 5 seconds
Source Settings:
- Source Queue: OFF (Respond after processing)
- Queue Buffer Size: 1000
- Response: None
- Process Batch: Yes
- Batch Response: First
- Max Processing Threads: 1
```

Database Reader Settings:
```
Driver: Please Select One
URL: $(databaseUrl)
Username: $(databaseUsername)
Password: $(databasePassword)
```

PROGRAMMING SUGGESTIONS
• 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

<table>
<thead>
<tr>
<th>Status</th>
<th>Data Type</th>
<th>Name</th>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>HL7 v2.x</td>
<td>Update Patient</td>
<td>92b11d35-6b63-455a-8e14-da3e39b936405</td>
</tr>
<tr>
<td>Enabled</td>
<td>HL7 v2.x</td>
<td>Get Patient Status</td>
<td>29037b-0927-4bdc-b262-7f2e5c1eb9a9</td>
</tr>
<tr>
<td>Enabled</td>
<td>HL7 v2.x</td>
<td>Create Patient</td>
<td>c9ef3d-0006-4108-9105-342f866c9c</td>
</tr>
</tbody>
</table>

Edit Channel - Get Patient Status

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

Programming Suggestions
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

Programming Suggestions
MAINTENANCE STRATEGIES
**Message Storage Impact**

- **Development**
  - Content: All
  - Metadata: All
  - Durable Message Delivery: On
  - Performance: 
    - Encrypt message content
    - Remove content on completion
    - Filtered only
    - Remove attachments on completion
  - Uses the most disk space which also slows processing time.

- **Production**
  - Content: Raw, Encoded, Sent, Response, Maps
  - Metadata: All
  - Durable Message Delivery: On
  - Performance: 
    - Encrypt message content
    - Remove content on completion
    - Filtered only
    - Remove attachments on completion
  - Preserves essential troubleshooting data. Pair with pruning options to keep disk usage in check.

- **Raw**
  - Content: Raw
  - Metadata: All
  - Durable Message Delivery: Reprocess only
  - Performance: 
    - Encrypt message content
    - Remove content on completion
    - Filtered only
    - Remove attachments on completion
  - Stores just the original source message. No mapped variables. Reprocess a message when you need to troubleshoot one.

- **Metadata**
  - Content: None
  - Metadata: All
  - Durable Message Delivery: Off
  - Performance: 
    - Encrypt message content
    - Remove content on completion
    - Filtered only
    - Remove attachments on completion
  - Deletes everything except what you see on the channel message log. *Cannot reprocess a message.*
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

```javascript
1. var xmlData = connectorMessage.getRawData();
2. var xmlDataWithPrettyPrintFormatting = XmlUtil.prettyPrint(xmlData);
3. channelMap.put('SourceXmlUnformatted', xmlData);
4. channelMap.put('SourceXmlPrettyPrintFormat', xmlDataWithPrettyPrintFormatting);
```
Poll #3
Wrapping Up
ADDITIONAL RESOURCES

“Best Practices and Vulnerabilities in Mirth Connect” white paper
Google “Mirth Connect Security White Paper Galen”

“10 Tips and Tricks To Make Mirth Connect Work For You” blog post
Google “Mirth Connect Tips and Tricks 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