



Health Information Exchange

Building cost-effective interface solutions

Presented By:

Morris Heights Health Center, Bronx, NY

- Clinical / Administrative: Lydia Gonzalez MD, MPH, FAAP
- Technical Lead: Ivan Pan
- Interface Developer: James Yu



Overview

- Clinical Motivations behind HIE
- Management planning and decisions
- Technical direction and implementation
- Lessons learned
- Demonstrations
- Q & A
- Credits



What is HIE?

- Health Information Exchange (HIE)

The exchange of electronic protected health information (PHI) across organizations within a region, community or hospital system.



Purpose of HIE?

- Patient Care
 - Improving care coordination
 - Better facilitation of patient care
 - Reduce medical costs
 - Minimize duplicate orders for diagnostic tests
 - Prevent medical errors
- Data/Report standardization
 - Interoperability between medical data from different sources
- Incentives
 - Requirement for Medical Home and Meaningful Use



What is RHIO?

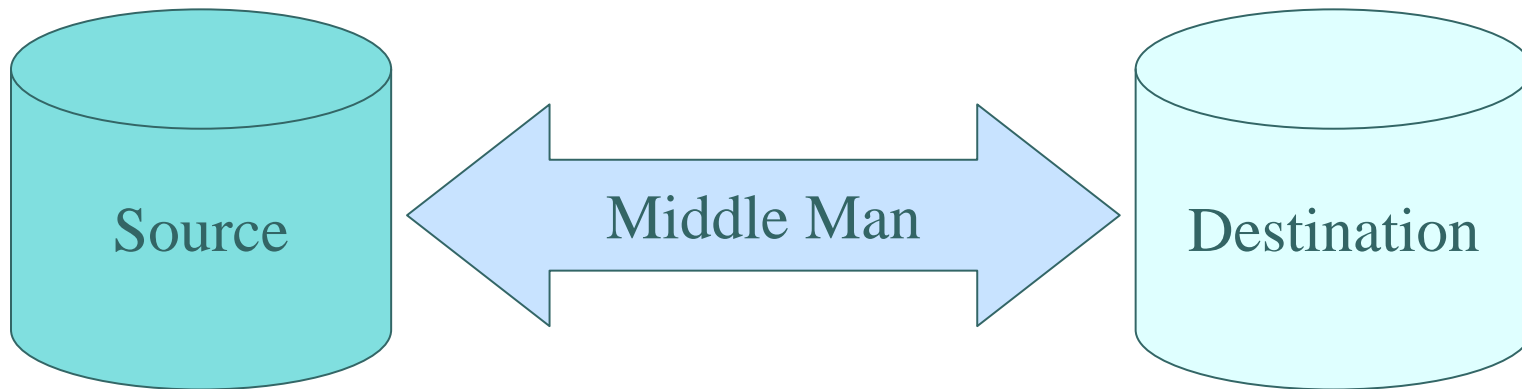
- A scenario that requires clinical information to be shared – the ER Prevent medical errors
- What and when to share?
 - Everything or only minimal?
- What about HIPAA?
- What it means for the future?



Bronx RHIO participation

- Collaboration of Health Centers and Hospitals in the Bronx region
- MHHC is one of those health centers
- Incentive for medical home and meaningful use

How does HIE Work?



- Data needs to be interoperable in order for it to flow from a source to a destination
 - The Interface Engine (Middle Man) reads and translates the data before sending it over to the Destination



The good old DTS

- What is it?
- The relations pre-defined in the export
- The benefits of DTS – simple, easy to setup relations
- The limitations – non-flexible definitions, tedious cross-reference tables, interface language that's from another planet



Interface code from IXP files

```
_BID_=UniversalServiceID~~4 // customization; use 5th instead of 2nd  
subcomponent
```

```
AcceptTruncateAppend={ Summary}  
_SAP_=_N_->Summary+=
```

```
#IFNOTEXIST _LABSUMMARYREPORT=_OBR_[]0->_BID_==;;_SAP__ORC_-  
>_OBR_[]0->_BID_#_SAP__OBR_[]0->_BID_#;  
!_OBR_[]1->_BID_==;;_SAP_", "#;  
!_ORC_->_OBR_[]1->_BID_==;;_SAP_", "#;  
_OBR_[]1->_BID_==;;_SAP__ORC_->_OBR_[]1->_BID_#_SAP__OBR_[]1-  
>_BID_#;  
!_OBR_[]2->_BID_==;;_SAP_", "#;  
!_ORC_->_OBR_[]2->_BID_==;;_SAP_", "#;  
_OBR_[]2->_BID_==;;_SAP__ORC_->_OBR_[]2->_BID_#_SAP__OBR_[]2-  
>_BID_#;  
!_OBR_[]3->_BID_==;;_SAP_", "#;
```



Assessing in-house skills

- LinkLogic and DTS
 - Can resolve and create basic EMR interfaces
- Basic HL7 format standard
- Ad hoc reporting off the EMR database - using SQL



Interface Engine requirements

- Budget needs to be cost-effective
- Learning Curve - Limited in-house resources and skill set, need some ease of use
- Control of interface engine is preferred
- Scalability and in-house staff development
- Product support and dependability



Interface Engine Selection

- Cloverleaf (formerly Quovadx)
 - Used by GE's CCG interface solution
 - Direct vendor
- Mirth
 - Open source interface engine
- Iguana
 - 30 day trial

Scoring Scale:

- 1. Limited
- 2. Standard
- 3. Extra value



Budget
Learning Curve
Control
Scalability
Support

Cloverleaf GE (CCG)

2

3

1

3

2

Cloverleaf (direct)

2

2

2

3

2

Mirth

3

2

2

3

2

Iguana

-

2

2

-

2

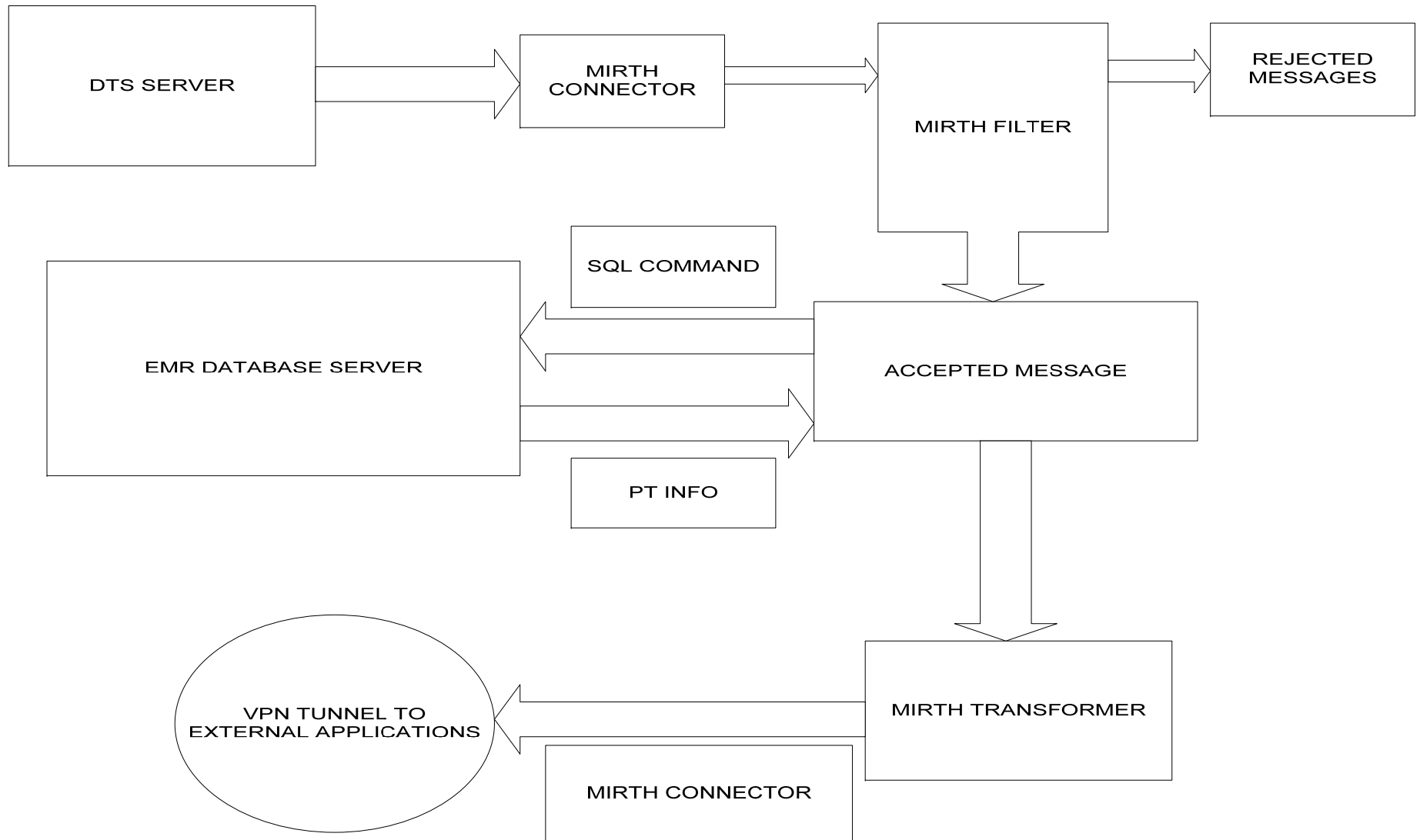


Our Ideal Interface Solution

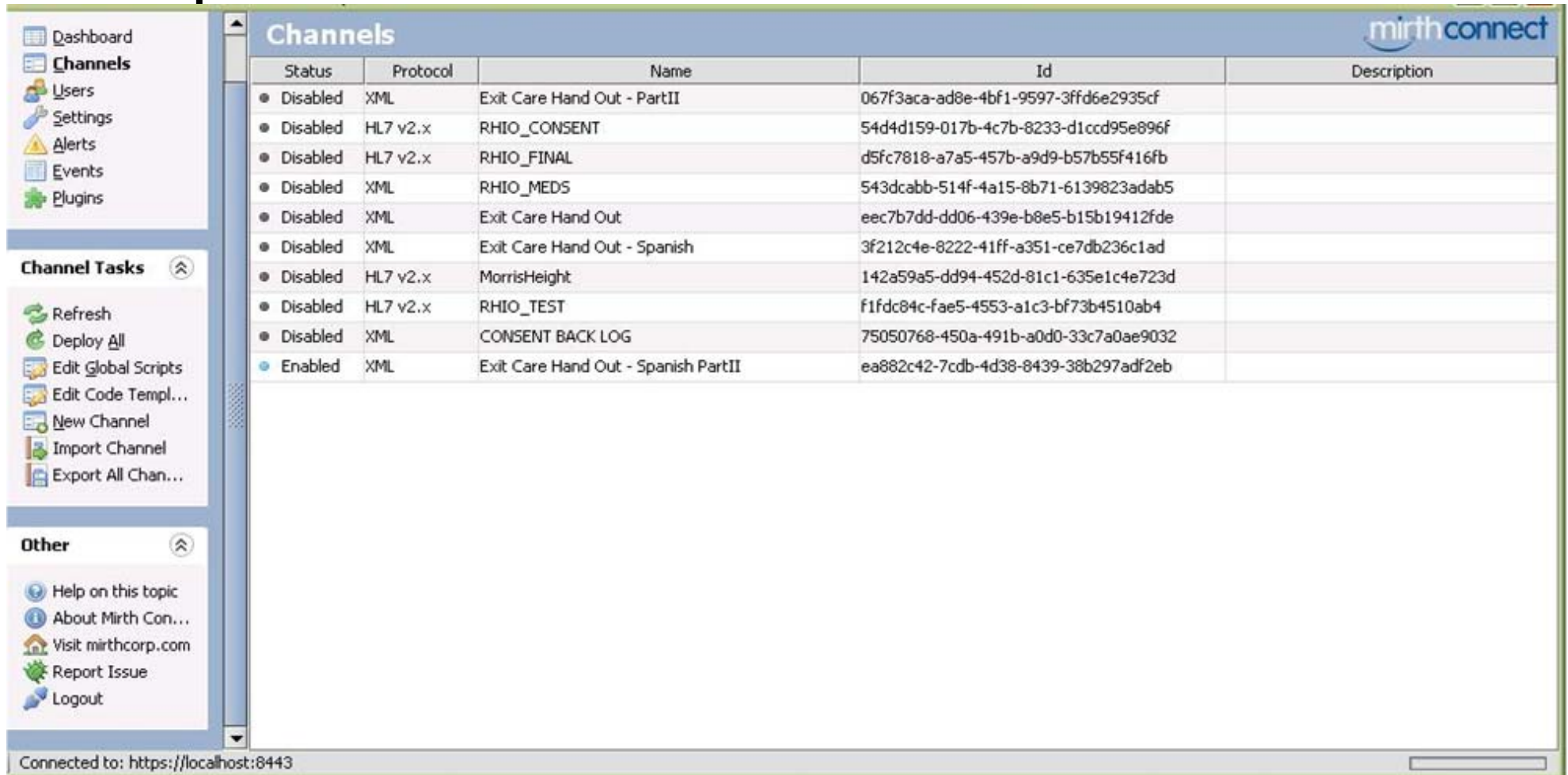
- Allows us to control what and how we send data
- Support service available when needed
- Capable of interfacing generically to any other system
- COST EFFECTIVE (free is a nice word)
- Our Selection:
 - MIRTH



Data Flow Diagram



Channel Definition



The screenshot displays the Mirth Connect web interface. The left sidebar contains navigation menus for Dashboard, Channels, Users, Settings, Alerts, Events, and Plugins. Below this are Channel Tasks (Refresh, Deploy All, Edit Global Scripts, Edit Code Templ..., New Channel, Import Channel, Export All Chan...) and Other (Help on this topic, About Mirth Con..., Visit mirthcorp.com, Report Issue, Logout). The main content area is titled 'Channels' and features a table with columns for Status, Protocol, Name, Id, and Description. The table lists ten channels, most of which are disabled. The status of each channel is indicated by a radio button: disabled channels have a grey radio button, and the 'Exit Care Hand Out - Spanish PartII' channel has a selected blue radio button.

Status	Protocol	Name	Id	Description
<input type="radio"/> Disabled	XML	Exit Care Hand Out - PartII	067f3aca-ad8e-4bf1-9597-3ffd6e2935cf	
<input type="radio"/> Disabled	HL7 v2.x	RHIO_CONSENT	54d4d159-017b-4c7b-8233-d1ccd95e896f	
<input type="radio"/> Disabled	HL7 v2.x	RHIO_FINAL	d5fc7818-a7a5-457b-a9d9-b57b55f416fb	
<input type="radio"/> Disabled	XML	RHIO_MEDS	543dcabb-514f-4a15-8b71-6139823adab5	
<input type="radio"/> Disabled	XML	Exit Care Hand Out	eec7b7dd-dd06-439e-b8e5-b15b19412fde	
<input type="radio"/> Disabled	XML	Exit Care Hand Out - Spanish	3f212c4e-8222-41ff-a351-ce7db236c1ad	
<input type="radio"/> Disabled	HL7 v2.x	MorrisHeight	142a59a5-dd94-452d-81c1-635e1c4e723d	
<input type="radio"/> Disabled	HL7 v2.x	RHIO_TEST	f1fdc84c-fae5-4553-a1c3-bf73b4510ab4	
<input type="radio"/> Disabled	XML	CONSENT BACK LOG	75050768-450a-491b-a0d0-33c7a0ae9032	
<input checked="" type="radio"/> Enabled	XML	Exit Care Hand Out - Spanish PartII	ea882c42-7cdb-4d38-8439-38b297adf2eb	

Connected to: <https://localhost:8443>

Input Connector

The screenshot displays the Mirth Connect web interface for editing a channel named 'RHIO_MEDS'. The interface is divided into a left sidebar and a main configuration area.

Left Sidebar:

- Mirth Connect:** Dashboard, Channels, Users, Settings, Alerts, Events, Plugins.
- Channel Tasks:** Validate Connector, Edit Filter, Edit Transformer (27), Import Connector, Export Connector, Export Channel.
- Other:** Help on this topic, About Mirth Connect, Visit mirthcorp.com, Report Issue, Logout.

Main Configuration Area:

Edit Channel - RHIO_MEDS (mirthconnect logo)

Summary | Source | Destinations | Scripts

Connector Type: Database Reader

Database Reader Configuration:

- Driver: Oracle 10g Release 2 (Insert URL Template)
- URL: jdbc:oracle:thin:@mhtemr:1521:emr
- Username: custom
- Password: [masked]
- Polling Type: Interval Time
- Polling Frequency (ms): 5000
- Polling Time (daily): 10:06 AM
- Process Results in Order: Yes No
- Use JavaScript: Yes No

Generate: Connection Select

SQL:

```
1 SELECT USRINFO_LOGINNAME AS USRINFO_LOGINNAME,
2 USRINFO_PVID AS USRINFO_PVID,
3 USRINFO_LASTNAME AS USRINFO_LASTNAME,
4 USRINFO_FIRSTNAME AS USRINFO_FIRSTNAME,
5 USRINFO_MIDDLENAME AS USRINFO_MIDDLENAME,
6 USRINFO_DEANUMBER AS USRINFO_DEANUMBER,
   USRINFO_LICNUMBER AS USRINFO_LICNUMBER,
```

Run On-Update Statment: Yes No

Generate: Connection Update

On-Update SQL:

```
1 UPDATE RXMEDS
2 SET RHIO_EXPORT_TIMESTAMP = TO_CHAR(SYSTIMESTAMP, 'HHMMSS.FF')
3 WHERE PRESCRIB_PTID=?(prescrib_ptid)
4
5
6
```

Field list on the right:

- usrinfo_loginnam
- usrinfo_pvid
- usrinfo_lastname
- usrinfo_firstname
- usrinfo_middlen
- usrinfo_deanum

Connected to: https://localhost:8443

Filters

Mirth Views

- Back to Channel

Filter Tasks

- Add New Rule
- Delete Rule
- Import Filter
- Export Filter
- Validate Script

Other

- Help on this topic
- About Mirth Connect
- Visit mirthcorp.com
- Report Issue
- Logout

Edit Channel - RHIO_FINAL - Consent Filter

#	Operator	Name	Type
0		New Rule	JavaScript

```
1 //basic database connection
2 var driver = "oracle.jdbc.OracleDriver";
3 var username = "rhio";
4 var password = "rhio";
5 var url = "jdbc:oracle:thin:@mhtemr:1521:rhio";
6 var dbConn = DatabaseConnectionFactory.createDatabaseConnection (driver, url, username, password);
7
8 //get the dcrId from PV1.19
9 var dcrId = msg['PV1']['PV1.19']['PV1.19.1'].toString();
10 logger.error("dcrId: "+dcrId);
11 //get the consent from RHIO_LOG_CONSENT
12 //there should always be one row
13 var consent = null;
14 var consentQuery = "SELECT * FROM RHIO_LOG_CONSENT WHERE DCRID="+dcrId;
15 var consentResult = dbConn.executeCachedQuery(consentQuery);
16 consentResult.first();
17 var con = consentResult.getString(5);
18 dbConn.close();
19 logger.error((con==null).toString());
20 if (con == null){
21     return false;
22 }
23 else {return true;}
24
25
26
27
28
29
30
31
32
```

Reference | Message Trees | Message Templates

Filter: All

- All
- Convert HL7 to XML
- Convert XML to HL7
- Convert X12 to XML
- Convert XML to X12
- Convert EDI to XML
- Convert XML to EDI
- Convert NCPDP to XML
- Convert XML to NCPDP
- Perform Database Query
- Perform Parameterized Database Query
- Perform Database Update
- Perform Parameterized Database Update

Available Variables

Visit Info

Connected to: https://localhost:8443

Transformers

Mirth Views

- Back to Channel

Transformer Tasks

- Add New Step
- Delete Step
- Import Transformer
- Export Transformer
- Validate Script

Other

- Help on this topic
- About Mirth Connect
- Visit mirthcorp.com
- Report Issue
- Logout

Edit Channel - RHIO_FINAL - Consent Transformer

#	Name	Type
0	New Step	JavaScript

```
1 msg['MSH']['MSH.9']['MSH.9.1'] = "ADT";
2 msg['MSH']['MSH.9']['MSH.9.2'] = "A08";
3 msg['EVN']['EVN.1']['EVN.1.1'] = "A08";
4
5 tmp['MSH'] = msg['MSH'];
6 tmp['PID'] = msg['PID'];
7 tmp['PV1'] = msg['PV1'];
8 tmp['EVN'] = msg['EVN'];
9
10 //basic database connection
11 var driver = "oracle.jdbc.OracleDriver";
12 var username = "rhio";
13 var password = "rhio";
14 var url = "jdbc:oracle:thin:@mhtear:1521:rhio";
15 var dbConn = DatabaseConnectionFactory.createDatabaseConnection (driver, url, username, password);
16
17 //get the dcrId from PV1.19
18 var dcrId = msg['PV1']['PV1.19']['PV1.19.1'].toString();
19
20 //get the consent from RHIO_LOG_CONSENT
21 //there should always be one row
22 var consent = null;
23 var consentQuery = "SELECT * FROM RHIO_LOG_CONSENT WHERE DCRID="+dcrId;
24 var consentResult = dbConn.executeCachedQuery(consentQuery);
25 consentResult.first();
26 var con = consentResult.getString(5);
27 if (con != null){
28     consent = createSegment('PD1',tmp,0);
29     consent['PD1.12']['PD1.12.1'] = con;
30 }
31 else {
32     delete tmp['PD1'];
33 }
34
```

Reference \ Message Trees \ Message Templates

Filter: All

- Convert HL7 to XML
- Convert XML to HL7
- Convert X12 to XML
- Convert XML to X12
- Convert EDI to XML
- Convert XML to EDI
- Convert NCPDP to XML
- Convert XML to NCPDP
- Perform Database Query
- Perform Parameterized Database Query
- Perform Database Update
- Perform Parameterized Database Update

Available Variables

Visit Info

Connected to: https://localhost:8443

Output Connectors

The screenshot displays the Mirth Connect web interface for editing a channel named 'RHIO_FINAL'. The interface is divided into several sections:

- Left Sidebar:** Contains navigation links for Dashboard, Channels, Users, Settings, Alerts, Events, and Plugins. Below this is a 'Channel Tasks' section with actions like 'Validate Connector', 'New Destination', 'Delete Destination', 'Clone Destination', 'Disable Destination', 'Move Dest. Down', 'Edit Filter', 'Edit Transformer (1)', 'Import Connector', 'Export Connector', and 'Export Channel'. At the bottom is an 'Other' section with links for help, about, visit mirthcorp.com, report issue, and logout.
- Header:** Shows 'Edit Channel - RHIO_FINAL' and the 'mirthconnect' logo.
- Navigation Tabs:** 'Summary', 'Source', 'Destinations', and 'Scripts'. The 'Destinations' tab is active.
- Table:** A table listing destinations with columns for Status, Destination, and Connector Type.

Status	Destination	Connector Type
Enabled	Visit Info	File Writer
Enabled	Consent	File Writer
Enabled	Problems	File Writer
Enabled	Allergy	File Writer
- Form Area:** Below the table, the 'Connector Type' is set to 'File Writer'. The 'File Writer' configuration includes:
 - Method: file (with a 'Test Write' button)
 - Directory: C:/James_Mirth_Test/Visit
 - File Name: \${ORIGINALNAME}.hl7
 - Anonymous: Yes (selected)
 - Username: anonymous
 - Password: masked with dots
 - Secure Mode: Yes (selected)
 - Passive Mode: Yes (selected)
 - Validate Connection: Yes (selected)
 - Append to file: Yes (selected)
 - File Type: ASCII (selected)
 - Encoding: Default
 - Template: \${message.encodedData}
- Destination Mappings:** A list of available mappings on the right side of the form, including Message ID, Raw Data, Transformed Data, Encoded Data, Message Source, Message Type, Message Version, Date, Formatted Date, Timestamp, Unique ID, Original File Name, Count, Entity Encoder, CDATA Tag, DICOM Message Raw Data, and Message with Attachment Data.
- Footer:** Shows 'Connected to: https://localhost:8443'.



Summary of Implementation

- Exporting a document when it's signed in the EMR
- Pull all information from EMR database that's related to the document in question
- Difficulties – finding the right documentation for coding
- Importance of project planning



Lessons Learned

- Interface - Learning curve (E4X JAVA script)
- Dynamically extract complex report-quality data from EMR production database using efficient SQL coding
- HL7 standard – not much of a standard
- Future projects
 - Interface engine to HRSA Careware
 - External EMR orders management tool
 - EMR handout reformatting and customization
 - Interface to Reporting (Data warehousing) and Business Intelligence (Oracle) solutions



Q & A





Credits

- Morris Heights Health Center
 - <http://www.mhhc.org/>
- Bronx RHIO
 - <http://www.bronxrhio.org/>
- Managing Interfaces with EMR
 - http://support.centricityservices.com/logician/emr_2005/documentation.html
- Mirth Web Site
 - www.mirthcorp.com
- HL7 Documentation
 - <http://www.hl7.org/implement/standards/index.cfm>

To request for additional information, demonstration, and collaborations on interface/EMR projects please email Ivan Pan at ipan@mhhc.org