IntegrateIT.us

Advinow Mirth Connect Channels Build Specifications

ver 2018.06.12

Revision History

Date	Author	Description
2018-06-04	Jeffrey Ritz	Initial version release.
2018-06-12	Jeffrey Ritz	Changed title, modified TOC and included the TCP CCD to File Writer channel specification and code templates.

Description

The purpose of this document is to provide the build specifications for the Mirth Connect (v3.5.0.8232) built channels (i.e. source and destination connectors, transformers, filters), code templates and other global configurations to support the Advinow requested interfaces functionality.

The HL7 to PostgreSQL Javascript channel and supporting configuration was developed to proper handle the receipt of Admit Discharge and Transfer (ADT) and Scheduling Information Unsolicited HL7 messages and updating the appropriate Mirth_Inbound schema tables.

The TCP CCD to File Writer channel, code templates and supportive configuration was developed to receive CCDA|CCD inbound documents to this channel inbound source receiver over TCP and will write these CCDA | CCDs to a specific local disk file location.

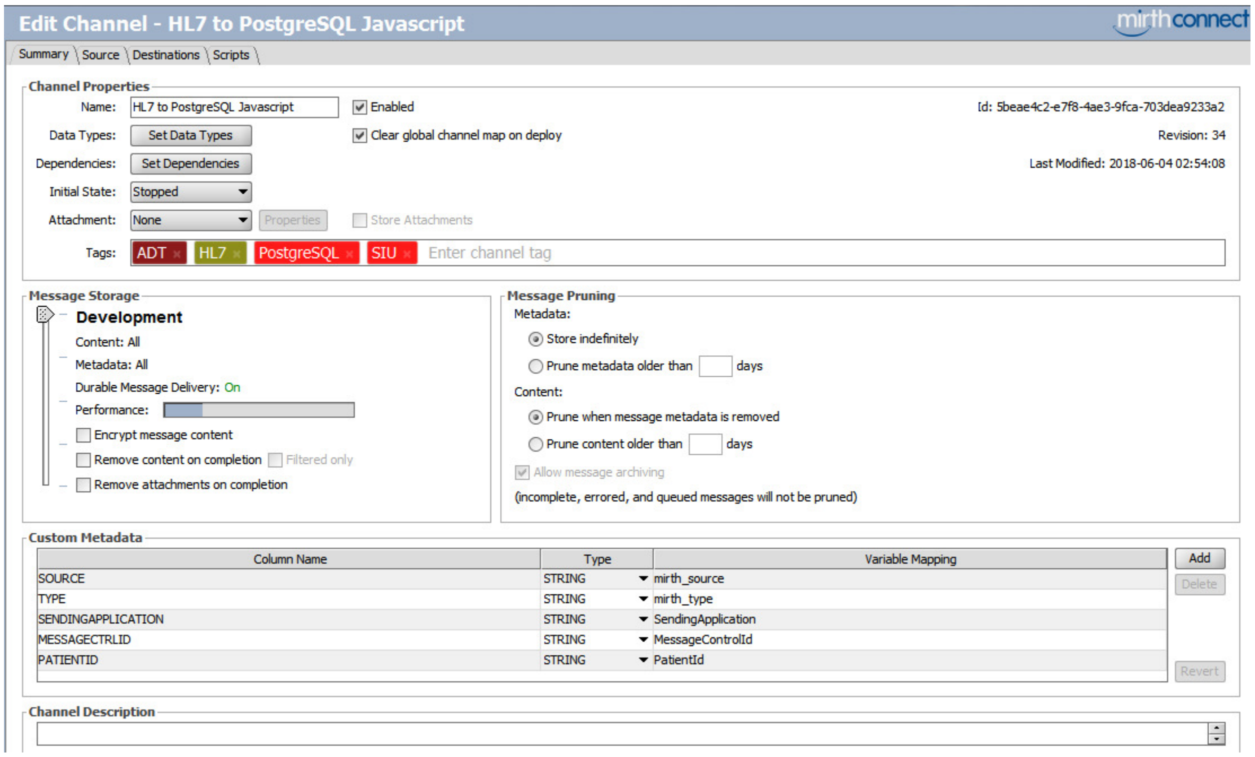
Definitions, acronyms, and abbreviations

The following table explains the definitions of terms, acronyms, and abbreviations that may appear in this manual.

Term	Meaning
ADT	Admission, Discharge and Transfer Messaging. The event driven set of ADT events provide the exchange of patient demographic information.
CCD	Continuity of Care Document. Continuity of Care Document (CCD) is an electronic document exchange standard for sharing patient summary information. Summaries include the most commonly needed pertinent information about current and past health status in a form that can be shared by all computer applications, including web browsers, electronic medical record (EMR) and electronic health record (EHR) software systems.
CCDA	Consolidated Clinical Document. Architecture. Consolidated CDA guide as "the single source for implementing the following CDA documents," including Continuity of Care Document (CCD), discharge summary, and many others.
HL7	Health Level Seven. HL7 refers to a set of international standards for transfer of clinical and administrative data between software applications used by various healthcare providers.
OID	Object identifier. An Object Identifier is a name used to identify an object.
SIU	Scheduled Information Unsolicited Messaging. The unsolicited transaction sets provide for the exchange of scheduling information between systems.
TCP	Transformation Control Protocol. The Transmission Control Protocol provides a communication service at an intermediate level between an application program and the Internet Protocol.
UTF-8	Unicode Transformation Format. The '8' means its 8-bit blocks to represent a character. UTF-8 is a character encoding capable of encoding all possible characters, or code points, defined by Unicode.
UUID	Universal Unique Identifier. A UUID is a 128-bit number used to uniquely identify some object or entity on the Internet.

Table of Contents

HL7 to PostgreSQL Javascript Channel Summary



Edit Channel - HL7 to PostgreSQL Javascript

Summary | Source | Destinations | Scripts

Channel Properties

Name: HL7 to PostgreSQL Javascript ☑ Enabled
Id: 5beae4c2-e7f8-4ae3-9fca-703dea9233a2
Data Types: Set Data Types ☑ Clear global channel map on deploy
Revision: 34
Dependencies: Set Dependencies
Last Modified: 2018-06-04 02:54:08
Initial State: Stopped
Attachment: None Properties ☐ Store Attachments
Tags: ADT HL7 PostgreSQL SIU Enter channel tag

Message Storage

Development
Content: All
Metadata: All
Durable Message Delivery: On
Performance:
☐ Encrypt message content
☐ Remove content on completion ☐ Filtered only
☐ Remove attachments on completion

Message Pruning

Metadata:
● Store indefinitely
○ Prune metadata older than days
Content:
● Prune when message metadata is removed
○ Prune content older than days
☑ Allow message archiving
(incomplete, errored, and queued messages will not be pruned)

Custom Metadata

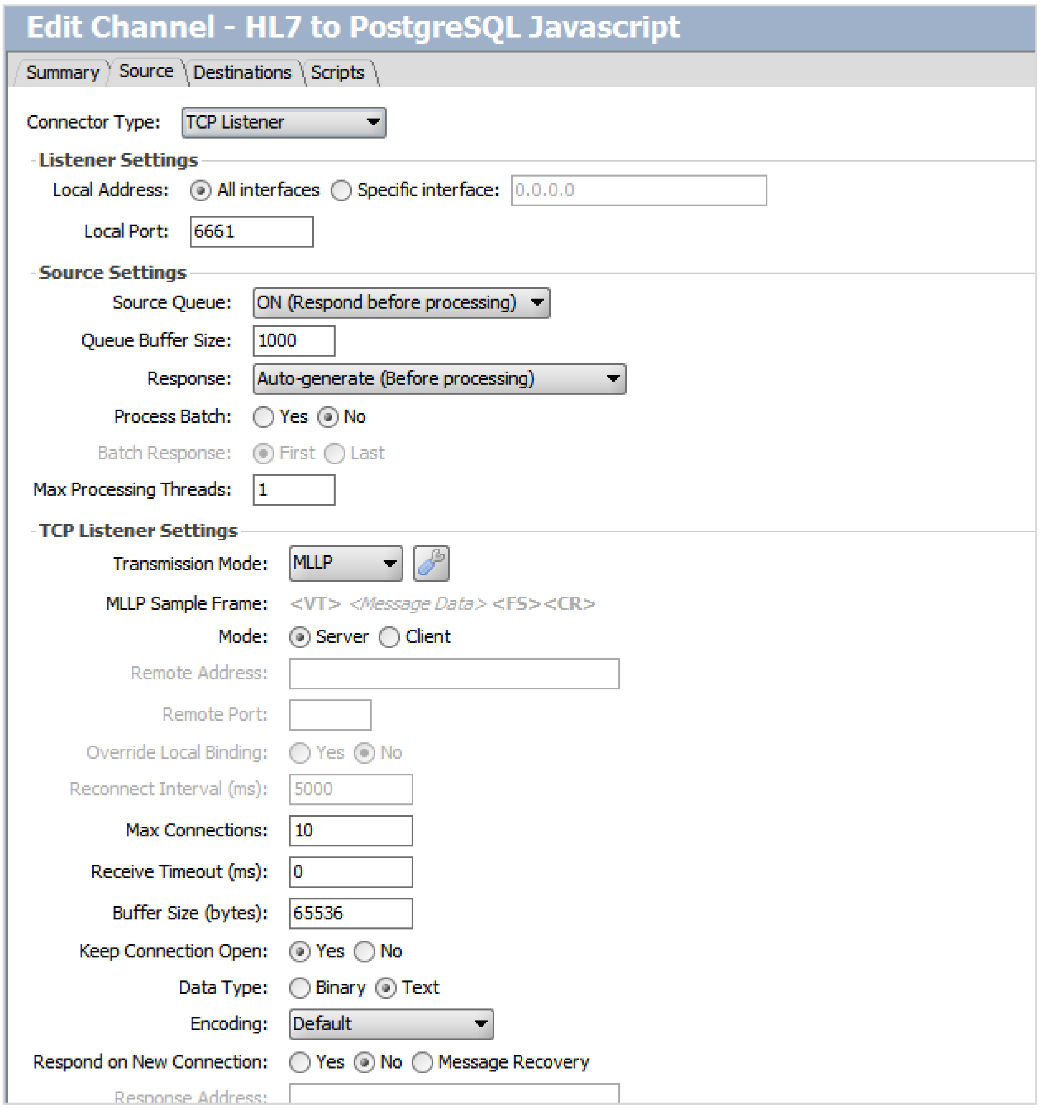
Column Name	Type	Variable Mapping
SOURCE	STRING	mirth_source
TYPE	STRING	mirth_type
SENDINGAPPLICATION	STRING	SendingApplication
MESSAGECTRLID	STRING	MessageControlId
PATIENTID	STRING	PatientId

Add Delete Revert

Channel Description

Name:	HL7 to PostgreSQL JavaScript
Set Data Types:	HL7 v2.x
Initiate State:	6661
Message Storage:	Development
Message Pruning:	
Metadata:	Store indefinitely
Content:	Prune when message metadata is removed
Custom Metadata:	
SENDINGAPPLICATION	SendingApplication
MESSAGECONTROLID	MessageControlId
PATIENTID	PatientId

Channel Source Connector


Edit Channel - HL7 to PostgreSQL Javascript

Connector Type: TCP Listener

Listener Settings

Local Address: [✓ All interfaces] Specific interface: 0.0.0.0

Local Port: 6661

Source Settings

Source Queue: ON (Respond before processing)

Queue Buffer Size: 1000

Response: Auto-generate (Before processing)

Process Batch: Yes [✓ No]

Batch Response: [✓ First] Last

Max Processing Threads: 1

TCP Listener Settings

Transmission Mode: MLLP

MLLP Sample Frame: <VT> *<Message Data>* <FS><CR>

Mode: [✓ Server] Client

Remote Address:

Remote Port:

Override Local Binding: Yes [✓ No]

Reconnect Interval (ms): 5000

Max Connections: 10

Receive Timeout (ms): 0

Buffer Size (bytes): 65536

Keep Connection Open: [✓ Yes] No

Data Type: Binary [✓ Text]

Encoding: Default

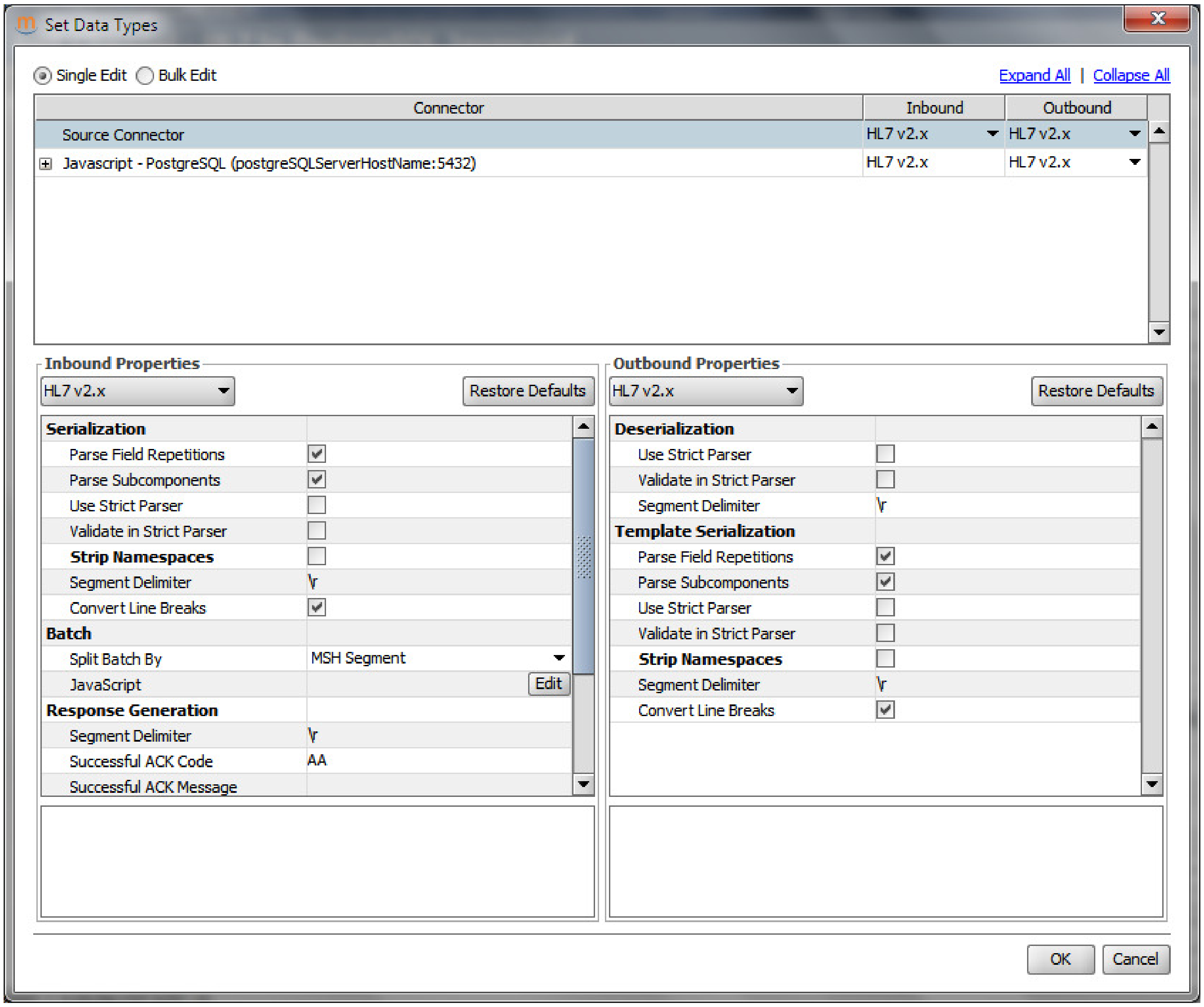
Respond on New Connection: Yes [✓ No] Message Recovery

Response Address:

Connector Type:	TCP Listener
Local Address:	All interfaces
Local Port:	6661
Source Queue:	ON (Respond before Processing)
Queue Buffer Size:	1000
Response:	Auto-generate (Before processing)
Process Batch:	No
Max Processing Threads:	1
TCP Listener Settings	
Transmission Mode:	MLLP
Mode:	Server
Max Connections:	10
Receive Timeout (ms):	0
Buffer Size (bytes):	65536
Keep Connection Open:	Yes
Data Type:	Text

Encoding:	Default
Respond on New	No

Source Connector – Set Data Type



Set Data Types

[✓ Single Edit] Bulk Edit

Expand All | Collapse All

Connector	Inbound	Outbound
Source Connector	HL7 v2.x	HL7 v2.x
⊞ Javascript - PostgreSQL (postgreSQLServerHostName:5432)	HL7 v2.x	HL7 v2.x

Inbound Properties

HL7 v2.x

Restore Defaults

Serialization	
Parse Field Repetitions	✓
Parse Subcomponents	✓
Use Strict Parser	☐
Validate in Strict Parser	☐
Strip Namespaces	☐
Segment Delimiter	\r
Convert Line Breaks	✓
Batch	
Split Batch By	MSH Segment
JavaScript	Edit
Response Generation	
Segment Delimiter	\r
Successful ACK Code	AA
Successful ACK Message	

Outbound Properties

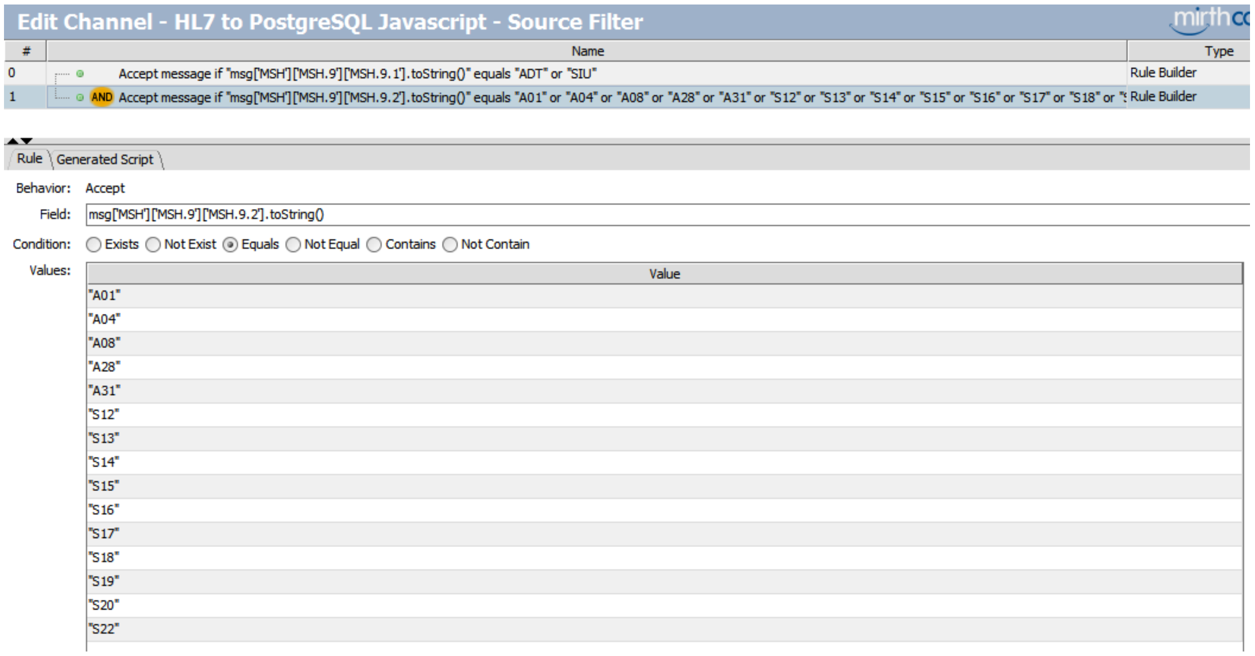
HL7 v2.x

Restore Defaults

Deserialization	
Use Strict Parser	☐
Validate in Strict Parser	☐
Segment Delimiter	\r
Template Serialization	
Parse Field Repetitions	✓
Parse Subcomponents	✓
Use Strict Parser	☐
Validate in Strict Parser	☐
Strip Namespaces	☐
Segment Delimiter	\r
Convert Line Breaks	✓

OK Cancel

Channel Source Filters



Edit Channel - HL7 to PostgreSQL Javascript - Source Filter

#	Name	Type
0	Accept message if "msg['MSH']['MSH.9']['MSH.9.1'].toString()" equals "ADT" or "SIU"	Rule Builder
1	AND Accept message if "msg['MSH']['MSH.9']['MSH.9.2'].toString()" equals "A01" or "A04" or "A08" or "A28" or "A31" or "S12" or "S13" or "S14" or "S15" or "S16" or "S17" or "S18" or "	Rule Builder

Rule | Generated Script

Behavior: Accept

Field: msg['MSH']['MSH.9']['MSH.9.2'].toString()

Condition: ( ) Exists ( ) Not Exist [✓ Equals] ( ) Not Equal ( ) Contains ( ) Not Contain

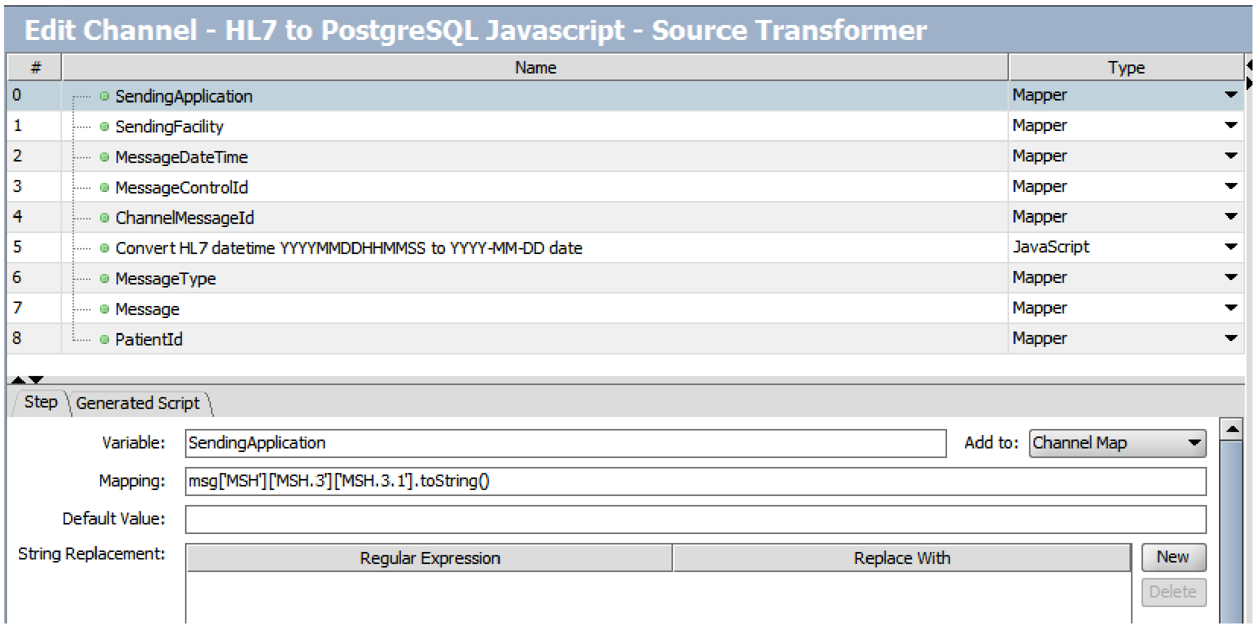
Values:

Value
"A01"
"A04"
"A08"
"A28"
"A31"
"S12"
"S13"
"S14"
"S15"
"S16"
"S17"
"S18"
"S19"
"S20"
"S22"

JavaScript Rule Builders

1. Accept message if ‘ADT’ or ‘SIU’
   AND
2. Accept message if ‘A01’, ‘A04’, ‘A08’, ‘A31’, ‘S12’, ‘S13’, ‘S14’, ‘S15’, ‘S16’, ‘S17’, ‘S18’, ‘S19’, ‘S20’, ‘S22’

Channel Source Transformers



Edit Channel - HL7 to PostgreSQL Javascript - Source Transformer

#	Name	Type
0	SendingApplication	Mapper
1	SendingFacility	Mapper
2	MessageDateTime	Mapper
3	MessageControlId	Mapper
4	ChannelMessageId	Mapper
5	Convert HL7 datetime YYYYMMDDHHMMSS to YYYY-MM-DD date	JavaScript
6	MessageType	Mapper
7	Message	Mapper
8	PatientId	Mapper

Step | Generated Script

Variable: SendingApplication
Add to: Channel Map
Mapping: msg['MSH']['MSH.3']['MSH.3.1'].toString()
Default Value:
String Replacement:

Regular Expression	Replace With

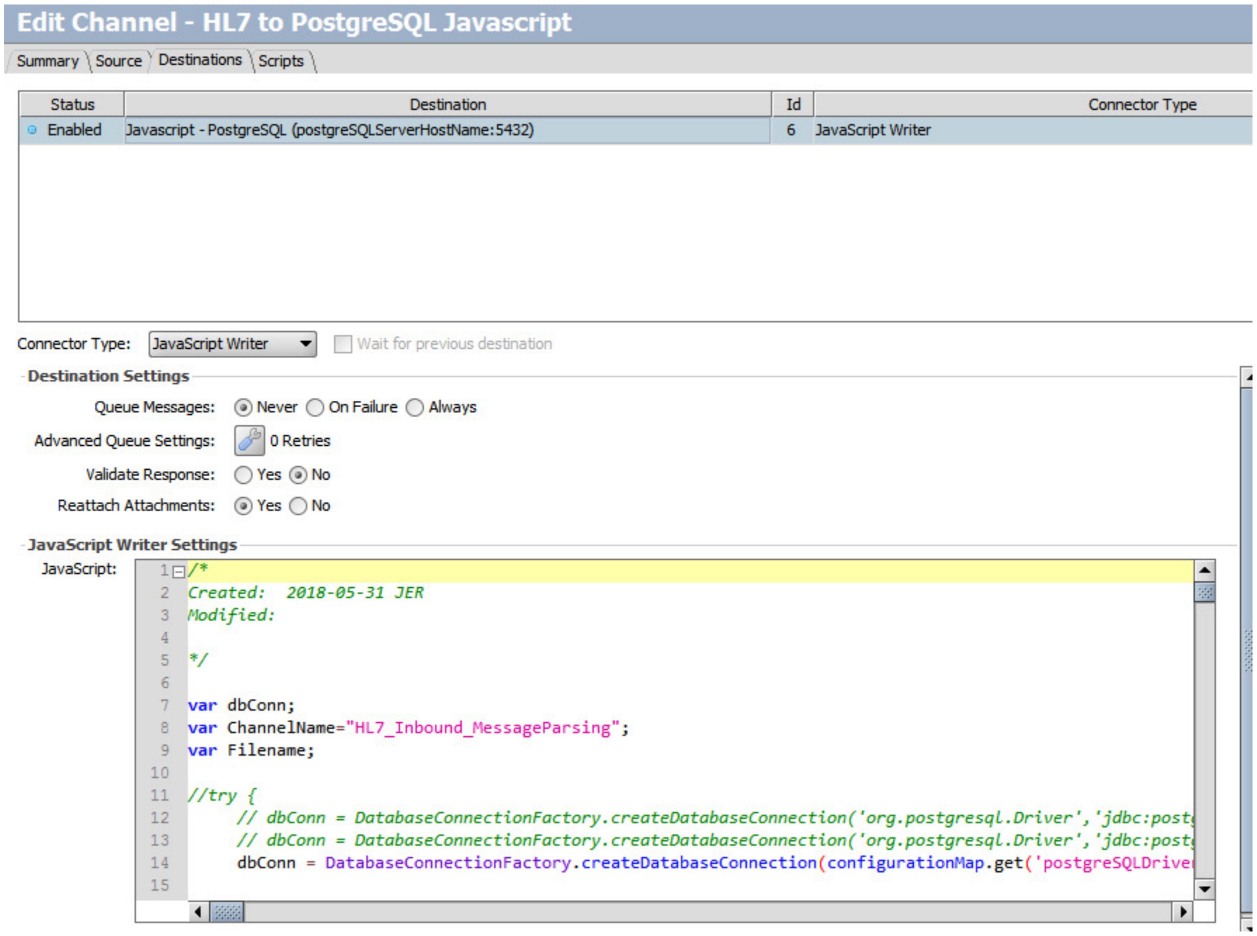
Transformer Descriptions

1.	SendingApplication	msg['MSH']['MSH.3']['MSH.3.1'].toString()
2.	SendingFacility	msg['MSH']['MSH.4']['MSH.4.1'].toString()
3.	MessageDateTime	msg['MSH']['MSH.7']['MSH.7.1'].toString()
4.	MessageControlId	msg['MSH']['MSH.10']['MSH.10.1'].toString()
5.	ChannelMessageId	connectorMessage.getMessageId()

6.	Convert HL7 datetime YYYYMMDDHHMMSS to YYYY-MM-DD date	//Convert date to consumable format
		//datetime will be yyyyMMddHHmmSS to yyyy-mm-dd
		var messageDate= convertDate($('MessageDateTime'),'yyyy-MM-dd');
		channelMap.put('MessageDate',messageDate);

7.	MessageType	msg['MSH']['MSH.9'].toString()
8.	Message	connectorMessage.getRawData().toString()
9.	PaitientId	msg['PID']['PID.3']['PID.3.1'].toString()

Channel Destination Connector


Edit Channel - HL7 to PostgreSQL Javascript

Summary | Source | Destinations | Scripts

Status	Destination	Id	Connector Type
Enabled	Javascript - PostgreSQL (postgreSQLServerHostName:5432)	6	JavaScript Writer

Connector Type: JavaScript Writer   Wait for previous destination

Destination Settings

Queue Messages: [✓ Never] On Failure Always

Advanced Queue Settings: 0 Retries

Validate Response: Yes [✓ No]

Reattach Attachments: [✓ Yes] No

JavaScript Writer Settings

JavaScript:

```
/*
Created:  2018-05-31 JER
Modified:

*/

var dbConn;
var ChannelName="HL7_Inbound_MessageParsing";
var Filename;

//try {
     // dbConn = DatabaseConnectionFactory.createDatabaseConnection('org.postgresql.Driver','jdbc:post
     // dbConn = DatabaseConnectionFactory.createDatabaseConnection('org.postgresql.Driver','jdbc:post
     dbConn = DatabaseConnectionFactory.createDatabaseConnection(configurationMap.get('postgreSQLDrive

```

Connector Type:	JavaScript Writer
Queue Messages:	Never
Advanced Queue Settings	0 Retries
Validate Response:	No
Reattach Attachments	Yes

JavaScript Writer Settings

```
/*
Desc:           JavaScript Writer – INSERT to Mirth_Inbound tables objects
Created:        2018-05-31 JER
Modified:
*/

var dbConn;
var ChannelName="HL7_Inbound_MessageParsing";
var Filename;

try {
        dbConn =
DatabaseConnectionFactory.createDatabaseConnection(configurationMap.get('postgreSQLDriverString'),config
urationMap.get('postgreSQLUrlHostString'),configurationMap.get('postgreSQLUser'),configurationMap.get('pos
tgreSQLPassword'));

        var result= dbConn.executeUpdate("INSERT INTO
\"Mirth_Inbound\".\"HL7_Message\"(\"sending_facility\",\"Channel_MessageId\",\"hl7message\",\"createdon
datetime\",\"status\",\"sending_application\",\"file_name\",\"data_direction\",\"msh_id\")
VALUES('"+$('SendingFacility')+"','"+$('ChannelMessageId')+"','"+$('Message')+"','"+$('MessageDate')+"','0','"+$
('SendingApplication')+"','"+$('originalFilename')+"','in','"+$('MessageControlId')+"')");

        var result = dbConn.executeCachedQuery("SELECT id AS HL7_InboundMessage_id,Sending_Facility AS
HL7_InboundMessage_sourcename, \"Channel_MessageId\" AS hl7messageid, hl7message AS
HL7_InboundMessage_hl7message, createdondatetime AS createdondatetime, status AS
HL7_InboundMessage_status,sending_application AS HL7_InboundMessage_Application ,file_name AS
HL7_File_Name FROM \"Mirth_Inbound\".\"HL7_Message\" where status=0 and data_direction='in' ");
        var resultSize = result.size();

        //      logger.info('result size in inbound parsing'+ resultSize);
        if(resultSize>0){

                for(var i = 0;i<resultSize;i++){
                        result.next();
                        var id = result.getString(1);
                        var SourceName = result.getString(2);
                        var HL7_id=result.getString(3);
                        var entry = result.getString(1) + "//" + result.getString(4);
                        var msg =result.getString(4);
                        var SendingApplication =result.getString(7);
                        Filename =result.getString(8);
                        var msgXML = new XML(SerializerFactory.getHL7Serializer().toXML(msg));
                        insertPID();
                        insertNK1();
                        insertPV1();
                        insertIN1();
                        insertGT1();
                        insertNTE();
                        insertMSH();
                        insertEVN();

                        logger.info("Row inbound parsing ==>" +entry);
                        var result2 = dbConn.executeUpdate("UPDATE \"Mirth_Inbound\".\"HL7_Message\"
SET status = 1 WHERE id="+id);
      }
        }

        // You may access this result below with $('column_name')
        return result;
}
catch(ex){
        var dateString = DateUtil.getCurrentDate('MM/dd/yyyy HH:mm:ss.hhh');
        var result = dbConn.executeUpdate("INSERT INTO
\"Mirth_Inbound\".\"HL7_Error_Log\"(\"Sending_Facility\",\"Channel_Message_Id\",\"Sending_Application\",\
"Error_Message\",\"Error_DateTime\",\"File_Name\",\"Channel_Name\",\"HL7_Message_Id\",data_direction)
values
```

```
('"+SourceName+"','"+HL7_id+"','"+SendingApplication+"','"+ex+"','"+dateString+"','"+Filename+"','"+ChannelN
ame+"','"+HL7_id+"','in')");
}
finally {
        if (dbConn) {
                dbConn.close();
        }
}

        function insertPID() {

                for each (seg in msgXML..PID) {
                        var PID_Set_ID = seg['PID.1']['PID.1.1'].toString().replace("'","''");
                        var PID_External_ID = seg['PID.2']['PID.2.1'].toString().replace("'","''");
                        var PID_Internal_ID = seg['PID.3']['PID.3.1'].toString().replace("'","''");
                        var PID_Alt_Patient_ID = seg['PID.4']['PID.4.1'].toString().replace("'","''");
                        var PID_Last_Name = seg['PID.5']['PID.5.1'].toString().replace("'","''");
                        var PID_Suffix = seg['PID.5']['PID.5.4'].toString().replace("'","''");
                        var PID_NameType = seg['PID.5']['PID.5.7'].toString().replace("'","''");
                        var PID_First_Name = seg['PID.5']['PID.5.2'].toString().replace("'","''");
                        var PID_Middle_Name = seg['PID.5']['PID.5.3'].toString().replace("'","''");
                        var PID_Mother_Maiden_Name = seg['PID.6']['PID.6.1'].toString().replace("'","''");
                        var PID_DOB = seg['PID.7']['PID.7.1'].toString().replace("'","''");
                        var PID_SEX = seg['PID.8']['PID.8.1'].toString().replace("'","''");
                        var PID_Race = seg['PID.10']['PID.10.1'].toString().replace("'","''");
                        var PID_Race_Text = seg['PID.10']['PID.10.2'].toString().replace("'","''");
                        var PID_Street = seg['PID.11']['PID.11.1'].toString().replace("'","''");
                        var PID_Country = seg['PID.11']['PID.11.6'].toString().replace("'","''");
                        var PID_Phone = seg['PID.13']['PID.13.1'].toString().replace("'","''");
                        var PID_Apartment = seg['PID.11']['PID.11.2'].toString().replace("'","''");
                        var PID_City = seg['PID.11']['PID.11.3'].toString().replace("'","''");
                        var PID_State = seg['PID.11']['PID.11.4'].toString().replace("'","''");
                        var PID_Zip = seg['PID.11']['PID.11.5'].toString().replace("'","''");
                        var PID_Email = seg['PID.13']['PID.13.4'].toString().replace("'","''");
                        var PID_AreaCode = seg['PID.13']['PID.13.6'].toString().replace("'","''");
                        var PID_LocalNumber = seg['PID.13']['PID.13.7'].toString().replace("'","''");
                        var PID_Work_Email = seg['PID.14']['PID.14.5'].toString().replace("'","''");
                        var PID_Work_AreaCode = seg['PID.14']['PID.14.6'].toString().replace("'","''");
                        var PID_Work_LocalNumber = seg['PID.14']['PID.14.7'].toString().replace("'","''");
                        var PID_Phone1 = seg['PID.13']['PID.13.1'].toString().replace("'","''");
                        var PID_Phone2 = seg['PID.14']['PID.14.1'].toString().replace("'","''");
                        var PID_Account_No = seg['PID.18']['PID.18.1'].toString().replace("'","''");
                        var PID_SSN = seg['PID.19']['PID.19.1'].toString().replace("'","''");
                        var PID_Ethnic_Group = seg['PID.22']['PID.22.1'].toString().replace("'","''");
                        var PID_Ethnic_Text = seg['PID.22']['PID.22.2'].toString().replace("'","''");
                        var PID_Birth_Place = seg['PID.23']['PID.23.1'].toString().replace("'","''");
                        var PID_Multiple_Birth_Indicator = seg['PID.24']['PID.24.1'].toString().replace("'","''");
                        var PID_Birth_Order = seg['PID.25']['PID.25.1'].toString().replace("'","''");
                        var PID_Patient_Death_Date_and_Time =
seg['PID.29']['PID.29.1'].toString().replace("'","''")
                        var PID_Patient_Death_Indicator = seg['PID.30']['PID.30.1'].toString().replace("'","''");
                        var PID_PrimaryLanguage = seg['PID.15']['PID.15.1'].toString().replace("'","''");
                        var PID_HL70005 = seg['PID.10']['PID.10.3'].toString().replace("'","''");
                        var PID_Address_Type = seg['PID.11']['PID.11.7'].toString().replace("'","''");
                        var PID_Home_PRN = seg['PID.13']['PID.13.2'].toString().replace("'","''");
                        var PID_Home_PH = seg['PID.13']['PID.13.3'].toString().replace("'","''");
                        var PID_Work_WPN = seg['PID.14']['PID.14.2'].toString().replace("'","''");
                        var PID_Work_PH = seg['PID.14']['PID.14.3'].toString().replace("'","''");
                        var PID_Languagetext = seg['PID.14']['PID.14.3'].toString().replace("'","''");
                        var PID_ISO0639 = seg['PID.15']['PID.15.3'].toString().replace("'","''");
                        var PID_HL70189 = seg['PID.22']['PID.22.3'].toString().replace("'","''");

                        var result = dbConn.executeUpdate("INSERT INTO
\"Mirth_Inbound\".\"HL7_PID_Segment\"(\"HL7_Message_Id\",\"Channel_Message_Id\",
\"First_Name\",\"Set_Id\",\"Patient_Id_Bc\",\"Patient_Id\",\"Alternative_Patient_Id_Bc\",\"Last_Name\",\"Mi
ddle_Initial\",\"Suffix\",\"Name_Type\",\"Mother's_Maiden_Name\",\"DOB\",\"Sex\",\"Race\",\"Race_Text\",
\"Address_Line_1\",\"Apartment\",\"City\",\"State\",\"Zip_Code\",\"Country\",\"Home_Phone_Number\",\"E
mail_Address\",\"Area_Code\",\"Local_Number\",\"Work_Email_Address\",\"Work_Phone_Number\",\"Work
_Area_Code\",\"Work_Local_Number\",\"Primary_Language\",\"Birth_Place\",\"Multiple_Birth_Indicator\",\"E
thnic_Group\",\"Ethnic_Text\",\"Patient_Account_Number\",\"SSN\",\"Patient_Death_Indicator\",\"Patient_D
eath_Date_And_Time\",\"Sending_Facility\",\"Sending_Application\",\"HL70005\",\"Address_Type\",\"Home_
PRN_PRS\",\"Home_PH_CP_Internet\",\"Work_WPN\",\"Work_PH_Internet\",\"Language_Text\",\"ISOo639\",
\"HL70189\",data_direction) VALUES("+id+","+HL7_id+",'"+PID_First_Name+"'
,'"+PID_Set_ID+"','"+PID_External_ID+"','"+PID_Internal_ID+"','"+PID_Alt_Patient_ID+"','"+PID_Last_Name+"','"
+PID_Middle_Name+"','"+PID_Suffix+"','"+PID_NameType+"','"+PID_Mother_Maiden_Name+"','"+PID_DOB+"','
"+PID_SEX+"','"+PID_Race+"','"+PID_Race_Text+"','"+PID_Street+"','"+PID_Apartment+"','"+PID_City+"','"+PID_
State+"','"+PID_Zip+"','"+PID_Country+"','"+PID_Phone1+"','"+PID_Email+"','"+PID_AreaCode+"','"+PID_LocalNu
mber+"','"+PID_Work_Email+"','"+PID_Phone2+"','"+PID_Work_AreaCode+"','"+PID_Work_LocalNumber+"','"+
PID_PrimaryLanguage+"','"+PID_Birth_Place+"','"+PID_Multiple_Birth_Indicator+"','"+PID_Ethnic_Group+"','"+P
ID_Ethnic_Text+"','"+PID_Account_No+"','"+PID_SSN+"','"+PID_Patient_Death_Indicator+"','"+PID_Patient_Dea
th_Date_and_Time+"','"+SourceName+"','"+SendingApplication+"','"+PID_HL70005+"','"+PID_Address_Type+"',
'"+PID_Home_PRN+"','"+PID_Home_PH+"','"+PID_Work_WPN+"','"+PID_Home_PH+"','"+PID_Languagetext+"','
"+PID_ISO0639+"','"+PID_HL70189+"','in')");
                }
        }

        function insertNK1() {

                for each (seg in msgXML..NK1) {
                        var NK1_Setid_Next_Of_Kin = seg['NK1.1']['NK1.1.1'].toString().replace("'","''");
                        var NK1_Next_Of_Kin_Last_Name = seg['NK1.2']['NK1.2.1'].toString().replace("'","''");
                        var NK1_Next_Of_Kin_First_Name = seg['NK1.2']['NK1.2.2'].toString().replace("'","''");
                        var NK1_Next_Of_Kin_Middle_Name =
seg['NK1.2']['NK1.2.3'].toString().replace("'","''");
                        var NK1_Next_Of_Kin_Relationship = seg['NK1.3']['NK1.3.1'].toString().replace("'","''");
                        var NK1_Next_Of_Kin_Address = seg['NK1.4']['NK1.4.1'].toString().replace("'","''");
                        var NK1_Next_Of_Kin_Apartment = seg['NK1.4']['NK1.4.2'].toString().replace("'","''");
                        var NK1_Next_Of_Kin_City = seg['NK1.4']['NK1.4.3'].toString().replace("'","''");
                        var NK1_Next_Of_Kin_State_Code = seg['NK1.4']['NK1.4.4'].toString().replace("'","''");
                        var NK1_Next_Of_Kin_Zip = seg['NK1.4']['NK1.4.5'].toString().replace("'","''");
                        var NK1_Next_Of_Kin_Country = seg['NK1.4']['NK1.4.6'].toString().replace("'","''");
                        var NK1_Home_Phone_Number = seg['NK1.5']['NK1.5.1'].toString().replace("'","''");
                        var NK1_Home_Phone_Email = seg['NK1.5']['NK1.5.3'].toString().replace("'","''");
                        var NK1_RelatioshipText = seg['NK1.3']['NK1.3.2'].toString().replace("'","''");
                        var NK1_HL70063 = seg['NK1.3']['NK1.3.3'].toString().replace("'","''");
                        var NK1_PRN_or_PRS = seg['NK1.5']['NK1.5.3'].toString().replace("'","''");
```

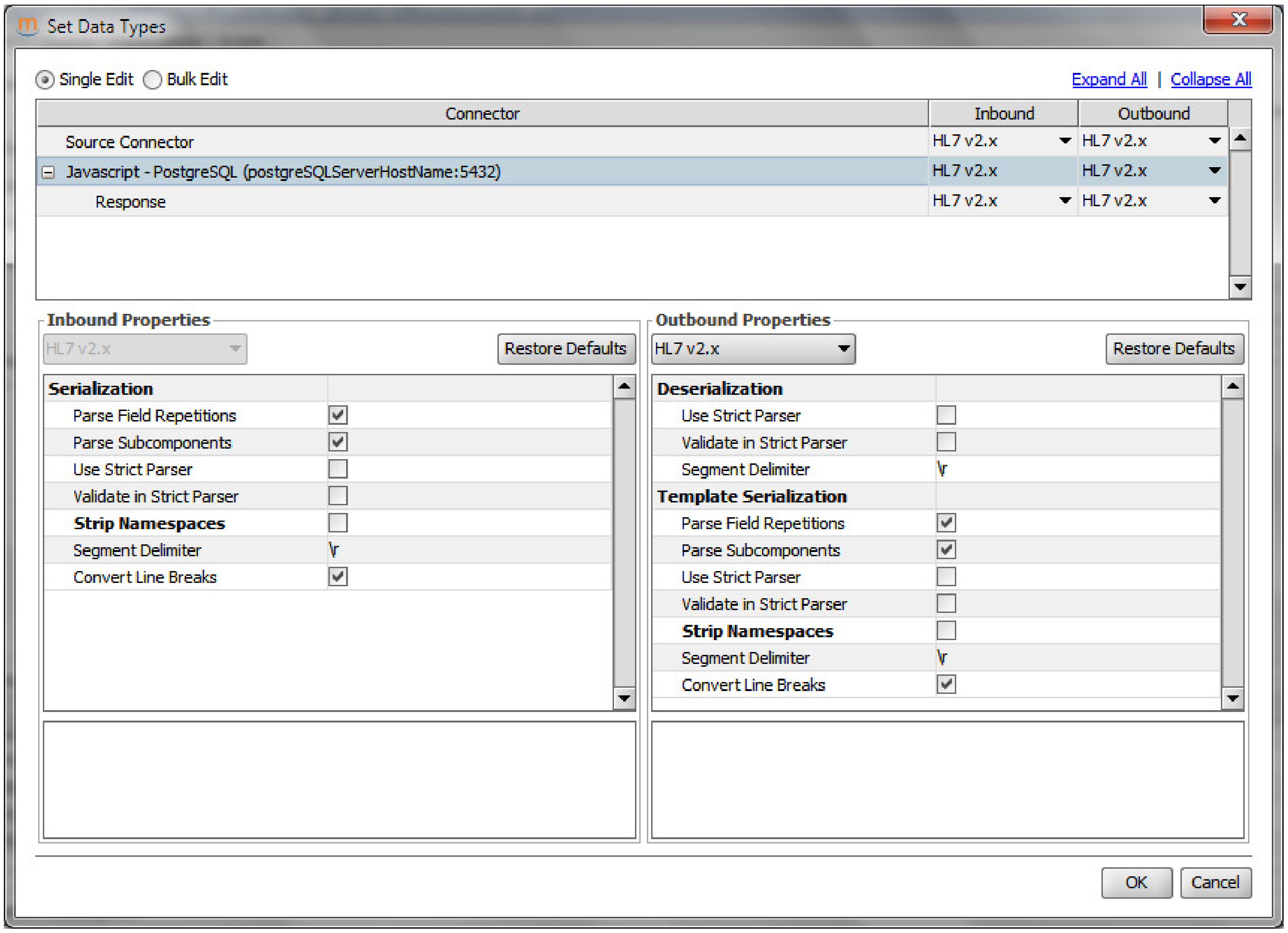
```
                    var result = dbConn.executeUpdate("INSERT INTO 
\"Mirth_Inbound\".\"HL7_NK1_Segment\"(\"set_id\",\"Last_Name\"
,\"First_Name\",\"Middle_Name\",\"Hl7_Relationship_Code\",\"Address_Line_1\",\"Apartment\",\"City\",\"St
ate\",\"zip_code\",\"Country\",\"Phone_Number\",\"PH_or_CP 
Internet\",\"HL7_Message_Id\",\"Sending_Facility\"
,\"Channel_Message_Id\",\"Sending_Application\",\"Relationship_Text\",\"HL70063\",\"PRN_or_PRS\",data_di
rection)values 
('"+NK1_Setid_Next_Of_Kin+"','"+NK1_Next_Of_Kin_Last_Name+"','"+NK1_Next_Of_Kin_First_Name+"','"+NK1
_Next_Of_Kin_Middle_Name+"','" + NK1_Next_Of_Kin_Relationship + 
"','"+NK1_Next_Of_Kin_Address+"','"+NK1_Next_Of_Kin_Apartment 
+"','"+NK1_Next_Of_Kin_City+"','"+NK1_Next_Of_Kin_State_Code+"','"+NK1_Next_Of_Kin_Zip+"','"+NK1_Next_
Of_Kin_Country 
+"','"+NK1_Home_Phone_Number+"','"+NK1_Home_Phone_Email+"','"+id+"','"+SourceName+"','"+HL7_id+"','"
+SendingApplication+"','"+NK1_RelatioshipText+"','"+NK1_HL70063+"','"+NK1_PRN_or_PRS+"','in')");
          }
     }

     function insertPV1() {

          for each (seg in msgXML..PV1) {
               var PV1_Set_ID_Patientid = seg['PV1.1']['PV1.1.1'].toString().replace("'","''");
               var PV1_Patient_Class = seg['PV1.2']['PV1.2.1'].toString().replace("'","''");
               var PV1_Point_Of_Care = seg['PV1.3']['PV1.3.1'].toString().replace("'","''");

               var PV1_Attending_Doctor_NPIID = seg['PV1.7']['PV1.7.1'].toString().replace("'","''");
               var PV1_Attending_Doctor_Last_Name = 
seg['PV1.7']['PV1.7.2'].toString().replace("'","''");
               var PV1_Attending_Doctor_First_Name = 
seg['PV1.7']['PV1.7.3'].toString().replace("'","''");
               var PV1_Attending_Doctor_Middle_Name = 
seg['PV1.7']['PV1.7.4'].toString().replace("'","''");
               var PV1_Attending_Doctor_Degree = seg['PV1.7']['PV1.7.7'].toString().replace("'","''");
               var PV1_Attending_Doctor_Professional_Suffix = 
seg['PV1.7']['PV1.7.21'].toString().replace("'","''");
               var PV1_Visit_Number = seg['PV1.19']['PV1.19.1'].toString().replace("'","''");
               var PV1_Servicing_Facility = seg['PV1.39']['PV1.39.1'].toString().replace("'","''");
               var PV1_Admit_Date_Time = seg['PV1.44']['PV1.44.1'].toString().replace("'","''");
               var PV1_Discharge_Date_Time = seg['PV1.45']['PV1.45.1'].toString().replace("'","''");
               var PV1_Visit_Indicator = seg['PV1.51']['PV1.51.1'].toString().replace("'","''");

               var result = dbConn.executeUpdate("INSERT INTO 
\"Mirth_Inbound\".\"HL7_PV1_Segment\"(\"Set_Id\",\"PatientClass\", 
\"Assigned_Location\",\"Attending_Doctor\",\"Last_Name\",\"First_Name\",\"Middle_Name\",\"Degree\",\"Pr
ofessional_Suffix\",\"Visitnumber\",\"Servicing_Facility\",\"Admit_Datetime\",\"Discharge_Datetime\",\"Visit_I
ndicator\",\"HL7_Message_Id\",\"Sending_Facility\",\"Channel_Message_Id\",\"Sending_Application\",data_di
rection) 
VALUES('"+PV1_Set_ID_Patientid+"','"+PV1_Patient_Class+"','"+PV1_Point_Of_Care+"','"+PV1_Attending_Doct
or_NPIID+"','"+PV1_Attending_Doctor_Last_Name+"','"+PV1_Attending_Doctor_First_Name+"','"+PV1_Attendi
ng_Doctor_Middle_Name+"','"+PV1_Attending_Doctor_Degree+"','"+PV1_Attending_Doctor_Professional_Suff
ix+"','"+PV1_Visit_Number+"','"+PV1_Servicing_Facility+"','"+PV1_Admit_Date_Time+"','"+PV1_Discharge_Date
_Time+"','"+PV1_Visit_Indicator+"','"+id+"','"+SourceName+"','"+HL7_id+"','"+SendingApplication+"','in')");
          }
```

```
    }

        function insertIN1() {

            for each (seg in msgXML..IN1) {
                var IN1_Setid_Insurance = seg['IN1.1']['IN1.1.1'].toString().replace("'","''");
                var IN1_ID_Number = seg['IN1.2']['IN1.2.1'].toString().replace("'","''");
                var IN1_Insurance_Company_Id = seg['IN1.3']['IN1.3.1'].toString().replace("'","''");
                var IN1_Insurance_Company_Name = seg['IN1.4']['IN1.4.1'].toString().replace("'","''");
                var IN1_Insurance_Company_Street = seg['IN1.5']['IN1.5.1'].toString().replace("'","''");
                var IN1_Insurance_Company_Street_2 =
seg['IN1.5']['IN1.5.2'].toString().replace("'","''");
                var IN1_Insurance_Company_City     = seg['IN1.5']['IN1.5.3'].toString().replace("'","''");
                var IN1_Insurance_Company_State    = seg['IN1.5']['IN1.5.4'].toString().replace("'","''");
                var IN1_Insurance_Company_Zip      = seg['IN1.5']['IN1.5.5'].toString().replace("'","''");
                var IN1_Insurance_Company_Contact_Person =
seg['IN1.6']['IN1.6.1'].toString().replace("'","''");
                var IN1_Insurance_Company_Contact_Phone_Number =
seg['IN1.7']['IN1.7.1'].toString().replace("'","''");
                var IN1_Insurance_Company_Contact_Email =
seg['IN1.7']['IN1.7.4'].toString().replace("'","''");
                var IN1_Insurance_Company_Contact_Areacode =
seg['IN1.7']['IN1.7.6'].toString().replace("'","''");
                var IN1_Insurance_Company_Contact_LocalNumber =
seg['IN1.7']['IN1.7.7'].toString().replace("'","''");
                var IN1_Group_Number = seg['IN1.8']['IN1.8.1'].toString().replace("'","''");
                var IN1_Group_Name = seg['IN1.9']['IN1.9.1'].toString().replace("'","''");
                var IN1_Plan_Effective_Date = seg['IN1.12']['IN1.12.1'].toString().replace("'","''");
                var IN1_Plan_Expiration_Date = seg['IN1.13']['IN1.13.1'].toString().replace("'","''");
                var IN1_Authorization_Number = seg['IN1.14']['IN1.14.1'].toString().replace("'","''");
                var IN1_Authorization_Date = seg['IN1.14']['IN1.14.2'].toString().replace("'","''");
                var IN1_Plan_Type = seg['IN1.15']['IN1.15.1'].toString().replace("'","''");
                var IN1_Insured_Last_Name = seg['IN1.16']['IN1.16.1'].toString().replace("'","''");
                var IN1_Insured_First_Name = seg['IN1.16']['IN1.16.2'].toString().replace("'","''");
                var IN1_Insured_Middle_Name = seg['IN1.16']['IN1.16.3'].toString().replace("'","''");
                var IN1_Insured_Suffix = seg['IN1.16']['IN1.16.4'].toString().replace("'","''");
                var IN1_Insured_Relationship_Code =
seg['IN1.17']['IN1.17.1'].toString().replace("'","''");
                var IN1_Insured_Relationship_Text = seg['IN1.17']['IN1.17.2'].toString().replace("'","''");
                var IN1_Insured_DOB = seg['IN1.18']['IN1.18.1'].toString().replace("'","''");
                var IN1_Insured_Street = seg['IN1.19']['IN1.19.1'].toString().replace("'","''");
                var IN1_Insured_City  = seg['IN1.19']['IN1.19.3'].toString().replace("'","''");
                var IN1_Insured_State  = seg['IN1.19']['IN1.19.4'].toString().replace("'","''");
                var IN1_Insured_Zip = seg['IN1.19']['IN1.19.5'].toString().replace("'","''");
                var IN1_Insured_Policy_Number  = seg['IN1.36']['IN1.36.1'].toString().replace("'","''");
                var IN1_Policy_Deductable = seg['IN1.37']['IN1.37.1'].toString().replace("'","''");
                var IN1_Insureds_Sex  = seg['IN1.43']['IN1.43.1'].toString().replace("'","''");
                var IN1_Signature_Code  = seg['IN1.50']['IN1.50.1'].toString().replace("'","''");
                var IN1_Signature_Code_Date  = seg['IN1.42']['IN1.42.1'].toString().replace("'","''");
                var IN1_Coordination_Of_Number  = seg['IN1.22']['IN1.22.1'].toString().replace("'","''");
                var IN1_WPN  = seg['IN1.7']['IN1.7.2'].toString().replace("'","''");
                var IN1_PH_FX_Internet  = seg['IN1.7']['IN1.7.3'].toString().replace("'","''");
                var IN1_HL70063  = seg['IN1.7']['IN1.7.3'].toString().replace("'","''");
				var result = dbConn.executeUpdate("INSERT INTO
\"Mirth_Inbound\".\"HL7_IN1_Segment\"(\"Set_Id\",\"Insurance_Plan_Id\",\"Insurance_Company_Id\",\"Insur
ance_Company_Name\",\"Street_Line_1\",\"Street_Line_2_And_Suite_Number\",\"City\",\"State\",\"Zip_Cod
e\",\"Insurance_Company_Contact_Person\",\"Insurance_Company_Contact_Phonenumber\",\"Email_Addres
s\",\"Area_Code\",\"Local_Number\",\"HL7_Message_Id\",\"Sending_Facility\",\"Channel_Message_Id\",\"Gro
up_Number\",\"Group_Name\",\"Plan_Effective_Date\",\"Plan_Expiration_Date\",\"Authorization_Number\",
\"Authorization_Date\",\"Plantype\",\"Insured’s_Last_Name\",\"First_Name\",\"Second_Or_Further_Given_N
ame\",\"Suffix\",\"Relationship_Code\",\"Relationship_Text\",\"Insurde’s_DOB\",\"Insurde’s_Address_Street\"
,\"Insurde’s_City\",\"Insurde’s_State\",\"Insurde’s_Zip_Code\",\"Coordination_Of_Benifits\",\"Policy_Number\
",\"Policy_Deductible\",\"Insurde’s_Sex\",\"Signature_Code\",\"Signature_Date\",\"Sending_Application\",\"W
PN\",\"PH_FX_Internet\",\"HL70063\",data_direction)VALUES('"+IN1_Setid_Insurance+"','"+IN1_ID_Number+"'
,'"+IN1_Insurance_Company_Id+"','"+IN1_Insurance_Company_Name+"','"+IN1_Insurance_Company_Street+"'
,'"+IN1_Insurance_Company_Street_2+"','"+IN1_Insurance_Company_City+"','"+IN1_Insurance_Company_Stat
e+"','"+IN1_Insurance_Company_Zip+"','"+IN1_Insurance_Company_Contact_Person+"','"+IN1_Insurance_Com
pany_Contact_Phone_Number+"','"+IN1_Insurance_Company_Contact_Email+"','"+IN1_Insurance_Company_C
ontact_Areacode+"','"+IN1_Insurance_Company_Contact_LocalNumber+"','"+id+"','"+SourceName+"','"+HL7_i
d+"','"+IN1_Group_Number+"','"+IN1_Group_Name+"','"+IN1_Plan_Effective_Date+"','"+IN1_Plan_Expiration_
Date+"','"+IN1_Authorization_Number+"','"+IN1_Authorization_Date+"','"+IN1_Plan_Type+"','"+IN1_Insured_L
ast_Name+"','"+IN1_Insured_First_Name+"','"+IN1_Insured_Middle_Name+"','"+IN1_Insured_Suffix+"','"+IN1_I
nsured_Relationship_Code+"','"+IN1_Insured_Relationship_Text+"','"+IN1_Insured_DOB+"','"+IN1_Insured_Str
eet+"','"+IN1_Insured_City+"','"+IN1_Insured_State+"','"+IN1_Insured_Zip+"','"+IN1_Coordination_Of_Number
+"','"+IN1_Insured_Policy_Number+"','"+IN1_Policy_Deductable+"','"+IN1_Insureds_Sex+"','"+IN1_Signature_C
ode+"','"+IN1_Signature_Code_Date+"','"+SendingApplication+"','"+IN1_WPN+"','"+IN1_PH_FX_Internet+"','"+I
N1_HL70063+"','in')");
			}
		}

		function insertGT1() {

			for each (seg in msgXML..GT1) {
				var GT1_Setid_Guarantor = seg['GT1.1']['GT1.1.1'].toString().replace("'","''");
				var GT1_Guarantor_Number = seg['GT1.2']['GT1.2.1'].toString().replace("'","''");
				var GT1_Guarantor_Last_Name = seg['GT1.3']['GT1.3.1'].toString().replace("'","''");
				var GT1_Guarantor_First_Name = seg['GT1.3']['GT1.3.2'].toString().replace("'","''");
				var GT1_Guarantor_Middle_Name = seg['GT1.3']['GT1.3.3'].toString().replace("'","''");
				var GT1_Guarantor_Suffix = seg['GT1.3']['GT1.3.4'].toString().replace("'","''");
				var GT1_Guarantor_NameType = seg['GT1.3']['GT1.3.7'].toString().replace("'","''");
				var GT1_Guarantor_Address1 = seg['GT1.5']['GT1.5.1'].toString().replace("'","''");
				var GT1_Guarantor_Address2 = seg['GT1.5']['GT1.5.2'].toString().replace("'","''");
				var GT1_Gurantor_City = seg['GT1.5']['GT1.5.3'].toString().replace("'","''");
				var GT1_Gurantor_State = seg['GT1.5']['GT1.5.4'].toString().replace("'","''");
				var GT1_Gurantor_Zip = seg['GT1.5']['GT1.5.5'].toString().replace("'","''");
				var GT1_Gurantor_Country = seg['GT1.5']['GT1.5.6'].toString().replace("'","''");
				var GT1_Gurantor_Home_Phone_Number =
seg['GT1.6']['GT1.6.1'].toString().replace("'","''");
				var GT1_Gurantor_Home_EmailAddress =
seg['GT1.6']['GT1.6.4'].toString().replace("'","''");
				var GT1_Gurantor_AreaCode= seg['GT1.6']['GT1.6.6'].toString().replace("'","''");
				var GT1_Gurantor_LocalNumber= seg['GT1.6']['GT1.6.7'].toString().replace("'","''");
				var GT1_Gurantor_Business_Phone_Number =
seg['GT1.7']['GT1.7.1'].toString().replace("'","''");
```

```
                    var GT1_Gurantor_Business_EmailAddress = 
seg['GT1.7']['GT1.7.4'].toString().replace("'","''");
                    var GT1_Gurantor_Business_AreaCode= 
seg['GT1.7']['GT1.7.6'].toString().replace("'","''");
                    var GT1_Gurantor_Business_LocalNumber= 
seg['GT1.7']['GT1.7.7'].toString().replace("'","''");
                    var GT1_Gurantor_DOB = seg['GT1.8']['GT1.8.1'].toString().replace("'","''");
                    var GT1_Gurantor_Sex = seg['GT1.9']['GT1.9.1'].toString().replace("'","''");
                    var GT1_Relationship_Id = seg['GT1.11']['GT1.11.1'].toString().replace("'","''");
                    var GT1_Relationship_Text = seg['GT1.11']['GT1.11.2'].toString().replace("'","''");
                    var GT1_Relationship_Coding_System = 
seg['GT1.11']['GT1.11.3'].toString().replace("'","''");
                    var GT1_Gurantor_SSN = seg['GT1.12']['GT1.12.1'].toString().replace("'","''");
                    var GT1_Gurtantor_Employer_Address = 
seg['GT1.17']['GT1.17.1'].toString().replace("'","''");
                    var GT1_Gurtantor_Employer_City = 
seg['GT1.17']['GT1.17.3'].toString().replace("'","''");
                    var GT1_Gurtantor_Employer_State = 
seg['GT1.17']['GT1.17.4'].toString().replace("'","''");
                    var GT1_Gurtantor_Employer_Zip = seg['GT1.17']['GT1.17.5'].toString().replace("'","''");
                    var GT1_Gurtantor_Employer_Phone = 
seg['GT1.18']['GT1.17.1'].toString().replace("'","''");
                    var GT1_Gurtantor_Marital_Status = 
seg['GT1.30']['GT1.30.1'].toString().replace("'","''");
                    var GT1_Gurtantor_Primary_Language = 
seg['GT1.36']['GT1.36.1'].toString().replace("'","''");
                    var GT1_Gurtantor_Language_Text = 
seg['GT1.36']['GT1.36.2'].toString().replace("'","''");
                    var GT1_Motherss_Maiden_Name = 
seg['GT1.42']['GT1.42.1'].toString().replace("'","''");
                    var GT1_Gurtantor_Ethinic_Group = 
seg['GT1.44']['GT1.44.1'].toString().replace("'","''");
                    var GT1_Gurtantor_Ethinic_Text = seg['GT1.44']['GT1.44.2'].toString().replace("'","''");
                    var GT1_Contact_Persons_Name = seg['GT1.45']['GT1.45.1'].toString().replace("'","''");
                    var GT1_Contact_Persons_Phone = seg['GT1.46']['GT1.46.1'].toString().replace("'","''");
                    var GT1_Contact_Relationship = seg['GT1.48']['GT1.48.1'].toString().replace("'","''");
                    var GT1_Gurtantor_Employer_organization = 
seg['GT1.51']['GT1.51.1'].toString().replace("'","''");
                    var GT1_Race = seg['GT1.55']['GT1.55.1'].toString().replace("'","''");
                    var GT1_Race_Text = seg['GT1.55']['GT1.55.2'].toString().replace("'","''");
                    var GT1_Gurtantor_Birth_Place = seg['GT1.56']['GT1.56.1'].toString().replace("'","''");
                    var GT1_Gurtantor_Birth_Place = seg['GT1.56']['GT1.56.1'].toString().replace("'","''");
                    var GT1_Home_PRN_PRS = seg['GT1.6']['GT1.6.2'].toString().replace("'","''");
                    var GT1_Home_PH = seg['GT1.6']['GT1.6.3'].toString().replace("'","''");
                    var GT1_WPN = seg['GT1.7']['GT1.7.2'].toString().replace("'","''");
                    var GT1_LanguageText = seg['GT1.36']['GT1.36.2'].toString().replace("'","''");
                    var GT1_ISOO639 = seg['GT1.36']['GT1.36.3'].toString().replace("'","''");
                    var GT1_HL70187 = seg['GT1.44']['GT1.44.3'].toString().replace("'","''");
                    var GT1_HL70005 = seg['GT1.55']['GT1.55.3'].toString().replace("'","''");
                    var GT1_PH_Internet = seg['GT1.7']['GT1.7.3'].toString().replace("'","''");

                    var result = dbConn.executeUpdate("INSERT INTO 
\"Mirth_Inbound\".\"HL7_GT1_Segment\"(\"Set_Id\",\"Patient_Id_Bc\",\"Last_Name\",\"First_Name\",\"Midd
le_Initial\",\"Suffix\",\"Name_Type\",\"Address_Line_1\",\"Apartment\",\"City\",\"State\",\"Zip_Code\",\"Cou
ntry\",\"Home_Phone_Number\",\"Email_Address\",\"Area_Code\",\"Local_Number\",\"Work_Phone_Numbe
r\",\"Work_Email_Address\",\"Work_Area_Code\",\"Work_Local_Number\",\"Guarantor_Employee_Address_
Street\",\"Guarantor_Gmployee_Address_City\",\"Guarantor_Employee_Address_State\",\"Guarantor_Employ
ee_Address_Zip\",\"Guarantor_Employee_Phone\",\"Guarantor_Martial_Status\",\"Primary_Language\",\"Con
tact_Person_Name\",\"Contact_Person_Phone\",\"Ethnic_Group\",\"Ethnic_Text\",\"Contact_Relationship\",\
"SSN\",\"Guarantor_Employer's_Organization_Name\",\"Guarantor_Birthplace\",\"Race\",\"Race_Text\",\"HL7
_Message_Id\",\"Sending_Facility\",\"Channel_Message_Id\",\"Mother's_Maiden_Name\",\"Sex\",\"Identifier
\",\"Text\",\"Coding_System\",\"Sending_Application\",\"Home_PRN_PRS\",\"Home_PH_CP_Internet\",\"Wor
k_WPN\",\"Language_Text\",\"ISO0639\",\"HL70187\",\"HL70005\",\"Work_PH_Internet\",data_direction)VAL
UES('"+GT1_Setid_Guarantor+"','"+GT1_Guarantor_Number+"','"+GT1_Guarantor_Last_Name+"','"+GT1_Guara
ntor_First_Name+"','"+GT1_Guarantor_Middle_Name+"','"+GT1_Guarantor_Suffix
+"','"+GT1_Guarantor_NameType+"','"+GT1_Guarantor_Address1+"','"+GT1_Guarantor_Address2+"','"+GT1_G
urantor_City+"','"+GT1_Gurantor_State+"','"+GT1_Gurantor_Zip+"','"+GT1_Gurantor_Country+"','"+GT1_Guran
tor_Home_Phone_Number+"','"+GT1_Gurantor_Home_EmailAddress+"','"+GT1_Gurantor_AreaCode+"','"+GT1
_Gurantor_LocalNumber+"','"+GT1_Gurantor_Business_Phone_Number+"','"+GT1_Gurantor_Business_EmailA
ddress+"','"+GT1_Gurantor_Business_AreaCode+"','"+GT1_Gurantor_Business_LocalNumber+"','"+GT1_Gurtan
tor_Employer_Address+"','"+GT1_Gurtantor_Employer_City+"','"+GT1_Gurtantor_Employer_State+"','"+GT1_G
urtantor_Employer_Zip+"','"+GT1_Gurtantor_Employer_Phone+"','"+GT1_Gurtantor_Marital_Status+"','"+GT1_
Gurtantor_Primary_Language+"','"+GT1_Contact_Persons_Name+"','"+GT1_Contact_Persons_Phone+"','"+GT1
_Gurtantor_Ethinic_Group+"','"+GT1_Gurtantor_Ethinic_Text+"','"+GT1_Contact_Relationship+"','"+GT1_Guran
tor_SSN+"','"+GT1_Gurtantor_Employer_organization+"','"+GT1_Gurtantor_Birth_Place+"','"+GT1_Race+"','"+G
T1_Race_Text+"','"+id+"','"+SourceName+"','"+HL7_id+"','"+GT1_Motherss_Maiden_Name+"','"+GT1_Gurantor
_Sex+"','"+GT1_Relationship_Id+"','"+GT1_Relationship_Text+"','"+GT1_Relationship_Coding_System+"','"+Sen
dingApplication+"','"+GT1_Home_PRN_PRS+"','"+GT1_Home_PH+"','"+GT1_WPN+"','"+GT1_LanguageText+"','"
+GT1_ISOO639+"','"+GT1_HL70187+"','"+GT1_HL70005+"','"+GT1_PH_Internet+"','in')");
                }
        }

        function insertNTE() {

                for each (seg in msgXML..NTE) {
                        var NTE_Setid = seg['NTE.1']['NTE.1.1'].toString().replace("'","''");
                        var NTE_Comment = seg['NTE.3']['NTE.3.1'].toString().replace("'","''");
                        logger.info('NTE_Set_iD: '+NTE_Setid);
                        logger.info('NTE_Comment: '+NTE_Comment);

                        var result = dbConn.executeUpdate("INSERT INTO
\"Mirth_Inbound\".\"HL7_NTE_Segment\"(\"Set_id\",\"Comment\",\"HL7_Message_Id\",\"Sending_Facility\",\
"Channel_Message_Id\",\"Sending_Application\",data_direction)values
('"+NTE_Setid+"','"+NTE_Comment+"','"+id+"','"+SourceName+"','"+HL7_id+"','"+SendingApplication+"','in')");
                }
        }

        function insertMSH() {

                for each (seg in msgXML..MSH) {
                        var MSH_Field_Separator = seg['MSH.1'].toString().replace("'","''");
                        var MSH_Encoding_Characters = seg['MSH.2'].toString().replace("'","''");
                        var MSH_Sending_Application = seg['MSH.3']['MSH.3.1'].toString().replace("'","''");
                        var MSH_Sending_Facility  = seg['MSH.4']['MSH.4.1'].toString().replace("'","''");
                        var MSH_Receiving_Application = seg['MSH.5']['MSH.5.1'].toString().replace("'","''");
                        var MSH_Receiving_Facility = seg['MSH.6']['MSH.6.1'].toString().replace("'","''");
```

```
            var MSH_Msg_DateTime = seg['MSH.7']['MSH.7.1'].toString().replace("'","''");
            var MSH_Msg_Type = seg['MSH.9']['MSH.9.1'].toString().replace("'","''");
            var MSH_Msg_Ctrl_Id = seg['MSH.10']['MSH.10.1'].toString().replace("'","''");
            var MSH_Processing_Type = seg['MSH.11']['MSH.11.1'].toString().replace("'","''");
            var MSH_Version_Id = seg['MSH.12']['MSH.12.1'].toString().replace("'","''");
            var MSH_Accept_Ack_Type_ID = seg['MSH.15']['MSH.15.1'].toString().replace("'","''");
            var MSH_Application_Ack_Type = seg['MSH.16']['MSH.16.1'].toString().replace("'","''");
            var MSH_Organization_Name = seg['MSH.22']['MSH.22.1'].toString().replace("'","''");
            var MSH_Type_Code = seg['MSH.22']['MSH.22.2'].toString().replace("'","''");

            var result = dbConn.executeUpdate("INSERT INTO
\"Mirth_Inbound\".\"HL7_MSH_Segment\"(\"Field_Separator\",\"Encoding_Character\",\"HL7_Message_Id\",\
"Channel_Message_Id\",\"Sending_Application\",\"Sending_Facility\",\"Receiving_Application\",\"Datetime\",\
"Message_Type\",\"Messagecontrol_Id\",\"Processing_Id\",\"Version_Id\",\"Accept_Acknowledgement_Type\
",\"Application_Acknowledgement_Type\",\"Organization_Name\",\"Type_Code\",data_direction)
VALUES('"+MSH_Field_Separator+"','"+MSH_Encoding_Characters+"','"+id+"','"+HL7_id+"','"+MSH_Sending_Ap
plication+"','"+MSH_Sending_Facility+"','"+MSH_Receiving_Application+"','"+MSH_Msg_DateTime+"','"+MSH_
Msg_Type+"','"+MSH_Msg_Ctrl_Id+"','"+MSH_Processing_Type+"','"+MSH_Version_Id+"','"+MSH_Application_
Ack_Type+"','"+MSH_Accept_Ack_Type_ID+"','"+MSH_Organization_Name+"','"+MSH_Type_Code+"','in')");
        }
    }

    function insertEVN() {

        for each (seg in msgXML..EVN) {

            var EVN_Event_Type_Code = seg['EVN.1']['EVN.1.1'].toString().replace("'","''");
            var EVN_Date_Time_Of_Event = seg['EVN.2']['EVN.2.1'].toString().replace("'","''");
            var EVN_Operator_Id = seg['EVN.5']['EVN.5.1'].toString().replace("'","''");

            var result = dbConn.executeUpdate("INSERT INTO
\"Mirth_Inbound\".\"HL7_EVN_Segment\"(\"Event_Type_Code\",\"Recorded_Datetime\",\"Operator_Id\",\"H
L7_Message_Id\",\"Sending_Facility\",\"Channel_Message_Id\",\"Sending_Application\",data_direction)values
('"+EVN_Event_Type_Code+"','"+EVN_Date_Time_Of_Event+"','"+EVN_Operator_Id+"','"+id+"','"+SourceName
+"','"+HL7_id+"','"+SendingApplication+"','in')");
        }
    }
```

Channel Destination Connector – Set Data Type



Set Data Types

[✓ Single Edit] Bulk Edit

Expand All | Collapse All

Connector	Inbound	Outbound
Source Connector	HL7 v2.x	HL7 v2.x
Javascript - PostgreSQL (postgreSQLServerHostName:5432)	HL7 v2.x	HL7 v2.x
Response	HL7 v2.x	HL7 v2.x

Inbound Properties

HL7 v2.x — Restore Defaults

Serialization	
Parse Field Repetitions	✓
Parse Subcomponents	✓
Use Strict Parser	☐
Validate in Strict Parser	☐
Strip Namespaces	☐
Segment Delimiter	\r
Convert Line Breaks	✓

Outbound Properties

HL7 v2.x — Restore Defaults

Deserialization	
Use Strict Parser	☐
Validate in Strict Parser	☐
Segment Delimiter	\r
Template Serialization	
Parse Field Repetitions	✓
Parse Subcomponents	✓
Use Strict Parser	☐
Validate in Strict Parser	☐
Strip Namespaces	☐
Segment Delimiter	\r
Convert Line Breaks	✓

OK Cancel

Channel Scripts

Preprocessor Script

```
// Modify the message variable below to pre-process data
return;
```

Postprocessor Script

```
// This script executes once after a message has been processed
// Responses returned from here will be stored as "Postprocessor" in the response map
return;
```

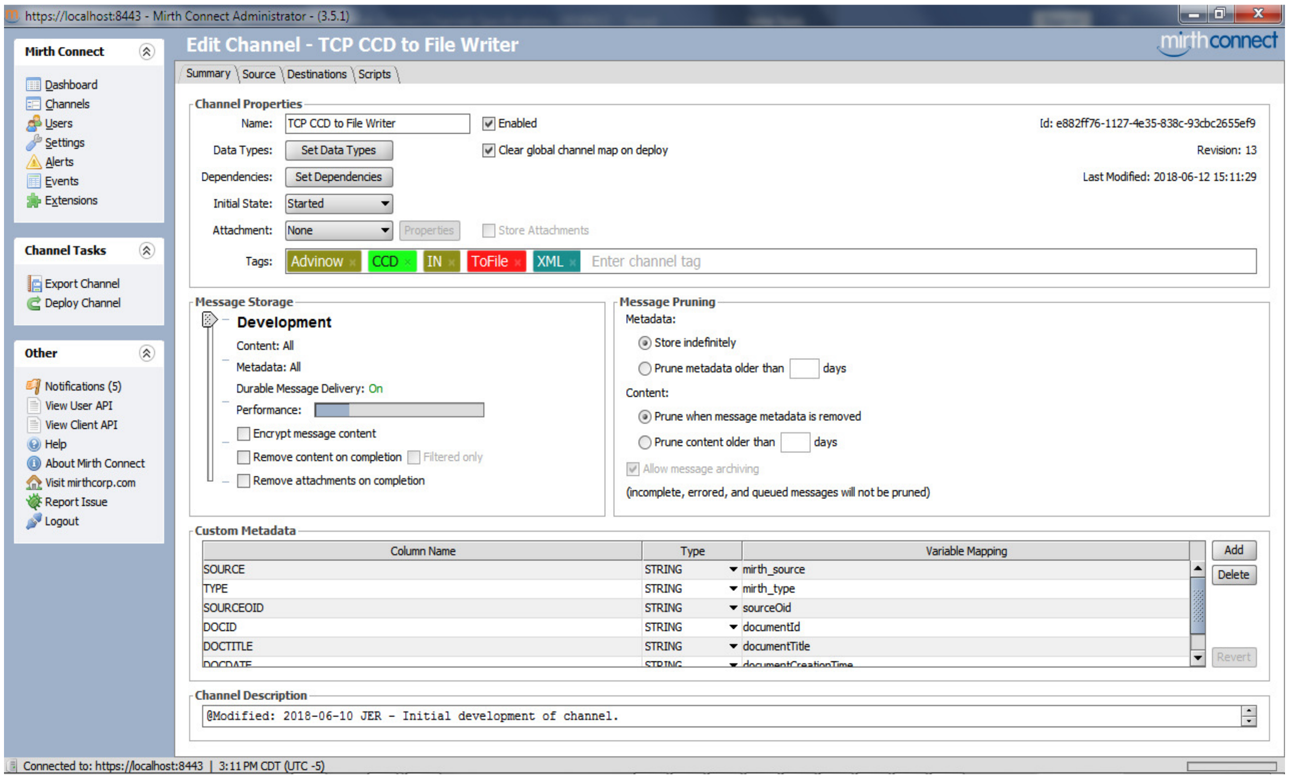
Deploy Script

```
// This script executes once when the channel is deployed
// You only have access to the globalMap and globalChannelMap here to persist data
return;
```

Undeploy Script

```
// This script executes once when the channel is undeployed
// You only have access to the globalMap and globalChannelMap here to persist data
return;
```

TCP CCDA to File Writer Channel Summary



Edit Channel - TCP CCD to File Writer

Channel Properties

Name: TCP CCD to File Writer ☑ Enabled — Id: e882ff76-1127-4e35-838c-93cbc2655ef9

Data Types: Set Data Types ☑ Clear global channel map on deploy — Revision: 13

Dependencies: Set Dependencies — Last Modified: 2018-06-12 15:11:29

Initial State: Started

Attachment: None ☐ Store Attachments

Tags: Advinow CCD IN ToFile XML Enter channel tag

Message Storage

Development

Content: All

Metadata: All

Durable Message Delivery: On

Performance:

☐ Encrypt message content

☐ Remove content on completion ☐ Filtered only

☐ Remove attachments on completion

Message Pruning

Metadata:

● Store indefinitely

○ Prune metadata older than days

Content:

● Prune when message metadata is removed

○ Prune content older than days

☑ Allow message archiving

(incomplete, errored, and queued messages will not be pruned)

Custom Metadata

Column Name	Type	Variable Mapping
SOURCE	STRING	mirth_source
TYPE	STRING	mirth_type
SOURCEOID	STRING	sourceOid
DOCID	STRING	documentId
DOCTITLE	STRING	documentTitle
DOCDATE	STRING	documentCreationTime

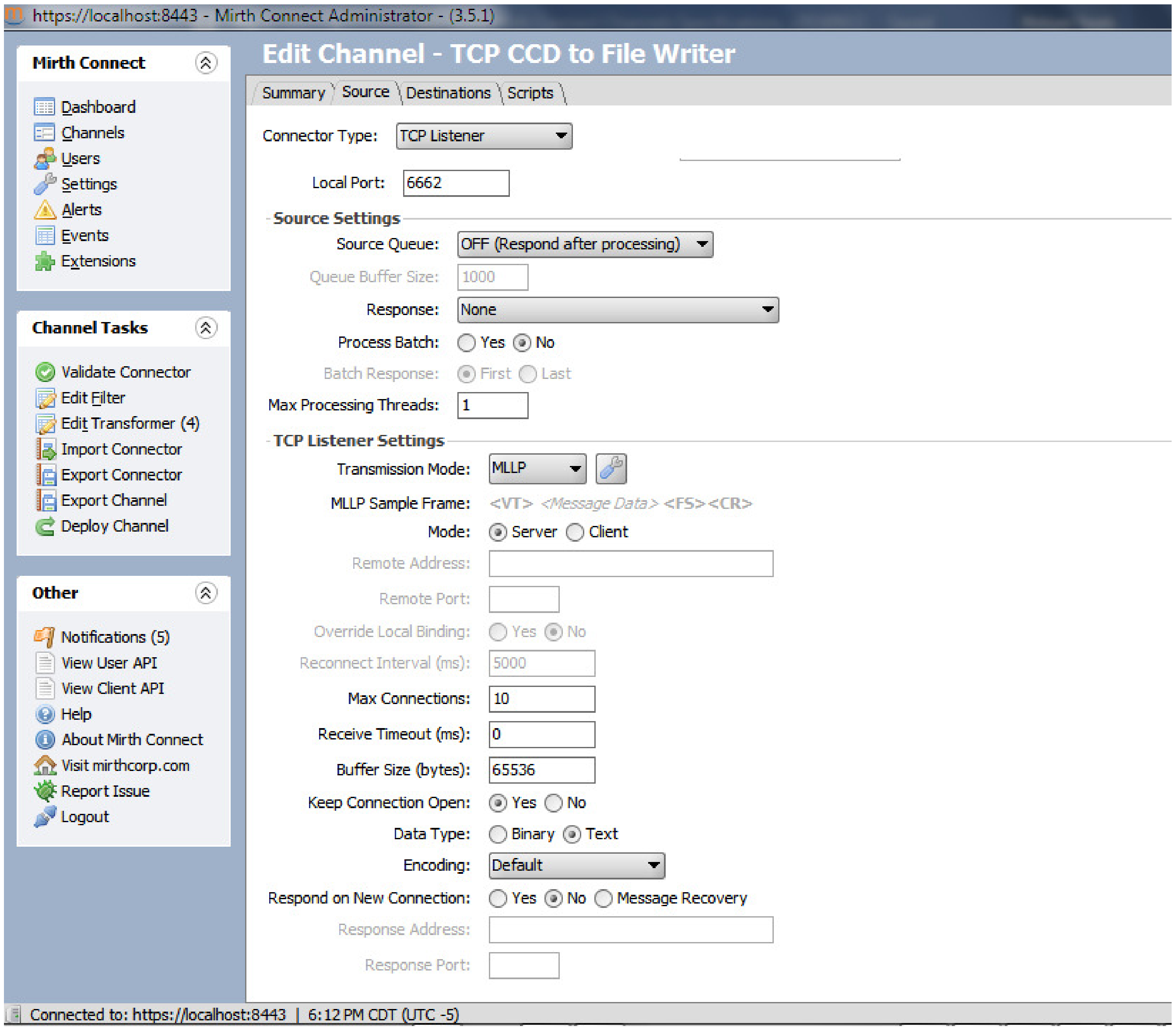
Channel Description

```
@Modified: 2018-06-10 JER - Initial development of channel.
```

Connected to: https://localhost:8443 | 3:11 PM CDT (UTC -5)

Name:	TCP CCD to File Writer
Enabled:	Checked – enabled
Clear Global channel Map on deploy:	Checked – enabled
Initial State:	Started
Attachment:	None
Message Storage:	Development
Message Pruning:	
Metadata:	Store indefinitely
Content:	Prune when message metadata is removed
Custom Metadata:	
SOURCEOID	sourceOid
DOCID	documentId
DOCTITLE	documentTitle
DOCDATE	documentCreationTime
PATIENTID	patientId

Channel Source Connector



Edit Channel - TCP CCD to File Writer

Connector Type: TCP Listener

Local Port: 6662

Source Settings

Source Queue: OFF (Respond after processing)

Queue Buffer Size: 1000

Response: None

Process Batch: Yes [✓ No]

Batch Response: [✓ First] Last

Max Processing Threads: 1

TCP Listener Settings

Transmission Mode: MLLP

MLLP Sample Frame: <VT> <Message Data> <FS><CR>

Mode: [✓ Server] Client

Remote Address:

Remote Port:

Override Local Binding: Yes [✓ No]

Reconnect Interval (ms): 5000

Max Connections: 10

Receive Timeout (ms): 0

Buffer Size (bytes): 65536

Keep Connection Open: [✓ Yes] No

Data Type: Binary [✓ Text]

Encoding: Default

Respond on New Connection: Yes [✓ No] Message Recovery

Response Address:

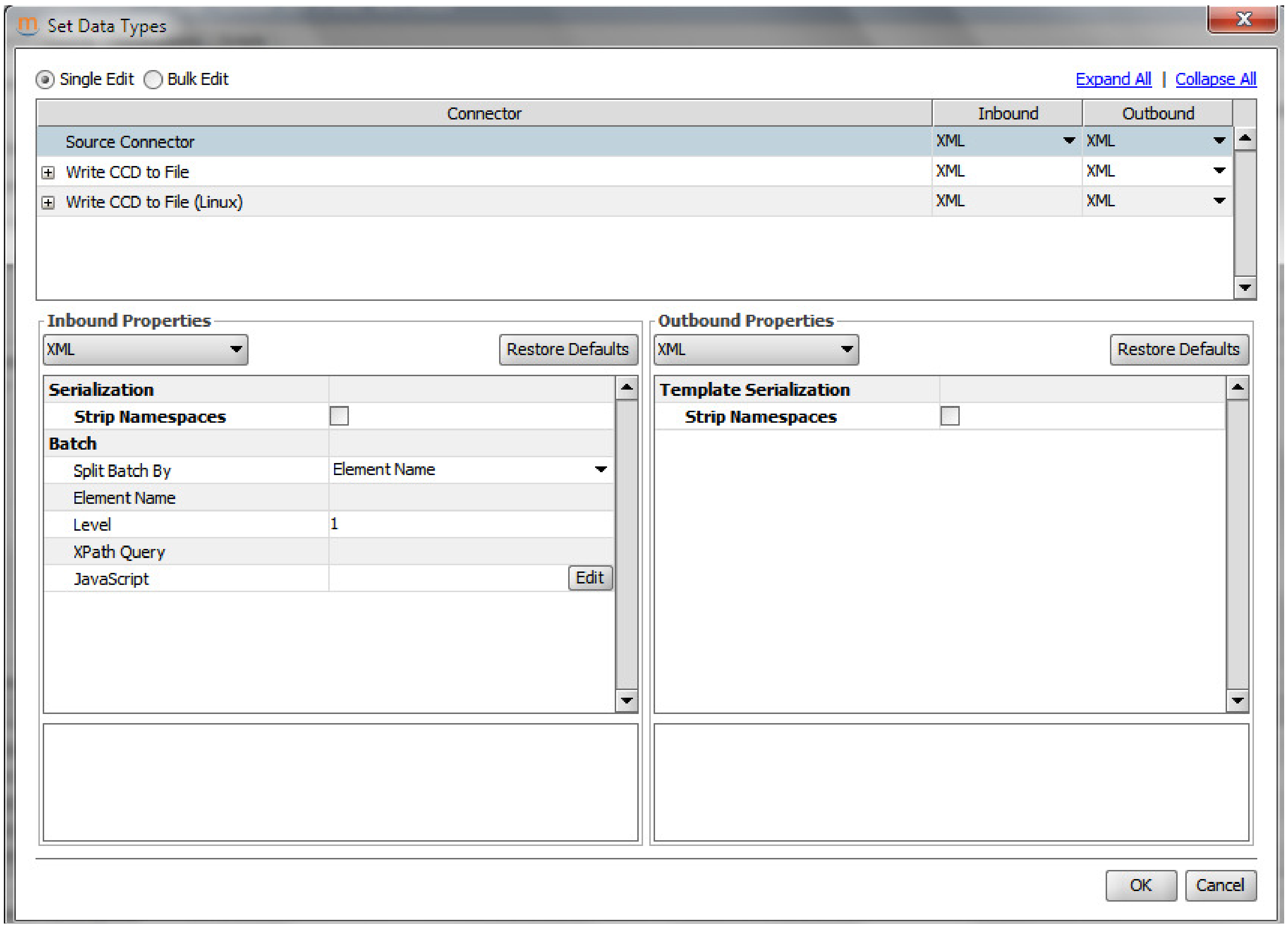
Response Port:

Connected to: https://localhost:8443 | 6:12 PM CDT (UTC -5)

Connector Type:	TCP Listener
Local Address:	All interfaces
Local Port:	**6662**
Source Queue:	OFF (Respond after Processing)
Response:	None
Process Batch:	No
Max Processing Threads:	1
TCP Listener Settings:	
Transmission Mode:	MLLP
Mode:	Server
Max Connections:	10
Receive Timeout (ms):	0
Buffer Size (bytes):	65536
Keep Connection Open:	Yes

Data Type:	Text
Encoding:	Default
Respond on New	No

Source Connector – Set Data Type



Set Data Types

(● Single Edit) ○ Bulk Edit

Expand All | Collapse All

Connector	Inbound	Outbound
Source Connector	XML	XML
⊞ Write CCD to File	XML	XML
⊞ Write CCD to File (Linux)	XML	XML

Inbound Properties

XML — Restore Defaults

Serialization	
Strip Namespaces	☐
Batch	
Split Batch By	Element Name
Element Name	
Level	1
XPath Query	
JavaScript	Edit

Outbound Properties

XML — Restore Defaults

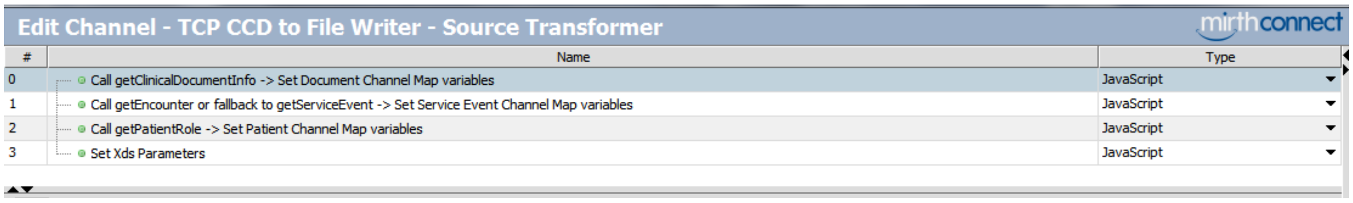
Template Serialization	
Strip Namespaces	☐

OK Cancel

Channel Source Filters

None

Channel Source Transformers



Edit Channel - TCP CCD to File Writer - Source Transformer

#	Name	Type
0	Call getClinicalDocumentInfo -> Set Document Channel Map variables	JavaScript
1	Call getEncounter or fallback to getServiceEvent -> Set Service Event Channel Map variables	JavaScript
2	Call getPatientRole -> Set Patient Channel Map variables	JavaScript
3	Set Xds Parameters	JavaScript

Transformer Descriptions

NOTE: R – required, R1 – if provided is required, NR – Not required

1. **Call getClinicalDocumentInfo -> Set Document Channel map variables**
   This transformer calls the getClinicalDocumentInfo function to reference the following document characteristics:
   - RootId and extension, Set Id rootId and extension, or Encounter rootId and extension -> to set the XDSDocumentEntry.uniqueId. (R)
   - Use document code, codeSystem, codeSystemName and codeSystemOID -> to set the XDSDocumentEntry.classCode. (R)
   - Use document title -> to set XDSDocumentEntry.contentTypeCode. (R)
   - Use document effectiveTime -> to set XDSDocumentEntry creationTime. (R)
   - Use document confidentialityCode -> to set XDSDocumentEntry confidentialityCode. (R)
   - Use document languageCode -> to set XDSDocumentEntry languageCode. (R)

   NOTE: for more information, reference https://gazelle.ihe.net/XDStarClient/rimihe-templates/rimihe-XDSDocumentEntry.html.

2. **Call getEncounter or fallback to getServiceEvent -> Set Service Event Channel Map variables**
   This transformer calls both getServiceEvent and getEncounter functions to reference the following document characteristics:
   - Use documentationOf.serviceEvent id root and extension or combination -> to set XDSDocumentEntry ServiceEvent called by getServiceEvent function. (R1)

   Conditional:
   - Use componentOf.encompassingEncounter effectiveTime low and high -> to set XDSDocumentEntry serviceStartTime and serviceStopTime by getEncounter function. (R)
   - Use documentationOf.serviceEvent effectiveTime low and high -> to set XDSDocumentEntry serviceStartTime and serviceStopTime by getServiceEvent function. (R)
   - Use currentDate_yyyyMMddhhmmss function -> to set to set XDSDocumentEntry serviceStartTime and serviceStopTime (R)

3. **Call getPatientRole -> Set Patient Channel Map variables**
   This transformer calls getPatientRole, getXdsSourcePatientId and getXdsSourcePatientInfo functions to reference the following document characteristics:

- Use recordTarget.patientRole id root, extension and assigningAuthority -> to set XDSDocumentEntry.patientId by getPatientRole function. (R)

  NOTE: for more information, reference https://gazelle.ihe.net/XDStarClient/rimihe-templates/rimihe-PatientId_XDSDocumentEntry.html.

- Use recordTarget.patientRole id, patient name, addr.streetAddress, city, state, postalCode, telecom, administrativeGenderCode, birthTime -> to set sourcePatientInfo by getXdsSourcePatientInfo and getXdsSourcePatientId functions. (R1)

4. Call getAuthor -> Set Author Channel Map variables

This transformer calls getAuthors, getPerformerand and getCustodian functions to reference the following document characteristics:

Conditional:

- Use author.assignedAuthor.representedOrganization id root, extension, name and authorizingDevice -> to set XDSDocumentEntry.author and XDSSubmissionSet.author authorPerson and authorInstitution. (R)
- Use performer.assignedEntity.representedOrganization id root, extension and name -> to set XDSDocumentEntry.author and XDSSubmissionSet.author authorPerson and authorInstitution. (R)
- Use custodian.representedOrganization id root, extension, assigningAuthorityName and name -> to set XDSDocumentEntry.author and XDSSubmissionSet.author authorPerson and authorInstitution. (R)

  NOTE: for more information, reference https://gazelle.ihe.net/XDStarClient/rimihe-templates/rimihe-Author_SubmissionSet.html.

5. Call getPerformer, getAuthors and getEncounters -> Set Performer and Authors Channel Map variables

This transformer calls getAuthors, getEncounters and and getPerformer functions to reference the following document characteristics:

Conditional:

- Use performer.functionCode code, codeSystem, codeSystemName, displayName -> to set XDSDocumentEntry.author and XDSSubmissionSet.author authorRole and authorSpecialty. (R1)
- Use encounterParticipant.assignedEntity typeCode or assignedEntity code, codeSystem, codeSystemName, displayName -> to set XDSDocumentEntry.author and XDSSubmissionSet.author authorRole and authorSpecialty. (R1)

Conditional:

- Use performer.assignedEntity.name -> to set XDSDocumentEntry.author authorPerson. (R)
- Use encompassingEncounter.encounterParticipant.assignedEntity.assignedPerson.name -> to set XDSDocumentEntry.author authorPerson. (R)

  NOTE: for more information, reference https://gazelle.ihe.net/XDStarClient/rimihe-templates/rimihe-Author_SubmissionSet.html.

6. Set Xds Parameters

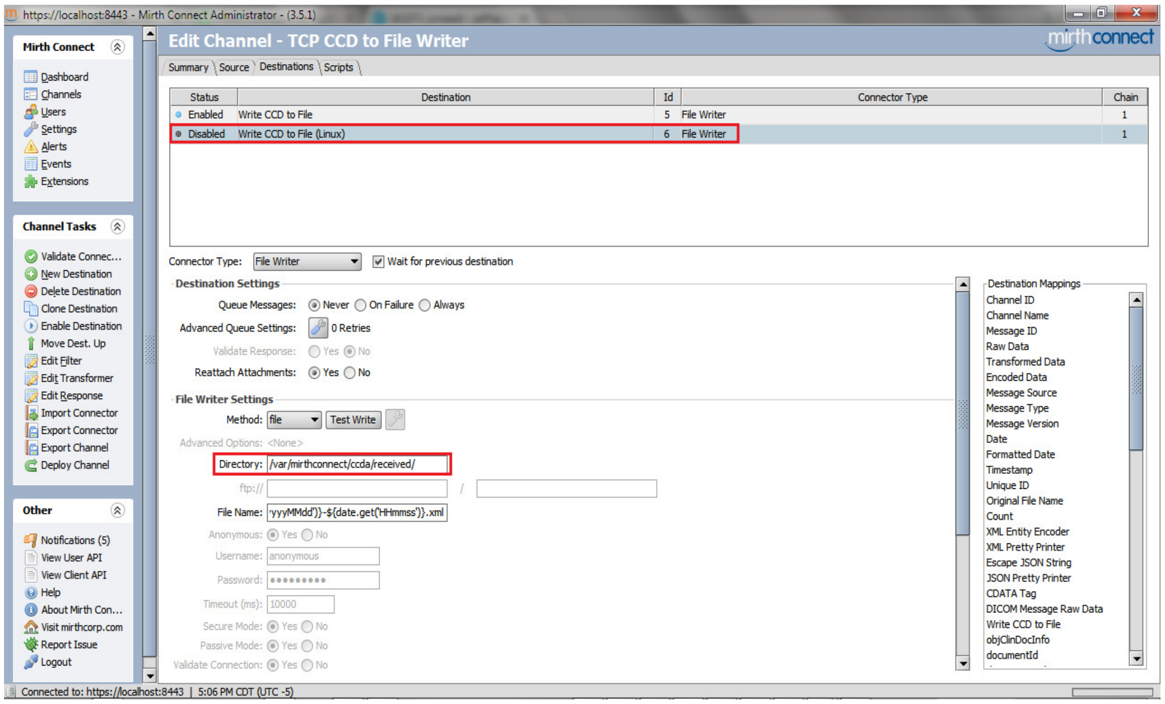
This transformer calls getClinicalDocumentInfo, getAuthor and getCustodian functions to reference the following document characteristics:

Conditional:

- Use id root -> to set XDSSubmissionSet.sourceId using getClinicalDocumentInfo function. (R)
- Use author.assignedAuthor.representedOrganization.id root -> to set XDSSubmissionSet.sourceId using getAuthor function. (R)
- Use custodian.assignedCustodian.representedCustodianOrganization.id root -> to set XDSSubmissionSet.sourceId using getCustodian function. (R)

NOTE: for more information, reference https://gazelle.ihe.net/XDStarClient/rimihe-templates/rimihe-SourceId_SubmissionSet.html.

Channel Destination Connector



Edit Channel - TCP CCD to File Writer

Status	Destination	Id	Connector Type	Chain
Enabled	Write CCD to File	5	File Writer	1
Disabled	Write CCD to File (Linux)	6	File Writer	1

Connector Type: File Writer ✓ Wait for previous destination

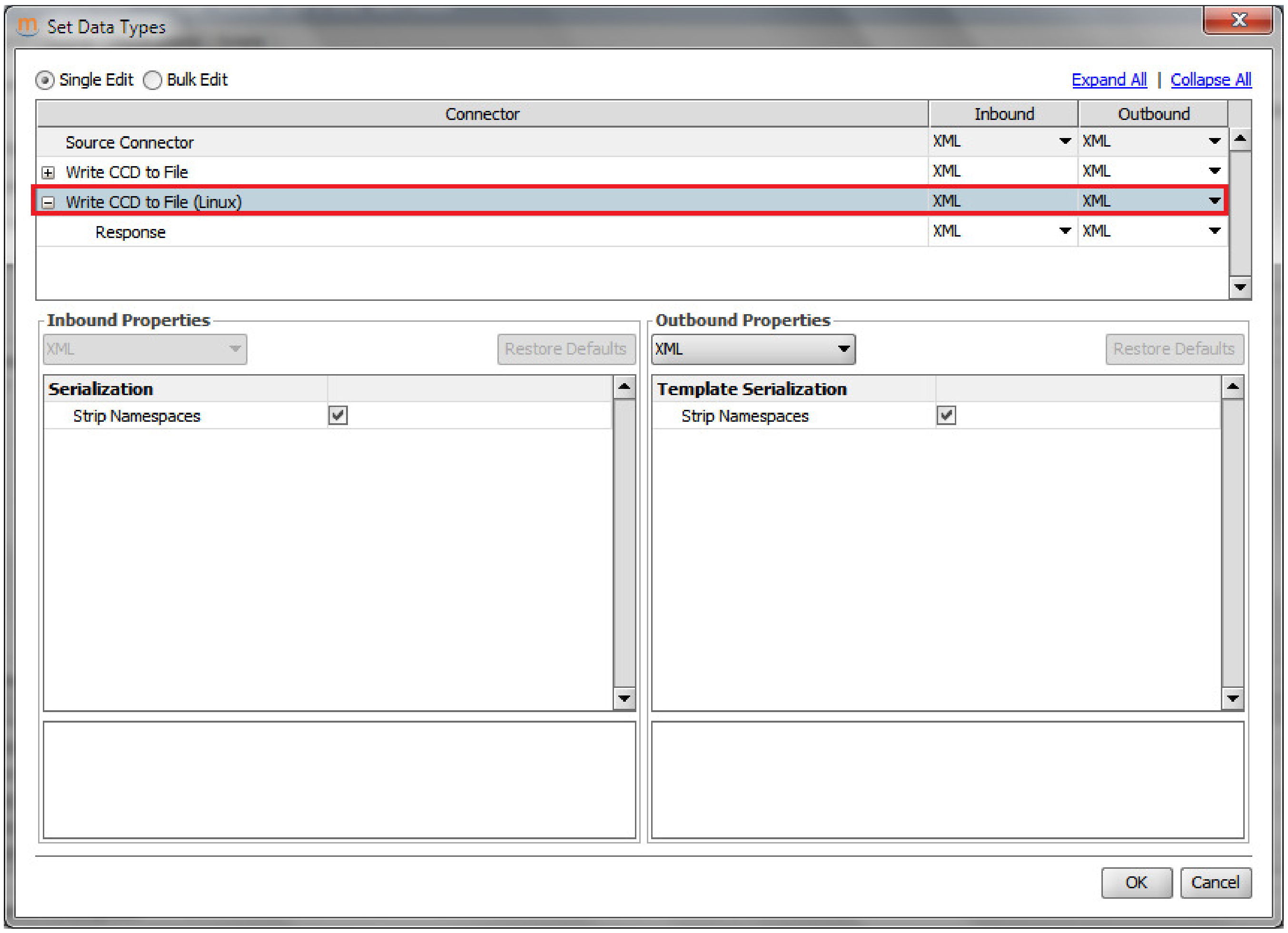
Destination Settings
Queue Messages: [✓ Never] On Failure Always
Advanced Queue Settings: 0 Retries
Validate Response: Yes [✓ No]
Reattach Attachments: [✓ Yes] No

File Writer Settings
Method: file
Advanced Options: <None>
Directory: /var/mirthconnect/ccda/received/
ftp://
File Name: 'yyyMMdd')}-${date.get('HHmmss')}.xml
Anonymous: [✓ Yes] No
Username: anonymous
Password: ●●●●●●●●●
Timeout (ms): 10000
Secure Mode: [✓ Yes] No
Passive Mode: [✓ Yes] No
Validate Connection: [✓ Yes] No

Connected to: https://localhost:8443 | 5:06 PM CDT (UTC -5)

Connector Type:	File Writer
Wait for previous destination:	Checked - Enable
Queue Messages:	Never
Advanced Queue Settings	0 Retries
Validate Response:	No
Reattach Attachments	Yes
File Writer Settings:	
Method:	File
Directory:	**/var/mirthconnect/ccda/received**
File Name:	ccda_${pid3}_${date.get('yyyyMMdd')}-${date.get('HHmmss')}.xml

Channel Destination Connector – Set Data Type



[✓ Single Edit] Bulk Edit

Connector	Inbound	Outbound
Source Connector	XML	XML
⊞ Write CCD to File	XML	XML
⊟ Write CCD to File (Linux)	XML	XML
Response	XML	XML

Inbound Properties

XML

Serialization	
Strip Namespaces	✓

Outbound Properties

XML

Template Serialization	
Strip Namespaces	✓

Channel Destination Filters

None

Channel Destination Transformers

None

Channel Scripts

Preprocessor Script

```
// Modify the message variable below to pre-process data
return;
```

Postprocessor Script

```
// This script executes once after a message has been processed
// Responses returned from here will be stored as "Postprocessor" in the response map
return;
```

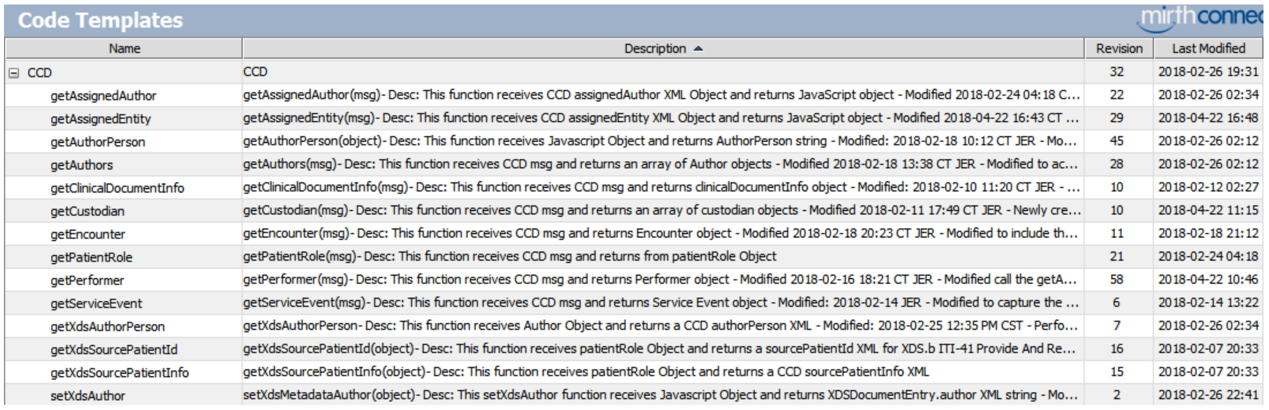
Deploy Script

```
// This script executes once when the channel is deployed
// You only have access to the globalMap and globalChannelMap here to persist data
return;
```

Undeploy Script

```
// This script executes once when the channel is undeployed
// You only have access to the globalMap and globalChannelMap here to persist data
return;
```

Code Templates – CCD

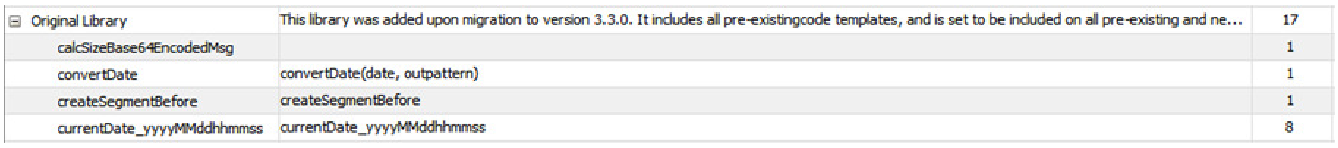


Code Templates

Name	Description	Revision	Last Modified
CCD	CCD	32	2018-02-26 19:31
getAssignedAuthor	getAssignedAuthor(msg)- Desc: This function receives CCD assignedAuthor XML Object and returns JavaScript object - Modified 2018-02-24 04:18 C...	22	2018-02-26 02:34
getAssignedEntity	getAssignedEntity(msg)- Desc: This function receives CCD assignedEntity XML Object and returns JavaScript object - Modified 2018-04-22 16:43 CT ...	29	2018-04-22 16:48
getAuthorPerson	getAuthorPerson(object)- Desc: This function receives Javascript Object and returns AuthorPerson string - Modified: 2018-02-18 10:12 CT JER - Mo...	45	2018-02-26 02:12
getAuthors	getAuthors(msg)- Desc: This function receives CCD msg and returns an array of Author objects - Modified 2018-02-18 13:38 CT JER - Modified to ac...	28	2018-02-26 02:12
getClinicalDocumentInfo	getClinicalDocumentInfo(msg)- Desc: This function receives CCD msg and returns clinicalDocumentInfo object - Modified: 2018-02-10 11:20 CT JER - ...	10	2018-02-12 02:27
getCustodian	getCustodian(msg)- Desc: This function receives CCD msg and returns an array of custodian objects - Modified 2018-02-11 17:49 CT JER - Newly cre...	10	2018-04-22 11:15
getEncounter	getEncounter(msg)- Desc: This function receives CCD msg and returns Encounter object - Modified 2018-02-18 20:23 CT JER - Modified to include th...	11	2018-02-18 21:12
getPatientRole	getPatientRole(msg)- Desc: This function receives CCD msg and returns from patientRole Object	21	2018-02-24 04:18
getPerformer	getPerformer(msg)- Desc: This function receives CCD msg and returns Performer object - Modified 2018-02-16 18:21 CT JER - Modified call the getA...	58	2018-04-22 10:46
getServiceEvent	getServiceEvent(msg)- Desc: This function receives CCD msg and returns Service Event object - Modified: 2018-02-14 JER - Modified to capture the ...	6	2018-02-14 13:22
getXdsAuthorPerson	getXdsAuthorPerson- Desc: This function receives Author Object and returns a CCD authorPerson XML - Modified: 2018-02-25 12:35 PM CST - Perfo...	7	2018-02-26 02:34
getXdsSourcePatientId	getXdsSourcePatientId(object)- Desc: This function receives patientRole Object and returns a sourcePatientId XML for XDS.b ITI-41 Provide And Re...	16	2018-02-07 20:33
getXdsSourcePatientInfo	getXdsSourcePatientInfo(object)- Desc: This function receives patientRole Object and returns a CCD sourcePatientInfo XML	15	2018-02-07 20:33
setXdsAuthor	setXdsMetadataAuthor(object)- Desc: This setXdsAuthor function receives Javascript Object and returns XDSDocumentEntry.author XML string - Mo...	2	2018-02-26 22:41

Name	**Description**
getAssignedAuthor	This function receives CCD assignedAuthor XML Object and returns JavaScript object.
getAssignedEntity	This function receives CCD assignedEntity XML Object and returns JavaScript object.
getAuthorPerson	This function receives Author JavaScript Object and returns AuthorPerson string.
getAuthors	This function receives CCD msg and returns an array of Author JavaScript objects.
getClinicalDocumentInfo	This function receives CCD msg and returns clinicalDocumentInfo JavaScript object.
getCustodian	This function receives CCD msg and returns an array of Custodian JavaScript objects.
getEncounter	This function receives CCD msg and returns Encounter JavaScript object.
getPatientRole	This function receives CCD msg and returns from patientRole JavaScript object.
getPerformer	This function receives CCD msg and returns Performer JavaScript objects.
getServiceEvent	This function receives CCD msg and returns Service Event JavaScript objects.
getXdsAuthorPerson	This function receives Author Object and returns Author Person JavaScript object.
getXdsSourcePatientId	This function receives patientRole JavaScript Object and returns a sourcePatientId String.
getXdsSourcePatientInfo	This function receives patientRole JavaScript Object and returns a XML SourcePatientInfo JavaScript object.
setXdsAuthor	This setXdsAuthor function receives JavaScript Array of Objects and returns author XML string.

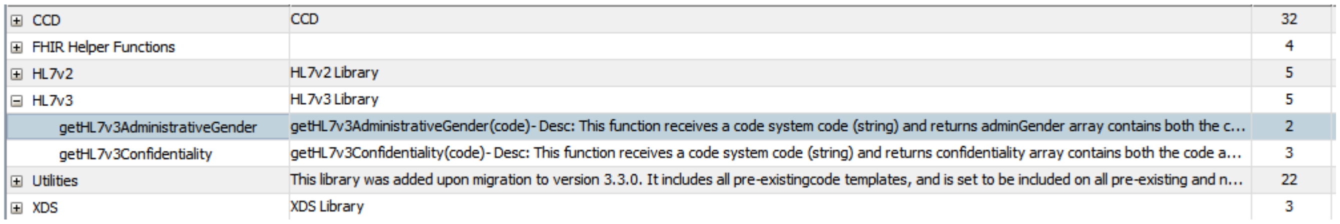
Code Templates – Utilities Library



⊟ Original Library	This library was added upon migration to version 3.3.0. It includes all pre-existingcode templates, and is set to be included on all pre-existing and ne...	17
calcSizeBase64EncodedMsg		1
convertDate	convertDate(date, outpattern)	1
createSegmentBefore	createSegmentBefore	1
currentDate_yyyyMMddhhmmss	currentDate_yyyyMMddhhmmss	8

Name	Description
calcSizeBased64EncodingMsg	This function is to calculate a Base64 Encoded file size.
convertDate	This function is to convert various dateTime formats using date object and pattern string to another specified dateTime formats.
currentDate_yyyyMMddhhmmss	This function is returns currentDate in yyyyMMddhhmmss string format.

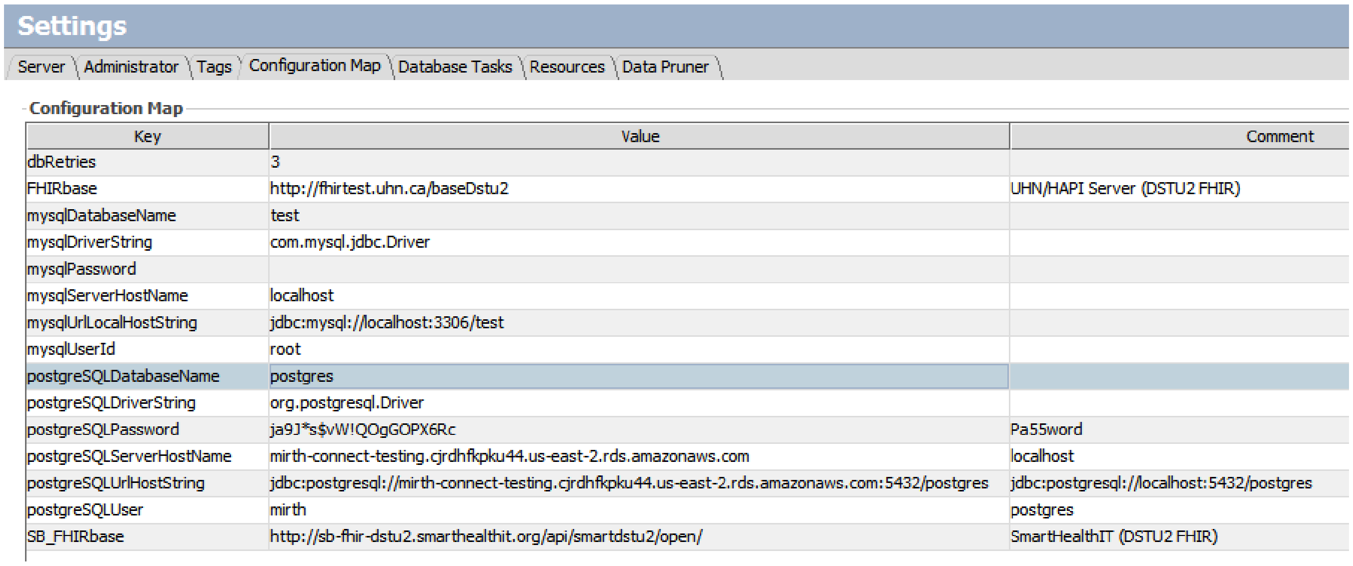
Code Templates – HL7v3 Library



⊞ CCD	CCD	32
⊞ FHIR Helper Functions		4
⊞ HL7v2	HL7v2 Library	5
⊟ HL7v3	HL7v3 Library	5
getHL7v3AdministrativeGender	getHL7v3AdministrativeGender(code)- Desc: This function receives a code system code (string) and returns adminGender array contains both the c...	2
getHL7v3Confidentiality	getHL7v3Confidentiality(code)- Desc: This function receives a code system code (string) and returns confidentiality array contains both the code a...	3
⊞ Utilities	This library was added upon migration to version 3.3.0. It includes all pre-existingcode templates, and is set to be included on all pre-existing and n...	22
⊞ XDS	XDS Library	3

Name	Description
getHL7v3AdministrativeGender	This function receives a code system code (string) and returns adminGender array contains both the HL7 Gender code and displayName.
getHL7v3Confidentiality	This function receives a code system code (string) and returns the HL7 Confidentiality array contains both the code and displayName.

Global Configuration Map



Key	Value	Comment
dbRetries	3	
FHIRbase	http://fhirtest.uhn.ca/baseDstu2	UHN/HAPI Server (DSTU2 FHIR)
mysqlDatabaseName	test	
mysqlDriverString	com.mysql.jdbc.Driver	
mysqlPassword		
mysqlServerHostName	localhost	
mysqlUrlLocalHostString	jdbc:mysql://localhost:3306/test	
mysqlUserId	root	
postgreSQLDatabaseName	postgres	
postgreSQLDriverString	org.postgresql.Driver	
postgreSQLPassword	ja9J*s$vW!QOgGOPX6Rc	Pa55word
postgreSQLServerHostName	mirth-connect-testing.cjrdhfkpku44.us-east-2.rds.amazonaws.com	localhost
postgreSQLUrlHostString	jdbc:postgresql://mirth-connect-testing.cjrdhfkpku44.us-east-2.rds.amazonaws.com:5432/postgres	jdbc:postgresql://localhost:5432/postgres
postgreSQLUser	mirth	postgres
SB_FHIRbase	http://sb-fhir-dstu2.smarthealthit.org/api/smartdstu2/open/	SmartHealthIT (DSTU2 FHIR)

postgreSQLDatabaseName	postgres
postgreSQLDriverString	org.postgresql.Driver
postgreSQLPassword	ja9J*s$vW!QOgGOPX6Rc
postgreSQLServerHostName	mirth-connect-testing.cjrdhfkpku44.us-east-2.rds.amazonaws.com
postgreSQLUrlHostString	jdbc:postgresql://mirth-connect-testing.cjrdhfkpku44.us-east-2.rds.amazonaws.com:5432/postgres
postgreSQLUser	mirth