

HL7 to PostgreSQL Javascript Mirth Connect Channel Specification

Revision History

Date	Author	Description
2018-06-04	Jeffrey Ritz	Initial version release.

Description

The purpose of this document is to provide the build specification for the Mirth Connect (v3.5.0.8232) application specific HL7 to PostgreSQL Javascript channel. This Mirth Connect specific channel was developed to properly handle the receipt of Admit Discharge and Transfer (ADT) and Scheduling Information Unsolicited HL7 messages and updating the appropriate Mirth_Inbound schema tables.

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.
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

Section	Pg.
Channel Summary	5
HL7 to PostgreSQL Javascript	5
Channel Source Connector	6
Source Connector – Set Data Type	7
Channel Source Transformers	7
Channel Destination Connector	9
JavaScript Writer Settings.....	9-18
Destination Connector – Set Data Type	19
Channel Scripts	20
Preprocessor Script	20
Postprocessor Script	20
Deploy Script	20
UnDeploy Script	20
Configuration Map	20

Channel Summary

[Summary](#) |
 [Source](#) |
 [Destinations](#) |
 [Scripts](#)

Edit Channel - HL7 to PostgreSQL Javascript

Channel Properties

Name:	<input type="text" value="HL7 to PostgreSQL Javascript"/>	<input checked="" type="checkbox"/> Enabled	Id: 5beae4c2-e7f8-4ae3-9fca-703dea9233a2
Data Types:	<input type="button" value="Set Data Types"/>	<input checked="" type="checkbox"/> Clear global channel map on deploy	Revision: 34
Dependencies:	<input type="button" value="Set Dependencies"/>		Last Modified: 2018-06-04 02:54:08
Initial State:	<input type="button" value="Stopped"/>		
Attachment:	<input type="button" value="None"/> <input type="button" value="Properties"/> <input type="checkbox"/> Store Attachments		
Tags:	<input type="button" value="ADT"/> <input type="button" value="HL7"/> <input type="button" value="PostgreSQL"/> <input type="button" value="SIU"/> <input type="text" value="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

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

Summary \ Source \ Destinations \ Scripts \

Connector Type: **TCP Listener**

Listener Settings

Local Address: ☒ All interfaces ☐ Specific interface:

Local Port:

Source Settings

Source Queue: **ON (Respond before processing)**

Queue Buffer Size:

Response: **Auto-generate (Before processing)**

Process Batch: ☐ Yes ☒ No

Batch Response: ☒ First ☐ Last

Max Processing Threads:

TCP Listener Settings

Transmission Mode: **MLLP**

MLLP Sample Frame:

Mode: ☒ Server ☐ Client

Remote Address:

Remote Port:

Override Local Binding: ☐ Yes ☒ No

Reconnect Interval (ms):

Max Connections:

Receive Timeout (ms):

Buffer Size (bytes):

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 "S19" or "S20" or "S22"	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	New	Delete

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. PatientId | 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
<input checked="" type="radio"/> 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:

```
1  /*
2  Created:  2018-05-31 JER
3  Modified:
4
5  */
6
7  var dbConn;
8  var ChannelName="HL7_Inbound_MessageParsing";
9  var Filename;
10
11 //try {
12 // dbConn = DatabaseConnectionFactory.createDatabaseConnection('org.postgresql.Driver', 'jdbc:postg
13 // dbConn = DatabaseConnectionFactory.createDatabaseConnection('org.postgresql.Driver', 'jdbc:postg
14 dbConn = DatabaseConnectionFactory.createDatabaseConnection(configurationMap.get('postgresqlDriver
15
```

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('postgresqlPassword'));
```

```
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')+"', '"+$('originalFilename')+"', 'in', '"+$('MessageControllId')+"')");
```

```
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+"", ""+ChannelName+"", ""+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_PRN\", \"Home_PH_CP_Internet\", \"Work_WPN\", \"Work_PH_Internet\", \"Language_Text\", \"ISO0639\",
\"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_RelationshipText = seg['NK1.3']['NK1.3.2'].toString().replace('','');
        var NK1_HL70063 = seg['NK1.3']['NK1.3.3'].toString().replace('','');
        var NK1_PRN_or_PRN = 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_RelationshipText+"','"+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_Phonenumbe\",\"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_Benefits\",\"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_Ethnic_Group =
seg['GT1.44']['GT1.44.1'].toString().replace("","");
        var GT1_Gurtantor_Ethnic_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_ISO639 = 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\", \"Country\", \"Home_Phone_Number\", \"Email_Address\", \"Area_Code\", \"Local_Number\", \"Work_Phone_Number\", \"Work_Email_Address\", \"Work_Area_Code\", \"Work_Local_Number\", \"Guarantor_Employee_Address_Street\", \"Guarantor_Employee_Address_City\", \"Guarantor_Employee_Address_State\", \"Guarantor_Employee_Address_Zip\", \"Guarantor_Employee_Phone\", \"Guarantor_Martial_Status\", \"Primary_Language\", \"Contact_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\", \"Work_WPN\", \"Language_Text\", \"ISO0639\", \"HL70187\", \"HL70005\", \"Work_PH_Internet\", data_direction)VALUES('"+GT1_Setid_Guarantor+"', '"+GT1_Guarantor_Number+"', '"+GT1_Guarantor_Last_Name+"', '"+GT1_Guarantor_First_Name+"', '"+GT1_Guarantor_Middle_Name+"', '"+GT1_Guarantor_Suffix+"', '"+GT1_Guarantor_NameType+"', '"+GT1_Guarantor_Address1+"', '"+GT1_Guarantor_Address2+"', '"+GT1_Guarantor_City+"', '"+GT1_Guarantor_State+"', '"+GT1_Guarantor_Zip+"', '"+GT1_Guarantor_Country+"', '"+GT1_Guarantor_Home_Phone_Number+"', '"+GT1_Guarantor_Home_EmailAddress+"', '"+GT1_Guarantor_AreaCode+"', '"+GT1_Guarantor_LocalNumber+"', '"+GT1_Guarantor_Business_Phone_Number+"', '"+GT1_Guarantor_Business_EmailAddress+"', '"+GT1_Guarantor_Business_AreaCode+"', '"+GT1_Guarantor_Business_LocalNumber+"', '"+GT1_Guarantor_Employer_Address+"', '"+GT1_Guarantor_Employer_City+"', '"+GT1_Guarantor_Employer_State+"', '"+GT1_Guarantor_Employer_Zip+"', '"+GT1_Guarantor_Employer_Phone+"', '"+GT1_Guarantor_Marital_Status+"', '"+GT1_Guarantor_Primary_Language+"', '"+GT1_Guarantor_Persons_Name+"', '"+GT1_Guarantor_Persons_Phone+"', '"+GT1_Guarantor_Ethnic_Group+"', '"+GT1_Guarantor_Ethnic_Text+"', '"+GT1_Guarantor_Relationship+"', '"+GT1_Guarantor_SSN+"', '"+GT1_Guarantor_Employer_organization+"', '"+GT1_Guarantor_Birth_Place+"', '"+GT1_Guarantor_Race+"', '"+GT1_Guarantor_Race_Text+"', '"+id+"', '"+SourceName+"', '"+HL7_id+"', '"+GT1_Motherss_Maiden_Name+"', '"+GT1_Guarantor_Sex+"', '"+GT1_Relationship_Id+"', '"+GT1_Relationship_Text+"', '"+GT1_Relationship_Coding_System+"', '"+SendingApplication+"', '"+GT1_Home_PRN_PRS+"', '"+GT1_Home_PH+"', '"+GT1_WPN+"', '"+GT1_LanguageText+"', '"+GT1_ISO0639+"', '"+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

√

Convert Line Breaks

☒

Outbound Properties

HL7 v2.x

Restore Defaults

Deserialization

Use Strict Parser

☐

Validate in Strict Parser

☐

Segment Delimiter

√

Template Serialization

Parse Field Repetitions

☒

Parse Subcomponents

☒

Use Strict Parser

☐

Validate in Strict Parser

☐

Strip Namespaces

☐

Segment Delimiter

√

Convert Line Breaks

☒

OK

Cancel

Page 19

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;
```

Configuration Map

Settings		
Server \ Administrator \ Tags \ Configuration Map \ Database Tasks \ Resources \ Data Pruner \		
- 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!QOGOPX6Rc	Pa55word
postgresqlServerHostName	mirth-connect-testing.cjrdhfkpu44.us-east-2.rds.amazonaws.com	localhost
postgresqlUrlHostString	jdbc:postgresql://mirth-connect-testing.cjrdhfkpu44.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!QOGOPX6Rc
postgresqlServerHostName	mirth-connect-testing.cjrdhfkpu44.us-east-2.rds.amazonaws.com
postgresqlUrlHostString	jdbc:postgresql://mirth-connect-testing.cjrdhfkpu44.us-east-2.rds.amazonaws.com:5432/postgres
postgresqlUser	mirth