ASTM E1394 Message Parsing

DRAFT

Theron W. Genaux

14-June-2024

This project demonstrates generically reading and creating ASTM E1394 messages.

In 2015 I started working with vendors and customers to help them interface to my company's new instrument. The message format was ASTM E1394 (ASTM). I started to wonder how LIS and middleware vendors were able to adapt to connecting with so many instruments where each instrument manufacturer developed their own message format based on the ASTM E1394 standard.

I had already seen some code that reads and writes ASTM messages, both on the web and proprietary. And I always thought that they were error prone to use and and overly complex. I started wondering how reading, writng, and processing ASTM messaging could be generalized so that one could adapt to all the variations that I've seen. I also wondered how to make it less error prone.

I played around mentally going through different ideas for a while until I realized that an ASTM record is in essence a recursive data structure, one recursion per separator. ASTM records have 3 separators; Field, Repeat-Field, and Components. Because it only has 3 delimiters, it is limited to only 3 levels of recursion. Note, HL7 2.x (Pipehat) has 4 separators and 4 levels of recursion. This code also supports HL7 2.x messages.

The following output is from a program that was developed to explore two ideas; generically reading and writing ASTM messages, and using bi-directional maps (Translation Map) to map database orders, patients, and results to create and read ASTM messages. The program supports round-tripping, such that the extracted content from one message can be used to recreate the equivelent original message. This program also works with HL7 Version 2.5 message files.

Input message:

```
H|\^&|||OCD^VISION^5.10.0.46252^JNumber|||||||P|LIS2-A|20240307151237
P|1|PID123456||NID123456^MID123456^OID123456|Brown^Bobby^B|White|1965010203
0400|U|||||||||||||||||||||||
O|1|SID101||ABO-
D|N|20240307151207|||||||||CENTBLOOD|||||||20240307151237|||F|||||
O|1|SID101||ABScr|N|20240307151207|||||||||CENTBLOOD|||||||20240307151237||
|F|||||
L||
```

Parse massage into record Position:Value pairs:

```
Delimiters:|\^&
H.1:H
H.5:OCD^VISION^5.10.0.46252^JNumber
H.5.1.1:OCD
H.5.1.2:VISION
H.5.1.3:5.10.0.46252
H.5.1.4:JNumber
H.12:P
H.13:LIS2-A
H.14:20240307151237
P.1:P
P.2:1
P.3:PID123456
P.5:NID123456^MID123456^OID123456
P.5.1.1:NID123456
P.5.1.2:MID123456
P.5.1.3:OID123456
P.6:Brown^Bobby^B
P.6.1.1:Brown
P.6.1.2:Bobby
P.6.1.3:B
P.7:White
P.8:19650102030400
P.9:U
O.1:O
O.2:1
O.3:SID101
O.5:ABO-D
O.6:N
O.7:20240307151207
O.16:CENTBLOOD
O.23:20240307151237
O.26:F
O.1:O
O.2:1
```

```
O.3:SID101
O.5:ABScr
O.6:N
O.7:20240307151207
O.16:CENTBLOOD
O.23:20240307151237
O.26:F
L.1:L
```

A bi-directional translation map is used to remap the Position:Value pairs into Key:Value pairs. It is also used to remap the Key:ValuePosition:Value pairs back into Position:Value pairs.

An Order record like this:

```
O|1|SID102\SID103||ABO FWD/RVS|||||||||||PACKEDCELLS\PLASMA
```

Is parsed into Position:Value pairs:

```
O.1:O
O.2:1
O.3:SID102\SID103
O.3.1:SID102
O.3.2:SID103
O.5:ABO FWD/RVS
O.16:PACKEDCELLS\PLASMA
O.16.1:PACKEDCELLS
O.16.2:PLASMA
```

It can be remapped into Key:Value pairs using a bi-directional translation map:

```
RecordType:O.1
SeqeNumber:O.2
OrderSampleIDs:O.3
OrderSampleID1:O.3.1
OrderSampleID2:O.3.2
OrderTestID:O.5    # Universal Test ID
OrderPriority:O.6
OrderReqDateTime:O.7
OrderActionCode:O.12
OrderSampleTypes:O.16
OrderSampleType1:O.16.1
OrderSampleType2:O.16.2
```

Remapping to Key:Value pairs:

```
RecordType:O
SeqeNumber:1
OrderSampleIDs:SID102\SID103
OrderSampleID1:SID102
OrderSampleID2:SID103
OrderTestID:ABO FWD/RVS
OrderSampleTypes:PACKEDCELLS\PLASMA
OrderSampleType1:PACKEDCELLS
OrderSampleType2:PLASMA
```

I used a bi-directional translation map to remap the message Position:Value pairs into Key:Value pairs.

Here we list the extracted message Key:Value pairs:

```
Delimiters:|\^&
H-RecordType:H
H-SenderName:OCD^VISION^5.10.0.46252^JNumber
H-Company:OCD
H-InstrumentModel:VISION
H-Version:5.10.0.46252
H-InstrumentID:JNumber
H-ProcessingID:P
H-VeresionNumber:LIS2-A
H-TimeStamp:20240307151237
P-RecordType:P
P-SeqNumber:1
P-PatientID:PID123456
P-PatientID3:NID123456^MID123456^OID123456
P-PatientName:Brown^Bobby^B
P-PatientNameLast:Brown
P-PatientNameFirst:Bobby
P-PatientNameMiddle:B
P-MothersMaiden:White
P-BirthDate:19650102030400
P-Sex:U
O-RecordType:O
O-SeqNumber:1
O-SampleIDs:SID101
O-Profiles:ABO-D
O-Priority:N
O-RequestedTimeStamp:20240307151207
```

```
O-SampleTypes:CENTBLOOD
O-ReportedTime:20240307151237
O-ReportType:F
O-RecordType:O
O-SeqNumber:1
O-SampleIDs:SID101
O-Profiles:ABScr
O-Priority:N
O-RequestedTimeStamp:20240307151207
O-SampleTypes:CENTBLOOD
O-ReportedTime:20240307151237
O-ReportType:F
L-RecordType:L
```

Recreate the message by first converting Key:Value pairs into record Position:Value pairs:

```
Delimiters:|\^&
H.1:H
H.5:OCD^VISION^5.10.0.46252^JNumber
H.5.1.1:OCD
H.5.1.2:VISION
H.5.1.3:5.10.0.46252
H.5.1.4:JNumber
H.12:P
H.13:LIS2-A
H.14:20240307151237
P.1:P
P.2:1
P.3:PID123456
P.5:NID123456^MID123456^OID123456
P.6:Brown^Bobby^B
P.6.1.1:Brown
P.6.1.2:Bobby
P.6.1.3:B
P.7:White
P.8:19650102030400
P.9:U
O.1:O
O.2:1
O.3:SID101
O.5:ABO-D
O.6:N
O.7:20240307151207
O.16:CENTBLOOD
O.23:20240307151237
```

```
O.26:F
O.1:O
O.2:1
O.3:SID101
O.5:ABScr
O.6:N
O.7:20240307151207
O.16:CENTBLOOD
O.23:20240307151237
O.26:F
L.1:L
```

Export Position/Value pairs into an ASTM message:

```
H|\^&|||OCD^VISION^5.10.0.46252^JNumber|||||||P|LIS2-A|20240307151237
P|1|PID123456||NID123456^MID123456^OID123456|Brown^Bobby^B|White|1965010203
0400|U
O|1|SID101||ABO-
D|N|20240307151207|||||||||CENTBLOOD|||||||20240307151237|||F
O|1|SID101||ABScr|N|20240307151207|||||||||CENTBLOOD|||||||20240307151237||
|F
L
```