Introduction to ASTM Message Formats

Theron W. Genaux

Draft

2-February-2023

# Introduction

This paper discusses ASTM E1394 based message formats. ASTM E1394 standard was first adopted in 1991 with the intent to be a flexible standard for LIS vendors and analyzer manufacturers to create compatible message formats.

This standard specifies the conventions for structuring the content of the message and for representing the data elements contained within those structures. It is applicable to all text-oriented clinical instrumentation. It has been specifically created to provide common conventions for interfacing computers and instruments in a clinical setting. [[1]](#footnote-1)

Unless otherwise stated, throughout this paper, *ASTM* refers to ASTM E1394 based message formats and the standard.

ASTM has since transferred the responsibility for maintaining this standard to the Clinical and Laboratory Standards Institute (CLSI). Which means that the latest version of the ASTM E1394 standard is actually an CLSI standard:

LIS02-A2 Specification for Transferring Information Between Clinical Laboratory Instruments and Information Systems; Approved Standard- Second Edition

The ASTM E1394 standard has been around since 1991, different systems will reference the version that they target. Below is a list of revisions:

* ASTM E1394-91
* ASTM E1394-97
* LIS02-A
* LIS02-A2

## Flexible Standard

The ASTM standard is not a ridged standard. The phrase "agreed upon between the sender and the receiver" appears several times in the ASTM standard. Additionally, the majority of fields are optional; the sender is not required to send the information and the receiver may ignore the information if it is not required for the processing of the message. This means there is significant variability between implementations.

The ASTM standard "has sufficient flexibility to permit the addition of fields to existing record types or the creation of new record types to accommodate new test and reporting mythologies."

A system may transmit a null value (devoid of text) for a field because:

1. it does not know the value
2. it knows the value is irrelevant to the receiving system
3. the value has not changed since the last transmission or any combination thereof

The ASTM standard designed to for supporting systems to be compatible, not interchangeable. Compatible so that it should be relatively easy to connect an analyzer to an LIS by leveraging existing drivers.

# Comma-Separated Values (CSV) Format

You may already be familiar the simple text format that separates fields with a delimiter, such as the comma-separated values (CSV) format. Although not limited to it, the CSV format is famous for storing tabular data using plain text, where each record has the same number and order of fields.

The simplest CSV file is a list of records. A record is an ordered array of related text fields, and each field is separated by a comma. Fields within a record are identified by their position in the record.

Let's define an order message based on the CSV format:

* A CSV order message consists of an array of records (lines)
* Each record is made up of comma separated fields and ends with a carriage return (CR)
* The text in a field cannot contain a comma or a carriage return
* Fields are identified by their position in the record
* The first field in a record identifies the type of data it contians
* A patient record begins with P and is followed by PatientID, LastName, FirstName, and MiddleInitial fields
* An order record begins with O and is followed by SampleID,Sampletype,and TestID fields

Below is our example CSV order message:

P,PID123456,Brown,Bobby,B<CR>  
O,SID304,CENTBLOOD,ABORh<CR>

The main advantages of such a format is that it is easy to create and parse, and therefore easy to implement.

Key points of our CSV message:

* The entire message is created from simple text
* Messages and records are easy to create and parse, either programmatically or manually with a text editor
* Any character can be used as the field delimiter as long as it does not appear in the text of a field

## ASTM Message Formats

Below is a minimal ASTM order message which contains only the information necessary for an order; sample ID, sample type, and a profile name.

H,\^&  
P,1  
O,1,SID101,,ABO,,,,,,,,,,,CENTBLOOD  
L,1,N

There is no difference between parsing the CSV message and this ASTM message, other than the minor differences in the ASTM patient and order records, and of course the two new records.

And for those who have seen ATSM messages before, I assure you that this is a valid ASTM message.

The first line is the Message Header record. The bare minimum header record is just 5 characters. The first character is the record type ID. The next 4 characters are the delimiter definitions; field delimiter (,), repeat delimiter (\), component delimiter (^), and the escape character (&). Delimiters are not fixed by the standard, but can be defined by the sender of the message as shown here and are then used throughout the message.

The standard uses a pipe (|) for the field delimiters in its examples and not the comma (,) as was used in the previous message. Below, we see the same message with the pipe (|) for the field delimiter. Per the ASTM standard, both messages are the same. By convention, the delimiters used in the standard are used by default for most implementations.

H|\^&  
P|1  
O|1|SID101||ABO|||||||||||CENTBLOOD  
L|1|N

The next record is the Patient record. Accession numbers and sample IDs can serve to identify a patient, negating the need for patient demographics in an order message. Therefore this record consists only of the record type ID (P) and the record sequence number. These two fields are separated by the field delimiter that was defined in the header record.

The third line is the Order record. The Order record contains multiple fields; the record type ID (O), record sequence number, sample ID, profile name, and the sample type.

The last line is the Termination record. It consists of 3 fields; record type ID (L), sequence number, and termination code.

An ASTM message is structured text:

* An ASTM message is an ordered list of records
* All ASTM records start with a record type ID
* The first record is always a Header record (H)
* The last record is always a Message Terminator record (L)
* An ASTM record is an ordered list of fields delimited by the field delimiter (|)
* A field is an unordered list of homogeneous field values delimited by the repeat field delimiter (\)
* A field value is an ordered list of components delimited by the component delimiter (^)

A record is an ordered list of fields delimited by the field delimiter (|). By convention, ASTM record notation numbers the fields in a record are numbered from 1 to N. This stems from the heading numbers in the standard that describes each record and field.

For example, the Patient record field headings are numbered from 7.1, 7.2, ... 7.35. The first field in a Patient record is denoted as P.1 and contains the record type ID (P). The second field is denoted as P.2 and is the record sequence number.

A field is an unordered list of homogeneous field values delimited by the repeat field delimiter (\). In the order record below, field O.1 is the Specimen ID, and O.16 is the Specimen Type. Field O.5 holds one or more Test IDs (repeated field values). The repeat field delimiter separates repeated field values. By convention, repeated fields are not included in the ASTM record notations used in literature. However, including them in the notation is useful when discussing software implementations. So, in the order record below, the first repeated field O.5, ABO is denoted as O.5.1 and the second, ABScr is denoted as O.5.2.

H|\^&  
P|1  
O|1|SID101||ABO\ABScr|||||||||||CENTBLOOD  
L|1|N

When field O.5 is parsed, "ABO\ABScr" is returned as the field value. The field value is then parsed for repeated fields using the repeat field delimiter to extract each repeated field values. In this case, the repeat field delimiter plays the same role as the comma does in the CSV format.

A field value is an ordered list of components delimited by the component delimiter (^). In the order message below, the patient name is in P.6 and contains 3 components; last name, first name, and middle initial. By convention, the last name is denoted as P.6.1 and the middle initial is denoted as P.6.3. If you include the repeat field number, the last name is denoted as P.6.1.1, first name is denoted as P.6.1.2, and the middle initial is denoted as P.6.1.3.

H|\^&  
P|1|PID123456|||Brown^Bobby^B|White|196501020304  
O|1|SID101||ABO\ABScr|||||||||||CENTBLOOD  
L|1|N

When field P.6 is parsed, "Brown^Bobby^B" is returned as the field value. This field value is then parsed for components using the component delimiter to extract the components. In this case, the component delimiter plays the same role as the comma does in the CSV format.

The escape character is the last of the 4 defined delimiters. The escape character is mostly used to create a sequence of characters to replace delimiters in the text of field values. This is similar to what is done in XML when &lt; is used to replace the reserved character *left angle bracket* (<) in text data.

Take for example the name of a profile that contains the repeat field delimiter, as in "ABO\Rh\ABScr", named after the analyses returned by the profile. If this profile name was placed in an order profile field as is, it would not be parsed as one profile, but three. To prevent this, embedded repeat field delimiters are replaced with an escape sequence, which will be converted back to the embedded repeat field delimiter when it is parsed by the receiver of the message.

When the sender writes "ABO\Rh\ABScr" into the order record, every repeat delimiter (\) is replaced with "&R&", resulting with a string of "ABO&R&Rh&R&ABScr". When the profile is read from the order record by the receiver of the message, every "&R&" is replaced with the repeat field delimiter (\), restoring the text data to it's original text.

Table 1: Recode Type IDs

| Type ID | Record Description | Level |
| --- | --- | --- |
| H | Message Header - contains information about the sender and defines delimiters | 0 |
| P | Patient - contains information about an individual patient | 1 |
| O | Order - when sent from an LIS, this record contains information about a test order. When sent by the instrument, it shall provide information about the test request. | 2 |
| R | Result - contains the results ofzl a single analytic determination. | 3 |
| M | Manufacture Information - the fields in this record are defined by the manufacturer. |  |
| Q | Request for information - used to request information, e.g. request outstanding orders for a sample. | 1 |
| L | Message Terminator - the last record in the message. A header record may be transmitted after this record signifying the start of a second message. | 0 |

* *ASTM E1381-02 Standard Specification for Low-Level Protocol to Transfer Messages Between Clinical Laboratory Instruments and Computer* *Systems (Withdrawn 2002)*

[Ortho Public Technical Documents](https://www.orthoclinicaldiagnostics.com/en-gb/home/technical-documents)

A Record Sequence Number is used in record types that may occur multiple tienes within a single message. The number used defines the z'th occurrence of the associated record type at a particular hierarchical level and is reset to one whenever a record of a greater hierarchical significance (lower number).

1. ASTM E1394 Standard Specification for Transferring Information Between Clinical Instruments and Computer Systems [↑](#footnote-ref-1)