

USE OF CARRIAGE RETURN LINE FEED

REQUEST

I have a client that is sending a fixed file format for the ANSI X12 835. In every segment at the 81st byte they send a CRLF. The CRLF can be for some segments right in the middle of a data element. They also send a segment terminator for that line at the end of the segment. When my translator reads this document, it errors stating that the next 3 bytes after the CRLF is an invalid segment tag. My interpretation of the X12 file format is that it is a variable length record format with the segment terminator being the end of record marker. Fixed file formats pose a problem due to a dual segment terminator per each segment therefore violating the definition of a segment composition and the segment terminator in X12. The versions affected are the HIPAA versions and addendums, specifically 004010X091 and 004010X091A.

REFERENCES

X12.6 Application Control Structure
X12.5 Interchange Control Structures

Version 004 Release 010
Version 004 Release 010

FORMAL INTERPRETATION

The CRLF characters described in this Request for Interpretation lie outside the scope of the X12 Standards. These characters are not defined in the X12 character set used in the interchange. The current X12 Standards (X12.5 and X12.6) are silent on how such characters are to be treated. Hence such treatment is implementation dependent.

DISCUSSION OF FORMAL INTERPRETATION

The X12 character set used to define data values is a graphical character set. In an interchange instance the X12 character set is extended by union with the set of delimiters defined for the interchange. These delimiters need not have common graphical representations.

Prior to the introduction of Binary Data into the X12 standards, wording was provided in the X12 Standard to ignore all extraneous bytes (those not representing the graphical character set or defined delimiters for the interchange), regardless of where in the data stream they appeared. Within an unfiltered binary data value, an X12 parser may or may not recognize and discard extraneous bytes that might have been supplied as part of a transmission packaging protocol. For this reason, the wording to ignore extraneous bytes was removed from the X12 standards.

Wording was not added to the X12 Standards to prevent an X12 parser from ignoring characters from the data stream that it deemed to be extraneous. In fact, use of a CRLF pair after each segment terminator, and use of a LF character following a segment terminator (perhaps a segment terminator defined as a CR) is a common practice in X12 data exchanges. One reason for this is that non-X12 knowledgeable display methods often 'look' better that way. Some

existing X12 parsers routinely ignore non-alphabetic characters following a segment terminator and preceding the next alphabetic character.

The CRLF characters described in this RFI are 'extraneous characters' supplied by a consistent algorithm, and a consistent algorithm could be employed by an X12-capable parser to remove those 'extraneous characters.' However, behavior of a non-X12 parser such as might be employed as part of a transmission protocol unpack operation might not be consistent, since the binary data may include data patterns that mimic the extraneous character pattern.

Some behaviors that a receiving entity might exhibit include:

- Front-end software might recognize such characters in the data stream prior to submitting them to an X12-capable parser. Upon such recognition, a receiving entity may employ some method to identify the algorithm required to remove the characters from the data stream, and attempt such removal. The application of such methods and algorithms, and the behavior of such methods and algorithms, is outside the scope of the X12 standards.
- An X12-capable parser might recognize such characters in the data stream. Upon such recognition, an X12-capable may employ some method to identify and apply an algorithm to ignore such characters in the data stream.

An X12-capable parser might recognize such characters in the data stream as invalid characters in the X12 syntax. The behavior of an X12-capable parser upon encountering such characters in the data stream is implementation dependent. In particular, per the description in the request for interpretation, such characters would appear within the ISA segment itself, and might prevent the parser from recognizing any of the content of the ISA segment.