**A****NSI MH10.8.15 – xxxx**

Committee Draft

**Notice**

This is a draft document and thus, is dynamic in nature. It does not reflect a consensus of the MH10.8.15 Committee and it may be changed or modified. Neither ANSI nor the MH10.8.15 Committee makes any representation or warranty, express of implied, with respect to the sufficiency, accuracy or utility of the information or opinion contained or reflected in the material utilized. ANSI and the MH10.8.15 Committee further expressly advise that any use of or reliance upon the material in question is at your risk and neither ANSI nor the MH10.8.15 Committee shall be liable for any damage or injury, of whatever nature, incurred by any person arising out of any utilization of the material. It is possible that this material will at some future date be included in a copyrighted work by ANSI.

***American National Standard  
for Material Handling - Specification for Handling Reader Output from ISO/IEC 15434 formatted AIDC Media***



***Material Handling Industry***

*8720 Red Oak Blvd., Suite 201*

*Charlotte, NC 28217-3992*

*Printed: XX/XXXX*

|  |  |
| --- | --- |
| American  National  Standard | This is a draft proposed American National Standard (ANS) of the accredited standards committee (ASC) MH 10. As such, this is not a completed standard. The MH 10/SC 8 Committee may modify this document as a result of comments received during public review and its approval as a standard.  Permission is granted to members of MH 10, its subcommittees, and their associated work groups to reproduce this document for the purposes of MH 10 standardization activities without further permission, provided this notice is included. All other rights are reserved. Any commercial or for-profit duplication is strictly prohibited.  Approval of an ANS requires verification by the American National Standards Institute (ANSI) that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer. Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered and that effort be made toward their resolution.  The use of an ANS is voluntary. Existence of the standard does not in any respect preclude anyone, whether he or she has approved the standards or not, from manufacturing, marketing, purchasing, or using items, processes, or procedures not confirming to the standards.  ANSI does not develop standards and in no circumstances will give interpretation on any ANS in the name of the ANSI. Address requests for interpretations to the secretariat or sponsor whose name appears on the title page of this standard.  **CAUTION NOTICE:** This ANS may be revised or withdrawn at any time. The ANSI procedures require periodic action to reaffirm, revise, or withdraw this standard. Purchasers of ANSs may receive current information on all standards by calling or writing the ANSI. |

Foreword v

Introduction vi

1 Scope 1

2 Normative references 1

3 Terms and definitions 1

4 xxxx Error! Bookmark not defined.

4.1 xxx Error! Bookmark not defined.

Foreword

Introduction

The ISO/IEC 15434 syntax for high-capacity AIDC media standard provides a standard manner of transferring data to and from high-capacity AIDC media. However this standard creates some difficulty for the systems integrator/end user to develop the interface to receive and process the data from AIDC readers through a computer keyboard wedge, USB port, Bluetooth device or other similar interfaces (e.g., interfaces to systems that do not use an RS-232 transmission protocol) and transmit to their Information Systems. This is due to the ISO/IEC 15434 specification of certain “unprintable” characters that are not easily processed by these hardware interfaces. This requires that the systems integrator/end user develop vendor specific interfaces to their Information Systems. The purpose of this standard is to provide a uniform alternate output in a popular format, XML, from AIDC readers to Information Systems for high capacity AIDC Media encoded per the ISO/IEC 15434 syntax, eliminating the need for vendor specific interfaces to Information Systems.

Readers enabled with the method described in this standard will be able to transmit the ISO/IEC 15434 syntax data using only "printable" characters thereby avoiding the "unprintable" characters used with the ISO/IEC 15434 syntax that can disrupt some data systems (e.g. systems not using an RS-232 transmission protocol).

ANSI MH10.8.15 — Specification for Handling Reader Output from ISO/IEC 15434 formatted AIDC Media

# Scope

This standard describes the requirements for a uniform alternate output format for transmission from AIDC readers to Information Systems for data encoded according to ISO/IEC 15434 *Syntax for high-capacity ADC media* ("15434 syntax"). This standard includes support for all current and future 15434 syntax assignments using a defined set of XML tags according to the XML protocol. This standard provides a process for converting 15434 syntax messages into XML-compliant output and includes several examples so that manufacturers of reader equipment can provide an optional standardized output for 15434 syntax data that can be interpreted unambiguously by application software.

# Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 646:1991, *Information technology -- ISO 7-bit coded character set for information interchange*

ISO/IEC 15434, *Information technology — Syntax for high capacity ADC media*

ISO/IEC 19762 (all parts), *Information technology -- Automatic identification and data capture (AIDC) techniques -- Harmonized vocabulary*

ISO 21067*, Packaging – Vocabulary (Is this needed?)*

# Terms and definitions

For the purposes of this document, the terms and definitions contained in ISO/IEC 19762 (all parts) apply.

Non-printable character

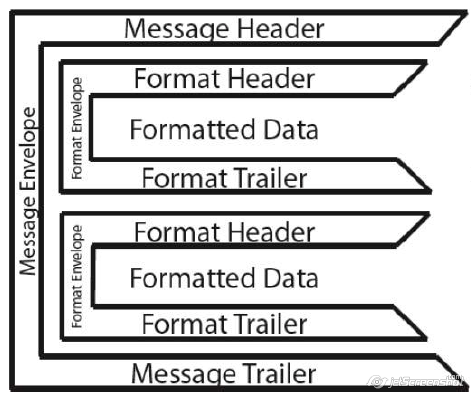
ISO 646 character not in the range of decimal 32 to 126

# Specification Details

## Symbol output encoding

Field length is the number of data characters in a data field. The character count of data characters is exclusive of overhead characters.

Based upon feedback into creating a standard XML format structure and to help maintain consistency with the ISO standard formatting, the following XML format structure: Table 1: ISO Message Format and Format Indicators, is recommended for all of the Formats.



|  |  |  |  |
| --- | --- | --- | --- |
| **Format Indicator** | **Variable Header Data** | **Format Trailer** | **Format Description** |
| 00 |  |  | Reserved for future use |
| 01 | **GS**vv | **RS** | Transportation |
| 02 |  |  | Complete EDI message / transaction |
| 03 | vvvrrr**FSGSUS** | **RS** | Structured data using ANSI ASC X12 Segments |
| 04 | vvvrrr**FSGSUS** | **RS** | Structured data using UN/EDIFACT Segments |
| 05 | **GS** | **RS** | Data using GS1 Application Identifiers |
| 06 | **GS** | **RS** | Data using ASC MH 10 Data Identifiers |
| 07 |  | **RS** | Free form text |
| 08 | vvvvrrnn |  | Structured data using CII Syntax Rules |
| 09 | **GS**ttt…t **GS**ccc…c **GS**nnn…n **GS** | **RS** | Binary data (file type) (compression technique) (number of bytes) |
| 10-11 |  |  | Reserved for future use |
| 12 | **GS** | **RS** | Structured data following Text Element Identifier rules |
| 12-99 |  |  | Reserved for future use |

Table — ISO Message Format and Format Indicators

The XML tags and their defined usage for this standard are described in the following table.

|  |  |
| --- | --- |
| **XML Tag** | **Usage** |
| MSG\_ISO\_IEC\_15434 | Message Envelope root element |
| FMT | Format Envelope element with a 2 character attribute indicating the associated Format Indicator |
| DATA | Data child element |
| BYTE | Non printable elements with an encoding attribute indicating the encoding method used for the associated data. Encode attribute values to be used: BASE16. |
| RS | Record separator |
| GS | Group separator |
| FS | File separator |
| US | United Separator |
| EOT | End of transmission |

Table : XML Tags

The following figure shows the basics of the use of XML tags to encode an ISO 15434 message.

|  |
| --- |
| <?xmlversion="1.0" encoding="UTF-8"?>  <MSG\_ISO\_IEC\_15434>  <RS/>  <FMT FI="03">  <DATA>vvvrrr</DATA>  <FS/>  <GS/>  <US/>  <DATA></DATA>  <RS/>  </FMT>  <FMT FI="06">  <GS/>  <DATA></DATA>  <GS/>  <DATA></DATA>  <GS/>  <DATA></DATA>  <RS/>  </FMT>  <EOT/>  </MSG\_ISO\_IEC\_15434> |

Figure : XML Format Structure

**Note:** Figure 1: XML Format Structure*, as well as the rest of the examples within this document, is only indented for readability and any spaces and/or tabs used for indentation will not be required by the scanner output.*

## Encoding of Binary Data

A non-printable character would be presented by <BYTE >xxxxx</BYTE>, where xxxxx is the associated value encode by the method indicated by the element’s attribute, but special tags will be used to represent FS, GS, RS, US, and LF, as format indicator “07” would require with two line feeds identifying paragraphs. 15434 Format “09” allows a message to include binary data. See Annex 3 for the XML method of encoding Format “09”.

## XML reserved characters

## There are three XML reserved characters that need to be converted if they appear in 15434 data. Table 3: displays the standard XML conversion for the XML reserved characters.

|  |  |  |  |
| --- | --- | --- | --- |
| **Character** | **Description** | **Entity Name** | **Entity Number** |
| < | Less than | &lt; | &#60; |
| > | Greater than | &gt; | &#62; |
| & | Ampersand | &amp; | &#38; |

Table : Example of XML Reserved Character Conversion

## Examples of ISO/IEC 15434 message encoding:

**First Example:**

The first example is from the new annex in ISO/IEC 15962 on 15434 support:

[)> RS 06 GS 25SUN043325711MH8031200000000001GS1T110780GSQ21GS4LUSRSEOT

Table : First XML Example

|  |
| --- |
| <?xmlversion="1.0" encoding="UTF-8"?>  <MSG\_ISO\_IEC\_15434>  <RS/>  <FMT FI="06">  <GS/>  <DATA>25SUN043325711MH8031200000000001</DATA>  <GS/>  <DATA>1T110780</DATA>  <GS/>  <DATA>Q21</DATA>  <GS/>  <DATA>4LUS</DATA>  <RS/>  </FMT>  <EOT/>  </MSG\_ISO\_IEC\_15434> |

**Second Example:**

The second example, is from the AIAG’s B11 Item-Level Radio Frequency Identification (RFID) Standard. It contains multiple “Format 06” (DI) format envelopes within a single 15434 Message envelope, to represent a series of “records,” each record being one component of a completed assembly:

|  |
| --- |
| **[)>RS**  **06GS25SUN98765432187654321A2B4C6D8ERS**  **06GSP34567812GS12V345678912GSTCC09030333333333RS**  **06GSP23456781GS2V234567891GSTBB09018222222222RS**  **06GSP12345678GS12V123456789GSTAA08274111111111RS**  **06GSP45678123GS12V456789123GSTDD09019444444444RS**  **06GSP56781234GS12V567891234GSTEE09016555555555RSEOT** |

Which is the 15434 representation of the following table:

Table : 15434 Representation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Assembly** | **DI** | **Part Number** | **DI** | **DUNS** | **DI** | **Trace Code** |
| **Gas Tank** | **25S** | **12345678** |  | **987654321** |  | **A2B4C6D8E** |
| **A** | **P** | **34567812** | **12V** | **345678912** | **T** | **CC09030333333333** |
| **B** | **P** | **23456781** | **12V** | **234567891** | **T** | **BB09018222222222** |
| **C** | **P** | **12345678** | **12V** | **123456789** | **T** | **AA08274111111111** |
| **D** | **P** | **45678123** | **12V** | **456789123** | **T** | **DD09019444444444** |
| **E** | **P** | **56781234** | **12V** | **567891234** | **T** | **EE09016555555555** |

Table : Second XML Example:

|  |
| --- |
| <?xmlversion="1.0" encoding="UTF-8"?>  <MSG\_ISO\_IEC\_15434>  <RS/>  <FMT FI="06">  <GS/>  <DATA>25SUN98765432187654321A2B4C6D8E</DATA>  <RS/>  </FMT>  <FMT FI="06">  <GS/>  <DATA>P34567812</DATA>  <GS/>  <DATA>12V 345678912</DATA>  <GS/>  <DATA>TCC09030333333333</DATA>  <RS/>  </FMT>  <FMT FI="06">  <GS/>  <DATA>P23456781</DATA>  <GS/>  <DATA>12V 234567891</DATA>  <GS/>  <DATA>TBB09018222222222</DATA>  <RS/>  </FMT>  <FMT FI="06">  <DATA>P12345678</DATA>  <GS/>  <DATA>12V 123456789</DATA>  <GS/>  <DATA>TAA08274111111111</DATA>  <RS/>  </FMT>  <FMT FI="06">  <GS/>  <DATA>P45678123</DATA>  <GS/>  <DATA>12V 456789123</DATA>  <GS/>  <DATA>TDD09019444444444</DATA>  <RS/>  </FMT>  <FMT FI="06">  <GS/>  <DATA>P56781234</DATA>  <GS/>  <DATA>12V567891234</DATA>  <GS/>  <DATA>TEE09016555555555</DATA>  <RS/>  </FMT>  <EOT/>  </MSG\_ISO\_IEC\_15434> |

## Mandatory Data (Is This needed?)

## Processing of Concatenated Symbols

In some cases a message may come from a Symbol that requires Concatenation of multiple bar codes. In these cases the concatenation processing is performed within the reader and the XML output will be the content of the concatenated symbols.

(Informative)  
Commonly Used Data Identifiers

Note: This is a partial listing. The complete listing is included in ANS MH10.8.2.

|  |  |  |
| --- | --- | --- |
| **Data Identifier** | **Identifier Description** | Application Identifier |
| 25S | Identification of a party to a transaction as identified in 18V, followed by the supplier assigned serial number. | DI25S |

ANNEX 1 Processing Steps

The translation from 15434 to XML is based on the following processing steps:

* The input message begins with “[)>”, triggering the first two output lines of XML:

<?xml version="1.0" encoding="iso-8859-1"?>

<MSG\_ISO\_IEC\_15434>

* The next input is always RS nn; set process variable inFormat = True and emit the XML line:

<FMT FI="nn">

* Set process variables foundEOT and inDataString to False
* Loop until either an EOT is found, or the end of the input message is reached:
  + If inDataString is True, and the next input character is not printable ASCII, then:
    - Set inDataString to False;
    - terminate the currently stored XML line with “</data>”, and
    - emit this line of XML.
    - Proceed to the immediately-following set of “If” statements, to process this same next input character
  + If the next character is RS, then:
    - Set inFormat to False, and emit the following two XML lines:

<RS/>

</FMT>

* + - Following this RS character, valid 15434 input will continue with either EOT or two decimal digits. If EOT, it will be processed by the next loop iteration, but if two decimal digits “nn” are found, then set inFormat to True, and emit

<FMT FI="nn">

and continue to the next loop iteration

* + Else, if the next character is EOT, then emit the two lines

<EOT/>

</MSG\_ISO\_IEC\_15434>

Set foundEOT = True, and exit the loop, because processing is completed

* + Else, if the next character is either GS, FS, or LF then emit a line consisting of <GS/>, <FS/>, or <LF/> and continue loop processing,
  + Else, if the next character is either <,> or & then replace the character with &lt, &gt or &amp and continue loop processing,
  + Else, if the next character is non-printable, then
    - begin a new XML line with

<Bytes>

* + - translate the character into two Hex characters from the set [0-9A-F]
    - continue translating to Hex until the next *printable* character is encountered;
    - terminate the current XML line with

<\Bytes>

and emit this XML line

* + - set inDataString to True (we’ll process this next char on next loop iteration)
    - start a new XML line with “<data>”
  + Else (the next character is a printable character)
    - If inDataString is False
      * Set inDataString to True
      * start a new XML line with “<data>”
    - add the printable character to the current XML line and continue

After exiting the main loop, if foundEOT is still false (this will be the outcome for Format 02 and Format 08, and maybe for some future 15434 formats too), then emit the following lines of XML:

</FMT> (emit this XML line only if inFormat is still True)

</MSG\_ISO\_IEC\_15434>

ANNEX 2 (Normative)  
Subset of ASCII/ISO 646 (Table of hexadecimal and decimal values)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **HEX** | **DEC** | **ASCII / ISO 646** | **HEX** | **DEC** | **ASCII / ISO 646** | **HEX** | **DEC** | **ASCII / ISO 646** |
| 00 | 00 | NUL | 30 | 48 | 0 | 60 | 96 | ' |
| 01 | 01 | SOH | 31 | 49 | 1 | 61 | 97 | a |
| 02 | 02 | STX | 32 | 50 | 2 | 62 | 98 | b |
| 03 | 03 | ETX | 33 | 51 | 3 | 63 | 99 | c |
| 04 | 04 | EOT | 34 | 52 | 4 | 64 | 100 | d |
| 05 | 05 | ENQ | 35 | 53 | 5 | 65 | 101 | e |
| 06 | 06 | ACK | 36 | 54 | 6 | 66 | 102 | f |
| 07 | 07 | BEL | 37 | 55 | 7 | 67 | 103 | g |
| 08 | 08 | BS | 38 | 56 | 8 | 68 | 104 | h |
| 09 | 09 | HT | 39 | 57 | 9 | 69 | 105 | i |
| 0A | 10 | **LF\*** | 3A | 58 | : | 6A | 106 | j |
| 0B | 11 | VT | 3B | 59 | ; | 6B | 107 | k |
| 0C | 12 | FF | 3C | 60 | **<\*** | 6C | 108 | l |
| 0D | 13 | CR | 3D | 61 | = | 6D | 109 | m |
| 0E | 14 | SO | 3E | 62 | **>\*** | 6E | 110 | n |
| 0F | 15 | SI | 3F | 63 | ? | 6F | 111 | o |
| 10 | 16 | DLE | 40 | 64 | @ | 70 | 112 | p |
| 11 | 17 | DC1 | 41 | 65 | A | 71 | 113 | q |
| 12 | 18 | DC2 | 42 | 66 | B | 72 | 114 | r |
| 13 | 19 | DC3 | 43 | 67 | C | 73 | 115 | s |
| 14 | 20 | DC4 | 44 | 68 | D | 74 | 116 | t |
| 15 | 21 | NAK | 45 | 69 | E | 75 | 117 | u |
| 16 | 22 | SYN | 46 | 70 | F | 76 | 118 | v |
| 17 | 23 | ETB | 47 | 71 | G | 77 | 119 | w |
| 18 | 24 | CAN | 48 | 72 | H | 78 | 120 | x |
| 19 | 25 | EM | 49 | 73 | I | 79 | 121 | y |
| 1A | 26 | SUB | 4A | 74 | J | 7A | 122 | z |
| 1B | 27 | ESC | 4B | 75 | K | 7B | 123 | { |
| 1C | 28 | **FS\*** | 4C | 76 | L | 7C | 124 | | |
| 1D | 29 | **GS\*** | 4D | 77 | M | 7D | 125 | } |
| 1E | 30 | **RS\*** | 4E | 78 | N | 7E | 126 | ~ |
| 1F | 31 | **US\*** | 4F | 79 | O | 7F | 127 | DEL |
| 20 | 32 | SP | 50 | 80 | P |  |  |  |
| 21 | 33 | ! | 51 | 81 | Q |  |  |  |
| 22 | 34 | “ | 52 | 82 | R |  |  |  |
| 23 | 35 | # | 53 | 83 | S |  |  |  |
| 24 | 36 | $ | 54 | 84 | T |  |  |  |
| 25 | 37 | % | 55 | 85 | U |  |  |  |
| 26 | 38 | **&\*** | 56 | 86 | V |  |  |  |
| 27 | 39 | ' | 57 | 87 | W |  |  |  |
| 28 | 40 | ( | 58 | 88 | X |  |  |  |
| 29 | 41 | ) | 59 | 89 | Y |  |  |  |
| 2A | 42 | \* | 5A | 90 | Z |  |  |  |
| 2B | 43 | + | 5B | 91 | [ |  |  |  |
| 2C | 44 | , | 5C | 92 | \ |  |  |  |
| 2D | 45 | - | 5D | 93 | ] |  |  |  |
| 2E | 46 | . | 5E | 94 | ^ |  |  |  |
| 2F | 47 | / | 5F | 95 | \_ |  |  |  |
| **\*These Characters have special handling within this standard.** | | | | | | | | |

**ANNEX 3**

**Format “09” encoding method**

Binary data is encoded using a subset of the printable characters to avoid non-printable characters and XML reserved characters. While not the most efficient method, two hexadecimal characters are used to represent each byte because of the ease of implementation.

Format “09” has the following structure.

09 GS ttt...t GS ccc...c GS nnn...n GS RS

where ttt is file type, such as “jpg” or “bmp”, ccc is the compression technique, if any, and nnn is the decimal number of bytes between the GS and the RS

The XML output contains two hex characters for every byte of data. Following is an example for a small photo shown below.



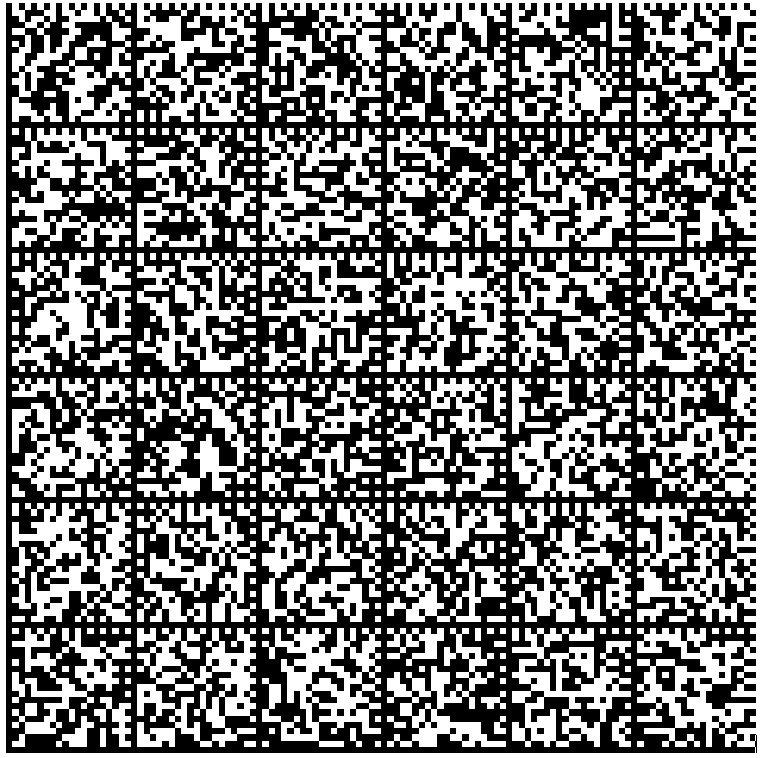
The file is in the JPG format and contains 877 bytes beginning and ending as follows

|  |  |  |  |
| --- | --- | --- | --- |
| Position | Character | Decimal ASCII value | Hex ASCII Value |
| 1 | ÿ | 255 | FF |
| 2 | Ø | 216 | D8 |
| 3 | ÿ | 255 | FF |
| 4 | à | 224 | E0 |
| 5 |  | 0 | 00 |
| 6 |  | 16 | 10 |
| 7 | J | 74 | 4A |
| 8 | F | 70 | 46 |
| . |  |  |  |
| . |  |  |  |
| . |  |  |  |
| 869 |  | 24 | 18 |
| 870 | ! | 33 | 21 |
| 871 | B | 66 | 42 |
| 872 | ˜ | 152 | 98 |
| 873 | c | 99 | 63 |
| 874 | » | 187 | BB |
| 875 | Ì | 204 | CC |
| 876 | ó | 243 | F3 |
| 877 | 2 | 50 | 32 |

The encoded data representation follows

[)> RS 09 GS JPG GS GS 877 GS ÿØÿà JF … !B˜c»Ìó2 RS EOT

The actual data encoded in a Data Matrix symbol is shown below.



The XML transmitted data representation follows.

|  |  |
| --- | --- |
| **XML Data** | **XML Data Annotation** |
| <?xmlversion="1.0" encoding="UTF-8"?> | Start of XML Data Stream |
| <MSG\_ISO\_IEC\_15434> | Start of ISO IEC 15434 Message |
| <RS/> | Record Separator |
| <FMT FI="09"> | Format = 09 |
| <GS/> | Group Separator |
| <DATA>JPG</DATA> | File Type (ttt) = JPG |
| <GS/> | Group Separator |
| <DATA></DATA> | Compression (ccc) = none This line |
| <GS/> | Group Separator |
| <DATA>877</DATA> | Decimal Number of Bytes (nnn) = 877 |
| <GS/> | Group Separator |
| <DATA>FFD8FFE000104A46…1821429863BBCCF332</DATA> | Data = 2 Character Hex data representation |
| <RS/> | Record Separator |
| </FMT> | End of Format |
| <EOT/> | End of Transmission |
| </MSG\_ISO\_IEC\_15434> | End of ISO IEC 15434 Message |