

X12 Standards

DISA and ASC X12 Standards

The ASC X12 Electronic Document Interchange (EDI) standards are published by the Data Interchange Standards Association, Inc. (DISA), who is the authorized Secretariat for Accredited Standards Committee X12 (ASC X12) of the American National Standards Institute (ANSI).

DISA Address

Data Interchange Standards Association,
Inc.
1800 Diagonal Road, Suite 200
Alexandria, VA 22314-2852
(703) 548-7005

Content of ASC X12 standards is proprietary, and Data Interchange Standards Association, Inc. claims a Copyright (c) to standards material with full control over distribution and use. Customware Solutions, Inc., dba AALtsys Technology, Inc., presents this material under license from DISA. Other use of this information is prohibited.

Note

DISA operates as a consortium committee for data exchange, but they are also a commercial entity. DISA license fees for standards table data run in the tens of thousands of US dollars. AAltSys has licensed table data for the following standards sets from DISA: 3060-X12, 3070-X12, 4010-X12, and 5010-X12. Requests to use other standards will require an agreement to pay the cost of table data licensing.

Function of EDI

The X12 standards define commonly used business transactions in a formal, structured manner called transaction sets. X12 standards differ from data exchange protocols, in that a protocol allows data to flow from one computer to another without regard to the structure or meaning of the exchange. EDI standards formats for data interchange provide an unambiguous structure for interpreting the documents exchanged by reference to standards, and without regard to the computer system which generates or receives the exchange.

Character sets

EDI consists of text data, and the text is restricted to using specific ASCII characters. A basic character set is defined for all exchanges, and this set may be extended with additional characters by mutual agreement of the trading partners. The characters described here are in universal use today, and using any additional characters is discouraged if not prohibited.

Basic characters

| | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------------|-----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Uppercase letters | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Numeric digits | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | | | | | | | | | | | | | | |
| Special characters | ! | " | & | ' | (|) | * | + | , | - | . | / | : | ; | ? | = | | | | | | | | | | |
| The space character | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Control characters: | BEL | HT | LF | VT | FF | CR | FS | GS | RS | US | NL | | | | | | | | | | | | | | | |
| ASCII hex values: | 07 | 09 | 0A | 0B | 0C | 0D | 1C | 1D | 1E | 1F | | | | | | | | | | | | | | | | |

Extended characters

| | |
|---------------------|---|
| Lowercase letters | a b c d e f g h i j k l m n o p q r s t u v w x y z |
| Other special chars | % ~ @ [] _ { } \ < > |
| National characters | # \$ |
| Control characters: | SOH STX ETX EOT ENQ ACK DC1 DC2 DC3 DC4 NAK SYN ETB |
| ASCII hex values: | 01 02 03 04 05 06 11 12 13 14 15 16 17 |

Delimiters

Elements and higher structures of the EDI transaction sets are distinguished by delimiters. Three delimiter values may be used: two levels of separators, for elements and sub-elements, and a terminator which indicates the end of a data segment. Delimiters are specified in the header of each interchange and shall not be used within a data element value elsewhere in the exchange. Actual characters used as delimiters are set within each exchange. Standards documentation indicates the use of a delimiter with symbols as shown in the sidebar.

Symbols representing delimiters

| Symbol | Type of delimiter |
|--------|-----------------------------|
| <tr> | Segment terminator |
| <gs> | Data element separator |
| <us> | Component element separator |

EDI standards do not impose requirements on delimiter choices other than to prohibit the use of a delimiter within a data element. Recommended delimiters are shown in the following table, along with values suggested in the standards and values in common use in industry:

| Delimiter | AAItsys recommends | Standards suggest | Industry practice |
|-----------|--|---|---|
| <tr> | Unix newline <i>0A</i> hex, 10 dec | any control char <i>1C</i> hex preferred | return/linefeed <i>0D0A</i> hex, CR/LF |
| <gs> | Unix backtick (`) <i>60</i> hex, 96 dec | special or control <i>1D</i> hex preferred | asterisk (*) tilde (~) |
| <us> | Unix carat (^) <i>5E</i> hex, 94 dec | special or control <i>1F</i> hex preferred | colon (:) |

Note

EDI data is exchanged in text files. With this in mind, AAItsys recommends delimiters which result in a text file that can be read in any Unix editor. Specifically the backtick and carat symbols are not included in the EDI character set, and therefore are not present in data elements. EDI standards recommendations do not result in a viewable file, and common industry practice produces corrupted files when collisions occur between data and delimiters.

Data Elements

The data element is the smallest named unit of information in the standard. Data elements are identified as either simple or component. A data element which occurs within a composite data structure is identified as a component data element. A data element which is outside the boundaries of a composite structure is a simple data element. The distinction between simple and component elements is based on the placement of the element in the set and not on the attributes of the element.

Each data element is assigned a unique reference identifier of from one to four digits. Data elements have three attributes: the type, the length minimum, and the length maximum. Data element types are as follows:

| Type | Code | Data Element Type Description |
|------------|-----------|---|
| Numeric | <i>Nn</i> | Numeric value with a predefined implied decimal point |
| Decimal | <i>R</i> | Numeric with an explicit decimal place |
| Identifier | <i>ID</i> | A unique coded value from a predetermined list |
| String | <i>AN</i> | A sequence of basic or extended characters |

| Type | Code | Data Element Type Description |
|--------------|-----------|---|
| Date | <i>DT</i> | A standard date in the format YYMMDD |
| Time | <i>TM</i> | A time in the format HHMMSSd...d |
| Binary | <i>B</i> | Numeric sequence of binary octets |
| Fixed String | <i>FS</i> | Space-padded sequence of basic or extended characters |

Composite Data Structures

The composite data structure is an intermediate unit of information in a segment. The definition of a composite data structure consists of two or more component data elements. In the actual data transmission the composite may consist of one or more component data elements. Adjacent component elements in a composite data structure are delimited by the component element separator *<us>*. Composite data structures are delimited from other adjacent elements in a segment by the data element separator *<gs>*.

Each composite data structure has a unique four-character reference identifier. The identifiers for composite data structures which appear in control segments start with an S; a C is the first letter of composite identifiers which are used in data segments.

A detail table defines the component data elements in a composite data structure. Each component referenced has a requirement and an ordinal sequence in the structure. In the actual data transmission omitted elements are indicated by including their delimiters to preserve this sequence, except for elements which are omitted at the end of the component data structure. An example composite data structure detail table is presented in the following table (Composite C022, Health Care Code Information):

| Ref | Ele | Element Name | Req | N | Cond | Relation | Tp | Min | Max |
|-----|------|-----------------------------------|-----|---|------|----------|----|-----|-----|
| 001 | 1270 | Code List Qualifier Code | M | Z | | | ID | 1 | 3 |
| 002 | 1271 | Industry Code | M | | | | AN | 1 | 30 |
| 003 | 1250 | Date Time Period Format Qualifier | C | Z | P | 3,4 | ID | 2 | 3 |
| 004 | 1251 | Date Time Period | C | | | | AN | 1 | 35 |
| . | . | ... | . | . | . | . | . | . | . |

Data Segment Structures

The data segment is an intermediate unit of related information in a transaction set. Simple data elements and composite data structures are the data parts of the segment. Each segment in a transmission starts with the segment identifier, followed by at least one data element or component structure, and ending with a segment terminator *<tr>*.

Each data segment has a unique two- or three-character identifier which also serves as a label for the segment in the data transmission. Segment labels are separated from the following data element by an element separator *<gs>*. The label is considered to be position zero of the segment, so that the first data element following the label is in position one.

A detail table defines the sequence of simple data elements and composite data structures in a segment. Each unit referenced has a requirement and an ordinal sequence in the segment. In the actual data transmission omitted elements are indicated by including their delimiters to preserve this sequence, except for elements which are omitted at the end of the segment. An example data segment detail table is presented following (Segment ADX, Adjustment):

| Seq | Ele | Element Name | Req | N | Cond | Relation | Tp | Min | Max |
|-----|-----|--------------|-----|---|------|----------|----|-----|-----|
|-----|-----|--------------|-----|---|------|----------|----|-----|-----|

| Seq | Ele | Element Name | Req | N | Cond | Relation | Tp | Min | Max |
|-----|-----|------------------------------------|-----|---|------|----------|----|-----|-----|
| 001 | 782 | Monetary Amount | M | Z | | | R | 1 | 15 |
| 002 | 426 | Adjustment Reason Code | M | Z | | | ID | 2 | 2 |
| 003 | 128 | Reference Identification Qualifier | C | Z | P | 3,4 | ID | 2 | 3 |
| 004 | 127 | Reference Identification | C | | | | AN | 1 | 30 |
| . | . | ... | . | . | . | . | . | . | . |

Transaction Sets

The transaction set is a complete unit of information exchanged between trading partners, representing a business document. Each transaction starts with a header segment (ST) and ends with a trailer segment (SE). At least one data segment is required between the header segment and the trailer. Each segment in the transaction set ends with the segment terminator `<tr>`.

The transaction set identifier uniquely identifies each transaction set. This identifier is the first data element of the transaction set header segment. The transaction set header and trailer segments contain a control number which must be identical for any given transaction. Transaction set control numbers should not repeat in the history of exchanges of the transaction set between two trading partners. The transaction set trailer segment also contains a count of the number of segments in the transaction including the ST and SE segments.

The sequence of data segments in a transaction set definition is presented in detail tables for the set. Up to three tables may be used to represent transaction header information, repeating details, and a summary area. Each segment in a set has a requirement designator, a position in the set definition, and a maximum occurrence. An example set detail table is shown below:

| N | A Pos | Seg | Segment Name | Req | MaxUse | Level | Repeat | Loop |
|---|--------|-----|-------------------------------|-----|--------|-------|--------|------|
| | 1 0010 | ST | Transaction Set Header | M | 1 | 0 | 0 | |
| | 1 0020 | BVP | Beginning Segment for Vehicle | M | 1 | 0 | 0 | |
| N | 1 0030 | G62 | Date/Time | M | 1 | 1 | 99 | G62 |
| | 1 0040 | VC | Motor Vehicle Control | M | 1 | 2 | 999 | VC |
| . | . | . | ... | . | . | . | . | . |

Repeating

Single data segments within a transaction set may repeat up to a specified maximum number of occurrences, as shown in the MaxUse column. The notation `>1` is used to show that the number of repeats for a segment is unlimited. Groups of two or more related data segments may be repeated as a loop. Loops may be either unbounded, or bounded by loop start (LS) and loop end (LE) segments.

Unbounded loops

The start of an unbounded loop is marked by the occurrence of the first segment of the loop. The beginning segment of an unbounded loop shall not appear anywhere else in a loop. The requirement for a loop is implicitly the requirement of the loop's first segment. If the requirement designator for the first segment of a loop is mandatory, then the loop must appear at least once in the transaction set. A loop may be repeated up to a specified maximum number of times. The notation `>1` designates an unlimited repeat.

A level entry indicates the nesting of loops, and the start of a loop structure is indicated by a loop label on the first segment of the loop. When unbounded loops are nested within loops, the inner loop shall not start at the same position as any outer loop. The inner loop shall not start with the same segment identifier as the start of any outer loop, nor may the inner loop contain a segment that is also the

beginning segment of any outer loop in the same structure. The inner loop must end before or on the same segment as its immediate outer loop.

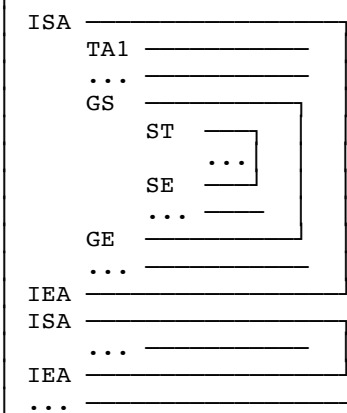
Bounded loops

The characteristics of unbounded loops also apply to bounded loops, except that bounded loops have no restriction on which segment begins the loop. For bounded loops, a unique loop identifier defined in the standard is used in the LS and LE segments to convey segment position or loop hierarchy, or both, within the transaction set.

Transmission Files

A transmission consists of a sequence of interchanges in a stream which are all addressed to a specific trading partner (as when receiving) or all addressed from a specific partner (as when sending).

The Sender and Receiver Identifiers in the ISA header of an interchange address the interchange envelope between partners. Therefore a transmission is analogous to a mail delivery to or a mail pickup from a mailbox. The sequence of control segments making up a transmission is displayed in the following figure:



ISA/IEA Each interchange is bounded by an ISA/IEA segment pair.

TA1 Transmission Acknowledgment segments, if used, immediately follow the ISA.

GS/GE Each Functional Group of similar transaction sets is bounded by a GS/GE segment pair.

ST/SE Each transaction set document starts with an ST and ends with an SE.

Each structure may repeat an arbitrary number of times in its position.