997 Functional Acknowledgment

VANTAGE GROUP® accepts functional acknowledgments for all EDI documents we send.
We send functional acknowledgments to trading partners that send us EDI documents.

For all EDI documents we send, we expect a functional acknowledgment within 24 hours of sending the document.

We send the functional acknowledgments as soon as the mapping process is complete.
If you have not received a functional acknowledgment within 24 hours, Please notify us at edi@vantage-group.com

Not Define:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes	Usage
	ISA	Interchange Control Header	M	1			Must Use
	GS	Functional Group Header	M	1			Must Use

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes	Usage
10	ST	Transaction Set Header	M	1			Must Use
20	AK1	Functional Group Response Header	M	1			Must Use
LOOP ID _ AK2					**999999**		
30	AK2	Transaction Set Response Header	O	1			Used
LOOP ID _ AK3					**999999**		
40	AK3	Data Segment Note	O	1			Used
50	AK4	Data Element Note	O	99			Used
60	AK5	Transaction Set Response Trailer	M	1			Must Use
70	AK9	Functional Group Response Trailer	M	1			Must Use
80	SE	Transaction Set Trailer	M	1			Must Use

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes	Usage

Summary:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes	Usage

Not Define:

GE	Functional Group Header	M	1	Must Use
IEA	Interchange Control Trailer	M	1	Must Use

ISA Interchange Control Header

Pos: **Max: 1**
Not Defined - Mandatory
Loop: N/A **Elements: 16**

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
ISA01	I01	Authorization Information Qualifier	M	ID	2/2	Must use
		Description:				
ISA02	I02	Authorization Information	M	AN	10/10	Must use
		Description:				
ISA03	I03	Security Information Qualifier	M	ID	2/2	Must use
		Description:				
ISA04	I04	Security Information	M	AN	10/10	Must use
		Description:				
ISA05	I05	Interchange Sender ID Qualifier	M	ID	2/2	Must use
		Description:				
ISA06	I06	Interchange Sender ID	M	AN	15/15	Must use
		Description:				
ISA07	I07	Interchange Receiver ID Qualifier	M	ID	2/2	Must use
		Description:				
ISA08	I08	Interchange Receiver ID	M	AN	15/15	Must use
		Description:				
ISA09	I09	Interchange Date	M	DT	6/6	Must use
		Description:				
ISA10	I10	Interchange Time	M	TM	4/4	Must use
		Description:				
ISA11	I11	Interchange Control Standards Identifier	M	ID	1/1	Must use
		Description:				
ISA12	I12	Interchange Control Version Number	M	ID	5/5	Must use
		Description:				
ISA13	I13	Interchange Control Number	M	N0	9/9	Must use
		Description:				
ISA14	I14	Acknowledgment Requested	M	ID	1/1	Must use
		Description:				
ISA15	I15	Usage Indicator	M	ID	1/1	Must use

		Description:				
ISA16	I16	Component Element Separato	M		1/1	Must use
		Description:				

GS Functional Group Header

Pos:	**Max: 1**
Not Defined - Mandatory	
Loop: N/A	**Elements: 8**

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
GS01	GS01	Functional Identifier Coder	M	ID	2/2	Must use
		Description:				
GS02	GS02	Application Sender’s Code	M	AN	2/15	Must use
		Description:				
GS03	GS03	Application Receiver’s Code	M	AN	2/15	Must use
		Description:				
GS04	GS04	Date	M	DT	8/8	Must use
		Description:				
GS05	GS05	Time	M	TM	4/8	Must use
		Description:				
GS06	GS06	Group Control Number	M	N0	1/9	Must use
		Description:				
GS07	GS07	Responsible Agency Code	M	ID	1/2	Must use
		Description:				
GS08	GS08	Version / Release / Industry Identifier Code	M	AN	1/12	Must use
		Description:				

ST Transaction Set Header

Pos: 10	**Max: 1**
Header - Mandatory	
Loop: N/A	**Elements: 2**

*See ASC X12 Nomenclature, to review the transaction set structure, including descriptions of segments, data elements, levels, and loops

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
ST001	143	Transaction Set Identifier Code **Description:**	M/Z	ID	3/3	Used
ST002	329	Transaction Set Control Number **Description:**	M	AN	4/9	Must use

AK1 Functional Group Response Header

Pos: 20	**Max: 1**
Header - Mandatory	
Loop: N/A	**Elements: 2**

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
AK1001	479	Functional Identifier Code	M/Z	ID	2/2	Used
		Description:				
AK1002	28	Group Control Number	M/Z	N0	1/9	Used
		Description:				

AK2 Transaction Set Response Header

Pos: 30	**Max: 1**
Header - Optional	
Loop: AK2	**Elements: 2**

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
AK2001	143	Transaction Set Identifier Code **Description:**	M/Z	ID	3/3	Used
AK2002	329	Transaction Set Control Number **Description:**	M/Z	AN	4/9	Used

AK3 Data Segment Note

Pos: 40	**Max: 1**
Header - Optional	
Loop: AK3	**Elements: 3**

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
AK3001	721	Segment ID Code	M	ID	2/3	Must use
		Description:				
AK3002	719	Segment Position in Transaction Set	M	N0	1/6	Must use
		Description:				
AK3003	447	Loop Identifier Code	O	AN	1/6	Used
		Description:				

AK4 Data Element Note

Pos: 50 **Max: 99**
Header - Optional
Loop: AK3 **Elements: 4**

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
AK4001	C030	Position in Segment	M			Must use
		Description:				
AK4001-1	722	Element Position in Segment	M	N0	1/2	Must use
		Description:				
AK4002	725	Data Element Reference Number	O	N0	1/4	Used
		Description:				
AK4003	723	Data Element Syntax Error Code	M	ID	1/3	Must use
		Description:				

AK5 Transaction Set Response Trailer

Pos: 60 **Max: 1**
Header - Mandatory
Loop: AK2 **Elements: 2**

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
AK5001	717	Transaction Set Acknowledgment Code	M	ID	1/1	Must use

Code	Purpose
A	Accepted
E	Accepted But Errors Were Noted
M	Rejected, Message Authentication Code (MAC) Failed
R	Rejected
W	Rejected, Assurance Failed Validity Tests
X	Rejected, Content After Decryption Could Not Be Analyzed

Description:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
AK5002	718	Transaction Set Syntax Error Code	O	ID	1/3	Used

Description:

AK9 Functional Group Response Trailer

Pos: 70 Max: 1
Header - Mandatory
Loop: N/A Elements: 4

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
AK9001	715	Functional Group Acknowledge Code **Code** **Purpose** A Accepted E Accepted, But Errors Were Noted. P Partially Accepted, At Least One Transaction Set Was Rejected R Rejected **Description:**	M	ID	1/1	Must use
AK9002	97	Number of Transaction Sets Included **Description:**	M	N0	1/6	Must use
AK9003	123	Number of Received Transaction Sets **Description:**	M	N0	1/6	Must use
AK9004	2	Number of Accepted Transaction Sets **Description:**	M	N0	1/6	Must use

SE Transaction Set Trailer

Pos: 80 **Max: 1**
Header - Mandatory
Loop: N/A **Elements: 2**

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
SE001	96	Number of Included Segments **Description:**	M	N0	1/10	Must use
SE002	329	Transaction Set Control Number **Description:**	M	AN	4/9	Must use

GE Functional Group Header

Pos: **Max: 1**
Not Defined - Mandatory
Loop: N/A **Elements: 2**

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
GE01	GE01	Number of Transaction Sets Included **Description:**	M	N0	1/6	Must use
GE02	GE02	Group Control Number **Description:**	M	N0	1/9	Must use

IEA Interchange Control Trailer

Pos:	**Max: 1**
Not Defined - Mandatory	
Loop: N/A	**Elements: 2**

Used

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max	Usage
IEA01	IEA01	Number of Included Functional Groups **Description:**	M	N0	1/5	Must use
IEA02	IEA02	Interchange Control Number **Description:**	M	N0	9/9	Must use

ASC X12 Nomenclature

Interchange and Application Control Structures

Interchange Control Structure

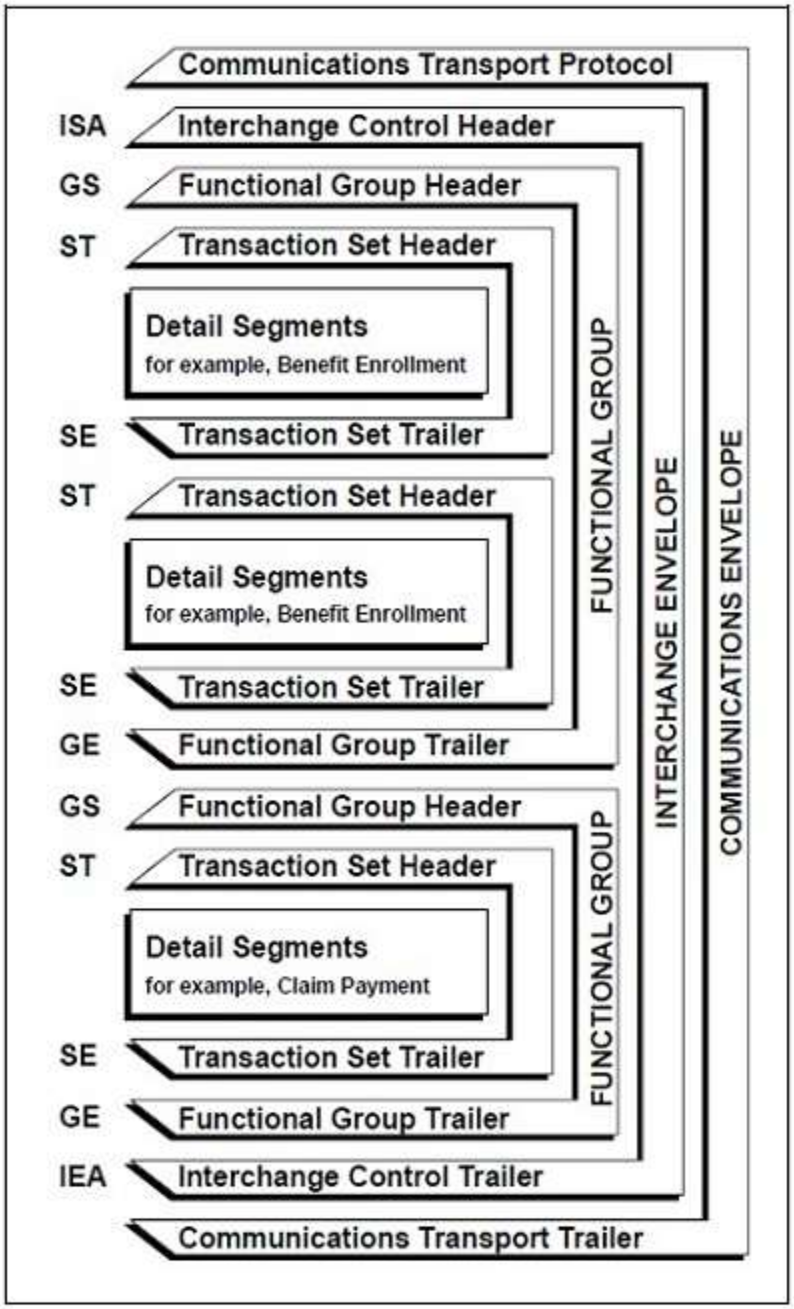


Figure A1. Transmission Control Schematic

The transmission of data proceeds according to very strict format rules to ensure the integrity and maintain the efficiency of the interchange. Each business grouping of data is called a transaction set. For instance, a group of benefit enrollments sent from a sponsor to a payer is considered a transaction set. Each transaction set contains groups of logically related data in units called segments. For instance, the N4 segment used in the transaction set conveys the city, state, ZIP Code, and other geographic information. A transaction set contains multiple segments, so the addresses of the different parties, for example, can be conveyed from one computer to the other. An analogy would be that the transaction set is like a freight train; the segments are like the train's cars; and each segment can contain several data elements the same as a train car can hold multiple crates. The sequence of the elements within one segment is specified by the ASC X12 standard as well as the sequence of segments in the transaction set. In a more conventional computing environment, the segments would be equivalent to records, and the elements equivalent to fields. Similar transaction sets, called “functional groups,” can be sent together within a transmission. Each functional group is prefaced by a group start segment; and a functional group is terminated by a group end segment. One or more functional groups are prefaced by an interchange header and followed by an interchange trailer. Figure A1, Transmission Control Schematic, illustrates this interchange control. The interchange header and trailer segments envelop one or more functional groups or interchange-related control segments and perform the following functions:

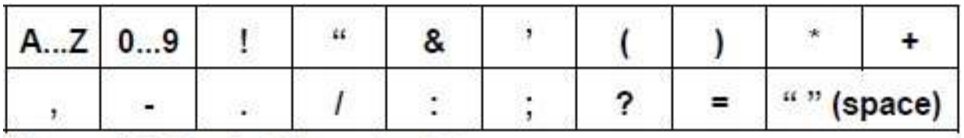
1. Define the data element separators and the data segment terminator.
2. Identify the sender and receiver.
3. Provide control information for the interchange.
4. Allow for authorization and security information.

Application Control Structure Definitions and Concepts

Basic Structure

A data element corresponds to a data field in data processing terminology. The data element is the smallest named item in the ASC X12 standard. A data segment corresponds to a record in data processing terminology. The data segment begins with a segment ID and contains related data elements. A control segment has the same structure as a data segment; the distinction is in the use. The data segment is used primarily to convey user information, but the control segment is used primarily to convey control information and to group data segments.

Basic Character Set



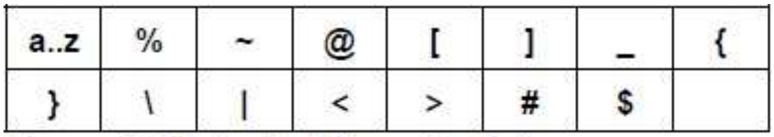
A...Z	0...9	!	“	&	’	(	)	*	+
,	-	.	/	:	;	?	=	“ ” (space)	

Figure A2. Basic Character Set

The section that follows is designed to have representation in the common character code schemes of EBCDIC, ASCII, and CCITT International Alphabet 5. The ASC X12 standards are graphic-character-oriented; therefore, common character encoding schemes other than those specified herein may be used as long as a common mapping is available. Because the graphic characters have an implied mapping across character code schemes, those bit patterns are not provided here.

The basic character set of this standard, shown in figure A2, Basic Character Set, includes those selected from the uppercase letters, digits, space, and special characters as specified below.

Extended Character Set



a..z	%	~	@	[	]	_	{
}	\	|	<	>	#	$	

Figure A3. Extended Character Set

An extended character set may be used by negotiation between the two parties and includes the lowercase letters and other special characters as specified in figure A3, Extended Character Set.

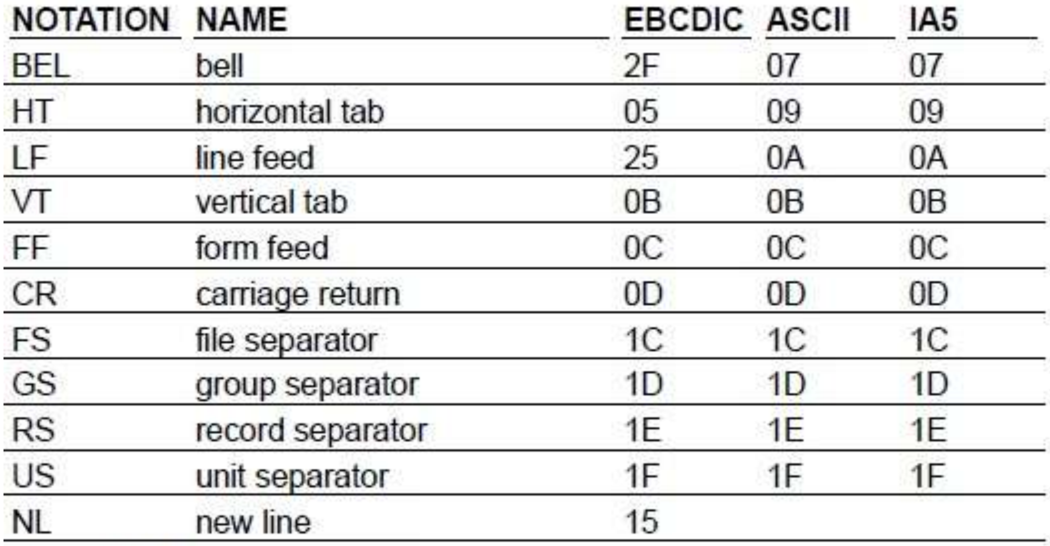
Note that the extended characters include several character codes that have multiple graphical representations for a specific bit pattern. The complete list appears in other standards such as CCITT S.5.

Use of the USA graphics for these codes presents no problem unless data is exchanged with an international partner. Other problems, such as the translation of item descriptions from English to French, arise when exchanging data with an international partner, but minimizing the use of codes with multiple graphics eliminates one of the more obvious problems.

Control Characters

Two control character groups are specified; they have only restricted usage. The common notation for these groups is also provided, together with the character coding in three common alphabets. In the matrix A1, Base Control Set, the column IA5 represents CCITT V.3 International Alphabet 5.

Base Control Set

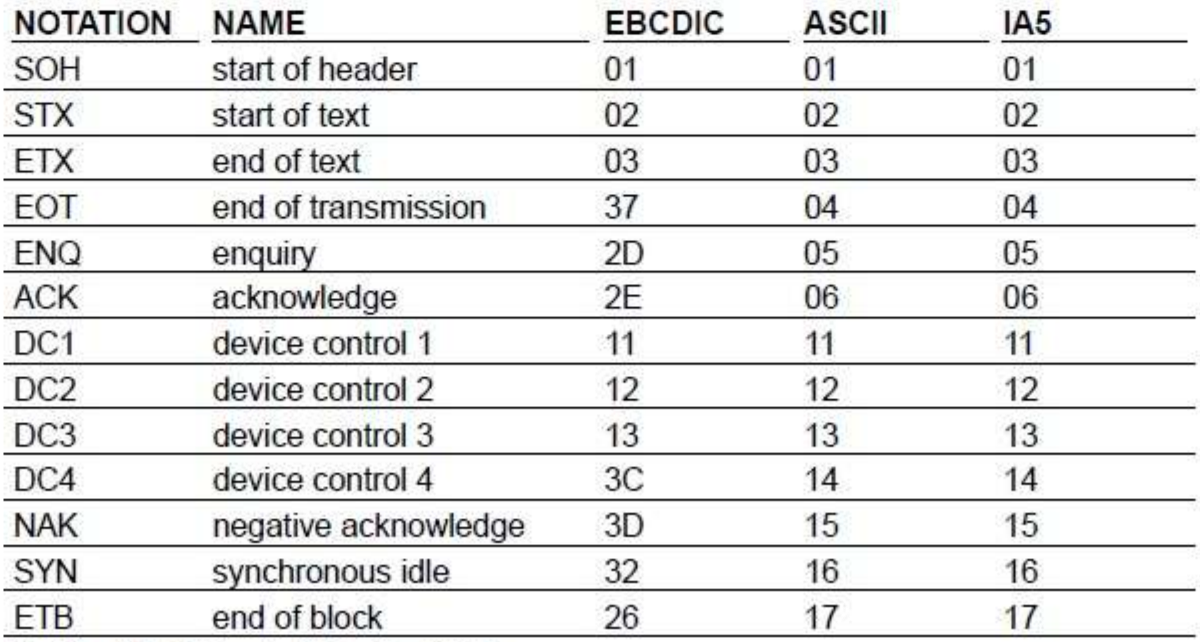


NOTATION	NAME	EBCDIC	ASCII	IA5
BEL	bell	2F	07	07
HT	horizontal tab	05	09	09
LF	line feed	25	0A	0A
VT	vertical tab	0B	0B	0B
FF	form feed	0C	0C	0C
CR	carriage return	0D	0D	0D
FS	file separator	1C	1C	1C
GS	group separator	1D	1D	1D
RS	record separator	1E	1E	1E
US	unit separator	1F	1F	1F
NL	new line	15		

Matrix A1. Base Control Set

The base control set includes those characters that will not have a disruptive effect on most communication protocols. These are represented by: The Group Separator (GS) may be an exception in this set because it is used in the 3780 communications protocol to indicate blank space compression.

Extended Control Set

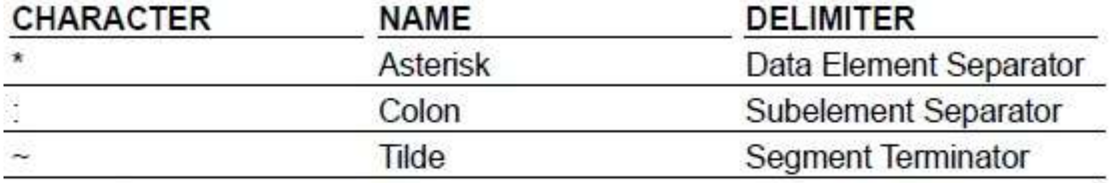


NOTATION	NAME	EBCDIC	ASCII	IA5
SOH	start of header	01	01	01
STX	start of text	02	02	02
ETX	end of text	03	03	03
EOT	end of transmission	37	04	04
ENQ	enquiry	2D	05	05
ACK	acknowledge	2E	06	06
DC1	device control 1	11	11	11
DC2	device control 2	12	12	12
DC3	device control 3	13	13	13
DC4	device control 4	3C	14	14
NAK	negative acknowledge	3D	15	15
SYN	synchronous idle	32	16	16
ETB	end of block	26	17	17

Matrix A2. Extended Control Set

The extended control set includes those that may have an effect on a transmission system. These are shown in matrix A2, Extended Control Set.

Delimiters



CHARACTER	NAME	DELIMITER
*	Asterisk	Data Element Separator
:	Colon	Subelement Separator
~	Tilde	Segment Terminator

Matrix A3. Delimiters

A delimiter is a character used to separate two data elements (or subelements) or to terminate a segment. The delimiters are an integral part of the data.

Delimiters are specified in the interchange header segment, ISA. The ISA segment is a 105 byte fixed length record. The data element separator is byte number 4; the component element separator is byte number 105; and the segment terminator is the byte that immediately follows the component element separator. Once specified in the interchange header, the delimiters are not to be used in a data element value elsewhere in the interchange. For consistency, this implementation guide uses the delimiters shown in matrix A3, Delimiters, in all examples of EDI transmissions.

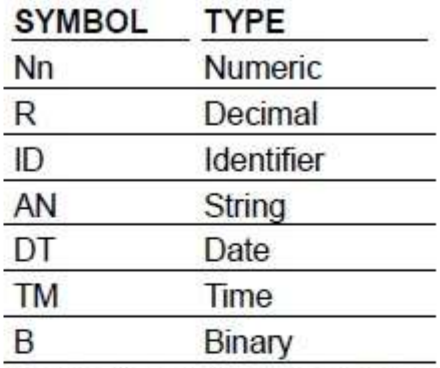
The delimiters above are for illustration purposes only and are not specific recommendations or requirements. Users of this implementation guide should be aware that an application system may use some valid delimiter characters within the application data. Occurrences of delimiter characters in transmitted data within a data element can result in errors in translation programs. The existence of asterisks (*) within transmitted application data is a known issue that can affect translation software.

Business Transaction Structure Definitions and Concepts

The ASC X12 standards define commonly used business transactions (such as a health care claim) in a formal structure called “transaction sets.” A transaction set is composed of a transaction set header control segment, one or more data segments, and a transaction set trailer control segment. Each segment is composed of the following:

- A unique segment ID
- One or more logically related data elements each preceded by a data element separator
- A segment terminator

Data Element



SYMBOL	TYPE
Nn	Numeric
R	Decimal
ID	Identifier
AN	String
DT	Date
TM	Time
B	Binary

Matrix A4. Data Element Types

The data element is the smallest named unit of information in the ASC X12 standard. Data elements are identified as either simple or component. A data element that occurs as an ordinarily positioned member of a composite data structure is identified as a component data element. A data element that occurs in a segment outside the defined boundaries of a composite data structure is identified as a simple data element. The distinction between simple and component data elements is strictly a matter of context because a data element can be used in either capacity.

Data elements are assigned a unique reference number. Each data element has a name, description, type, minimum length, and maximum length. For ID type data elements, this guide provides the applicable ASC X12 code values and their descriptions or references where the valid code list can be obtained. Each data element is assigned a minimum and maximum length. The length of the data element value is the number of character positions used except as noted for numeric, decimal, and binary elements.

The data element types shown in matrix A4, Data Element Types, appear in this implementation guide.

Numeric

A numeric data element is represented by one or more digits with an optional leading sign representing a value in the normal base of 10. The value of a numeric data element includes an implied decimal point. It is used when the position of the decimal point within the data is permanently fixed and is not to be transmitted with the data.

This set of guides denotes the number of implied decimal positions. The representation for this data element type is “Nn” where N indicates that it is numeric and n indicates the number of decimal positions to the right of the implied decimal point.

If n is 0, it need not appear in the specification; N is equivalent to N0. For negative values, the leading minus sign (-) is used. Absence of a sign indicates a positive value. The plus sign (+) should not be transmitted.

EXAMPLE

A transmitted value of 1234, when specified as numeric type N2, represents a value of 12.34.

Leading zeros should be suppressed unless necessary to satisfy a minimum length requirement. The length of a numeric type data element does not include the optional sign.

Decimal

A decimal data element may contain an explicit decimal point and is used for numeric values that have a varying number of decimal positions. This data element type is represented as “R.”

The decimal point always appears in the character stream if the decimal point is at any place other than the right end. If the value is an integer (decimal point at the right end) the decimal point should be omitted. For negative values, the leading minus sign (-) is used. Absence of a sign indicates a positive value. The plus sign (+) should not be transmitted.

Leading zeros should be suppressed unless necessary to satisfy a minimum length requirement. Trailing zeros following the decimal point should be suppressed unless necessary to indicate precision. The use of triad separators (for example, the commas in 1,000,000) is expressly prohibited. The length of a decimal type data element does not include the optional leading sign or decimal point.

EXAMPLE

A transmitted value of 12.34 represents a decimal value of 12.34.

Identifier

An identifier data element always contains a value from a predefined list of codes that is maintained by the ASC X12 Committee or some other body recognized by the Committee. Trailing spaces should be suppressed unless they are necessary to satisfy a minimum length. An identifier is always left justified. The representation for this data element type is “ID.”

String

A string data element is a sequence of any characters from the basic or extended character sets. The significant characters shall be left justified. Leading spaces, when they occur, are presumed to be significant characters. Trailing spaces should be suppressed unless they are necessary to satisfy a minimum length. The representation for this data element type is “AN.”

Date

A date data element is used to express the standard date in either YYMMDD or CCYYMMDD format in which CC is the first two digits of the calendar year, YY is the last two digits of the calendar year, MM is the month (01 to 12), and DD is the day in the month (01 to 31). The representation for this data element type is “DT.” Users of this guide should note that all dates within transactions are 8-character dates (millennium compliant) in the format CCYYMMDD. The only date data element that is in format YYMMDD is the Interchange Date data element in the ISA segment, and also used in the TA1 Interchange Acknowledgment, where the century can be readily interpolated because of the nature of an interchange header.

Time

A time data element is used to express the ISO standard time HHMMSSd..d format in which HH is the hour for a 24 hour clock (00 to 23), MM is the minute (00 to 59), SS is the second (00 to 59) and d..d is decimal seconds. The representation for this data element type is “TM.” The length of the data element determines the format of the transmitted time.

EXAMPLE

Transmitted data elements of four characters denote HHMM. Transmitted data elements of six characters denote HHMMSS.

Composite Data Structure

The composite data structure is an intermediate unit of information in a segment. Composite data structures are composed of one or more logically related simple data elements, each, except the last, followed by a sub-element separator. The final data element is followed by the next data element separator or the segment terminator. Each simple data element within a composite is called a component. Each composite data structure has a unique four-character identifier, a name, and a purpose. The identifier serves as a label for the composite. A composite data structure can be further defined through the use of syntax notes, semantic notes, and comments. Each component within the composite is further characterized by a reference designator and a condition designator. The reference designators and the condition designators are described below.

Data Segment

The data segment is an intermediate unit of information in a transaction set. In the data stream, a data segment consists of a segment identifier, one or more composite data structures or simple data elements each preceded by a data element separator and succeeded by a segment terminator.

Each data segment has a unique two- or three-character identifier, a name, and a purpose. The identifier serves as a label for the data segment. A segment can be further defined through the use of syntax notes, semantic notes, and comments. Each simple data element or composite data structure within the segment is further characterized by a reference designator and a condition designator.

Syntax Notes

Syntax notes describe relational conditions among two or more data segment units within the same segment, or among two or more component data elements within the same composite data structure.

Semantic Notes

Simple data elements or composite data structures may be referenced by a semantic note within a particular segment. A semantic note provides important additional information regarding the intended meaning of a designated data element, particularly a generic type, in the context of its use within a specific data segment. Semantic notes may also define a relational condition among data elements in a segment based on the presence of a specific value (or one of a set of values) in one of the data elements.

Comments

A segment comment provides additional information regarding the intended use of the segment.

Reference Designator

Each simple data element or composite data structure in a segment is provided a structured code that indicates the segment in which it is used and the sequential position within the segment. The code is composed of the segment identifier followed by a two-digit number that defines the position of the simple data element or composite data structure in that segment.

For purposes of creating reference designators, the composite data structure is viewed as the hierarchical equal of the simple data element. Each component data element in a composite data structure is identified by a suffix appended to the reference designator for the composite data structure of which it is a member. This suffix is a two-digit number, prefixed with a hyphen, that defines the position of the component data element in the composite data structure.

EXAMPLE

· The first simple element of the CLP segment would be identified as CLP01.

· The first position in the SVC segment is occupied by a composite data structure that contains seven component data elements, the reference designator for the second component data element would be SVC01-02.

Condition Designator

This section provides information about X12 standard conditions designators. It is provided so that users will have information about the general standard. Implementation guides may impose other conditions designators.

Data element conditions are of three types: mandatory, optional, and relational. They define the circumstances under which a data element may be required to be present or not present in a particular segment.

DESIGNATOR	DESCRIPTION
M- Mandatory	The designation of mandatory is absolute in the sense that there is no dependency on other data elements. This designation may apply to either simple data elements or composite data structures. If the designation applies to a composite data structure, then at least one value of a component data element in that composite data structure shall be included in the data segment.
O- Optional	The designation of optional means that there is no requirement for a simple data element or composite data structure to be present in the segment. The presence of a value for a simple data element or the presence of value for any of the component data elements of a composite data structure is at the option of the sender.
X- Relational	Relational conditions may exist among two or more simple data elements within the same data segment based on the presence or absence of one of those data elements (presence means a data element must not be empty). Relational conditions are specified by a condition code (see table below) and the reference designators of the affected data elements. A data element may be subject to more than one relational condition. The definitions for each of the condition codes used within syntax notes are detailed below:

CONDITION CODE	DEFINITION
P- Paired or Multiple	If any element specified in the relational condition is present, then all of the elements specified must be present.
R- Required	At least one of the elements specified in the condition must be present.
E- Exclusion	Not more than one of the elements specified in the condition may be present.
C- Conditional	If the first element specified in the condition is present, then all other elements must be present. However, any or all of the elements not specified as the first element in the condition may appear without requiring that the first element be present. The order of the elements in the condition does not have to be the same as the order of the data elements in the data segment.
L- List Conditional	If the first element specified in the condition is present, then at least one of the remaining elements must be present. However, any or all of the elements not specified as the first element in the condition may appear without requiring that the first element be present. The order of the elements in the condition does not have to be the same as the order of the data elements in the data segment.

Control Segments

A control segment has the same structure as a data segment, but it is used for transferring control information rather than application information.

Loop Control Segments

Loop control segments are used only to delineate bounded loops. Delineation of the loop shall consist of the loop header (LS segment) and the loop trailer (LE segment). The loop header defines the start of a structure that must contain one or more iterations of a loop of data segments and provides the loop identifier for this loop. The loop trailer defines the end of the structure. The LS segment appears only before the first occurrence of the loop, and the LE segment appears only after the last occurrence of the loop. Unbounded looping structures do not use loop control segments.

Transaction Set Control Segments

The transaction set is delineated by the transaction set header (ST segment) and the transaction set trailer (SE segment). The transaction set header identifies the start and identifier of the transaction set. The transaction set trailer identifies the end of the transaction set and provides a count of the data segments, which includes the ST and SE segments.

Functional Group Control Segments

The functional group is delineated by the functional group header (GS segment) and the functional group trailer (GE segment). The functional group header starts and identifies one or more related transaction sets and provides a control number and application identification information. The functional group trailer defines the end of the functional group of related transaction sets and provides a count of contained transaction sets.

Relations among Control Segments

The control segment of this standard must have a nested relationship as is shown and annotated in this subsection. The letters preceding the control segment name are the segment identifier for that control segment. The indentation of segment identifiers shown below indicates the subordination among control segments.

GS Functional Group Header, starts a group of related transaction sets.
ST Transaction Set Header, starts a transaction set.
LS Loop Header, starts a bounded loop of data segments but is not part of the loop.
LS Loop Header, starts an inner, nested, bounded loop.
LE Loop Trailer, ends an inner, nested bounded loop.
LE Loop Trailer, ends a bounded loop of data segments but is not part of the loop.
SE Transaction Set Trailer, ends a transaction set.
GE Functional Group Trailer, ends a group of related transaction sets.

More than one ST/SE pair, each representing a transaction set, may be used within one functional group. Also more than one LS/LE pair, each representing a bounded loop, may be used within one transaction set.

Transaction Set

The transaction set is the smallest meaningful set of information exchanged between trading partners. The transaction set consists of a transaction set header segment, one or more data segments in a specified order, and a transaction set trailer segment. See figure A1, Transmission Control Schematic.

Transaction Set Header and Trailer

A transaction set identifier uniquely identifies a transaction set. This identifier is the first data element of the Transaction Set Header Segment (ST). A user assigned transaction set control number in the header must match the control number in the Trailer Segment (SE) for any given transaction set. The value for the number of included segments in the SE segment is the total number of segments in the transaction set, including the ST and SE segments.

Data Segment Groups

The data segments in a transaction set may be repeated as individual data segments or as unbounded or bounded loops.

Repeated Occurrences of Single Data Segments

When a single data segment is allowed to be repeated, it may have a specified maximum number of occurrences defined at each specified position within a given transaction set standard. Alternatively, a segment may be allowed to repeat an unlimited number of times. The notation for an unlimited number of repetitions is ">1."

Loops of Data Segments

Loops are groups of semantically related segments. Data segment loops may be unbounded or bounded.

Unbounded Loops

To establish the iteration of a loop, the first data segment in the loop must appear once and only once in each iteration. Loops may have a specified maximum number of repetitions. Alternatively, the loop may be specified as having an unlimited number of iterations. The notation for an unlimited number of repetitions is ">1."

A specified sequence of segments is in the loop. Loops themselves are optional or mandatory. The requirement designator of the beginning segment of a loop indicates whether at least one occurrence of the loop is required. Each appearance of the beginning segment defines an occurrence of the loop.

The requirement designator of any segment within the loop after the beginning segment applies to that segment for each occurrence of the loop. If there is a mandatory requirement designator for any data segment within the loop after the beginning segment, that data segment is mandatory for each occurrence of the loop. If the loop is optional, the mandatory segment only occurs if the loop occurs.

Bounded Loops

The characteristics of unbounded loops described previously also apply to bounded loops. In addition, bounded loops require a Loop Start Segment (LS) to appear before the first occurrence and a Loop End Segment (LE) to appear after the last occurrence of the loop. If the loop does not occur, the LS and LE segments are suppressed.

Data Segments in a Transaction Set

When data segments are combined to form a transaction set, three characteristics are applied to each data segment: a requirement designator, a position in the transaction set, and a maximum occurrence.

Data Segment Requirement Designators

A data segment, or loop, has one of the following requirement designators for health care and insurance transaction sets, indicating its appearance in the data stream of a transmission. These requirement designators are represented by a single character code.

DESIGNATOR	DESCRIPTION
M- Mandatory	This data segment must be included in the transaction set. (Note that a data segment may be mandatory in a loop of data segments, but the loop itself is optional if the beginning segment of the loop is designated as optional.)
O- Optional	The presence of this data segment is the option of the sending party.

Data Segment Position

The ordinal positions of the segments in a transaction set are explicitly specified for that transaction. Subject to the flexibility provided by the optional requirement designators of the segments, this positioning must be maintained.

Data Segment Occurrence

A data segment may have a maximum occurrence of one, a finite number greater than one, or an unlimited number indicated by “>1.”

Functional Group

A functional group is a group of similar transaction sets that is bounded by a functional group header segment and a functional group trailer segment. The functional identifier defines the group of transactions that may be included within the functional group. The value for the functional group control number in the header and trailer control segments must be identical for any given group. The value for the number of included transaction sets is the total number of transaction sets in the group. See figure A1, Transmission Control Schematic.

Envelopes and Control Structures

Interchange Control Structures

Typically, the term “interchange” connotes the ISA/IEA envelope that is transmitted between trading/business partners. Interchange control is achieved through several “control” components. The interchange control number is contained in data element ISA13 of the ISA segment. The identical control number must also occur in data element 02 of the IEA segment. Most commercial translation software products will verify that these two fields are identical. In most translation software products, if these fields are different the interchange will be “suspended” in error.

There are many other features of the ISA segment that are used for control measures. For instance, the ISA segment contains data elements such as authorization information, security information, sender identification, and receiver identification that can be used for control purposes. These data elements are agreed upon by the trading partners prior to transmission and are contained in the written trading partner agreement. The interchange date and time data elements as well as the interchange control number within the ISA segment are used for debugging purposes when there is a problem with the transmission or the interchange. Data Element ISA12, Interchange Control Version Number, indicates the version of the ISA/IEA envelope. The ISA12 does not indicate the version of the transaction set that is being transmitted but rather the envelope that encapsulates the transaction. An Interchange Acknowledgment can be denoted through data element ISA14. The acknowledgment that would be sent in reply to a “yes” condition in data element ISA14 would be the TA1 segment. Data element ISA15, Test Indicator, is used between trading partners to indicate that the transmission is in a “test” or “production” mode. This becomes significant when the production phase of the project is to commence. Data element ISA16, Subelement Separator, is used by the translator for interpretation of composite data elements. The ending component of the interchange or ISA/IEA envelope is the IEA segment. Data element IEA01 indicates the number of functional groups that are included within the interchange. In most commercial translation software products, an aggregate count of functional groups is kept while interpreting the interchange. This count is then verified with data element IEA01. If there is a discrepancy, in most commercial products, the interchange is suspended. The other data element in the IEA segment is IEA02 which is referenced above.

Functional Groups

Control structures within the functional group envelope include the functional identifier code in GS01. The Functional Identifier Code is used by the commercial translation software during interpretation of the interchange to determine the different transaction sets that may be included within the functional group. If an inappropriate transaction set is contained within the functional group, most commercial translation software will suspend the functional group within the interchange.

The Application Sender’s Code in GS02 can be used to identify the sending unit of the transmission. The Application Receiver’s Code in GS03 can be used to identify the receiving unit of the transmission.

The functional group contains a creation date (GS04) and creation time (GS05) for the functional group. The Group Control Number is contained in GS06. These data elements (GS04, GS05, AND GS06) can be used for debugging purposes during problem resolution. GS08,Version/Release/Industry Identifier Code is the version/release/sub-release of the transaction sets being transmitted in this functional group. The GS08 does not represent the version of the interchange (ISA/IEA) envelope but rather the version/release/sub-release of the transaction sets that are encompassed within the GS/GE envelope.
The Functional Group Control Number in GS06 must be identical to data element 02 of the GE segment. Data element GE01 indicates the number of transaction sets within the functional group. In most commercial translation software products, an aggregate count of the transaction sets is kept while interpreting the functional group. This count is then verified with data element GE01.