

ANSI ASC X12 Standards Overview Tutorial

Learn About Key e-Commerce Trends and
Technologies at Your Own Pace

Welcome	3
How To Use This Tutorial.....	3
Tutorial Objectives.....	4
Standards 101	4
Why Standardize?.....	4
What is a Standard?.....	5
What is ANSI? ANSI ASC X12?	5
What are the “X12 Standards”?	6
What is a Transaction Set?	6
Structure	7
Segments and Sequence	7
Beginning and Ending	7
Unique Segments.....	8
Transaction Set Segment Requirements Designation	8
MAX Use.....	8
Loops	9
Nested Loops.....	10
Summary	10
Segment Structure.....	11
Data Segment Types	13
Summary	14
Qualifiers and Values In ANSI ASC X12	14
Data Element Types	15
Composite Structures.....	16
Summary	17
Envelope Process	17
Envelope Structure.....	17
Transaction Set Envelopes.....	18
Functional Group Envelopes.....	19
Functional Group Sub Addressing	20
Interchange Envelopes.....	21
Summary of Envelopes	21
Group Level FA.....	22
Error Reporting	23
ANSI Wrap-up	23

ANSI X12—this overview provides need-to-know information on the ANSI X12 standards for electronic data interchange (EDI), including transaction sets, data elements and functional acknowledgements.

Welcome

This tutorial is an overview of the ANSI ASC X12 Standard format.

The topics covered in this module are:

- **Objectives of this tutorial and how to use it**
- **ANSI ASC X12 101**—some basic information about Standards and ANSI Transaction Sets—structure, segments, loops, etc.
- **Segments**—structure, interpretations, conditions, etc.
- **Data Elements**—what they are and how to use them
- **Message Structure and Enveloping**—packaging it all
- **Functional Acknowledgments**—who needs them?

How To Use This Tutorial

This tutorial provides basic information about ANSI and ANSI ASC X12. It should be used as a prerequisite to understanding what ANSI ASC X12 is and how ANSI ASC X12 is developing standards to meet the needs of electronic commerce solutions.

It is only the beginning. This tutorial will introduce terms and concepts that you will find necessary to form an awareness of the ANSI ASC X12 standards.

The best way to use this tutorial is to read through a module and its subtopics, though not necessarily at one sitting. Each module in this tutorial builds on information presented in earlier modules but an individual subtopic can be used as reference outside of the linear progression of the course. You will find summaries at the end of each module and exercises throughout so you can evaluate your understanding of the material.

Please send all feedback to Client Training at: training@gxs.com

Tutorial Objectives

After completing this tutorial, you should be able to discuss:

- What is ANSI and ANSI ASC X12
- How and why the ANSI ASC X12 standard was developed
- What is the basic structure of the ANSI ASC X12 format
- How ANSI ASC X12 documents are read
- What Functional Acknowledgments are and why are they used
- How the ANSI ASC X12 standard may be used in providing electronic commerce solutions

Standards 101

Why Standardize?

There are almost as many business computer programs as there are businesses. In the early days, each business had its own programs for tracking merchandise, ordering, invoicing, accounts payable, receivable, and other business needs. We soon realized that:

The computer programs of one business couldn't talk to those of another which meant that all data that was received would need to be re-entered.

The programs in one department of a business couldn't talk to those of another in the same business—order entry couldn't talk to invoicing which couldn't talk to accounts receivable. Required data needed to be re-entered two, three or more times.

The answer was to standardize the data that was read by a computer program so that the data could be read by all programs with that standard. Can you read the purchase order below?

Human readable purchase order:

<i>Quantity</i>	<i>Unit</i>	<i>Price</i>	<i>No.</i>	<i>Description</i>
3	CSE	12.75	6900	CELLULOSE SPONGES
12	EA	.45	P450	PLASTIC PAILS
4	EA	.95	1640Y	DISH DRAINER
1	DZ	3.00	1507	6" FLOWER POTS

Standards translate the 'human readable' invoice to a 'machine readable' format (ANSI X12 version/release 004010*):

```
IT1*1*3*CA*12.75**VC*6900 N/L
IT1*2*12*EA*.457**VC*P450 N/L
IT1*3*4*EA*.95**VC*1640Y N/L
IT1*4*1*DZ*3**VC*1507 N/L
```

***Note: All examples in this tutorial are based on ANSI ASC X12 version/release 004010 unless otherwise noted**

What is a Standard?

A **standard** is a method of coding data to facilitate Electronic Data Interchange (EDI).

It provides:

- Rules of syntax
- Definition of the data organization
- Editing rules and conventions
- Published public documentation (i.e., a standards manual)

This provides the standards user with:

- **An open system**—where trade is possible with anyone who uses the same standard
- **Reduced implementation effort**—the implementation of a standard can, itself, be standardized
- **Third-party interfaces**—software and network applications can be written that address specific business needs and conform to a single standard

What is ANSI? ANSI ASC X12?

ANSI is an abbreviation for the American National Standards Institute that has been coordinating standards in the United States since 1918. ANSI offers an open forum for all concerned to identify needs, create plans to meet those needs, and come to an agreement on the proposed standards.

The Institute has a number of committees including the ANSI Accredited Standards Committee X12 (**ANSI ASC X12**). This committee is a voluntary standards committee that consists of subcommittees representing both private and public sectors in many industries. The subcommittees use a consensus process to propose a new standard or changes to existing standards. These standards enable the electronic exchange of business transactions.

The standard that has been recommended by this committee is known as the **ANSI ASC X12** Standard. It is sometimes called the ANSI X12 Standard or simply the X12 Standard.

What are the “X12 Standards”?

So, you might ask, what are the X12 Standards? How do I get them and what do I get?

You can order the X12 Standard by contacting:

Publications Department
Data Interchange Standards Association
7600 Leesburg Pike, Suite 340
Falls Church, VA 22043

Phone: (703) 970-4480

FAX: (703) 970-4488

e-mail: publications@disa.org

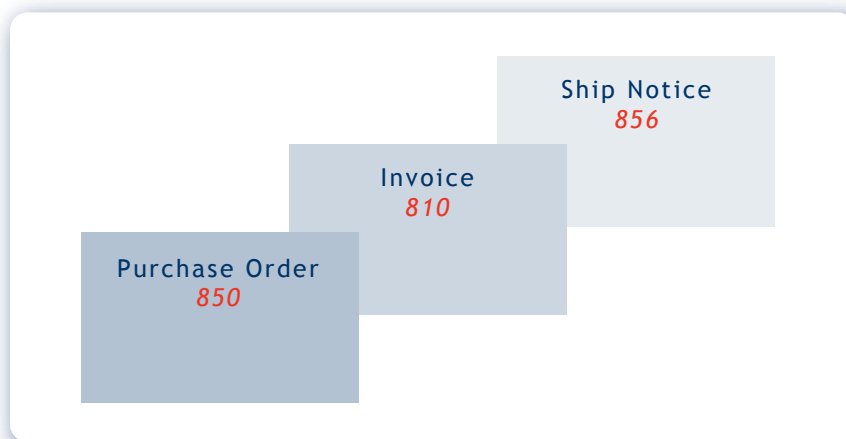
Internet: www.disa.org

You will receive the X12 Standards manual that includes:

- **Transaction Set Tables**—Transaction Sets are commonly used business transactions (PO, Invoice, etc.). They are made up of Segments which are made up of Data Elements.
- **Segment Directory**—A listing of all possible segments.
- **Data Element Dictionary**—A definition for all possible data elements.
- **Code Sources**—ANSI ASC X12 allows industry-specific codes. These codes are not maintained by the ANSI ASC X12 committee, but by industry organizations. A list of where to obtain these industry codes can be found in the ASC X12 Standards manual.
- **Interactive Control Structure**—These are syntax rules at the application level that have been agreed upon by the United Nations Economic Commission for Europe (UN/ECE) for Electronic Data Interchange for Administration, Commerce, and Transport (EDIFACT).

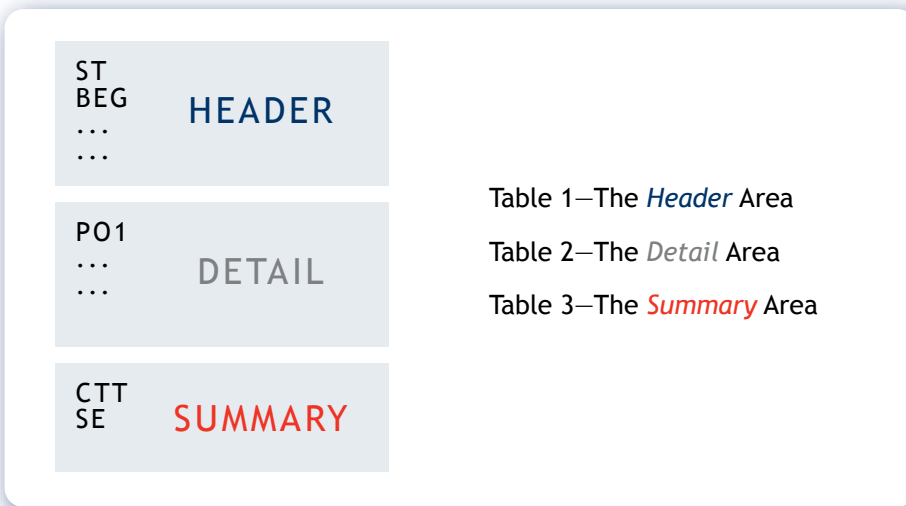
What is a Transaction Set?

A Transaction Set is a single business document such as a Purchase Order, Invoice, or Shipment Notice. There are hundreds of Transaction Sets available in the ANSI ASC X12 standards. Each set of transaction data is identified by a three digit code number.



Structure

Many Transaction Sets have three parts. The segments that may be used in each of these parts, within a specific document (i.e., invoice), are specified in associated tables defined in the X12 Standards document. For example:



Segments and Sequence

The tables show which segments may be used in a Transaction Set and the required sequence of the segments. For example, the Purchase Order Transaction Set table shows:

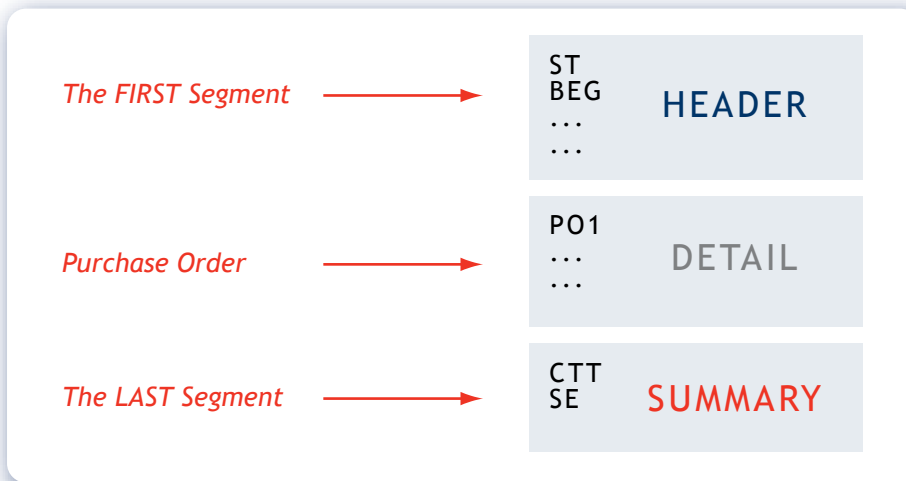
<i>ID</i>	<i>TITLE</i>
ST	Transaction Set Header
BEG	Beginning Segment for Purchase Order
CUR	Currency
REF	Reference Identification
PER	Administrative Communications Contract
TAX	Tax Reference
FOB	F.O.B. Related Information
CTP	Pricing Information
PAM	Period Amount
CSH	Sales Requirements
TC2	Commodity
etc.	
etc.	

Beginning and Ending

Transaction Sets:

- Begin with the Transaction Set Header (ST) segment
- End with the Transaction Set Trailer (SE) segment

These two segments are the innermost level of the three levels of envelopes within the 850 Purchase Order.



Enveloping is discussed later in this course.

Unique Segments

Many Transaction Sets have a unique beginning segment that immediately follows the ST segment. Here are a few examples of transaction sets that contain unique beginning segments:

Transaction Set:----- **Beginning Segment:**
Purchase Order----- **850 BEG Segment**
Price/Sales Catalog----- **832 BCT Segment**
Ship Notice----- **856 BSN Segment**

Transaction Set Segment Requirements Designation

The standard provides a requirements designator for each segment. The requirements designator indicates if the segment is required, optional or conditional.

Mandatory At least one occurrence must appear in the Transaction Set.
Optional May be used if needed.
Floating Floating segment that may appear anywhere in the transaction set between the ST and SE segments (valid in Standard version/releases 003060 and earlier).

MAX Use

The Transaction Set table shows specific attributes of a segment. The MAX use column of the segment detail table shows how many times you may repeat a particular segment at its location in the transaction. This example is for an 850 Purchase Order.

Loops

Loops show how a specified group of segments (two or more segments) may repeat in a Transaction Set.

<i>SEG ID</i>	<i>REQ. DES.</i>	<i>MAX USE</i>
ST	M	1
BEG	M	1
CUR	0	1
REF	0	>1
PER	0	3
TAX	0	>1
FOB	0	>1
CTP	0	>1
PAM	0	10
CSH	0	5
TC2	0	>1

One example of a loop is the N1 (Name and Address) loop within table 1 of the Purchase Order. The table below shows that the N1 loop includes the N1 segment through the PKG segment. The loop is optional, but if any segment in the loop is used, the first segment within the loop becomes mandatory. For example, IF any segment in the N1 loop is used, it MUST begin with the first segment (N1) in the loop. The 200 means that up to 200 different names and addresses may be entered using this loop.

If a segment, other than the first segment in a loop, has a USAGE of M (Mandatory), it is only mandatory if the loop is used. This example is for an 850 Purchase Order.

<i>SEG ID</i>	<i>REQ. DES.</i>	<i>MAX USE</i>	<i>LOOP</i>
ST	M	1	
BEG	M	1	
CUR	0	1	
REF	0	>1	
PER	0	3	
TAX	0	>1	
FOB	0	>1	
etc.			
etc.			
N1	0	1	N1/200
N2	0	2	
N3	0	2	
N4	0	>1	
NX2	0	>1	
REF	0	12	
PER	0	>1	
SI	0	>1	
FOB	0	1	
TD1	0	2	
TD5	0	12	
TD3	0	12	
TD4	0	5	
PKG	0	200	
etc.			

Nested Loops

Loops may be nested. This means that a loop may exist inside another loop. The brackets identify the segments that are part of the loop. Here is an 850 Purchase Order example:

POS NO.	SEG ID		USAGE	MAX USE	LOOP REP
LOOP	ID - PO1				100000
010	PO1	Baseline Item Data	M	1	}
015	LIN	Item Identification	O	>1	
018	SI	Service Characteristic Id.		>1	
020	CUR	Currency	O	1	
025	CN1	Contract Information	O	1	
030	PO3	Additional Item Detail	O	25	
LOOP	ID - CTP				
040	CTP	Pricing Information	O	1	}
043	CUR	Currency	O	1	
045	PAM	Period Amount	O	1	}
049	MEA	Measurements	O	10	
LOOP	ID - PID				1000
050	PID	Product/Item Description	O	1	}
060	MEA	Measurements	O	10	
070	PWK	Paperwork	O	25	}
090	PO4	Item Physical Description	O	>1	

In the example above:

There is one main outer PO1 loop. The PO1 loop can repeat up to 100,000 times. The example shows that an inner CTP loop exists which does not have a limit on how many times it can repeat within a single PO1 loop. Another inner loop is the PID loop which may repeat up to 1,000 times within a single PO1 loop.

Notice that the PAM and MEA segments are members of the outer PO1 loop.

Summary

Here is what we have covered in this module:

A transaction set (i.e., 850, 810, etc.) is a standard for a business document. For example, 850 is used for a purchase order, 810 for an invoice.

Transaction sets all begin with a mandatory ST segment and end with a mandatory SE segment.

Transaction sets may contain a unique, mandatory beginning segment (e.g., BEG in the PO).

The other segments in the transaction set are mainly 'generic' segments that can appear in many transaction sets.

The Standard contains specifications for:

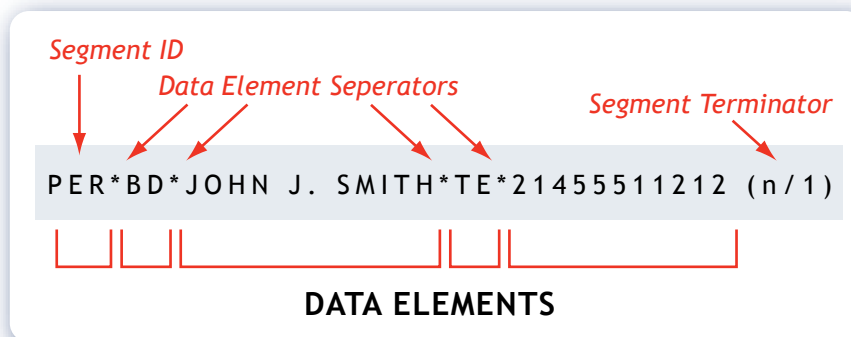
- Which segments may be used in the transaction set
- Sequence in which the segments must appear
- Which segments are mandatory or optional
- Maximum use of repeating segments
- Loop structure and usage

Segment Structure

Segments are a collection of logically-related data elements in a fixed, defined sequence.

Segments contain:

- Variable length data elements
- Data element separator or delimiter
- Segment terminator



The segment terminator (in this example (n/l)) is for optics only. In ANSI, a non-printable character can be used as a segment terminator. Whatever segment terminator character is used, that character cannot be used in the actual content of the data.

The structure of a data segment is determined by its data element attributes:

- All mandatory (M) data elements must contain data.
- Optional (O) or relational (X) data elements may or may not contain data depending on the requirements of a particular transmission.

Data elements must be accounted for by their position in the segment. If an optional or relational data element does not have data, that data element must still be accounted for in its position by using the appropriate number of data element separators to 'skip over' the empty field.

Name (N102) is RELATIONAL, And Not Used Here

N1*ST**92**42168 (n/1)

When there is a relational condition (indicated by an 'X') the syntax notes define the relations. Here is a list of the element ID prefixes and their meaning.

P Paired or Multiple

If any element specified in the relational condition is present, then all must be present.

R Required

At least one specified in the relational condition must be present.

E Exclusion

Not more than one can be present.

C Conditional

If the first element specified in the condition is present, then all others must be present.

L List Conditional

If the first element specified is present, then at least one of the remaining elements must be present.

Here are some examples of data elements with the relations for each defined by the prefix:

P0304

R0203

E010405

C010405

L010405

The end of each segment is uniquely determined by the Data Segment Terminator. Optional or Relational data elements without data that appear at the end of a data segment DO NOT need additional data element separators to correctly position the data. For example:

Indicates No More Data

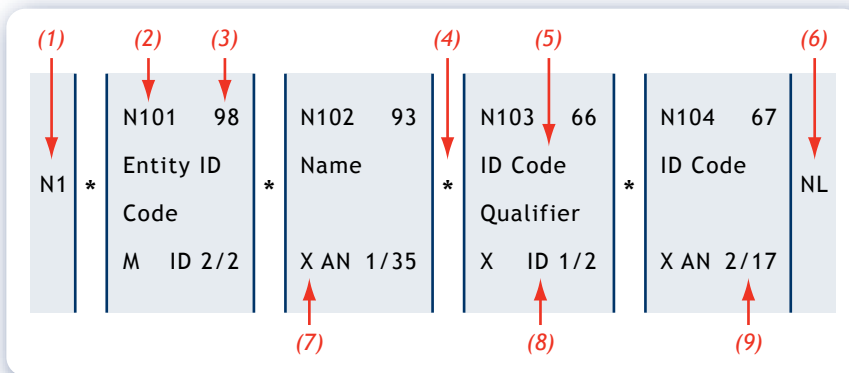
N1*BY*DON'S DISTRIBUTION(n/1)

Not Needed

N1*BY*DON'S DISTRIBUTION***(n/1)

Data Segment Types

In releases prior to October 1993, data segments are defined in diagrams. However, many industry implementation guidelines use the diagram format to define data segment usage. Here is an example of these diagrams:



Notes:

- **R0203**—At least one of N102 or N103 is required.
- **P0304**—If either N103 or N104 is present, then the other is required.

- (1) Data Segment Identifier
- (2) Reference Designator within Segment
- (3) Data Dictionary Number
- (4) Data Element Separator
- (5) Data Element Title
- (6) Data Segment Terminator
- (7) Condition Designator (M, O, X) or Semantic Note Designator (Z)
- (8) Data Element Type (N, B, R, AN, ID, DT, TM)
- (9) Data Element Length (Minimum/Maximum)

Data Segments (post-1993)

REF.	ELE. ID		ATTRIBUTES		
01	98	Entity Identifier Code	M	ID	2/3
02	93	Name	X	AN	1/60
03	66	Identification Code Qualifier	X	ID	1/2
04	67	Identification	X	AN	2/80
05	706	Entity Relationship Code	O	ID	2/2
06	98	Entity Identifier Code	O	ID	2/3

SYNTAX NOTES

COMMENTS

Summary

Here is what we have covered in this module:

1. Data segments contain a collection of data elements in a defined order.
2. All data elements are separated by a data element separator character.
3. There is a positional organization to the data within a segment.
4. The data segment is terminated using a segment terminator character.
5. Data element usage within a segment is mandatory (M), optional (O) or relational (X).

Qualifiers and Values In ANSI ASC X12

ANSI ASC X12 uses a pair of data elements together. The first data element is a qualifier. It defines what the second element (the value) means.

Qualifiers are typically ID (code value) fields. For example, in the 'PER' segment, the pair of data elements PER03 and PER04 determines the method of contacting someone. The qualifier (365) is a code (ID field) with values:

- BN** Beeper Number
- FX** Facsimile
- TE** Telephone
- TL** Telex
- TM** Telemail
- EM** Electronic Mail
- CP** Cell Phone

These codes can be used to qualify a particular value to show what kind of communication number is in a particular PER segment:

```

PER*CR*JOHN J. JOHNSON*TE*2145551212 → (Telephone)
PER*CR*SAM SMITH*TL*0198237667 → (Telex)
PER*CR*BILL JONES*EM*BJONES@ACMECO.COM → (E-mail)
PER*PE*ED ENGINEER*BN*FR1001.53 → (Beeper Number)
PER*QIR*IVAN INSPECTOR*CP*3013339999 → (Cell Phone)

```

Data Element Types

Numeric

Numeric data element types contain only numeric characters and a minus sign if they are negative. The digit after “N” defines the number of implied decimal positions. Therefore, N0 indicates a numeric value with no decimal position (a whole number); N2 indicates numeric value with implied two decimal positions.

<i>ATTRIBUTES</i>		<i>SAMPLE DATA</i>
N0	01/04	1 123 1234 -1234 are OK 1.23 A12 12345 are ILLEGAL values
N2	01/04	1 (value is 0.01) is OK 123 (value is 1.23) is OK -123 (value is -1.23) is OK -1234 (value is -12.34) is OK 1.23 A12 -12345 are ILLEGAL

Decimal

Decimal data element types contain only numeric characters, a decimal point (if needed), and a minus sign if negative.

<i>ATTRIBUTES</i>		<i>SAMPLE DATA</i>
R	01/04	1.00 .123 .0005 -1.234 123 (value is 123.) are OK 1.2345 A12.3 -12345 are ILLEGAL values

String

String data element types contain alphanumeric characters, including imbedded spaces.

<i>ATTRIBUTES</i>		<i>SAMPLE DATA</i>
AN	01/08	12345 TOM A12 AL SMITH are OK JOHN Q. PUBLIC is an ILLEGAL value

Date

The date in the format: YYMMDD or CCYYMMDD. The eight-digit date is allowed for version/releases 003072 and later.

<i>ATTRIBUTES</i>		<i>SAMPLE DATA</i>
DT	08/08	February 21, 2000 would be 20000221
DT	06/06	February 21, 2000 would be 000221

Time

The time in the format: HHMMSSd...d (24-hour clock). The length of the data element determines the significance of the time specified. The minimum length is four, therefore at least HHMM are specified. If the maximum length is six, then up to HHMMSS can be transmitted. A longer maximum length allows for decimal portions of seconds to be specified.

Binary

A sequence of octets ranging in value from 00000000 to 11111111. This data element type has no defined maximum length. The length is specified by the preceding data element.

ID

If ANSI ASC X12 maintains the list of ID codes, the code values follow the data element definition in the data element dictionary.

If ANSI ASC X12 does not maintain the ID code list, the source is in the Appendix of the ANSI ASC X12 manual. ID codes of this type include:

- Dunn & Bradstreet codes
- Medical codes
- Airport codes
- Telecommunications codes
- Petroleum codes
- Bureau of Census codes
- Government (tax, traffic, armed services) codes
- Pharmaceuticals codes

Composite Structures

What is a Composite Data Element Structure?

A Composite Data Element Structure is the intermediate unit of information in a segment. It is made up of component data elements separated by sub-element separators. Components have the condition designators of:

M-----Mandatory

Ø-----Optional

X-----Relational

A Semantic Note can be associated with a data element. A Semantic Note provides important additional information regarding the intended use of a data element. If there is a Semantic Note associated with a data element, a “/Z” follows the requirement designator.

Ø/Z-Optional with Semantic Note

X/Z-Relational with Semantic Note

Here is an example of a Composite Data Element Structure, the C003 (Composite Medical Procedure Identifier):

REF	ELE. ID	NAME	ATTRIBUTES		
01	235	Product/Service ID Qualifier	M	ID	2/2
02	234	Product/Service ID	M	AN	1/48
03	1339	Procedure Modifier	O/Z	AN	2/2
04	1339	Procedure Modifier	O/Z	AN	2/2
05	1339	Procedure Modifier	O/Z	AN	2/2
06	1339	Procedure Modifier	O/Z	AN	2/2
07	352	Description	O/Z	AN	1/80

SEMANTIC NOTES

03 C003-03 modifies the value in C003-02
04 C003-04 modifies the value in C003-02
05 C003-05 modifies the value in C003-02
06 C003-06 modifies the value in C003-02
07 C003-07 is the description of the procedure indentified in C003-02

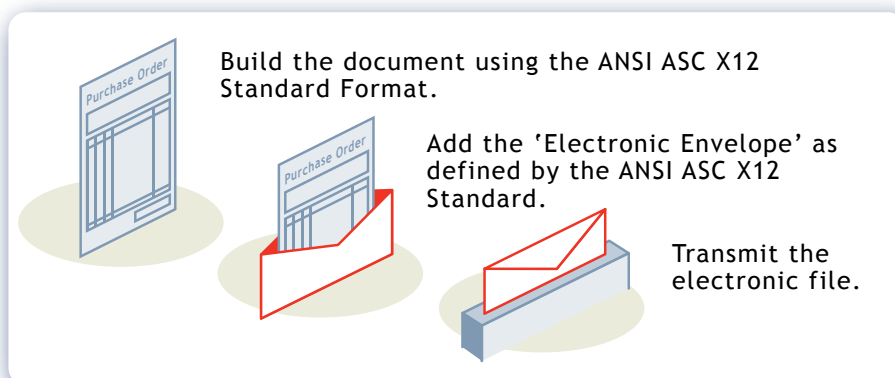
Summary

Here is what we have covered in this module:

1. Data element qualifiers and values
2. Data element attributes:
 - Usage: mandatory (M), optional (O) or relational (X)
 - Data element types: N, R, AN, DT, TM, B, ID
 - Field Length: minimum/maximum
3. Composite Data Elements and their components

Envelope Process

There are three significant steps for creating and sending data using the ANSI ASC X12 format:

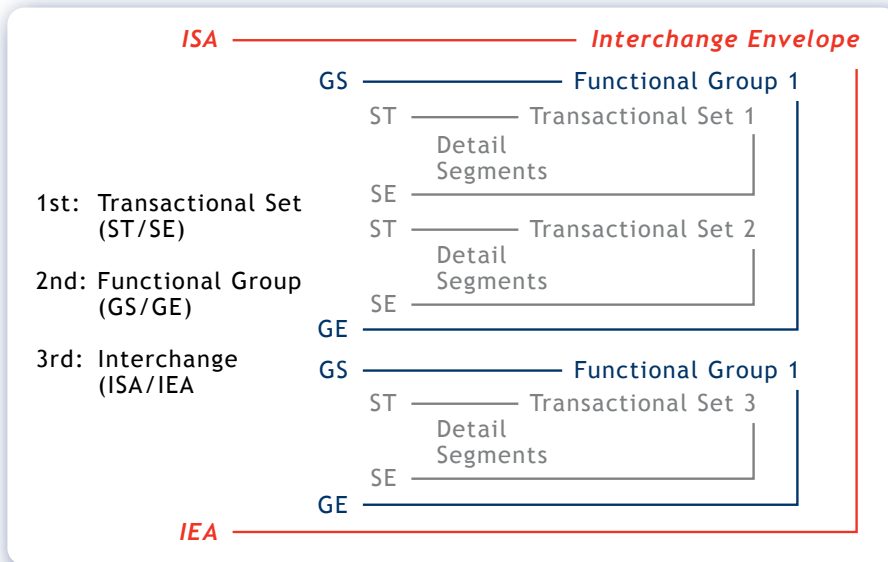


Envelope Structure

For every message there are three levels of enveloping:

- Transaction Set
- Functional Group
- Interchange

The following diagram identifies the three levels and how they relate to each other:



Envelope Levels:

1st: Transaction Set (ST/SE) 2nd: Functional Group (GS/GE) 3rd: Interchange (ISA/IEA)

Transaction Set Envelopes

The innermost level is the Transaction Set identified by the ST/SE segments. The ST segment always has two data elements. A third data element is optional for version/releases 004020 and later. They are:

- Transaction Set ID (e.g., 850)
- Control Number (e.g., 1001)
- Implementation Convention Reference (e.g., X099) optional and available for version/releases 004020 and later

The SE segment contains the **Number of Included Segments** in the transaction set and the same **Control Number** as the ST segment.

Transaction Set ID

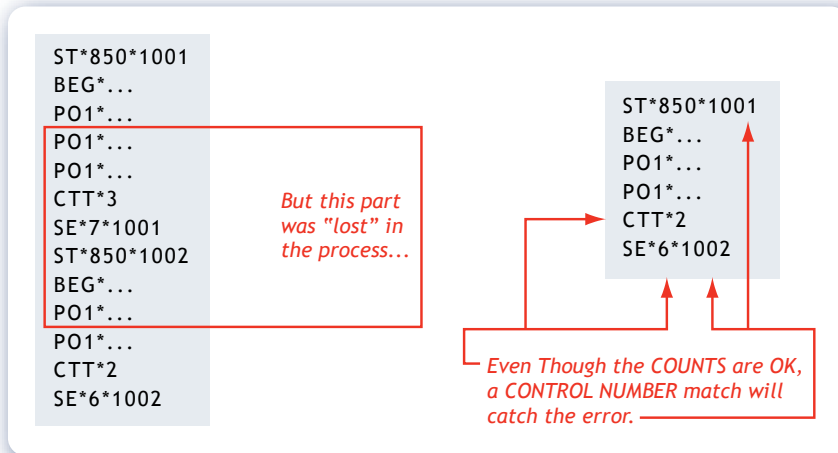
The Transaction Set ID identifies the transaction set being enveloped using the three-digit Transaction Set ID code. Examples of these codes are:

- 850 Purchase Order
- 810 Invoice
- 997 Functional Acknowledgement

Number of Included Segments and Control Number

The Number of Included Segments and Transaction Control Number provide data integrity for transaction control via segment counts and control numbers.

Let's suppose that this is part of a PO that was sent...

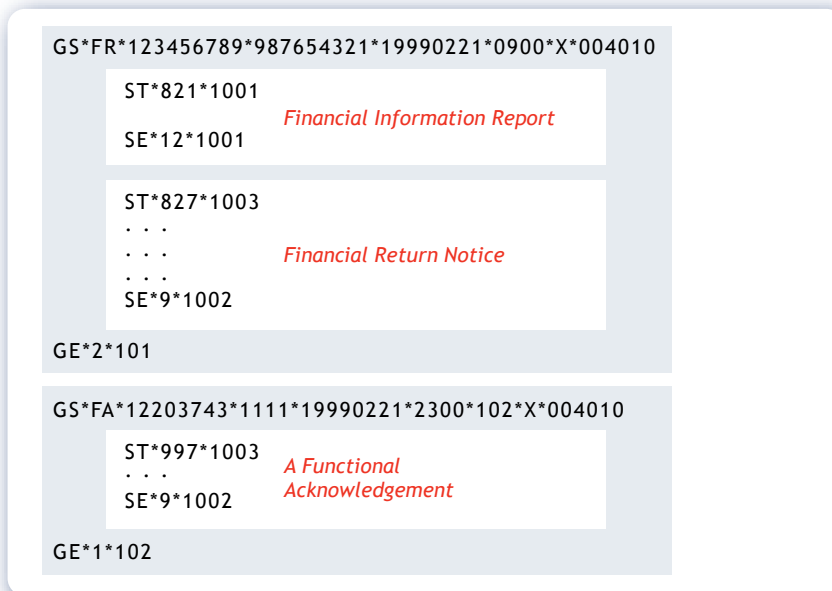


Implementation Convention Reference

The Implementation Convention Reference (e.g., X099) is optional and used for industry and legislative guidelines based on version/release 004020 and later.

Functional Group Envelopes

The second (middle) level of enveloping is the Functional Group Envelope. Its purpose is to group similar types of Transaction Sets within a transmission.



The definition of 'similar' varies by the version/release of ANSI ASC X12 being used. For example, the November 1987 (and prior) release for the PO group may contain: the Planning Schedule (830), the Purchase Order (850), the Purchase Order Acknowledgment (855), the Purchase Order Change (865), the Order Status Inquiry (869) or the Order Status Report (870).

For later versions/releases and for new Transaction Sets that are being developed, most Transaction Sets are being assigned to a unique Functional Group.

The Functional Group Envelope is defined by the GS/GE segments. The GS segment has a number of data elements. An example of some are:

- Functional Group Set ID
- Format and version of the document, and date/time stamp numbers

Functional Group Set ID

The Functional Group Set ID identifies the Function Group Set being enveloped using a two-character Functional Group Set ID code. Refer to the Standards manual for a complete list of Functional ID codes. Examples of these codes are:

- PO** Purchase Order
- SH** Shipment Notice/Manifest
- FA** Functional Acknowledgment

Controls

The Transaction Sets Count and Functional Group Control Number provides data integrity for transaction control via the transaction.

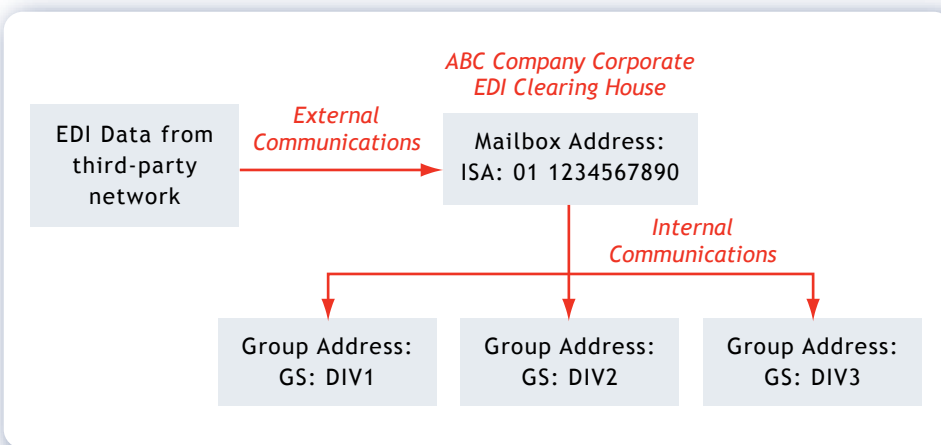
Format and Version

The format and version of the document are used to identify the ANSI ASC X12 release being used.

Example: X for the format and 004010 for the version/release.

Functional Group Sub-Addressing

Functional Group envelopes also contain a sub-addressing capability. The data that is sent to a particular receiver is addressed to the mailbox address on the ISA. Many companies route the Functional Group data internally, so the Functional Group segment has a provision for user-defined addresses in the GS02 and GS03 elements.



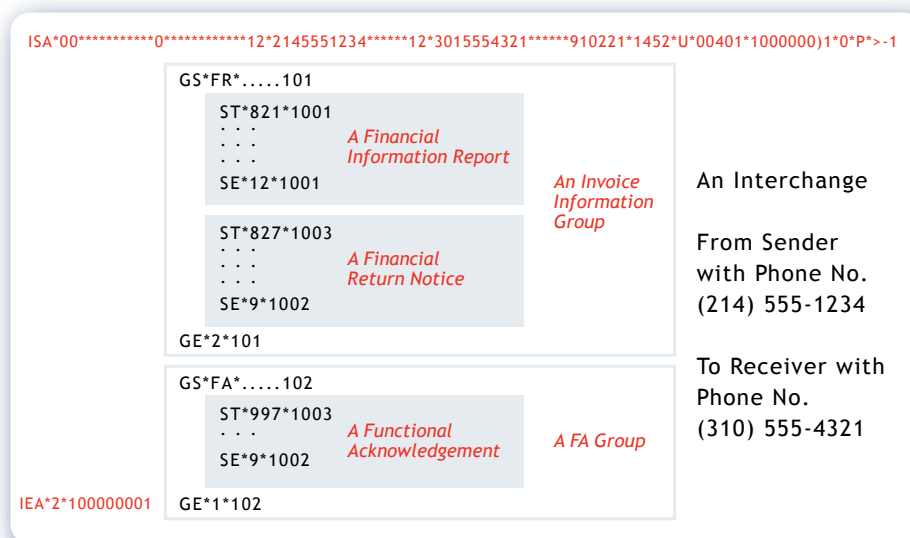
Interchange Envelopes

The outermost level is the Interchange envelope that is defined by ISA and IEA segments. The ISA/IEA data elements start with the letter I and are found at the end of the data element dictionary in the Standards manual. The Interchange envelope encloses the data from one sender to one receiver.

The ISA is a fixed length segment. Some items contained in the ISA/IEA segments are:

- Structured mailbox addresses of the sender and receiver
- Interchange control numbers
- Counts of the Functional Groups within the interchange
- Time/date stamp (similar to functional group, but does not include the century)
- Version of the interchange envelope
- Characters in the ISA segment used for data element separators, sub-element separators, and segment terminators

Here is an example of the Interchange, Group and Transaction Set Envelopes:



Summary of Envelopes

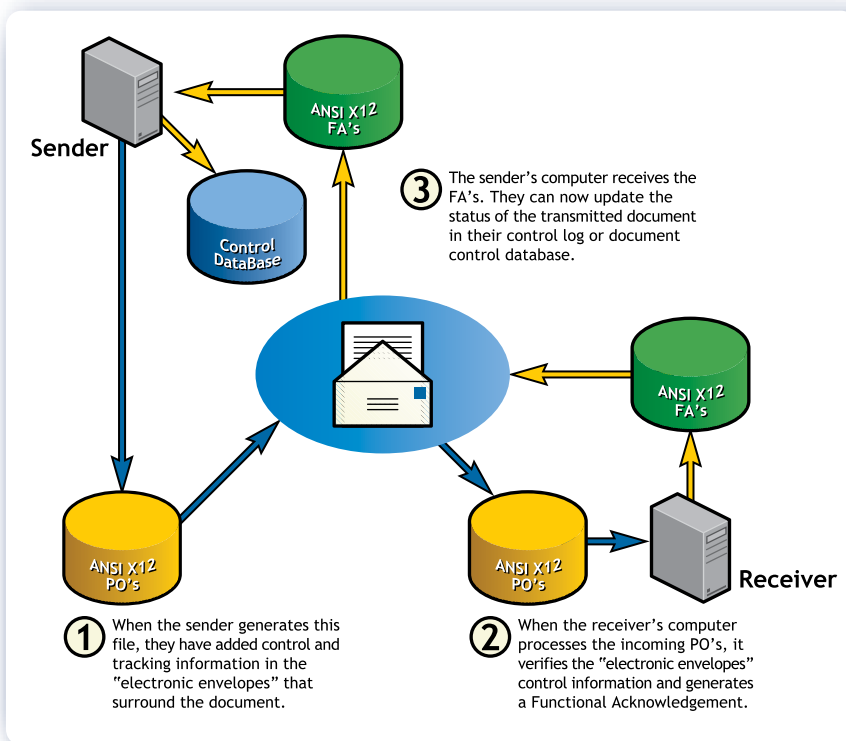
- Three levels of envelopes: Transaction Set, Functional Group and Interchange
- Control Numbers and Segment/Transaction/Group counts: ensure integrity of transmission
- Specific mailbox addressing on ISA
- Sub-addressing and routing on Group envelope
- Controls for loss or damage to data during transmission if any of the above is missing

Group Level FA

Functional Acknowledgments are used to provide “end-to-end” acknowledgment of the receipt of a document.

It is the responsibility of the receiver’s computer to check the syntax and control numbers of the transmission, then to build and transmit a Functional Acknowledgment back to the sender. One Functional Acknowledgment is created for each Functional Group received. The receiver can acknowledge at the Functional Group or transaction set level. If acknowledging at the Functional Group level, one AK1 and one AK9 segment are sent. If acknowledging at the transaction set level, within the AK1/AK9 segments, one set of AK2 and AK5 segments are sent for each transaction set within the Functional Group identified in the AK1 segment.

- Accepted
- Rejected
- Accepted with errors
- Partially accepted (at least one transaction set was rejected)



Error Reporting

It is possible to identify specific errors in transaction sets by using:

- the optional AK3 Segment for segment errors
- the optional AK4 Segment for data element errors

ST*997* . . .	
AK1* . . .	Identify the Group
AK2* . . .	Identify the Set
AK3* . . .	Identify SEGMENT and report error
AK4* . . .	Identify ELEMENT and report error
AK4* . . .	Identify ELEMENT and report error
AK3* . . .	Identify SEGMENT and report error
AK5* . . .	Report on the Set
AK9* . . .	Report on the Group
SE*10* . . .	

Envelope Levels:

1st: Transaction Set (ST/SE) 2nd: Functional Group (GS/GE) 3rd: Interchange (ISA/IEA)

ANSI Wrap-Up

Some things to keep in mind:

- ANSI ASC X12 committee developed the ANSI ASC X12 standard as the EDI standard for North America.
- Transaction Sets are defined for each document in the standard and are made up of segments. The standard states the sequence of the segments, which are mandatory, optional or floating (for older versions), how often they may repeat, and information about looping.
- Segments are a collection of data elements, whether simple or composite, in a defined order. The order is laid out in the data segment table.
- Data elements are sometimes paired as qualifier and value. A composite data element consists of two or more component data elements separated by sub-element separators.
- There are three levels of envelopes: transaction set, functional group, interchange. These levels help maintain transmission integrity through a system of control numbers and segment/transaction/group counts.
- Functional Acknowledgements provide end-to-end status information on a document.