# EDI Standards

Two organizations communicating electronically with EDI must follow the same rules for how they format the data. This is necessary so they know (or their computers know) where to find particular pieces of information within the message.

Fortunately, there are such rules that govern various aspects of formatting EDI data. These rules – called EDI Standards – ensure that everyone participating in EDI is, in effect, speaking the same language.

# Setting the Standards – Where EDI Standards Come From?

In the United States, organizations traditionally followed the ANSI standards for EDI document formats. ANSI – the American National Standards Institute – is a private, non-profit organization that oversees the development of voluntary consensus standards for products, services, processes, systems and more. A specific ANSI committee, ANSI ASC X12, developed the most common standard for EDI. The standard is often referred to simply as ǲASC Xͳʹ,ǳ or even just ǲXͳʹ.ǳ

Outside the U.S., the international EDI standard is EDIFACT (Electronic Data Interchange for Administration, Commerce and Transport), developed under the United Nations.

# The Elements of EDI Standards – What Standards Address

Whether following ANSI ASC X12 or EDIFACT, the standards define the structure of data within various documents. A single EDI transmission may include one or more documents, such as invoices, insurance claims, etc. In the ASC X12 standards, each document types are defined, referenced by a three-digit number, such as 837 for a healthcare claim, 810 for an invoice, and so on.

Traditionally, EDI specifically represented one formatting structure for data following the industry standards. Today, however, EDI can include many different formats of these documents, including XML, CSV, fixed length and more. The information – individual data elements, such as dates, item numbers, order quantities and the like – is organized within the message, according to the EDI standards, into groups and smaller units. The combination of data that forms a single message or document is called a ǲtransaction set,ǳ according to the Xͳʹ standard.

EDI standards are generally independent of communication methods. That is, EDI can be transmitted using any number of methodologies, also called protocols. A fairly common method used today is called AS2 (Applicability Statement 2). AS2 provides specific security measures for data transmitted over the Internet.

# Data Translation

In the course of sending and receiving EDI data, software is involved which runs through a set of actions to verify and accept the files. This process is called ǲdata translation.ǳ Data translation includes managing the [EDI standards](http://www.1edisource.com/learn-about-edi/what-is-edi/edi-standards) and trading partner requirements, supporting different communications protocols, implementing the conversion of inbound and outbound transactions, validating transactions when they arrive or before they are sent to your trading partners, sending and receiving Functional Acknowledgements, and more. Translation – Taking Action on Inbound and Outbound Data Sending and receiving EDI transactions involves a number of actions above and beyond the transmission itself. These are all different functions of EDI data translation.

# Translation of inbound EDI transmissions includes these steps:

The file must be broken down ȋor ǲparsedǳȌ to identify everything it contains, and what actions should be taken. The individual transactions and their senders are identified. The senders are validated as legitimate trading partners. The file structure and individual data fields of each transaction are validated as proper according to the EDI standards. Typically, a Functional Acknowledgement (FA) is sent to the trading partner. Only after translation is complete is the [data](http://www.1edisource.com/learn-about-edi/what-is-edi/data-mapping) [mapped and applied](http://www.1edisource.com/learn-about-edi/what-is-edi/data-mapping) to your internal business system. A similar process occurs for preparing an outbound file for transmission, ensuring that the data fields and file structure are valid. The translator will prepare the transaction(s) and await receipt of the appropriate acknowledgements. **Data Mapping:** Prior to sending information via EDI, the information likely resides on a server

or mainframe, in a database or in some other back-end software application. As long as it is possible to import and export files from the application, data can be extracted for an outbound EDI transaction and inbound data can be injected automatically or entered manually. Directing which data elements from an application go where in a file, and vice versa, is known as ǲdata mapping.ǳ **Mapping** – the Foundation of System Integration

In simplest terms, mapping data is the act of converting information from one location – the data source – to another format for another application – the destination. In EDI terms, data mapping properly converts a userǯs application data into an ED) file format, and vice versa. Mapping outbound data involves establishing what information from the business or accounting software goes where in an EDI file. The data must be properly arranged and formatted so that it conforms to both the [EDI standards](http://www.1edisource.com/learn-about-edi/what-is-edi/edi-standards) and the trading partnerǯs requirements. [EDI software like IBM Sterling](http://www.1edisource.com/edi-products/edi-software) Integrator allows you to map EDI data, and then save that map like a template to use over and over again. You can create multiple data maps, for particular EDI transactions and for each trading partner.

Once inbound data is received, it must also be converted to a format your business or accounting system can understand. EDI software can often map inbound EDI data directly into a variety of destination types: Applications Readable documents Specific layouts like fixed length or delimited records, XML • One or more database tables A combination of these.

**Business documents** – These are any of the documents that are typically exchanged between businesses. The most common documents exchanged via EDI are purchase orders, invoices and Advance Ship Notices. But there are many, many others such as bill of lading, customs documents, inventory documents, shipping status documents.

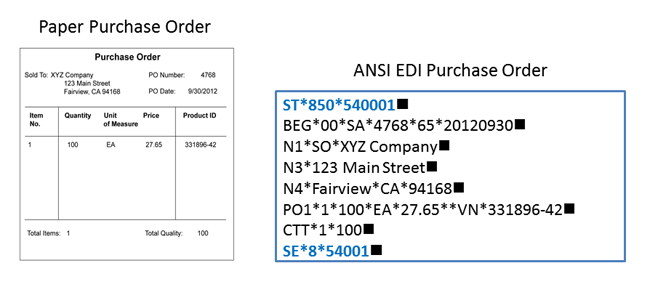
**Standard format**– Because EDI documents must be processed by computers rather than

humans, a standard format must be used so that the computer will be able to read and understand the documents. A standard format describes what each piece of information is and in what format (e.g. integer, decimal, mmddyy). Without a standard format, each company would send documents using its company-specific format and, much as an English-speaking person probably doesnǯt understand Japanese, the receiverǯs computer system doesnǯt understand the company-specific format of the senderǯs format.There are several ED) standards in use today, including ANS), EDIFACT, TRADACOMS and XML. And, for each standard there are many different versions, e.g.

ANSI 5010 or EDIFACT version D12, Release A. When two businesses decide to exchange EDI documents, they must agree on the specific EDI standard and version. Businesses typically use an EDI translator – either as in-house software or via an EDI service provider – to translate the EDI format so the data can be used by their internal applications and thus enable straight through processing of documents.

**Business partners** – The exchange of EDI documents is typically between two different

companies, referred to as business partners or trading partners. For example, Company A may buy goods from Company B. Company A sends orders to Company B. Company A and Company B are business partners.



There are 3 steps to sending EDI documents – Prepare the documents, Translate the documents into EDI format, Transmit the EDI documents to your partner.

**Step 1**: Prepare the documents to be sent

The first step is to collect and organize the data. For example, instead of printing a purchase order, your system creates an electronic file with the necessary information to build an EDI document. The sources of data and the methods available to generate the electronic documents can include:

Human data entry via screens

Exporting PC-based data from spreadsheets or databases Reformatted electronic reports into data files

Enhancing existing applications to automatically create output files that are ready for translation into an EDI standard

Purchasing application software that has built-in interfaces for EDI files

**Step 2**: Translate the documents into EDI format

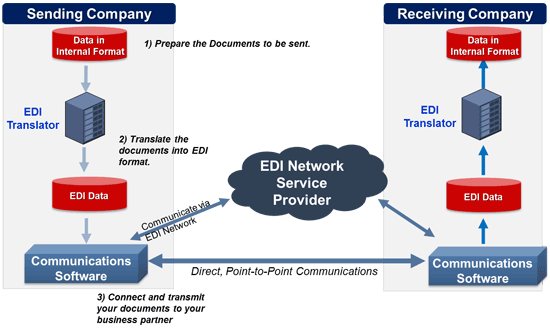
The next step is to feed your electronic data through translator software to convert your internal data format into the EDI standard format using the appropriate segments and data elements. You can purchase EDI translation software that you manage and maintain on your premises. This requires specialized mapping expertise in order to define how your internal data is to be mapped (i.e. correlated) to the EDI data. Translation software is available to suit just about any computing environment and budget, from large systems that handle thousands of transactions daily to PC- based software that need only process a few hundred transactions per week.

Alternatively, you can use the translation services of an EDI service provider. In that case, you send your data to the provider, who handles translation to and from the EDI format on your behalf.

**Step 3**: Connect and transmit your EDI documents to your business partner

Once your business documents are translated to the appropriate EDI format they are ready to be transmitted to your business partner. You must decide how you will connect to each of your partners to perform that transmission. There are several ways, the most common of which include

1) to connect directly using AS2 or another secure internet protocol, 2) connect to an EDI Network provider (also referred to as a VAN provider) using your preferred communications protocol and rely on the network provider to connect to your business partners using whatever communications protocol your partners prefer, or 3) a combination of both, depending on the particular partner and the volume of transactions you expect to exchange. To learn more about the various options.



# The EDI file structure

Transaction Sets (Messages) can be organized by grouping them by their topics or functions. A group of Transaction Sets is called a Functional Group. Functional Groups themselves constitute an interchange. An Interchange is an electronic equivalent of a letter envelope. It contains the destination and sender ID of the EDI message, and the date/time stamp of when it was sent.

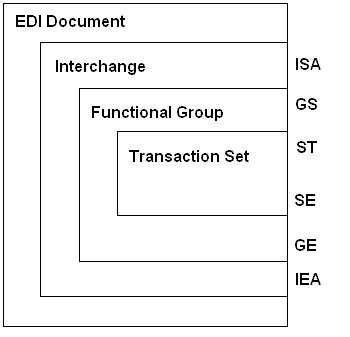
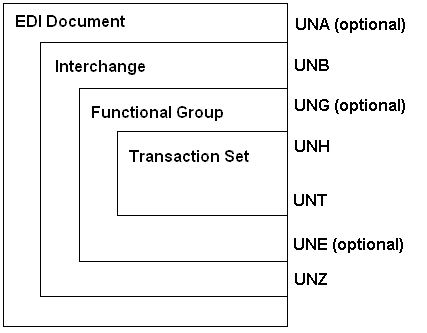
 

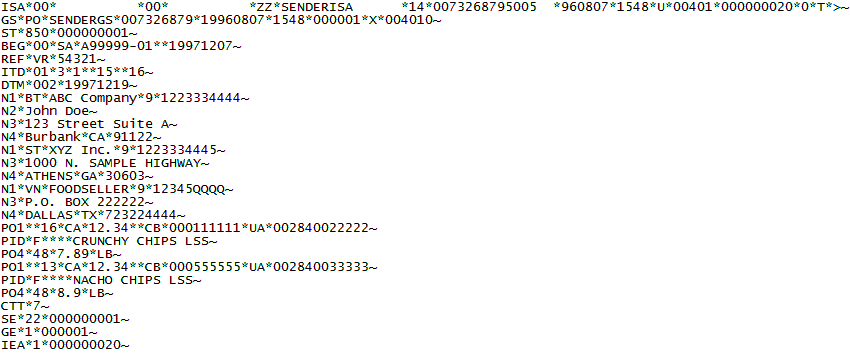
Figure 4a: The ASC X12 file structure Figure 4b: The UN/EDIFACT file structure

The Control Segments are the header and trailer segments that mark the start and end of their controlling structure.

The Header/Trailer control segments of the Interchange are the ISA/IEA in X12, and UNB/UNZ in UN/EDIFACT.

The Header/Trailer control segments of the Functional Group are the GS/GE in X12, and UNG/UNE in UN/EDIFACT.

The Header/Trailer control segments of the Transaction Set are the ST/SE in X12, and UNH/UNT in UN/EDIFACT.

Figure 5a: An example of an ASC X12 EDI file

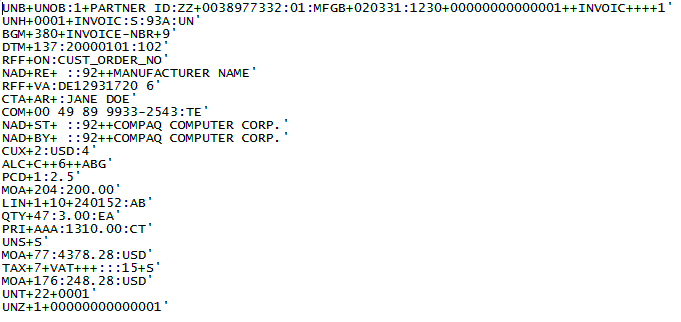
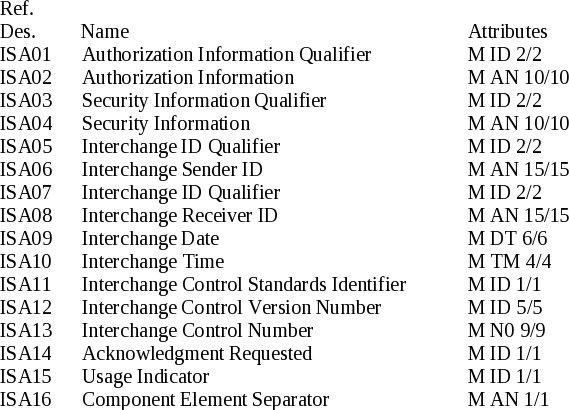
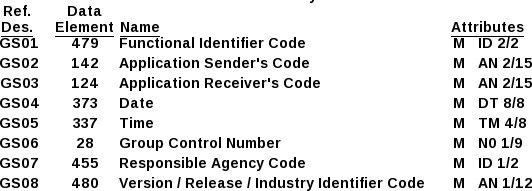


Figure 5b: An example of an UN EDIFACT file

# Envelope, Header, and Trailer Segments



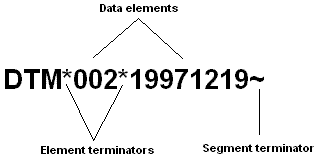


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Soft Share Tree Level |  |  |  |  |
| Segment/ Example Elements |
| Example Values / Comments |
| Name | Description |
|  |  |
|  |
|  |
| 1 | ISA | Interchange Information | Identifies the sender and  receiver using QID's. There is only one ISA | The ISA and IEA  segments are called the envelope |
| 1 | ISA05 | Sender Qualifier ID | Two digits that specify what kind of sender identification is used in segment ISA06 | 01 = Duns # 08 =UCC EDI #  12 =phone #  14 = Duns + suffix # |

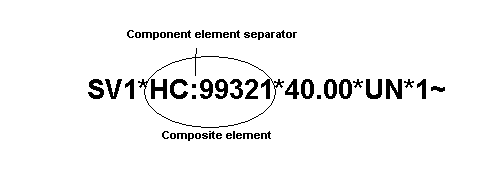
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | ZZ = mutually defined |
| 1 | ISA06 | Interchange Sender ID | A number that uniquely identifies the sender | The Duns #, UCC EID #, phone #, Duns + suffix #, or mutually defined # |
| 1 | ISA 07 | Receiver Qualifier ID | Two digits that specify what kind of receiver identification is used in segment ISA08 | 01 = Duns # 08 =UCC EDI #  12 =phone #  14 = Duns + suffix # ZZ = mutually  defined |
| 1 | ISA08 | Interchange Receiver ID | A number that uniquely identifies the receiver | The Duns #, UCC EID #, phone #, Duns + suffix #, or mutually defined # |
| 1 | ISA13 | Interchange Control Number | Sequential assigned number by the sender for this transmission | Will match the ISA15 number in the envelope report on EC Grid |
| 1 | ISA15 | Usage Indicator | Identifies this transmission's purpose for production or test | Will be P for  Production or T for Test |
| 2 | GS | Functional Group Header | Identifies the kind of business document in this transmission | There can be more than one GS, but this is very rare |
| 2 | GS01 | Functional Identifier Code | Two letter abbreviation for the type of transaction being sent | PO, Purchase Order SH, Ship Notice  IN, Invoice  FA, Functional Acknowledgement PC, Purchase Order Change  RA, Remit Advice  RS, Order Status |
| 2 | GS02 | Application Sender's Code | The identification number of the sender | Same as ISA06 |
| 2 | GS03 | Application | The identification number of the | Same as ISA08 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Receiver's Code | receiver |  |
| 2 | GS06 | Group Control Number | Sequentially assigned by sender for each functional group | Will match the control number in the AK1 segment of the 997 for a summary level 997 |
| NA | ST | Transaction Set Header | Marks beginning of a document, for example, one Purchase Order | A transaction set is one business document, and there can be multiple documents (common for  Purchase Orders) |
| NA | ST01 | Transaction Set ID Code | Three digit number for the type of transaction set (i.e. document being sent) | 850 for Purchase Order  810 for Invoice Redundant with GS01 but uses 3 digits instead of 2  letters |
| NA | ST02 | Transaction Set Control Number | Sequentially assigned number by  the sender for each transaction set |  |
| NA | SE | Transaction Set Trailer | Marks the end of a document, for example, one Purchase Order |  |
| NA | SE01 | Number of Included Segments | The number of segments included in the transaction set |  |
| NA | SE02 | Transaction Set Control Number | The control number of the transaction set just ended | Must match the control number in ST02 of this transmission |
| NA | GE | Functional Group Trailer | Marks the end of the functional group |  |
| NA | GE01 | Number of Transaction Sets | The number of transaction sets included in this functional group just ended |  |
| NA | GE02 | Group Control Number | The control number of the functional group just ended | Must match the control number in |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | GS06 of this transmission |
| NA | IEA | Interchange Control Trailer | Marks the end of one functional group(s) and the end of an EDI Transmission | The ISA and IEA segments are called the "envelope" |
| NA | IEA01 | Number of Included Functional Groups | Number of GS segments | This is very rarely more than one |
| NA | IEA02 | Interchange Control Number | The same number as ISA13 | Must match the ISA13 number in this transmission |

An EDI file is a binary computer file that contains data arranged in units called data elements. Data elements are separated by element terminators; and a group of data elements make up a data segment. Data segments are separated by segment terminators.

**Figure 1**: An example of a data segment (DTM) having two elements.

Some elements (known as composite elements) are made up of sub-elements (component elements).

**Figure 1a**: An example of a data segment (SV1) having four elements. The first element is a composite element with two sub-elements (or component elements) separated by a component element separator, which is ":" (a colon).

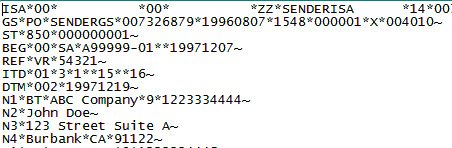


Figure 1b: A section of an EDI file with segments ISA, GS, ST, BEG, REF, ITD, DTM, N1, N2, N3, N4

A block of data segments with data that are inter-dependent to each other is called a group or a loop. An example of a loop is the N1 loop, which may hold a company's address information.

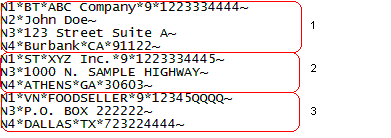
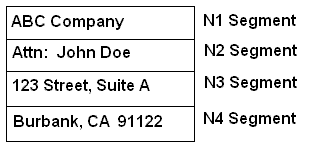


Figure 2: An N1 Loop Figure 2a: A section of an EDI file with three instances of the N1 loop. The first N1 loop consists of segments N1, N2, N3, and N4. The second and third instances of the loop consist of segments N1, N3 and N4.



In an EDI document, each section is described by a particular segment. Below is the set of EDI segments that would describe the purchase order above when using the ANSI standard. Each segment begins with a segment ID (e.g., ST, BEG, N1) that describes the type of data elements that follows. The elements within each segment are separated by a data element separator, in this case the Ǯ\*ǯ.

ST\*850\*1001 ST, to indicate start of a transaction set – in this case the 850 purchase ord BEG\*00\*SA\*4768\*65\*20120930 BEG, to indicate the beginning of the PO, specifically

N1\*SO\*XYZ Company N1, a name segment

N3\*123 Main Street N3, to provide street address

N4\*Fairview\*CA\*94168 N4, to provide city/state/zip PO1\*1\*100\*EA\*27.65\*\*VN\*331896-42 PO1, to provide line item detail CTT\*1\*100 CTT, to provide summary data for the PO

SE\*8\*1001