Electronic Data Interchange Overview (includes HIPAA considerations)

by Michael Foster, STC Network Systems

EDI is a file based format that predates the modern Internet and what we now think of as "E-commerce". EDI was developed to reliably move data from one system to another in an low-cost, standardized format. EDI was never implemented on a wide scale back then so the same issues followed e-commerce into today's Internet. Also, there are many non-standard variations to the original EDI, many created for specific industries or modified by certain corporations and organizations in their effort to "improve" the original.

EDI: File Format

EDI is a file format that provides a standard of data format with the goal that an EDI file generated by one system will be deliverable to a second system, and that system will be able to read the data it needs from the file.

Unfortunately, the more developers who have input, the more preferred file formats will be output. Those formats are workable, but none are the same and some are not even close. The trouble this causes is two-fold.

First, each new format brings with it the preconceptions and experience (or lack of) that the creator of the format has. This can cause problems as expected data may be missing or extraneous data included when the communication from this system with another is attempted. Also some transactional logic is implied in within the dataset.

Secondly, having a great diversity in the way data is presented means that the room for error is greatly increased. As form and content are dissimilar, converting something small like a date to a date time stamp introduces a bogus time stamp to make the format compliant. This small error in the data is generally acceptable, but as more are introduced, the potential for them impacting significant data increases.

This is where standardization enters the discussion. In the same way that proponents of cXML, ebXML, and other XML that purports itself as a standard, EDI is a standard for handling data. As with reading left to right, top to bottom is our standard for writing (along with all of the grammar and punctuation rules) help us integrate this document, so do the electronic data standards help us interpret what those electronic documents mean.

EDI: Data Structure

Where an EDI file contains data, the EDI standard contains Data Structures. Each EDI document type contains a structure for presenting a type of data in a logical fashion. This is part and parcel of the solution that EDI is providing. Structured data allows the receiver to scrape out the data that is important quickly. This is not unique to the EDI standard, but it is a basic part of what EDI is.

When you unwrap an EDI file you can see that the data structure is roughly modeled after the paper that EDI is replacing. This is a plus in that systems that started out based on paper orders and invoices are easy to map to EDI. We also see systems that are not based on paper, and sometimes they have unique requirements that are not so easy to map. In any case, EDI has a stable and predictable structure that was designed around the flexibility needs of commercial transactions.

EDI: System Solution

Many people have a their favorite format for data presentation. They may be adamant about the superiority of cXML, or CSV, or even EDI. In the long run, though, an integrator should avoid having favorites. It is the integrator's job to be the interpreter for these formats. From time to time, an integrator will need to recommend one or the other for a specific task of implementation.

EDI provides solution to e-commerce problems. Getting business information to your partner in the transaction; cheaply, quickly and reliably, are key to the solution. By following the best practices of EDI or whatever solution that you have chosen will help you create this solution.

Usage of EDI Specifications

When two trading partners agree to send each other electronic documents. And they begin to describe what EDI documents they will exchange and how the documents will flow, they should also exchange EDI specification documents. EDI usage or specification documents describe what fields and what segments a trading partner will send or expect to convey the information necessary to complete a transaction. It doesn't matter if we are ordering widgets, or invoicing, or transmitting catalog data, or checking insurance claims eligibility, the EDI needs to contain the data that the two parties need to communicate. To explain this, and document it to that both trading partners know what is expected, we create an EDI usage specification.

What are EDI Specifications?

In the larger sense, the EDI specification is the set of rules that define each document type, the segments they contain, and the size and type of data in the elements. When we talk about the EDI Standard Specification, we are talking about the whole set of valid EDI document types. If something is valid EDI, then it complies with the EDI Standard Specification. However, this large, all encompassing specification is not useful in coordinating the exchange of documents between two trading partners.

What goes in an EDI Usage Specification?

So we take EDI Standard Specification and we reduce it. We remove the unused document types and the unused segments within the documents we will use. And we even remove the unused elements and encoded values within the segments we will use. At this point, we have created an EDI Usage Specification. We can also add to the specification by including values that we need, like designating the an optional value or ID as required and specifying what type is should be.

How is an EDI specification Used?

Now that we have a usage specification we can use it to do to basic things. First, it is the source of information for us to set-up our integration for our EDI document exchange. And Second it is the format that we validate our EDI documents with to determine a valid transaction.

EDI Usage Specifications can be a source of data and integration documentation. This can become extremely valuable when the choices and information about how your EDI interface works is recorded into the notes of the Usage Specification.

Beyond a visual validation of looking at the EDI file and comparing it with the specification, many times a specification can be found in the form of an SEF file. The SEF file can be used in an EDI validation application. This allows a potentially large and unwieldy EDI file to be scanned for compliance and accuracy.

There may be other uses, but these are the main two, integration data repository, tool for EDI validation.

EDI Structure

Two important concepts in EDI are not exactly parts. They are the structures of Enveloping and Segments:

- *Enveloping*, as we discuss in another post, is a way for an EDI file to be self-determining. In the first segment of an EDI file, the ISA segment, all of the information can be found to allow a system designed to interpret EDI to know where it is going, where it came from, what version it is, and what characters it is using as delimiters.

- *Segments* following this first segment are not fixed position. White space is not used unless it is part of the data in the element. Each segment has a header element that indicates the definition used for the rest of the data elements in that segment. A segment may not need to use all of the possible elements in its definition. When this is the case, the segment terminates with the segment delimiter or terminator.

The parts:

- *Standard* is a concept that has more than one meaning, There is the standard that is the overall X12 compliance rules that regulate what is and may be an EDI document. But there is also the concept of a specific use case, or usage specification. Both of these are referred to as 'Standards' but in this description of the parts of EDI, the standard is the first concept. This is that when an EDI document is parsed, it must comply with this standard of segments, delimiters, and formats, or it is rejected as being corrupt.
- *Version* is a sub-component of the standard. It is declared in the ISA, and lets the parser know what version of the EDI standard will be contained inside the envelope. This is important as to format and structure of the data that has changed as technology and needs have matured.
- *Doctype* is not declared in the ISA, but is declared in the ST. We may also know when we see the functional declaration in the GS. The doctype determines which segments and in what order and structure may be present. Not all EDI documents will need all segment types, so the doctype is very important when parsing the EDI file. This is covered more in the ST Enveloping section.
- *Segments* are the most basic part that makes and EDI file and EDI file is the segment. There is a more detailed discussion of the nuances of segments here. Segments are defined by their type, the characters that are located before the first delimiter in the segment. And segments are terminated by a special delimiter called either the segment delimiter or segment terminator.
- *Element* exist inside the segment. Elements hold the data. Elements are separated by element delimiters. On rare occasions some elements can have sub-elements. There is a more detailed discussion of elements here.
- *Envelope Segments* There are three pairs of special segments that must be found, and can only be found, in the enveloping structure. Enveloping contains the origin and destination identification, as well as content definition identification values. A more detailed discussion of enveloping can be found here.
- *Files* are not really a part of EDI, but when EDI is formed, it is generally saved in a file. The file can contain any mix of EDI document types and going to any number of destination. Generally, for sanity's sake, this is not done. But there is nothing in the name of the EDI file that matters to the contents. There is no really expected extension like ".edi" for instance, and there is not need for white space removal from the beginning or end of an EDI file.
- *Documents* are another matter. An EDI document is the segments that reside within the ST envelope. As such, they are a set of data that has a single format, (EDI standard), a single sender and single receiver. Thus when an EDI document is discussed, we are really discussing the payload of the EDI envelopes.

STC Network Systems
Web Site Services & Consulting

STC Network Systems • PO Box 3334, La Habra, CA 90632-3334 • (562) 888-3270 • www.stcllp.com

Typical EDI Documents

PO - Purchase Orders (ANSI-EDI 850, UCS-EDI 875) detail the items, quantities, actual cost or estimated cost, Terms, Notes, Ship-to locations, Ship Dates, Cancel Dates, etc.

INV - Invoices (ANSI-EDI 810, UCS-EDI 880) are issued by the trading partner who has provided products and/or services as a request for payment. These EDI invoices include most of the purchase order information as well as Invoice Number, Invoice Date and Total Amount Due.

FA - Functional Acknowledgments (EDI 997) are short documents returned to the sender of an EDI transmission. They indicate that the transmission was received, but do not indicate agreement with the document.

POA - Purchase Order Acknowledgements (EDI 855) are used to provide seller's acknowledgment and acceptance or rejection of a buyer's purchase order. This can be an acknowledgement that the entire ordered quantity will be shipped and the date when it will be shipped. It can also be a line by line acknowledgment of partial or complete quantities and single or multiple shipping dates. The 855 Document can also be used as notification of a vendor-generated order or Reverse Purchase Order. The 855 Reverse PO is used in support of Vendor Managed Inventory (VMI). The vendor creates this Reverse PO to indicate to the buyer that the supplier has created a purchase order to maintain inventory levels as agreed to in the VMI agreement between the two parties. There is usually a minimum time period for the 855 to be sent prior to the sending of the 856 - Advance Ship Notice, which is sent when the goods are shipped.

ASN - Advance Shipping Notices (EDI 856) tell the customer all the details about the shipment: what items were sent, how many, when they were sent, etc. The ASN document has information about each carton and its contents including the carton number, weight, cube, etc. There is almost always a UCC128 label on each carton, which has information about the contents of the carton and its destination and shipping method. There are several barcodes so that important data items can be scanned. One of the most important codes on the UCC128 label is the carton number (128 code), which is how this carton information is referenced back to the ASN document.

POCR - The Purchase Order Change Request (EDI 860) document is sent by a customer to communicate changes to a previous Purchase Order. The 860 can be used to cancel the entire order, add and delete items, alter item quantities and prices, and change dates specifying order shipment, delivery, etc.

TXT - The Text Message (EDI 864) document is used to communicate plain free-form text messages. The 864 is often used to communicate information regarding problems with a received document. It is also used to communicate general business information about the trading relationship that is not document specific.

ADJ - The Credit/Debit Adjustment (EDI 812) document is used to communicate details of credits and debits for products. It may reference a specific Purchase Order and Invoice and include detailed information such as item identification and quantity. Standard codes are used to indicate the reason for the credit or debit request. Some common reasons are defective product, non-receipt of goods, return of goods, order quantity shortage or overage, and pricing error.

O.R. - The Organizational Relationships (EDI 816) document is used to communicate location address and relationship information. On the most frequently exchanged EDI documents (the PO, ASN, and invoice) locations are most often identified only by codes in order to avoid the cost of transmitting full addresses. Location codes are unique to a specific trading partner. The 816 document tells trading partners the address to associate with a particular location code. The trading partner receiving the 816 can use it to maintain a list of the sender's location codes and associated addresses. An alternate format of the 816 is used to show organizational relationships, such as stores serviced by a specific Distribution Center or warehouse. Trading of 816's begins with a full set of the trading partner's addresses, after which changes may be sent on a weekly or monthly basis, or as needed.

PORA - The Payment Order/Remittance Advice (EDI 820) document is used in connection with Electronic Funds Transfer (EFT) in one of two ways. In the first, a Remittance Advice is sent to the trading partner and a Payment Order is sent to the sender's bank to initiate transmission of the payment to the trading partner's bank. The second option sees a combined Payment Order/Remittance Advice transmitted to the sender's bank, which in turn sends the payment and remittance data to the trading partner's bank, which then informs the trading partner that the payment has been received. The 820 includes such data as payer and payee identification, including bank and account IDs, seller's invoice ID, adjustment amounts, reason codes, and billed and paid amounts.

AA - The Application Advice (EDI 824) document is used to inform trading partners of errors in EDI documents they have sent, most often invoices. The 824 differs from the 997 Functional Acknowledgement in that it is created as the result of error checking by the trading partner's business application program, whereas the 997 generally indicates EDI standard problems rather than business rule errors. In contrast to the 864 Message Text document, the 824 provides a fixed format for the identification of specific data items in error and for their suggested correction, in addition to free-form text message.

PAD - The Product Activity Data (EDI 852) document is used to report sales, inventory, and ordering information. Data is reported by item and may be broken out by store location. The 852 is usually sent on a weekly basis. The data is useful to the selling trading partner for product planning purposes. Normally the purchaser's buyer must approve the sending of 852 data to the seller.

Rules and Imperfections

While EDI X12 format rules are strict real world implementations sometimes try to bend them. EDI X12 was designed to make various systems talk to each other. While most software vendors try to follow rules to the letter some let them slip away. We call those cases ‘imperfections’ as most of them are not impossible to fix or workaround. This short chapter will try to list some of the typical implementation imperfections. Some of the cases listed here are not clear breakings of the rules but you may find them annoying.

Example 1:

EDI X12 data that is coming from mainframe or some other old systems is broken in lines of 80 characters. This is easily fixable by removing carriage return and line feed before data is being fed into translator. Removal of carriage return and line feeds will make EDI data as continues stream.

Example 2:

EDI files have no predefined size limits. Typical files range from 5Kbt to 5Mbt. Sending 1Gbt is not a rule breaker but having multiple smaller files instead can speed up the process. Many software packages allocate memory based on incoming data size, huge files can simply make processing system run out of memory or make it run much slower and cut memory for other services and programs on the same machine.

Example 3:

EDI files should have enveloping segments ISA, GS, ST, SE, GE, IEA. You may encounter EDI documentation from your trading partner that does not list or mention them at all. It is because trading partner mutually assumes that you know about those segments and will not even mention them in they documentation.

While EDI X12 formats are based on the standards in real world there are many deviations from the standard. Some deviations are so prevalent they become de facto sub standards in they own right. One of the best examples is EDI X12 837 Healthcare Claim.

Officially there is only one EDI X12 837. But due to very different requirements for institutional, dental and professional medical claims there are subsets for 837I, 837D and 837P. There is also probably a few hundred if not thousand variations of these.

Many trading partners you might need to deal with will provide you with they own document called "Companion Guide". These additional documents list specific requirements to their business needs.

In order to combat these deviations many independent software vendors release they products with Map Editors, Mappers, Mapping tools, etc. Tools allow flexible customization and mapping of translation and validation rules to your requirements.

Other messages that have number of deviations are EDI X12 835, 834, 810, 850, 855. As a rule of thumb more complex the message structure is – more possible deviations of it are in real world implementations.

What does it mean for your implementation? Why those deviations matter? Let's take 837 as an example. It is important to understand that every EDI X12 837 is unique within context of trading partner. Yes, they are all based on standard 837. Yes, they are all targeted for specific sectors: 837P, 837I, 837D, and more etc. So you logically assume they are all identical. If you take two EDI X12 837s from two different trading partners chances are you will find at least few different segments in each. That means in most cases you need your implementation to include specifics.

Only EDI files used internally by the organization may not contain ISA, GS, ST, SE, GE, IEA segments because VAN or some other third party gateway performs all the file preprocessing and routing and might be stripping them off. This is not an error but many shrink-wrapped EDI software packages will not be able to deal with files missing the envelope. They will consider EDI data as invalid.

Once you open EDI file in some text editor like Notepad, it looks cryptic. However all valid EDI X12 files have same properties:

1. They all start with three letters: "ISA"

2. After 106 characters there is word "GS". Sometimes "GS" is on the second line. If "GS" is on the second line then there might be 108 characters between "ISA" and "GS" (there are two non-printable carriage return and line feed characters between lines). There are also few exceptions to this rule. Example: if your segment separator is carriage return then you have 105 characters in ISA and one non-printable carriage return character at the end of line with GS following it.

3. ISA segment is 106 characters long and sometimes longer. Even 108 characters if you have ~ tilde segment separator and carriage return and line feed characters at the end of segment (after the tilde).

Most translators read and break down EDI files using these two points above. Most common scenarios why translator might not be able to process the file:

A. That it is not an EDI X12 file. Period. It is flat file, XML or scanned print image form or something else but EDI file.

B. If you copy and paste EDI file from Internet Browser window in most cases resulting EDI file will not be valid anymore. This is due to how spaces are displayed in the Internet Browsers. Instead of 103 characters between ISA and GS you might end up having much less.

C. Usually EDI files are sent over secure connections using encryption/decryption communications. Sometimes not just communications but EDI files are encrypted themselves. Encryption leaves ISA/GS part valid but all segments in the file unreadable. Unless EDI file is decrypted translator will not be able to read it correctly.

D. Some EDI files come from mainframe computer systems. Occasionally they are formatted into columns of 80 characters long. Extra carriage return and line feed characters break segments into multiple lines making EDI file invalid. Those files cannot be processed by some translators but other applications are able to process them.

E. Rarely EDI software communication packages combine multiple EDI messages of various types into one file. There are two variations of this combination:

Example 1:

810 invoices and 997 acknowledgements with the same separators in one file. This is valid EDI file however output of translator might not look right. Example: in translations from EDI to flat files translator will try to flatten EDI loops and encounter few loops repeating independent of each other (as per example above: 810 invoices will repeat separately from 997 loops). It is like trying to combine two spreadsheets that have no common fields; at the end combined spreadsheet will have mix of lines that do not tie up.

Example 2:

810 invoices and 997 acknowledgements with the different separators in one file. This is not possible to process by many translators. Usually translators detect separators at the start of the EDI file. If separators change somewhere in the middle of the file, translator will continue processing the file using old separators found at the file start. Scenario when various EDI messages are mixed in one file with different separators is very rare but we mention it here to complete possible list of files that cannot be processed.

Envelopment Elements

The ISA Segment is the first segment in any EDI document. Once you understand what it is saying, This long random seeming string will make sense, and be very helpful in helping you relate EDI documents to the real world task that you need to do.

Here is an ISA segment.

```
ISA^00^          ^00^          ^01^9012345720000  ^01^9088877320000  ^080902^1523^U^00401
^000000001^0^T^|~
```

This is one long string, there is no linefeed or line break.

Like all segments, the first element is 0 and identifies the segment type. This is an ISA segment.

There are 16 ISA elements, here are the definition for the ISA:

ISA_01 is a qualifier. The Authorization Information Qualifier can be one of 7 values from 00 to 06. Most of the time, it is 00 which means that ISA_02 will not contain meaningful data. This field must be populated with one of these values.

ISA_02 is for Authorization Information. If the ISA_01 is 00, this will normally be left blank. If ISA_01 is one of the other 6 valid values, this field must be populated. The value here will be used to determine the authorization of the document. I have only seen this used with one trading partner. You might not see this at all.

ISA_03 is also a Qualifier. It is the Security Information Qualifier and can be one of two values, 00 or 01. Most of the time, it is 00 which means that ISA_04 contains no password. This field must be populated with one of these values.

ISA_04 is for Security Information. This is also called the password. If the ISA_03 is 01, this must contain a value. Most of the time, this will be blank.

ISA_05 is an Interchange ID Qualifier. There are two of these. They qualify or define the Interchange ID field that they precede. These are required to be present. There are 41 valid options, and I won't define them all here. Some common usages are:

01 = DUNS number

12 = Phone number

14 = Duns Number with Suffix

16 = DUNS with 4 character suffix

ZZ = mutually defined.

ISA_06 is the Sender's Interchange Identifier. This value will identify the sender to the receiver. The receiver should have this as a unique value. Thus some times it is necessary to have more than one option in case the value you have chosen is already in used by some other Trading Partner that the receiver exchanges documents with.

ISA_07 is an Interchange ID Qualifier. It follows the same rule as ISA_05 but qualifies the receiver instead of the sender.

ISA_08 is the Receiver's Interchange Identifier. This values tells the sender's system where the message is going, and tells the receiver's system that this message is for them. The sender will need to have a unique value for each trading partner.

ISA_09 is the Interchange date. This is the date that the ISA envelope is created. Sometimes this value is derived from the contents of the document, but ideally this should be the date of the interchange envelope.

ISA_10 is the interchange time. This is the time that the ISA envelope is created. Sometimes this values is derived from the document, or is defaulted to midnight. Ideally this should be the time of the interchange envelope.

ISA_11 is the Standard Identifier. This will always be 'U' identifying this as the US EDI standard.

ISA_12 is the Standard Version Number. This relates to the major version of the EDI standard that is used. This value needs to be accurate so that the receiver's EDI system can expect the right size of fields in the GS segments. This may not look like what it is, as you see this '00401′ for a version of '4010′. This only identifies the major version, not distinguishing between 4010, 4011 and 4012.

ISA_13 is the Interchange control number. This is the way that an EDI system identifies the envelope. This control number should not be random, but an incrementing number. It is 9 digits padded with zeroes. The combination of ISA Sender and Receiver Identifiers and the Control number should identify a distinct interchange transmission. When 9 '9′s are reached, the control number should roll back to 1. Nine zeros should be avoided as some systems that may internally trim the zeros will them be presented with a null value.

ISA_14 is the Acknowledgment Request. This lets the EDI file make a request for a functional acknowledgment or 997 to be transmitted. This is a binary option of 0 meaning "don't send a 997." and 1 meaning "send a 997." In a perfect world any ISA with a 1 in this would get a 997 returned. In the real world, most EDI systems require some setup to enable the 997, thus it is good to discuss this before sending the request.

ISA_15 is the usage indicator. This can be one of three values. P is for Production Data, T is for Test Data, and I is for Information Definition. This is important as a EDI system should only process interchanges with the P as production data.

ISA_16 are the delimiters. You may remember from the delimiter discussion that these are in a fixed position.

There are only a few things that you need to worry about as you start working with EDI and the ISA. If you just need to do a quick edit for a test or to resubmit a document, here is what is important.

- Pay attention to the ISA_05, 06, 07 and 08. These are the sender and receiver fields.
- Increment you control numbers to make sure that you EDI files don't get booted as dupes.
- Make sure the ISA and IEA have matching control numbers.

The GS segment is the second mandatory enveloping segment. It shares some properties with the ISA segment. There is a sender and receiver, version Identifier, time stamp and control number. But one thing that the GS has that the ISA does not is a Functional Identifier.

That's right, Functional. This means that within the GS envelope are only EDI messages or documents that have the same function.

Here is a GS segment:

```
GS*PO*901234572000*908887732000*081031*0835*1*T*004010!
```

Like all segments, the first element is 0 and identifies the segment type. This segment has a GS in this element, making it a GS segment.

Unlike the ISA segment, that does not change in any way from standard to standard, the GS may change a bit.

Here are the definitions of the GS segment for the 3020 X12 standard:

GS_01 is the Functional Identifier. This designates the type of messages this envelope contains. If it contains Purchase Orders, the GS_01 is "PO" if it contains Invoices, it is "IN" as so forth.

GS_02 is the Sender Identifier. This does not have to be the same as the ISA Sender, but it can be. This element has the same restrictions on content, a 15 character alpha-numeric. But unlike the ISA sender, this one is not fixed in size. This means that the delimiters fit around the value without including any white space. In many basic implementations, the value in the ISA sender, and the GS sender are the same value.

GS_03 is the Receiver Identifier. This does not have to be the same as the ISA receiver, but it can be. This element has the same restrictions on content, a 15 character alpha-numeric. But unlike the ISA sender, this one is not fixed in size. This means that the delimiters fit around the values without including any white space. In many basic implementations, the values in the ISA receiver and the GS receiver are the same value.

GS_04 is the Date. It should be the date of the creation of the GS envelope. Many times it is the same date as the ISA date.

GS_05 is the Time. It should be the time of the creation of the GS envelope. Many times it is the same as the ISA time.

GS_06 is the Control Number. This is a 1 to 9 numeric value. This is a number that needs to be unique inside the ISA envelope only. Some implementations use an incrementing number that increments with the usage like the ISA control number. Others just have a count of the number of GS envelopes inside an ISA. Both are acceptable. All that is required is that the GS control number be unique among other GS control numbers within the ISA envelope that it is contained in.

GS_07 is the Agency Code. Basically for any X12 document that will be "T" or "X". I have only seen "X" indicating that it is really using X12. Even when it is not. Most of the time I ignore this value.

GS_08 is a Version identification value. Just like the ISA_12, this identifies the standard, but identifies it to the next level. Thus if you are 4010, this will be 004010 and so on.

Now look at the size of the Date and Time elements in GS_04 and GS_05. In the 3020 version, the Date is 6 char, and in 4010 it is 8 char. The time also grows from a 6 to a possible 8 char size.

Version 4010 was considered the Y2K standard where the dates moved from a 6 char date, to and 8 char date to accommodate the century change.

Other than this change, the GS segments and usage are the same.

There are really a few things to pay attention to as you begin to work with GS envelopes. And if you just need to do a quick edit and resubmit you may not need to do much.

STC Network Systems • PO Box 3334, La Habra, CA 90632-3334 • (562) 888-3270 • www.stcllp.com

- Pay attention to the GS_02 and GS_03, these are the sender and receiver and need to be correct to route documents and match with the profile for most systems.
- Make sure the GS_01 is the type of document that you are really sending. Some systems get cranky when they get an Invoice in a Purchase Order envelope.
- And as always, make sure your GS and GE control numbers match and are unique inside the ISA-IEA envelope.

The 3rd layer of the EDI envelope is the ST segment. The ST segment contains a Document Identifier, and a control number. The ST and its partner the SE segment define the beginning and ending of the Document. The SE segment contains a counter of segments within the document, and the corresponding control number to the ST. Relatively simple compared to the ISA and the GS.

Document Identifier

The document Identifier is not encoded. That is to say, for an 850 the ST document ID is "850". For an 810, the Document Identifier is "810" and so on. Thus, if you open an EDI file and don't recognize the GS_01, skip to the ST_01 and it will tell you what standard you will need to find to read this document.

Good form would dictate that the GS_01 and the ST_01 would specify the same type of document. The GS would specify it on a functional level, and the ST on the standard level. Thus they would not be identical strings but would indeed convey the same message about what the document(s) would be. Some EDI translator or processing software will validate this relationship and some may not. But in the end consistency is the best policy.

Control Number

The control number is different for the ST than we saw in both the ISA and GS. The ST_02 is a 3 to 9 digit number. This means that it will start with 001 not 000000001. Of course you can still use a 000000001 but don't be surprised when you see less than 9 digits in the ST-SE control number. Whichever you chose, you must do the same for both the ST and SE.

(Using 000 or 000000000 may be allowed, but is not recommended as any application that trims the zeroes will get an empty string and produce an error. Be nice and only use values greater than zero.)

Segment Counter

The SE segment counter is the part of the ST-SE that gives people hand editing an EDI document the most trouble. If the counter is inaccurate it indicates that the document is corrupted. This should fail in all good translators as processing a corrupt EDI file is saying, "Its okay if we get bad data." This counter represents the number of segment between the ST and before the SE.

And it is just that simple. Remember that you can have more than one GS-GE envelope inside an ISA-IEA, and you can have more than one ST-SE inside a GS-GE, but you can only have one document of one type inside a ST-SE envelope.

Terms and Abbreviations

TERM	DEFINITIONS
4010	Standard version 4, release 1 (major, first Y2K compliant standard, widest release, HIPAA, etc) [see also **X12**]
AA	Application Advice
ADJ	Credit/Debit Adjustment
ANSI	American National Standards Institute
ASC	Accredited Standards Committee [see also **X12**]
ASN	Advanced Shipping Notice
BHT	Basic Header Transaction segment
CG	Companion Guide (specific partner requirements as local, vendor, or industry supplements and rules)
CR	Credit
EAN	European Article Number
EDI	Electronic Data Interchange
EFT	Electronic Funds Transfer (to or from banking institution)
FA	Functional Acknowledgement (automated receipt)
GE	Group (functional) End, middle segment, envelope [paired with **GS**]
GS	Group (functional) Start, middle segment, envelope [paired with **GE**]
GS1	Global Standards Group (company registrar, product code provider) [see also **UPC**, **EAN**]
ICS	Interchange Control Structure
IEA	Interchange End Authorization, outer segment, envelope [paired with **ISA**]
INV	Invoice
ISA	Interchange Start Authorization, outer segment, envelope [paired with **IEA**]
O.R.	Organizational Relationships (name, address, contacts, etc.)
PAD	Product Activity Data
PO	Purchase Order
POA	Purchase Order Acknowledgement [see also **PO**]
POCR	Purchase Order Change Request
PORA	Payment Order/Remittance Advice
SE	Segment (data) End, inner segment, envelope [paired with **ST**]
ST	Transaction (data) Start, inner segment, envelope [paired with **SE**] (contains type no. or message numeric name)
TXT	Text Message
UCS	Uniform Communication Standard (grocery and retail industry, subset of ANSI) [see also **ANSI**]
UPC	Universal Product Code [see also **EAN]**
V.A.N.	Value Added Network (common, shared electronic mailbox or exchange/transfer service)
VICS	Voluntary Inter-industry Commerce Standards (subset of ANSI) [see also **ANSI**]
X12	Exchange Format #12 (standard) [see also **ASC**]

STC Network Systems • PO Box 3334, La Habra, CA 90632-3334 • (562) 888-3270 • www.stcllp.com

HIPAA

The Health Insurance Portability and Accountability Act (HIPAA) was enacted by the U.S. Congress in 1996.
Key EDI transactions are:

* 837: Medical claims with subtypes for Professional, Institutional, and Dental varieties.

* 820: Payroll Deducted and Other Group Premium Payment for Insurance Products

* 834: Benefits enrollment and maintenance

* 835: Electronic remittances

* 270/271: Eligibility inquiry and response

* 276/277: Claim status inquiry and response

* 278: Health Services Review request and reply

These standards are X12 compliant, and are grouped under the label X12N. Essentially they are EDI X12 4010 or 5010 release transactions. All the basic standard EDI X12 rules listed in this document apply to the HIPAA as well because HIPAA are EDI X12 transactions.

Converting from Wrapped to Unwrapped and back

When you need to do this, you need to do this fast, not manually. Sure the manual way works with small files, but if you are trying to edit a big invoice or order, you will want to do something snappier than walking through the file segment at a time. Here are two ways to do it with common tools.

To unwrap:

1. open the file. Type "vi [file name]"
2. Hit ":" to get the command prompt
3. Type "%" to do a every line replace. (may not be needed as this is wrapped, but it doesn't hurt.)
4. Type "s" for a substitution.
5. Type "/" to begin the string to find. (you may need to escape characters like ~ and | with a . I will show this in the example)
6. Type the delimiter to be replaced.
7. Type "/" to begin the replaced string (again with the possible need to escape characters.)
8. Type the delimiter again, as we don't want to lose it, then follow it will a [crtl]+v and [crtl]+m
9. Type "/" to end the substitution
10. Type "g" to make it global or indicating all of the instances not just the first one.

It looks like this: "%s/[delimiter]/[delimiter]^M/g"

Put another way, here is what I typed for our example: "%s/~/~^M/g" (notice that I am escaping the ~ with a)

To re-wrap:

1. open the file. Type "vi [file name]"
2. Hit ":" to get the command prompt
3. Type "%" to do a every line replace. (may not be needed as this is wrapped, but it doesn't hurt.)
4. Type "s" for a substitution.
5. Type "/" to begin the string to find. (you may need to escape characters like ~ and | with a . I will show this in the example)
6. Type a backslach () followed by "n" to designate the newlines.
7. Type "/" to begin the replaced string (again with the possible need to escape characters.)
8. Type "//" to begin and end the substitution.
9. Type "g" so that it will get any multiples of the newline.

It looks like this: "%s/n//g"