

EDIFACT Standards Overview Tutorial

Learn About Key E-commerce Trends and Technologies at Your Own Pace



Welcome!3

How To Use This Tutorial3

Tutorial Objectives.....3

Part 1: Why use EDI Standards?.....4

Part 2: What is EDIFACT?4

Part 3: Message Definition.....6

Part 4: Message Structure: Segment Tables6

Part 5: Message Structure: Segment Groups.....8

Part 6: Segments.....9

Part 7: Segment Terminators and Delimiters10

Part 8: What is Mapping?10

Part 9: Data Elements: Simple and Composite11

Part 10: Composite Data Elements: Qualifier and Value.....13

Part 11: Message Structure and Electronic Enveloping14

Part 12: Envelope Architecture.....14

Part 13: Electronic Enveloping.....15

Part 14: The Message Envelope15

Part 15: The Functional Group Envelope.....17

Part 16: The Interchange Envelope.....17

Part 17: The CONTRL Message18

EDIFACT Wrap-up19

EDIFACT Glossary19

EDIFACT Self-Test I: After Completion of Part 521

EDIFACT Self-Test II: After Completion of Part 1022

EDIFACT Self-Test III: After Completion of Part 1723

Welcome!

This tutorial is an overview of the EDIFACT Standard format.

The topics covered in this tutorial are:

- Objectives of this tutorial and how to use it
- EDIFACT 101—some basic information about EDIFACT Standards development
- Messages—structure, segments, types, 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 Acknowledgements—who needs them?

Terms marked in **red** are defined in the EDIFACT Glossary.

How To Use This Tutorial

This tutorial provides information about the development of EDIFACT and basic EDIFACT concepts. It should be used as a prerequisite to understanding what EDIFACT is, and how EDIFACT is developing 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 EDIFACT.

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

Please send all feedback to training@gxs.com.

Tutorial Objectives

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

- What are EDIFACT and EDIFACT Standards
- How and why the EDIFACT standard was developed
- What is the basic structure of the EDIFACT format
- How the EDIFACT documents are read
- What are Functional Acknowledgements and why are they used
- How the EDIFACT standard may be used in providing electronic commerce solutions

Part 1: Why use EDI Standards?

EDI standards facilitate electronic data interchange (EDI) by providing:

- Rules of syntax
- Definition of the data organization
- Editing rules and conventions
- Published public documentation

EDI standards:

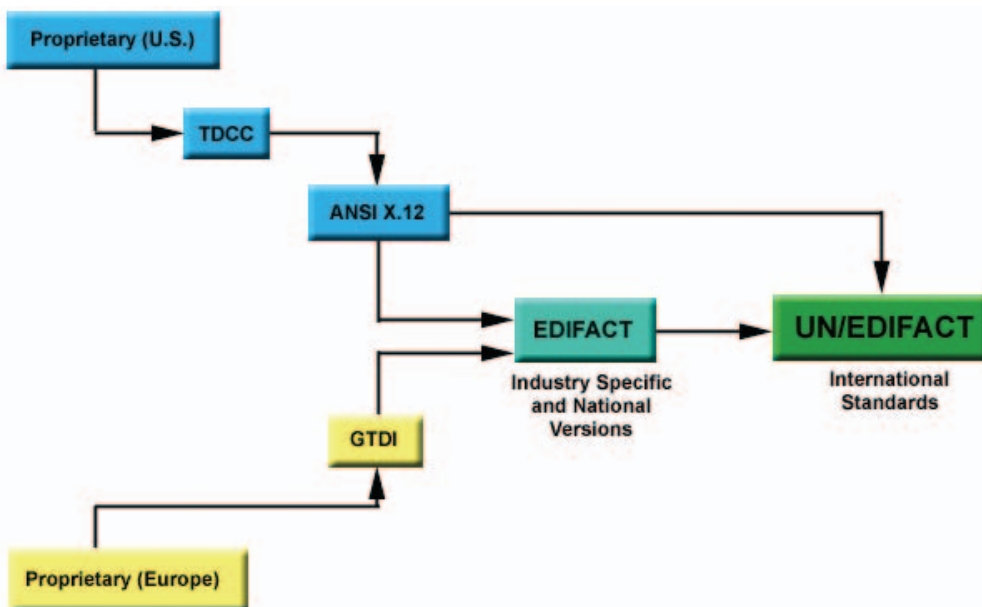
- Allow an 'open' system
- Reduce implementation effort
- Provide 'third-party interfaces'

Part 2: What is EDIFACT?

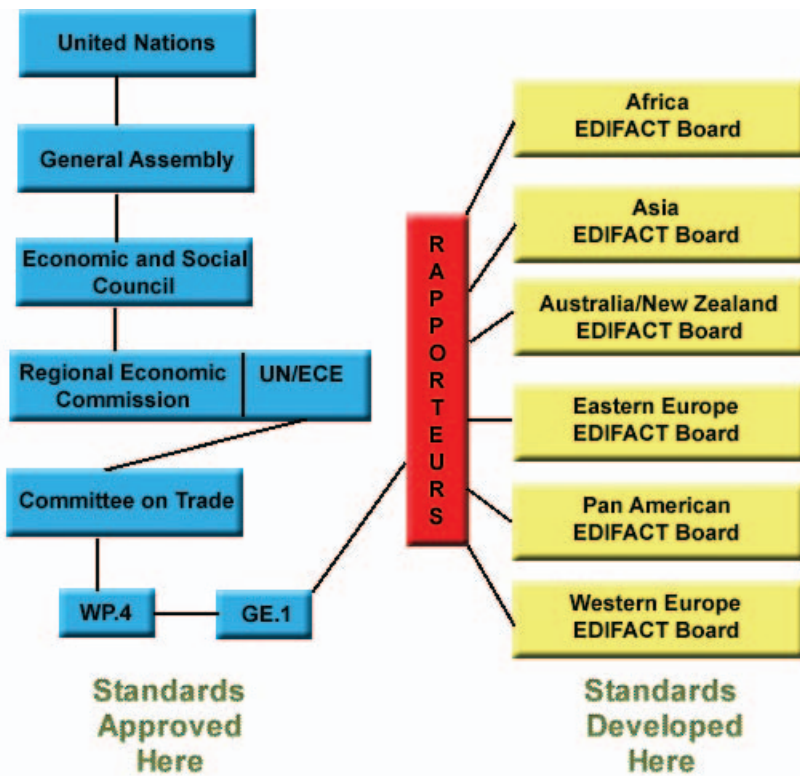
EDIFACT is an acronym for **EDI For Administration, Commerce and Transport**. It coordinates international standardization by working through the **UN/ECE** (United Nations/Economic Commission for Europe). It provides:

- an international EDI standard
- a set of syntax rules
- data elements, segments and codes
- messages

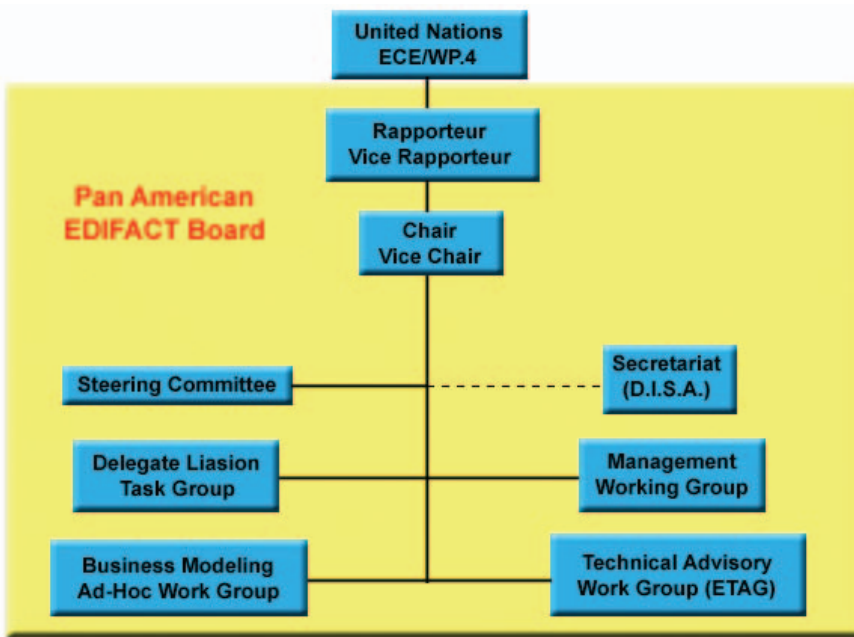
As shown in the following diagram, EDIFACT is the product of the evolution in bringing the Proprietary Standards of the U.S. and Europe together to form a single international EDI standard.



In order to bring about the evolution of the EDIFACT standard, the UN has created UN/ECE to coordinate this effort. The organizational structure of the UN/ECE is made up of the following board members:



The organizational structure of the regional boards are all structured in a similar fashion. Here is the organizational structure of the Pan American Board:



Part 3: Message Definition

A Message is a single business document. Each message is identified by a six character name. From the buyer-side these include:

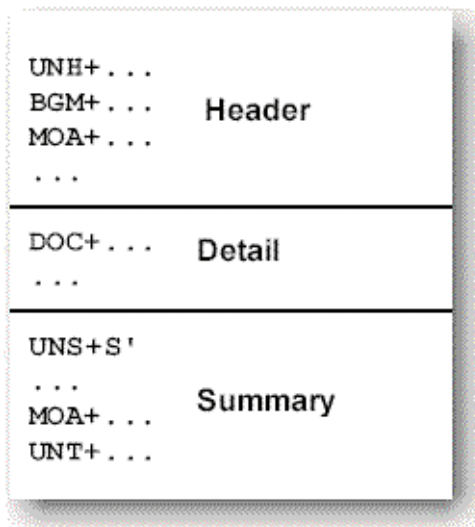
- **ORDERS**—Purchase Orders
- **CUSDEC**—Customs Declaration
- **IFTMIN**—Instruction Message
- **REMADV**—Remittance Advice
- **PAYORD**—Payment Order

Seller-side messages include:

- **IFTMAN**—Arrival Notice
- **CUSRES**—Custom Response
- **INVOIC**—Invoices

Messages are made up of a collection of sequenced **segments** within defined **areas**. Some segments may be used in more than one area. The segments that can be used in each area are defined by the EDIFACT documentation. EDIFACT provides a hierarchical structure for messages.

Messages begin with the Message Header (UNH) Segment and end with the Message Trailer (UNT) Segment. These two segments are the first, and innermost, level of the three levels of “electronic envelopes” within EDIFACT. Here is an example of an Extended Payment Order (PAYEXT) message that illustrates this structure:



Part 4: Message Structure: Segment Tables

The message structure is defined in **segment tables**. These give the ‘rules’ of the message. They also show which segments are used in a particular message and the order in which the segments must appear.

Here is an example of a segment table for the Extended Payment Order (PAYEXT):

Position	Tag	Name
0010	UNH	Message Header
0020	BGM	Beginning of Message
0030	DTM	Date/time/period
0040	BUS	Business function
0060	RFF	Reference
0070	DTM	Date/time/period
0080	FTX	Free text
0090	PAI	Payment instructions
0100	FCA	Financial charges allocation
0120	MOA	Monetary amount
0130	CUX	Currencies
0140	DTM	Date/time/period
0150	RFF	Reference
	etc.	
	etc.	

Segment tables specify if a segment must appear in a message. This is done using the **‘Requirements Designator’** field. Each segment in the table is designated as either **Mandatory (M)** or **Conditional (C)**. Mandatory means that at least one occurrence of the segment must appear in the message. Conditional means a segment may be used, if needed, but it is not required.

Segment tables also specify how many times a particular segment may repeat. This is called the **‘Repetition’** field. Here is the requirements designators and repetition in as displayed in the table for the Extended Payment Order (PAYEXT) message:

Position	Tag	Name	Req	Rept
0010	UNH	Message Header	M	1
0020	BGM	Beginning of Message	M	1
0030	BUS	Business Function	C	1
0040	DTM	Date/Time/Period	M	4
0060	RFF	Reference	M	1
0070	DTM	Date/Time/Period	C	1
0080	FTX	Free Text	C	5
0090	PAI	Payment Instructions	C	1
0100	FCA	Financial Charges Allocation	C	1
0120	MOA	Monetary Amount	M	1
0130	CUX	Currencies	C	1
0140	DTM	Date/Time/Period	C	2
0150	RFF	Reference	C	1
	etc.			
	etc.			

Part 5: Message Structure: Segment Groups

When collections of segments repeat as a group, they are called **segment groups**. If you are familiar with ANSI X.12, these are the equivalents of loops.

Here is an example of segment groups for the Extended Payment Order (PAYEXT):

Position	Tag	Name	Req	Rept
0010	UNH	Message Header	M	1
0020	BGM	Beginning of Message	M	1
0030	BUS	Business Function	C	1
0040	DTM	Date/Time/Period	M	4
0050	Segment Group 1		C	2
0060	RFF	Reference	M	1
0070	DTM	Date/Time/Period	C	1
0080	FTX	Free Text	C	5
0090	PAI	Payment Instructions	C	1
0100	FCA	Financial Charges Allocation	C	1
0110	Segment Group 2		M	1
0120	MOA	Monetary Amount	M	1
0130	CUX	Currencies	C	1
0140	DTM	Date/Time/Period	C	2
0150	RFF	Reference	C	1
	etc.			
	etc.			

Segment groups may be 'nested'. This means that a segment group is fully contained within another segment group.

Here is an example of a **Nested Segment Group**:

Position	Tag	Name	Req	Rept
0280	Segment Group 6		C	9999
0290	DOC	Beginning of Message	M	1
0310	DTM	Date/Time/Period	C	5
0320	RFF	Reference	C	5
0330	NAD	Name and Address	C	2
0340	Segment Group 7		C	5
0350	CUX	Currencies	M	1
0360	DTM	Date/Time/Period	C	1
0370	Segment Group 8		C	100
0380	AJT	Adjustment Details	M	1
0390	MOA	Monetary Amount	C	1
0400	RFF	Reference	C	1
	etc.			
	etc.			

Both Segment Group 7 segments (CUX, DTM) and Segment Group 8 segments (AJT, MOA, RFF) are within Segment Group 6 (Document Details).

- Take Self Test I to check your understanding at this point. Self Test I can be found at the end of this tutorial.

Part 6: Segments

A **segment** is a collection of logically related **data elements** in a fixed, defined sequence. Segments contain:

- a three-character alphanumeric code that identifies the segment. This is called the **segment tag**.
- **variable length data elements**. These can be either **simple** or **composite**.

Segments must be separated by a data element separator (data element delimiter), which is normally + and :, and terminated by a segment terminator, normally '.

All segments are fully documented in the United Nations Trade Data Interchange Directory (UNTDID). These tables list the segment position, segment tag, segment name. Segment tables also specify if a segment must appear in a message using the requirements designator M (Mandatory) or C (Conditional), and how many times a particular segment may repeat (repetition field).

In EDIFACT, there are two kinds of segments:

- **Service Segments**
- **Generic Segments**

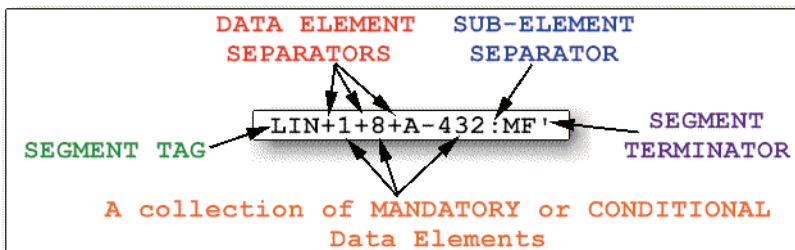
Service Segments are:

- Envelopes (UNB-UNZ, UNG-UNE, UNH-UNT)
- Delimiter String Advice (UNA)
- Section Separator (UNS)

Generic Segments are:

- DOC to identify and specify documents
- MOA for monetary amounts
- DTM for dates and times
- NAD for name and address data

Here is a sample segment:



Part 7: Segment Terminators and Delimiters

The end of each segment is determined by the Data Segment Terminator. In EDIFACT the standard data segment terminator is ‘.

Indicates no more data

`NAD+BY+123456789::16'`

Optional or conditional data elements that are not used must be accounted for by their position with the segment. Here is an example:

`NAD+BY+123456789::16'`

Indicates optional data not used

However, optional or conditional 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. Here is an example:

`NAD+BY+123456789::16++++++'`

Not Correct

Part 8: What is Mapping?

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

1. The computer applications of one business couldn't talk to those of another. This meant re-entering all data that was received.
2. The applications 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. This meant re-entering required data multiple times.

The solution 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:

QTY	UNIT	NO.	DESCRIPTION	PRICE
3	CASE	6900	CELLULOSE SPONGES	12.75
12	EACH	P450	PLASTIC PAILS	.457
4	EACH	1640Y	YELLOW DISH DRAINER	.94

Standards translate the 'human readable' invoice to a 'machine readable' format:

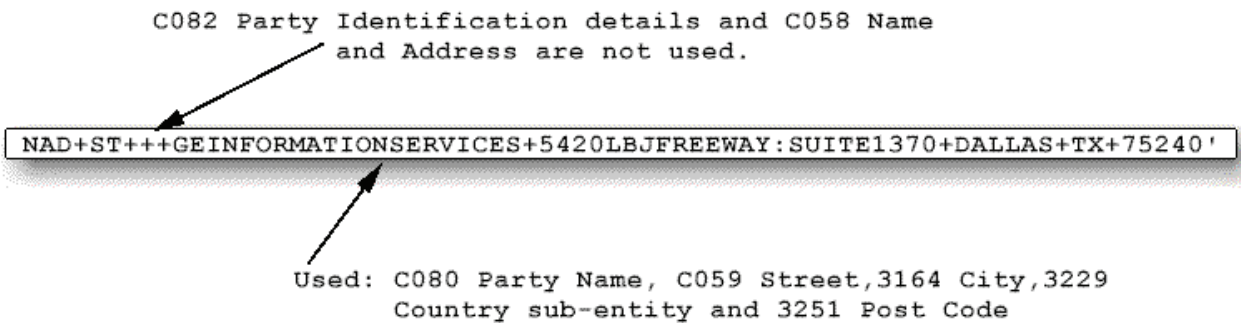
```
LIN+1++6900:MF'  
IMD+F+++::CELLULOSE SPONGES'  
QTY+21:3:CA'  
PRI+CAL+12.75'  
  
LIN+2++P450:MF'  
IMD+F+++::PLASTIC PAILS'  
QTY+21:12:EA'  
PRI+CAL+0.457'  
  
LIN+3++1640:MF'  
IMD+F+++::YELLOW DISH DRAINER'  
QTY+21:4:EA'  
PRI+CAL+0.94'
```

Part 9: Data Elements: Simple and Composite

A **simple** data element contains one piece of information. The **composite** data element contains more than one piece of information, usually containing qualifiers.

In EDIFACT all mandatory data elements must contain data. Conditional data elements may or may not contain data, depending on the requirements of the particular transmission.

Since data elements must be accounted for by their position in the segment, if an optional or conditional data element does not have data, that data element must still be accounted for in its position within the segment by using the appropriate number of data element separators to 'skip over' the empty field. For example:



Here is a list of data element types and the rules that apply to them:

Numeric

A numeric segment may contain only digits, a decimal point and, if negative, a minus sign.

Attributes	Sample Data
n..4	Permitted: 1 123 1234 -1234 1.1 1.234 Not Permitted: A12 12345
n4	Permitted: 1234 -1234 Not Permitted: 1 123 A12 -12345
n8	Permitted 20000214 (dates are numerics)

If the numeric is given as a decimal, the number must have a digit before and after the decimal point. For example: 2.0 is correct (as is 2), however, 2. is wrong. 0.50 is correct (as is 0.5), .50 is wrong.

Alphabetic

An Alphabetic segment contains the specified number of alpha characters, including imbedded blanks. Leading spaces must be preserved.

Attributes	Sample Data
a..8	Permitted: ABCD TOM ABC ALSMITH Not Permitted: DONGREENWOOD
a4	Permitted: ABCD WXYZ Not Permitted: A A123 BCDEF

Alphanumeric

Alphanumeric segments contain the specified number of alphanumeric characters (including imbedded blanks). Leading spaces must be preserved.

Attributes	Sample Data
a..8	Permitted: ABCD TOM ABC ALSMITH Not Permitted: DONGREENWOOD
a4	Permitted: ABCD WXYZ Not Permitted: A A123 BCDEF

Different types of data elements also have specific rules they must follow. The data element dictionary usually specifies the codes (Identifiers) by using the words 'coded' or 'qualifier' in the data element name:

6345 Currency, coded	C an..3
6343 Currency qualifier	C an..3

Part 10: Composite Data Elements: Qualifier and Value

In EDIFACT, the composite data element is made up of 2 or more pieces of data (known as **components**) which form a single data unit. Typically the first data element is the **value**, which is being qualified. The second data element is typically the **qualifier**. These are typically ID (code values) fields. The qualifier gives additional definition to the value.

Here is an example of a composite data element. This data element is in regard to financial institution information. This is the information provided in the segment detail:

3035	Party Qualifier	M	an..3
C078	Account Identification	C	
3194	Account holder number	C	an..17
3192	Account holder name	C	an..35
3192	Account holder name	C	an..35
6345	Currency, coded	C	an..3
C088	Institution Identification	C	
3433	Institution name identification	C	an..11
1131	Code list qualifier	C	an..3
3055	Code list responsible agency, coded	C	an..3
3434	Institution branch number	C	an..17
1131	Code list qualifier	C	an..3
3055	Code list responsible agency, coded	C	an..3
3432	Institution name	C	an..70
3436	Institution branch place	C	an..17
3207	Country, coded	C	an..3

This is how the Party Qualifier data element (3035) is displayed in the message:

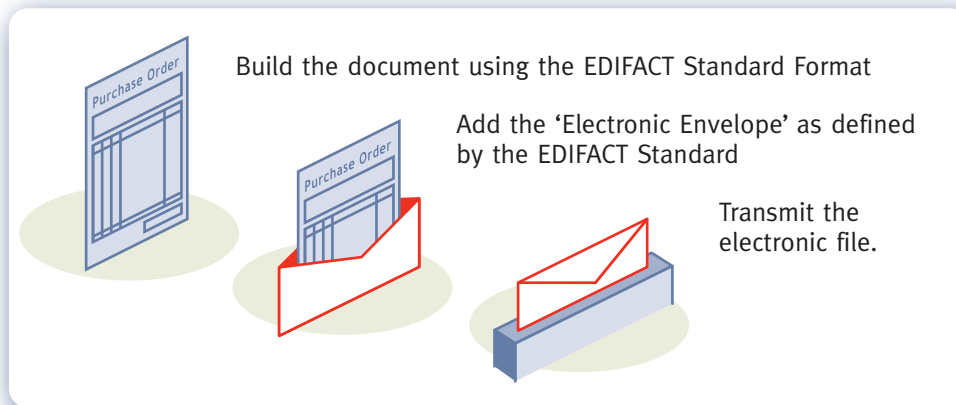
FII+BK+2160644555:W.G.CAFIERO+111902039:25:19::::NATIONS BANK: RICHARDSON, TX'					
DE3194	DE3192	DE3433	DE 1131	DE3432	DE3436
C078			C088		

The composite data elements (C078 and C088) are made up of various conditional components from the segment table. Because they are conditional not all of the data elements are used. All components are separated by a sub-element qualifier (:).

- Take Self Test II to check your understanding at this point. Self Test II can be found at the end of this tutorial.

Part 11: Message Structure and Electronic Enveloping

There are three significant steps for creating and sending data using EDIFACT:



Part 12: Envelope Architecture

Levels and Character Sets

In EDIFACT there are two levels in which messages may be transmitted. The use of a particular level designates which character set will be used:

- LEVEL A (UNA): only upper case; only printable characters
- LEVEL B (UNB): upper and lower case; includes non-printing characters for delimiters

The UNA Interchange is transmitted as a single string of 9 characters prior to the UNB Interchange segment. UNA is optional, and if not used, the defaults shown below apply:

	LEVEL A	LEVEL B	
Character 1	:	IS1	Sub-element delimiter
Character 2	+	IS3	Data element delimiter
Character 3	. or ,	Same	Decimal point indicator
Character 4	?	Not Used	Release character (space if not used)
Character 5	Reserved for future use		Space
Character 6	'	IS4	Segment terminator

A RELEASE CHARACTER is used when there is a need to transmit a message in which a character is designated as a delimiter, indicator or terminator, but it also represented in the message data. For example:

The name of a company is **DON'S DISTRIBUTION COMPANY**. The apostrophe is designated as the segment terminator. Question: How do you transmit the apostrophe in **DON'S** in your message?

Solution: Use a release character (normally ?). So the data would look as follows:

DON'S DISTRIBUTION COMPANY

It is also possible to release the release character. The question: **WHEN WILL YOU SEND THE SHIPMENT?** would be transmitted: **WHEN WILL YOU SEND THE SHIPMENT??**

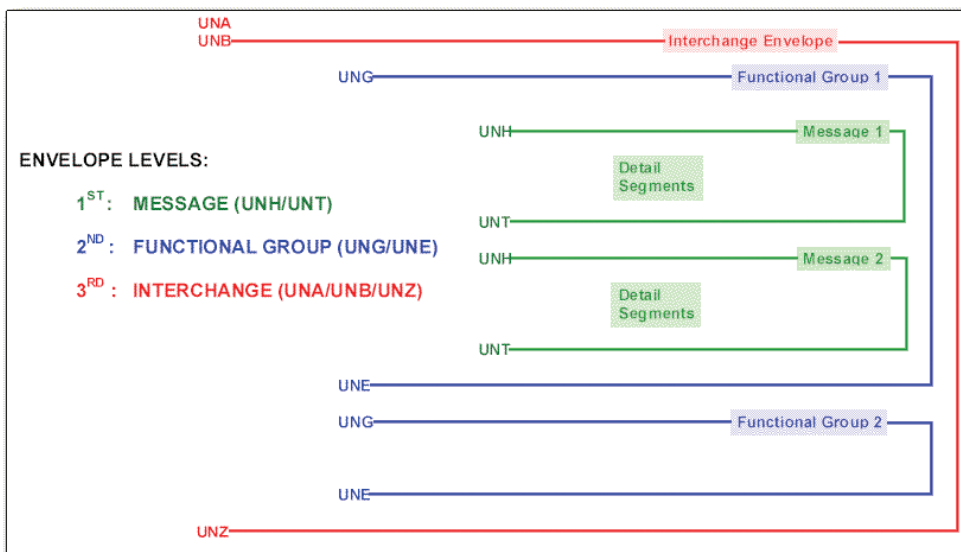
Part 13: Electronic Enveloping

EDIFACT has two required levels of envelopes:

- **Interchange (UNB/UNZ):** a set from one sender's mailbox address to another sender's mailbox address
- **Message (UNH/UNT):** the envelope around one particular message

In addition, there is one optional envelope level: **Functional Group (UNG/UNE)**. It is used to group like messages together and for sub-addressing within an organization. In the US ANSI X.12 standards this group level is where the message format and version are specified. Use of the UNG/UNE is mandatory to/from North America.

The following diagram illustrates Electronic Enveloping:



Part 14: The Message Envelope

The innermost envelope level is around each message. It is defined by the UNH/UNT segments.

The UNH segment has four Data Elements:

- **Message Reference Number (M):** assigned by the sender's computer and is part of the CONTROL mechanism.

- **Message Identifier (M):** Composite which identifies the message being enveloped using the six character message code (e.g., PAYEXT, REMADV) and the Version/Release data.
- **Common Access Reference Number (C):** Relates multiple transactions together.
- **Status of the Transfer (C):** Sequences a series of related messages.

The UNT segment has two Data Elements:

- **Number of Segments in a Message (M):** count of segments in the message (including UNH and UNT).
- **Message Reference Number (M):** same one that was used in the UNH for that particular message.

Here is an example of how the CONTROL mechanism in the UNH element is used to validate message data:

Suppose this message was sent:

```

UNH+1001+ORDERS:D:94B:UN'
BGM+105+12345+...'
DTM+4:931102:101'
LIN+1+...'
LIN+2+...'
LIN+3+...'
UNS+S'
CNT+2:3'
UNT+9+1001'
UNH+1002+ORDERS:D:94B:UN'
BGM+105+12346+...'
DTM+4:931102:101'
LIN+1+...'
LIN+2+...'
UNS+S:2'
CNT+2:2'
UNT+8+1002'

```

But this part was 'lost' in the process

And this is what was received:

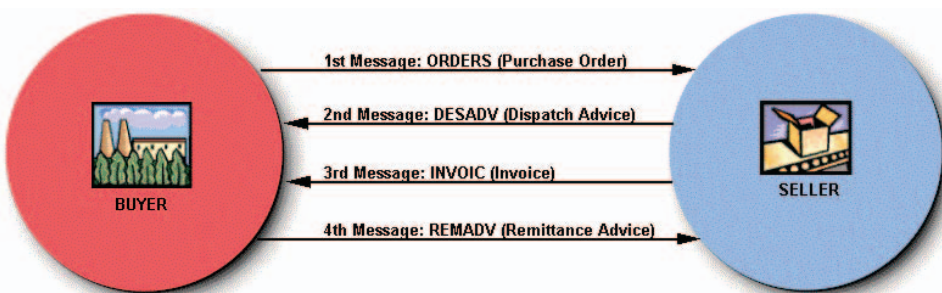
```

UNH+1001+ORDERS:D:94B:UN'
BGM+105+12345+...'
DTM+4:931102:101'
LIN+1+...'
LIN+2+...'
UNS+S:2'
CNT+2:2'
UNT+8+1002'

```

Even though the COUNTS are correct, a CONTROL NUMBER match will catch the error

The **Common Access Reference Number** is used to identify a series of related EDIFACT messages. For example, one purchase may involve a message exchange that requires four messages to accomplish the complete business transaction as given here:



- For Message #1: UNH+2348+ORDERS:D:94B:UN+10381+1:C'
- For Message #2: UNH+156009+DESADV:D:94B:UN+10381+2'
- For Message #3: UNH+156078+INVOIC:D:94B:UN+10381+3'
- For Message #4: UNH+2451+REMADV:D:94B:UN+10381+4:F'

Part 15: The Functional Group Envelope

Functional Group Envelope

The second (middle) envelope level is around each functional group. It is defined by the UNG/UNE segments. The use of the UNG and UNE envelopes is mandatory for EDI to/from North America.

This envelope groups like types of messages within a transmission. Here are a few examples of the data elements in the functional group envelope:

- **Functional Group (M)**
- **Message Identifier (M)**
- **Date/Time Stamp (M):** Relates multiple transactions together.
- **Status of the Transfer (C):** Sequences a series of related messages.
- **Group Reference Number (M)**
- **Controlling Agency (M)**
- **Message Version (M)**
- **Application Password (C)**

The UNE segment includes:

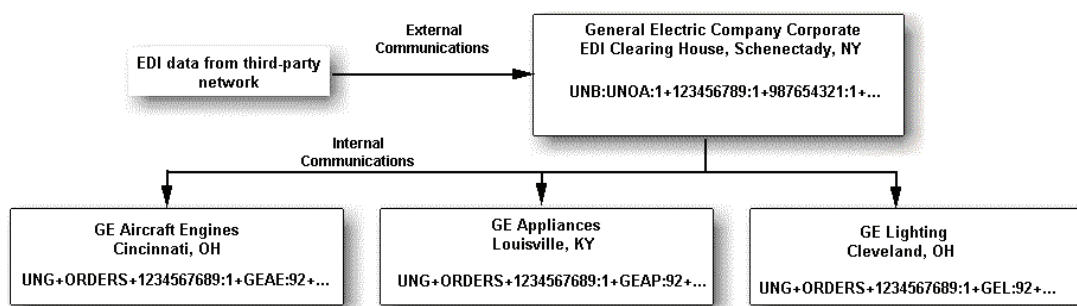
- **Number of Segments in a Message (M)**
- **Message Reference Number (M)**

Functional Group Sub-Addressing

Functional Group envelopes contain a sub-addressing capability. The data that is sent to a particular receiver is addressed to the mailbox address on the UNB.

Many companies want to route a group of data internally, so the UNG segment has a provision for user defined addresses in the S006 and S007 elements.

Here is a diagram that illustrates the sub-addressing function:



Part 16: The Interchange Envelope

The outermost level of the message envelope structure is the interchange envelope. It is defined by the UNA, UNB and UNZ segments.

This envelope is used to identify data sent from one sender to one receiver:

The UNA segment contains:

- Delimiter String Advice
- Examples of included data elements

The UNB segment contains:

- Date/Time Stamp (M)
- Interchange Control Numbers (M)\
- Password and Application Reference (C)
- Processing Priority Reference (C)
- Acknowledgment Request Indicators (C)
- Communications Agreement ID (C)
- Test Indicators (C)

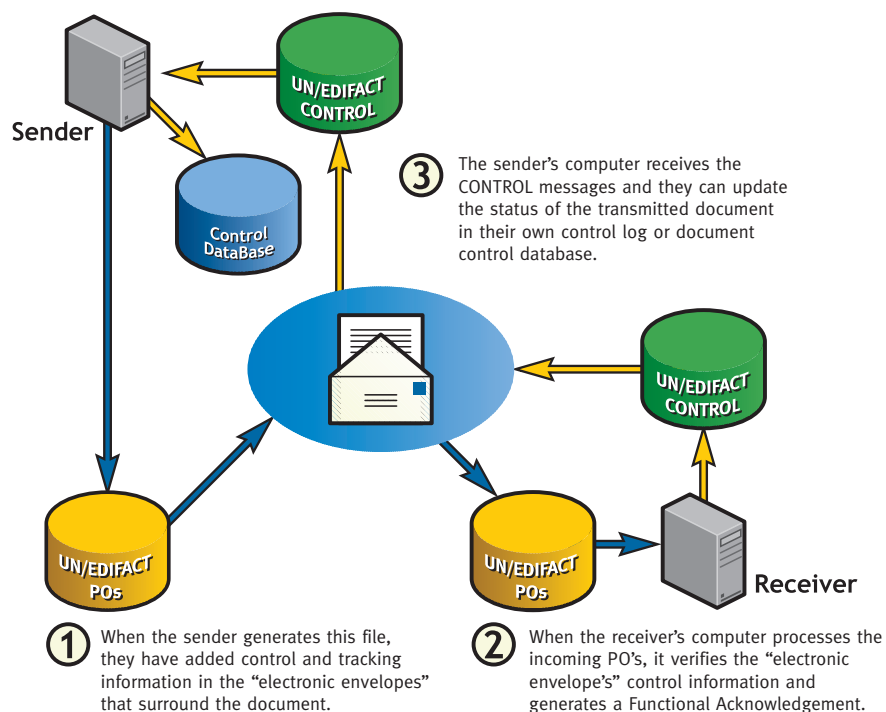
The UNZ segment includes:

- Interchange Control Numbers (M)
- Counts of Messages or Groups in the Interchange (M)

Part 17: The CONTRL Message

It is the responsibility of the receiver's computer to check the syntax and control numbers of the transmission and to build and transmit back to the sender this Functional Acknowledgment.

The **EDIFACT CONTRL** message will provide this functionality.



- Take Self Test III to check your understanding at this point. Self Test III can be found at the end of this tutorial.

EDIFACT Wrap-up

Some things to keep in mind:

- EDIFACT was originally developed from a base of U.S. ANSI standards and UN GTDI standards
- The syntax of both standards are remarkably similar
- Many segments are similar in the two standards
- Translation software can readily generate/accept either EDIFACT or ANSI ASC X.12 data
- A key issue will be the organizational improvements needed to develop/maintain EDIFACT standards within the U.S. and Canada.

Thank you for using the GXS EDIFACT Tutorial. Please send all feedback to training@gxs.com.

EDIFACT Glossary

Area—Areas contain groups of segments that perform specific functions. The areas, in turn, make up a message. To be complete a message must contain a header area, detail area and summary area.

Component—A composite data element made up of 2 or more pieces of data.

Composite Data Element—Two or more related data items separated by a delimiter character, grouped together to form a unit or piece of information as defined in the data dictionary of a system of EDI Standards, and contained in an EDI message.

Conditional—Attribute that indicates that the data element may or may not contain data, depending on the message.

Data Element—One or more data items, forming a unit or piece of information as defined in the data dictionary of a system of EDI Standards, and contained in an EDI message or transaction set. The term “data element” is often abbreviated as “DE” followed immediately by the data element number (i.e., data element 128 would be abbreviated as DE128) in some texts

Data Element Separator (Delimiter)—Character that separates the individual data elements in a segment.

EDIFACT—Also known as UN/EDIFACT, is the acronym for EDI for Administration, Commerce and Transport. It is the international message standard for EDI being developed through the cooperation of the United Nations and the Economic Commission for Europe (UN/ECE).

Mandatory—Attribute that indicates that the data element must contain data.

Nested Segment Group—Segment Groups that reside within other segment groups.

Qualifier—Typically an ID (code value) field, it is used to qualify a particular value in a segment.

Repetition Field—Entry in a segment table which states how many times a particular segment may repeat.

Requirements Designator—Usage indicator of segment detail: Mandatory, Optional, or Floating.

Segment—A part of an EDI message or transaction set, made up of a number of logically related data elements in a fixed, defined, sequence, separated by a delimiter, conveying a part of the business transaction being made.

Segment Tables—The message structure is defined in a listing known as a ‘segment table’. These give the ‘rules’ of the message.

Segment Tag—A three character alphanumeric code that identifies the segment.

Segment Terminator—Character at the end of a segment that indicates no more data follows.

Simple Data Element—A data element contains one piece of information.

Service Segment—Service Segments are Envelopes (UNB-UNZ, UNG-UNE, UNH-UNT), Delimiter String Advice (UNA), and Section Separators (UNS).

UN/ECE—The United Nations/Economic Commission for Europe that oversees the development of the EDIFACT standard.

Value—Typically, the first data element which is being qualified.

Variable Length Data Element—Data element whose length changes depending on the number of characters in the actual data. Minimum and maximum size of the element is given in the element table.

EDIFACT Self-Test I: After Completion of Part 5

Select the most appropriate answer to the following questions:

1. A primary benefit of EDIFACT is that...

- it is an industry-to-industry only standard.
- it is from the UN.
- it is both industry-to-industry and country-to-country.
- it is a form of data encryption.

2. Messages:

- have all the of the next three features.
- collections of sequenced segments within defined areas.
- being with the UNH segment and end with the UNT segment.
- identified by a six character name.

3. Message structure is defined in the:

- segment group.
- segment table.
- segment rules.
- segment tag.

4. Segments contain...

- a segment tag.
- simple or composite variable length data elements.
- the data element separators + or :.
- all of the above.

EDIFACT Self-Test II: After Completion of Part 10

Select the most appropriate answer to the following questions:

1. Which is true of data elements:

- All optional or conditional elements must be accounted for.
- All conditional elements are dependent on message requirements.
- All mandatory data elements must contain data.
- All of the above.

2. EDIFACT uses two separate pieces of data in a single element:

- Value and Qualifier.
- Simple and Composite.
- Numeric and Alphanumeric.
- Composite and Component.

3. If there is an alphanumeric data element with an attribute of any all of these would be correct except:

- KEN D
- END 64
- 48597
- TOM12

4. Optional data elements without data at the end of a segment...

- need additional data element separators to correctly position the data.
- must be accounted for by their position with the segment.
- do not need additional data element separators to correctly position the data.
- all of the above.

EDIFACT Self-Test III: After Completion of Part 17

Select the most appropriate answer to the following questions:

1. The characteristics of Level A EDIFACT transmissions are:

- upper case, includes non-printing characters for delimiters.
- upper and lower case.
- upper case with only printable characters.
- none of the above.

2. A release character...

- allows the use of a delimiter or terminator within data.
- is normally the '?'.
- can release a release character.
- all of the above.

3. EDIFACT has two required levels of envelopes:

- Interchange and Functional Group.
- Message and Interchange.
- Functional Group and Message.
- Sub-Addressing and Message.

4. For an incoming message, it is the responsibility of the receiver's computer to...

- check syntax, then build and transmit a Contrl message.
- check control numbers, then build and transmit a Contrl message.
- build and transmit a Contrl message.
- check syntax and control numbers, then build and transmit a Contrl message.