

Paragraph 2

EDIFACT

EDIFACT

We are all familiar with the following terms: invoice, statement of account, purchase order. Normally we send these as paper documents, which implies difficulty in writing the paper, difficulty in forwarding the paper and difficulty in using the information of the paper.

As most companies use data processing writing on the paper and most companies need to receive the information in their own data processing systems, it is obvious to exchange this information between data processing machines. This facilitated exchange of data: Saving time printing papers, saving time dispatching papers and saving time entering information into the data processing system.

In order to exchange information between EDP-machines an agreed set of standard data elements is necessary. It is, for example, necessary to know which information a purchase order or an invoice must contain, and in which order the information must be sent.

The International Standard Organisation (ISO) and United Nations (UN) have made syntax rules for electronic data exchange called EDIFACT, electronic Data Exchange For Administration, Commerce & Transport.

The Paneuropean user organisation for EDI within the construction industry EDIBUILD has developed a common documentation for electronic data exchange of supply, DECIDE/S (Documentation European Construction Industry Data Exchange/Supply).

EDIBUILD is responsible for the translation and rewriting of the English master version of DECIDE/S.

EDI-Byg - the Danish version of EDIBUILD - has made an agreement with EAN Danmark which determines that the documentation for the construction industry will be an addition to the standards described in HANCOM 97 for the retail trade.

EDIFACT terms etc.

EDIFACT contains the following syntaxes:

dispatches

messages

segments

data elements/fields

A **physical dispatch** contains all information in one dispatch. A physical dispatch may contain several logical dispatches.

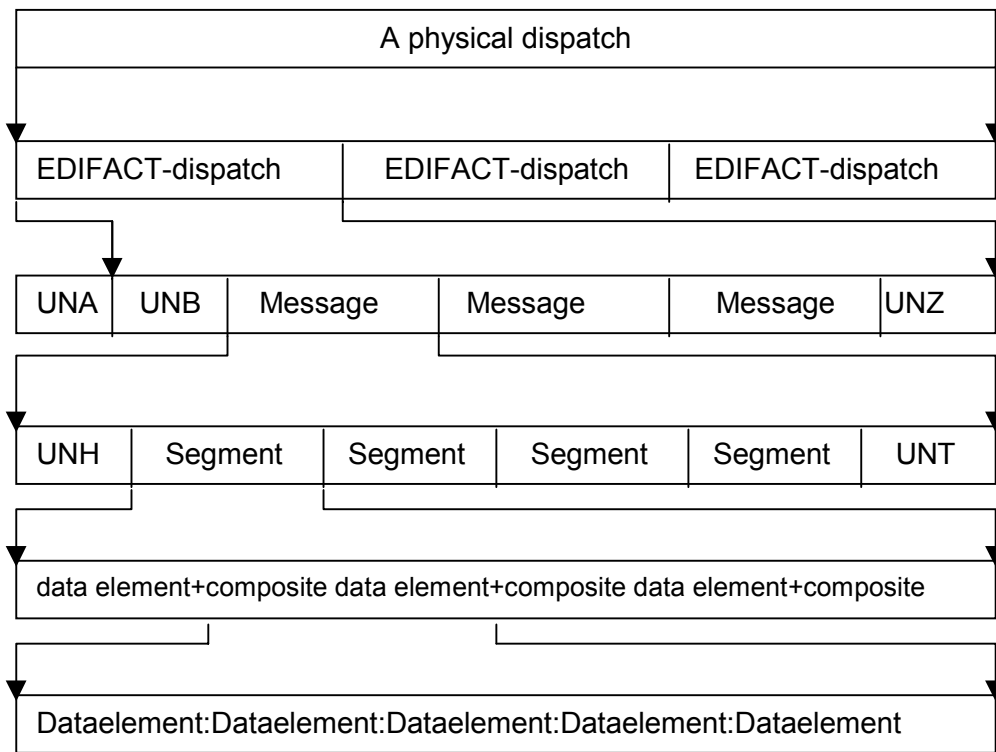
A **logical dispatch** (EDIFACT dispatch) contains all information sent from **one** sender to **one** recipient. You may only send to **one** recipient in **one** logical dispatch.

A **message** is an order, an invoice, or a statement of account, i.e. information typically stated on the same piece of paper.

A **segment** is a group of connected data elements/fields, e.g. article number. This article number may be the manufacturer's article number, the wholesaler's article number or the plumber's article number. These numbers are grouped as a 'logical' article number.

Data elements (fields) are individual information sent in one segment, e.g. a wholesaler article number.

In the figure on the next page showing the EDIFACT syntaxes five different syntaxes are used. UNA, UNB & UNZ are segments used to indicate the beginning and the end of the dispatch. UNH & UNT are segments used to indicate the beginning and the end of a message.



A logical dispatch can only have one recipient.

A UNB (beginning) segment and a UNZ (ending) segment delimit a dispatch. In front of the UNB segment there **MUST** always be a UNA segment indicating which characters are used in the management of the syntax in the EDIFACT. LEC-EDI is able to use and forward all UNA segments provided they observe the rules indicated in the next chapter.

The UNB & UNZ segments are the 'envelope' in which data are forwarded. LEC-EDI uses the sender and the recipient etc. 'mentioned' on the 'envelope'. The content of the 'envelope' **is a matter between the sender and the recipient**.

The number of messages in one dispatch is unlimited.

A UNH (beginning) segment and a UNT (ending) segment delimit each message. LEC-EDI does not control this information.

The number of segments varies from message to message. LEC-EDI only controls the segment 'on the envelope', i.e. in the UNB & UNZ segments.

EDIFACT-syntax

Message names and segment names are always written in CAPITAL letters when transmitted.

To distinguish between data elements and between segments a + is used. To indicate the ending of a segment a ´ (single quotation mark) is used.

When composite data elements are used, i.e. several fields describing the same thing, : is used as a separator between each data field of the composite data element.

If one of the reserved separators is used (+, :, ´), a ? must be put in front of it. For example: 10?+20=30. Question mark is represented by ?? . The first ? restores the original meaning of the next ?.

If a so-called 'release' character is used, it does not count in the field length. If a field consists of 20 alphanumeric characters and for example 2 'release' characters are needed, the field may be up to 22 positions.

If a field consists of, for example, 20 characters and only 5 characters are needed, only 5 characters are sent.

If a field has zeros in front, these are not sent, neither are fields with blanks in front or at the end.

Data elements (fields) can be neglected omitting 'the value' of the fields. The result might be 24500+45++25600, where the field value between 45 and 25600 has been omitted. If you use a composed data element, the result might be 3650:3575::3500 where the field value between 3575 and 3500 has been omitted.

The segments can be neglected continuing directly with the subsequent segment name or the UNT (the trailer) segment.

You are NEVER to send 24500+45+++++´, but instead send 24500+45´, i.e. make an abbreviation if possible. Furthermore, you are NEVER to send 24500:45::::´, but instead send 24500:45´.

Floating decimal points are used in fields where it is relevant to use decimals. The decimal character is indicated in the UNA-segment, usually by means of a comma. Therefore, 1.785,75 is indicated as 1785,75.

In the UNB segment yymmdd and hhmm MUST be filled in completely. I.e. 645 is not enough to indicate time. The correct indication is 0645.

If more messages are to be sent in the same dispatch, a UNT segment from the previous message must be followed by a UNH segment from the subsequent message. There is no limit as to the number of EDIFACT-messages which may be sent in the same physical dispatch.

The same message can be seen several times in one dispatch. However, the contents of the messages must vary.

All messages between a UNB segment and a UNZ segment **MUST** be for the same recipient.

Transmission protocols make sure that the sequence of the forwarded records is always correct.

UNA-service segment

Dispatches in EDIFACT-format **MUST** always start with a 'UNA'. This service segment is used to indicate which control characters are used in the dispatch. This means that it is used to indicate which characters are used as separators in composite data elements (default :), ordinary data elements (default +), decimal separation (default ,), release indicator (default ?), A blank character is reserved for future purposes and segment ending (default `).

Example: **UNA:+,? `**

The example uses the default values. It is the forwarded characters in the UNA segment that are valid.

When LEC-EDI receives data, the UNA segment is copied in front of each UNB segment before the data are delivered in the recipient's mailbox.

Example of input to LEC-EDI:

UNA'UNB1.....'UNZ1'UNB2.....'UNZ2'

The above input after sorting:

UNA'UNB1.....'UNZ1'
UNA'UNB2.....'UNZ2'

When data are forwarded to LEC-EDI, there must only be one UNA segment. This UNA segment **MUST** be placed as the first three characters in the 'physical' dispatch.

When data are extracted from LEC-EDI, there may be several, even different, UNA segments in the data extracted since there may be data from various senders. It may be an advantage if the sender and the recipient always use the same UNA characters, but this must be stipulated between the two parties.

If the UNA segment is completed, this is checked by LEC-EDI. The UNA segment MUST contain:

UNA	3 alphanumeric characters
component data element separator	1 alphanumeric character
data element separator	1 alphanumeric character
decimal separator	1 alphanumeric character
'release' character	1 alphanumeric character
reserved	1 blank
segment terminator	1 alphanumeric character

This means that it is NOT allowed to 'abbreviate' the UNA segment using the syntax rules in the chapter concerning syntax.

Example of data at the recipient:

```
UNA'UNB.....  
.....'UNZ'  
UNA'UNB.....'UNZ'UNB.....  
.....'UNZ'UNB.....  
.....'UNZ'  
UNA'UNB.....  
.....'UNZ'
```

The UNB-service segment

The UNB segment must always be filled in. The UNB segment is described in the examples and in the catalogue segment.

The UNZ-service segment

The UNZ segment must always be filled in. The UNZ segment is described in the examples and in the catalogue segment.

General Considerations

In this connection it is **IMPORTANT** to notice that the **UNB and UNZ segments DO NOT start/terminate** in particular positions of the records.

Furthermore, it is **IMPORTANT** to notice that the recipient (a new UNB segment) may change in the middle of a record.

An example of data as they appear when received by LEC-EDI:

```

UNA'UNB.....
.....'UNZ'UNB.....
.....'UNZ'UNB.....
.....'UNZ'

```

There may only be one UNA segment in one physical dispatch to LEC-EDI. Each set of coherent UNB and UNZ segments indicates a sender/recipient combination. There may be one or more recipients in the above physical dispatch.

An example of data as they appear at the recipient:

```

UNA'UNB.....
.....'UNZ'
UNA'UNB.....UNZ'UNB.....
.....'UNZ'UNB.....
.....'UNZ'
UNA'UNB.....
.....'UNZ'

```

The UNA segments may be different, but the recipient must ALWAYS be able to see which UNA values are valid.

Message 'envelope'

Schedule

UNB+UNH+segments+UNT+UNH+segments+UNT+UNZ

The plus signs (+) indicate the separation of various elements. Please note that in the example there are two messages in the dispatch. There may be a random number of messages in one dispatch provided that the sender and the recipient are the same.

A UNH and a UNT segment surround each message in a dispatch. The UNH segment is sent to indicate which message is being sent and the UNT segment is sent to indicate that the message is terminated.

The UNH and UNT segment must ALWAYS be filled in.

The UNH and UNT are described in detail later.