## T.2 STRUCTURE OF AN EDIFACT TRANSMISSION

This section is a summary of the ISO 9735 document: "*EDIFACT application level syntax rules*", first released on 1988-07-15, amended and reprinted on 1990-11-01, and *Addendum 1* of 1992.

The EDIFACT syntax rules set the standards for structuring data into segments, segments into messages, and messages into an interchange.

### T.2.1 STRUCTURE OF AN INTERCHANGE

An interchange may consist of the following segments:

| | | |
|---|---|---|
| UNA | | Conditional |
| UNB | Interchange Header | Mandatory |
| UNG | Functional Group Header | Conditional |
| UNH | Message Header | Mandatory |
| USER DATA SEGMENTS | | |
| UNT | Message Trailer | Mandatory |
| UNE | Functional Group Trailer | Conditional |
| UNZ | Interchange trailer | Mandatory |

Segments starting with "UN" are called service segments. They constitute the envelope or the "packing" of the EDIFACT messages.

The UNA segment defines the separator characters used in the transmission, if they are not the default set for the character set defined in the UNB segment. See section 1.4 below.

The UNB segment identifies the sender and receiver of the transmission, specifies the character set used, and carries other "housekeeping" data for the transmssion.

The UNG and UNE segments are used only if the transmission carries several groups of message of different types. For book and serials applications, EDItEUR recommends that a transmission should be limited to carrying only one group of messages of a single type. The UNG and UNE segments should not be used.

The UNA, UNB and UNZ segments will normally be generated in outgoing transmissions, and processed in incoming transmissions, by a standard EDI software package. The user application need not be aware of their content. They are not, therefore, specified in detail in the present document.

User data segments contain the information itself, in a format specific to each message type.

## T.2.2 STRUCTURE OF A MESSAGE

Each data segment has a specific place within the sequence of segments in the message. Data segments may occur in any of the following three sections of the message:

a. **Header section** - A segment occurring in this section relates to the *entire message*.

b. **Detail section** - A segment occurring in this section relates to the detail information only.

c. **Summary section** - Only segments containing *totals* or *control information* may occur in the summary section, eg invoice total amount, number of lines in a purchase order, etc.

The sequence of the three message sections can be represented by the following simple example:

*Header section*

UNH…..
BGM…..
………..
………..

*Detail section*

LIN…….
QTY…..
………..
………..

*Summary section*

CNT…..
UNT…..

The same segment type may occur in more than one of the message sections, e.g. in the header and in the detail section, and/or more than once in the same section.

Some segments may be repeated a certain number of times at their specific location in the message. The status, *mandatory* or *conditional*, and the maximum number of repetitions of segment types are indicated in the message specification.

Within a message, specific groups of functionally related segments may be repeated; these groups are referred to as “segment groups”. The maximum number of repetitions of a particular segment group at a specific location is included in the message specification.

A segment group may be nested within other segment groups, provided that the inner segment group terminates before any outer segment group terminates.

## T.2.3 SEGMENT STRUCTURE

A segment consists of:

- A segment tag, which identifies the segment type
- Data element separators
- Simple, composite, or component data elements
- A segment terminator

Data elements can be defined as having a fixed or variable length.

A composite data element contains two or more component data elements.

A component data element is a simple data element used in a composite data element.

A data element can be qualified by another data element, the value of which is expressed as a code that gives specific meaning to the data. The data value of a qualifier is a code taken from an agreed set of code values.

## T.2.4 CHARACTER SETS AND SEPARATORS

EDIFACT standards define a number of character sets, coded in the UNB segment as UNOA, UNOB, UNOC, UNOD etc. EDItEUR has adopted UNOC as the standard character set for book and serials trading. This character set permits the representation of a full repertoire of special characters, including accents, for most European languages which use the Latin alphabet. It corresponds to the international standard character set ISO 8859.1.

EDItEUR recommends that the following four characters should be used as standard separators:

| | | |
|---|---|---|
| Apostrophe | ' | = segment terminator |
| Plus sign | + | = segment tag and data element separator |
| Colon | : | = component data element separator |
| Question Mark | ? | = release character; immediately preceding one of the separator characters, the release character returns the separator to its normal meaning |

For example, 10?+10=20 means 10+10=20. Question mark is represented by ??

These are NOT the default separators for the UNOC character set. Consequently the UNA segment must be sent at the start of each transmission to explicitly state the separator values.

When using the UNOC character set, the EDIFACT syntax version number (DE 0002) in the UNB segment must be set to 3.

**Example of an EDIFACT segment:**

DTM+137:19940101:102'

| | | |
|---|---|---|
| DTM | = | segment tag, identifying the “Date/time/period” segment |
| + | = | segment tag and data element separator |
| 137 | = | date qualifier, indicating that the date is the document/message date/time |
| : | = | separator for component data elements within a composite (here, date qualifier and date) |
| 19940101 = | | date, in the format specified by the date format qualifier |
| : | = | separator for data elements within a composite (here, date and date format qualifier) |
| 102 | = | date format qualifier, indicating the format of the date (CCYYMMDD) |
| ' | = | segment terminator. |

## T.2.5 COMPRESSION OF DATA

In data elements for which the EDIFACT Directory specifies variable length and no other restrictions, non-significant character positions (eg leading zeroes and trailing spaces) are suppressed.

In the following examples, TAG = segment tag; DE = data element; CE = component data element.

**Exclusion of segments.** Conditional segments containing no data are omitted (including their segment tags).

**Exclusion of data elements by omission.** Data elements are identified by their sequential position within the segment as stated in the EDIFACT Directory. If a conditional data element is omitted and followed by another data element, its position must be indicated by retention of its data element separator.

**Example:**

| | |
|---|---|
| TAG+DE+DE+DE+CE:CE:CE' | Complete segment including all data elements |
| TAG+DE+ +DE+CE:CE:CE' | The second data element has been omitted |

**Exclusion of data elements by truncation.** If one or more conditional data elements at the end of a segment are omitted, the segment is truncated by placing the segment terminator after the last data element for which data is present.

**Example:**

| | |
|---|---|
| TAG+DE+DE+DE+DE' | Segment including all data elements |
| TAG+DE+DE' | Segment truncated after the first two data elements |

**Exclusion of component data elements by omission.** If a conditional CE is omitted and followed by another CE, its given position must be represented by its CE separator.

**Example:**

| | |
|---|---|
| TAG+DE+DE+CE:CE:CE' | Segment including all CE's |
| TAG+DE+DE+CE::CE' | Segment with the penultimate CE omitted |

**Exclusion of component data elements by truncation.** One or more conditional CE's at the end of a composite DE may be excluded by truncation by the DE separator or, if at the end of a segment, by the segment separator.

**Example:**

| | |
|---|---|
| TAG+DE+DE+CE:CE:CE' | Segment including last CE |
| TAG+DE+DE+CE:CE' | Segment truncated by the segment separator |

### T.2.6 REPRESENTATION OF NUMERIC VALUES

**Decimal sign.** The decimal sign is represented by a full point on the line (.). The decimal sign is not counted as a character when computing the maximum field length of a data element. When a decimal sign is transmitted, there must be at least one digit before and after the decimal sign.

To assist in-house file designers and data interchange partners, the following lengths may be used as a guideline:

| Numeric Class | Format | Integer Digit | Decimal Digit |
|---|---|---|---|
| Amounts | n..18 | 15 | 3 |
| Control Values | n..18 | 14 | 4 |
| Cubes | n..9 | 5 | 4 |
| Currency Rates | n..12 | 6 | 6 |
| Other Range Value | n..18 | 15 | 3 |
| Percentages | n..10 | 6 | 4 |
| Percentage Range Value | n..18 | 14 | 4 |
| Quantities | n..15 | 12 | 3 |
| Rate per Unit | n..15 | 12 | 3 |
| Tax Rates | n..17 | 13 | 4 |
| Unit Prices | n..15 | 11 | 4 |
| Unit Price Basis | n..9 | 6 | 3 |
| Weights | n..18 | 15 | 3 |

**Triad separator.** Triad separators shall not be used in interchange. (Allowed: 2500000. Not allowed: 2,500,000 or 2.500.000 or 2 500 000)

**Sign.** Numeric data element values are to be sent as positive. Although conceptually a deduction is negative, it is represented by a positive value: eg in a credit note all values are sent as positive amounts, and the application software will take note of the message name code (DE 1001) and process the values accordingly. In addition some data element and code combinations will lead to implied negative values, eg DE 5463 with code value "A" ( allowance) in an ALC segment in an invoice. Again, the values are sent as positive amounts.

If, however, a value has to be explicitly represented as negative, it must be sent immediately preceded by a minus sign, eg -112. The minus sign is not counted as a character when computing the maximum field length of the data element.

## T.2.7 DOCUMENTATION CONVENTIONS

### Format and picture of data elements

The following conventions apply in the present documentation:

| | |
|---|---|
| a | alphabetic characters |
| n | numeric characters |
| an | alpha-numeric characters |
| a3 | 3 alphabetic characters, fixed length |
| n3 | 3 numeric characters, fixed length |
| an3 | 3 alpha-numeric characters, fixed length |
| a..3 | up to 3 alphabetic characters |
| n..3 | up to 3 numeric characters |
| an..3 | up to 3 alpha-numeric characters |

### Layout of message subset specifications

In sections T.3 onwards, the message subsets required for this application are defined and illustrated. Complete message specifications will be found in the *EDItEUR Reference Manual*, Part III.

To keep the application guidelines as clear and compact as possible, while making it easy to cross-refer to the full specifications if so required, the following conventions have been used:

(a) Only those segments which are used in this application are included in the message definition.

(b) Each segment is identified by its EANCOM segment number, and by its EDIFACT segment code (eg **UNH**) and name (eg **MESSAGE HEADER**).

(c) Each data element or composite data element is identified by its EDIFACT code number (eg 1001) and name (eg Document/message code).

(d) The status of each data element *for the purposes of this application* is shown by one of the following codes:

M Mandatory. This includes data elements which are *mandatory* in EDIFACT; data elements which, although not mandatory in EDIFACT, are *required* in EANCOM; and data elements which EDItEUR considers to be *required for the application*, although both EDIFACT and EANCOM treat them as optional or conditional.

D Dependent. Data elements which are used only under specified conditions, when they may in fact become mandatory.

O Optional. Data elements whose presence is optional, ie to be agreed between trading partners.

N Not used. This category includes data elements which are *not used* in EANCOM, and data elements which are *optional* or *dependent* in EANCOM, but which EDItEUR considers to be unnecessary for the application.

(e) Where a data element is not used in this application, details of its format are omitted.

(f) Where a composite data element is not used in this application, its constituent data elements are not listed.