

# Electronic data interchange for administration, commerce and Transport (EDIFACT) - Application level syntax rules

## 1 Scope

This International Standard gives syntax rules for the preparation of messages to be interchanged between partners in the fields of administration, commerce and transport.

## 2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

*ISO 31-0 : 1981, General principles concerning quantities, units and symbols.*

*ISO 646 : 1983, Information processing - ISO 7-bit coded character set for Information interchange.*

*ISO 2382-1 : 1984, Data processing - Vocabulary - Part 01: Fundamental terms.*

*ISO 2382-4: 1987, Data processing - Vocabulary - Part 04: Organization of data.*

*ISO 6523 : 1984, Data interchange - Structures for the identification of organizations.*

*ISO 6937-2 : 1983, Information processing - Coded character sets for text communication - Part 2: Latin alphabetic and non-alphabetic graphic characters.*

*ISO 7372 : 1986, Trade data interchange - Trade Data Elements Directory.*

*ISO 7498 : 1984, Information processing systems - Open Systems Interconnection - Basic Reference Model.*

*ISO 8859 : 1987, Information processing - 8 bit sing-byte coded graphic character sets –  
Part 1: Latin Alphabet No. 1.  
Part 2: Latin alphabet No. 2.*

**ISO 9735: 1988 (E)**  
**(First edition 1988-07-15)**

### **3 Definitions**

For the purpose of this International Standard, the definitions in annex A apply.

### **4 Syntax levels**

This International Standard specifies syntax levels A and B which are identical in all respects except for the character sets used. As requirements for additional syntactical features appear, further levels may be added.

Unless interchange partners agree to use other or additional characters, level A shall use only the character set specified in 5.1, and level B only the character set specified in 5.2.

The conditional Service String Advice, UNA, (see annex B) provides the capability to specify the separator and other service characters used in the interchange in case they differ from those in clause 5.

### **5 Character sets**

For the characters in the sets below, the 7-bit codes in the basic code table in ISO 646 shall be used, unless the corresponding 8-bit codes in ISO 6937 and ISO 8859 or other bit codes are specifically agreed between the interchanging partners. See clause 4.

#### **5.1 Level A character set**

|                        |        |
|------------------------|--------|
| Letters, upper case    | A to Z |
| Numerals               | 0 to 9 |
| Space character        |        |
| Full stop              | .      |
| Comma                  | ,      |
| Hyphen/minus sign      | -      |
| Opening parenthesis    | (      |
| Closing parenthesis    | )      |
| Oblique stroke (slash) | /      |
| Equals sign            | =      |

|               |  |
|---------------|--|
|               | Reserved for use as:                     |
| 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 characters ' + : ? restores their normal meaning. For example, 10? + 10 = 20 means 10 + 10 = 20. Question mark is represented by ??.

The following characters are part of the level A character set but cannot be used internationally in telex transmissions.

|                   |   |
|-------------------|---|
| Exclamation mark  | ! |
| Quotation mark    | " |
| Percentage sign   | % |
| Ampersand         | & |
| Asterisk          | * |
| Semi-colon        | ; |
| Less-than sign    | < |
| Greater-than sign | > |

## 5.2 Level B character set

This character set is not intended for transmission to telex machines.

|                        |        |
|------------------------|--------|
| Letters, upper case    | A to Z |
| Letters, lower case    | a to z |
| Numerals               | 0 to 9 |
| Space character        |        |
| Full stop              | .      |
| Comma                  | ,      |
| Hyphen/minus sign      | -      |
| Opening parenthesis    | (      |
| Closing parenthesis    | )      |
| Oblique stroke (slash) | /      |
| Apostrophe             | '      |
| Plus sign              | +      |
| Colon                  | :      |
| Equals sign            | =      |
| Question mark          | ?      |
| Exclamation mark       | !      |
| Quotation mark         | "      |
| Percentage sign        | %      |
| Ampersand              | &      |
| Asterisk               | *      |
| Semi-colon             | ;      |
| Less-than sign         | <      |
| Greater-than sign      | >      |

Information separator IS 4 segment terminator  
 Information separator IS 3 data element separator  
 Information separator IS 1 component data element separator

## 6 Structures

### 6.1 Interchange structure

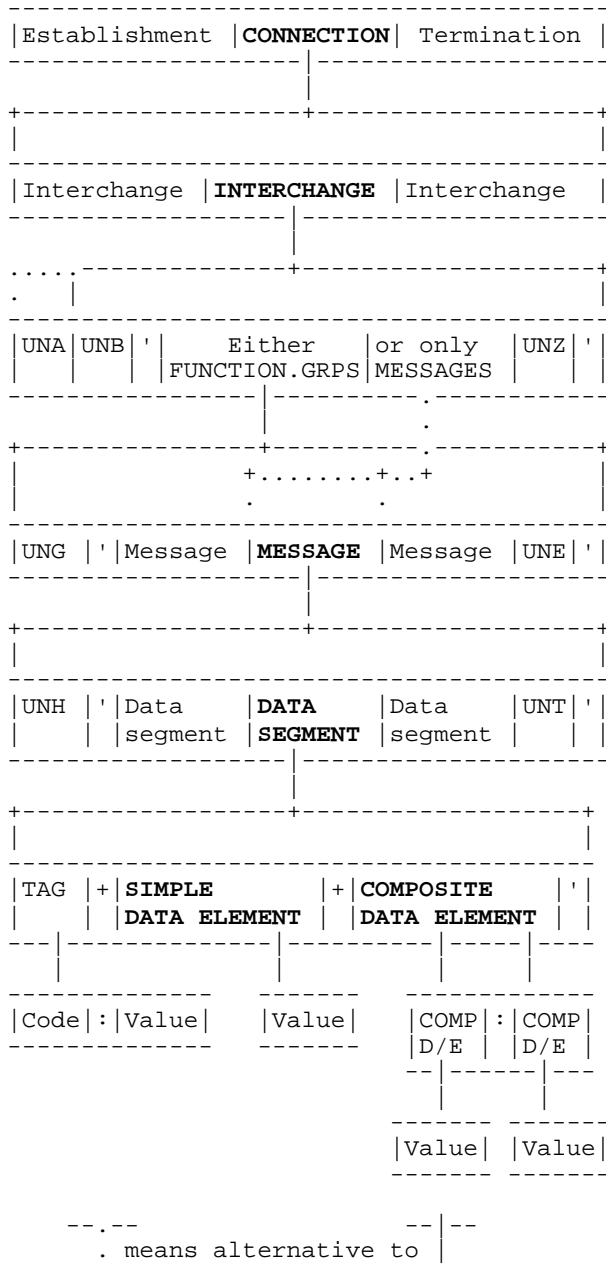
The Service String Advice, UNA, and the service segments UNB to UNZ shall appear in the order stated below in an interchange. There may be several functional groups or messages within an interchange and several messages in a functional group. A message consists of segments. The structures for segments and for data elements therein are shown in 6.2 and 6.3. The contents of the service segments are shown in annex B. See also figure 1.

An interchange consists of:

|       |                          |             |             |
|-------|--------------------------|-------------|-------------|
| ----- | Service String Advice    | UNA         | Conditional |
| ----- | Interchange Header       | UNB         | Mandatory   |
| ----- | Functional Group Header  | UNG         | Conditional |
| ----- | Message Header           | UNH         | Mandatory   |
|       | User Data Segments       | As required |             |
| ----- | Message Trailer          | UNT         | Mandatory   |
| ----- | Functional Group Trailer | UNE         | Conditional |
| ----- | Interchange Trailer      | UNZ         | Mandatory   |

In addition to the above service segments, the service segment UNS can, when required, be used to divide a message into sections. See annex B.

**ISO 9735: 1988 (E)**  
**(First edition 1988-07-15)**



A CONNECTION contains one or more interchanges. The technical protocols for establishment, maintenance and termination etc. are not part of this standard.

An INTERCHANGE contains:  
 - UNA, Service String Advice, if used  
 - UNB, Interchange Header  
 - Either only Functional Groups, if used, or only Messages  
 - UNZ, Interchange Trailer

A FUNCTIONAL GROUP contains:  
 - UNG, Functional Group Header  
 - Messages of the same type  
 - UNE, Functional Group Trailer

A MESSAGE contains:  
 - UNH, Message Header  
 - Data segments  
 - UNT, Message Trailer

A SEGMENT contains:  
 - A Segment TAG  
 - Simple data elements or  
 - Composite data elements or both as applicable

A SEGMENT TAG contains:  
 - A segment code and, if explicit indication, repeating and nesting value(s). See 8.1 and 9.

A SIMPLE DATA ELEMENT contains:  
 - A single data element value

A COMPOSITE DATA ELEMENT contains:  
 - Component data elements

A COMPONENT DATA ELEMENT contains:  
 - A single data element value

UNA, UNB, UNZ, UNG, UNE, UNH and UNT are Service segments, see 6.1 and annex B. In the diagram, the level A separators/terminators have been used, see 5.1.

**Figure 1 - Hierarchical structure of an interchange**

EXAMPLES illustrating segment sequences:

- a) A single message within a functional group: UNA UNB UNG UNH...Data Segments...UNT UNE UNZ
- b) A single message without functional grouping: UNA UNB UNH... Data Segments...UNT UNZ
- c) A single message without functional grouping and without UNA: UNB UNH...Data Segments...UNT UNZ

## 6.2 Order of segments and groups of segments within a message

A message structure diagram and the order of the segments following the processing rules in the ECE Message Design Guidelines is shown in annex C.

## 6.3 Segment structure

|                                   |  |
|-----------------------------------|--|
| Segment Tag, composed of          | Mandatory  |
| Segment Code                      | Mandatory component data element   |
| Component data element separator  | Conditional  |
| Nesting and repeating indication  | Conditional component data element(s)  |
| Data element separator            | Mandatory  |
| Simple or composite data elements | Mandatory or conditional as specified in the relevant segments directory;<br>see 6.4 |
| Segment Terminator                | Mandatory  |

## 6.4 Data element structure

|  |   |
|--|---|
| Simple Data Element, or<br>Composite Data Element with<br>Component data elements and<br>Component data element separators | Mandatory or conditional as specified in the relevant<br>segments directory |
| Data element separator   | Mandatory (see restriction below)   |
|  | Mandatory (see restriction below)   |

There shall be no component data element separator after the last component data element in a composite data element and no data element separator after the last data element in a segment.

## 7 Compressing

In data elements for which the Data Elements Directory specifies variable length and there are no other restrictions, insignificant character positions shall be suppressed. In the case of insignificant characters, leading zeroes and trailing spaces shall be suppressed.

Note, however, that a single zero before a decimal sign is significant (see 10.1) and that a zero may be significant (e.g. to indicate a temperature) if so stated in the data elements specification.

When compressing messages, the rules below shall be followed.

In the following examples of the rules, "Tag" represents a segment tag, "DE" a data element and "CE" a component data element. The separators in level A in 5.1 are used.

### 7.1 Exclusion of segments

Conditional segments containing no data shall be omitted (including their segment tags).

### 7.2 Exclusion of data elements by omission

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

Tag+DE+DE+++DE+DE+DE '

  |\_|\_\_\_\_\_ These two data elements are omitted

### **7.3 Exclusion of data elements by truncation**

If one or more conditional data elements at the end of a segment are omitted, the segment may be truncated by the segment terminator, i.e. contiguous trailing data element separators are not required to be transmitted.

Tag+DE+DE+++DE ' \_\_\_\_\_ Using the example from 7.2, the last two data elements have been omitted and the segment truncated

### **7.4 Exclusion of component data elements by omission**

Component data elements are identified by their given sequential positions within a composite data element. If a conditional component data element is omitted and is followed by another component data element, its given position must be represented by its component data element separator.

Tag+DE+CE:CE+CE:::CE ' |\_|\_\_\_\_\_ Two component data elements omitted in the last composite data element

### **7.5 Exclusion of component data elements by truncation**

One or more conditional component data elements at the end of a composite data element may be excluded by truncation by the data element separator or, if at the end of a segment, by the segment terminator.

Tag+DE+CE+CE ' |\_|\_\_\_\_\_ Using the example from 7.4, the last component data element in the first composite data element has been omitted and also three component data elements in the last composite data element. In both cases the composite data elements have been truncated, indicated in the first case by the data element separator and in the second case by the segment terminator.

## **8 Repetition**

### **8.1 Repetition of segments**

Within a given message type, either explicit or implicit repetition techniques shall be used and this decision shall be taken during message design. The two techniques shall not be mixed within the same message.

Indication of repetition shall either be explicit as a component data element being part of the segment tag composite data element that heads a segment (see 8.1.1 and 9. 1) or be implicitly understood from the sequence of the segments as stated in the relevant message specification (see 8.1.2).

Segments at level 0 (see annex C) shall not be repeated and their tags include no repeating indication,

Service segments (see annex B), excluding TXT, shall not be repeated and their tags include no repeating indication.

#### **8.1.1 Explicit indication of repetition**

In the segment tag, the first component data element shall be the segment code and the last of the subsequent component data elements shall indicate the incidence of repetition of the segment. See 9.1.

#### **8.1.2 Implicit indication of repetition**

The segments within a message shall appear in the order stated in the message type specification. Therefore it can be implicitly understood which segments are repeated, identified by their ordinal positions.

## 8.2 Repetition of data elements

Data elements (DE) shall not be repeated within a segment more than the number of times prescribed in the relevant segment directory. If less, the exclusion rule in 7.2 to 7.5 shall apply.

Tag+...+DE1+DE1+++... ' | | \_\_\_\_\_ 2 of prescribed up to 4 repeats of DE1 omitted.

It is, however, sometimes practical to structure repeatable elements as component data elements (CE) in composite elements, thereby allowing truncation by the data element separator. This may also apply to specified repeatable sequences of data elements, e.g. the sequence CE1:CE2:CE3.

Tag+...+CE1:CE2:CE3:CE1:CE2:CE3+... ' | \_\_\_\_\_ Truncation by the data element separator after 2 sequences.

## 9 Nesting of segments

A segment may depend on a segment on a higher hierarchical level in the message structure and consequently be nested in that segment.

Within a given message type, either explicit or implicit nesting techniques shall be used and this decision shall be taken during message design. The two techniques shall not be mixed within the same message.

Indication of nesting shall either be explicit as component data elements being part of the segment tag composite data element that heads a segment (see 9.1) or be implicitly understood from the sequence of the segments as stated in the relevant message specification<sup>\*)</sup> (see 9.2).

Service segments (see annex B) and other segments at level 0 (see annex C) shall not be nested and their tags include no nesting indication.

### 9.1 Explicit Indication of nesting

In the segment tag, the first component data element shall be the segment code and be followed by conditional component data elements indicating both the level and the incidence of repetition of the segment as stated in 8.1.1.

The number of component data elements used for this purpose depends upon the hierarchical level in which the segment appears in the message structure diagram. See annex C. After the segment code, the next component data element (which is for the first control count) shall be used if the segment appears at level one, the second as well if it appears at level two, the third as well at level 3, etc.

When a conditional segment on a higher level is not used in an application, the level indication shall show component data element separators for the levels not used and the segment shall appear before segments which include an indication at that level. See examples below.

---

<sup>\*)</sup> See ECE Message Design Guidelines





## 9.2 Implicit nesting indication

The order of the segments specified in the message structure diagram (top to bottom, left to right) shall be followed strictly. Thereby the nesting relation between the segments is implicitly evident and no further indication is required for processing.

## 10 Representation of numeric data element values

### 10.1 Decimal sign

The ISO representation for decimal mark is the comma (,) but a point on the line (.) is allowed. See ISO 31/0-1981. Both these characters are part of the Level A and B sets in clause 5 and both alternatives are allowed.

When the Service String Advice, UNA, is used, its third character specifies the one character used in the interchange to represent a decimal sign and thus overrides the above alternative use.

The decimal sign shall not be counted as a character of the value when computing the maximum field length of a data element. However, allowance has to be made for the character in transmission and reception.

When a decimal sign is transmitted, there shall be at least one digit before and after the decimal sign. For values represented by integers only, neither a decimal sign nor decimal zeroes are used unless there is a need to indicate the degree of precision.

Preferred: 0,5 and 2 and 2,0            Not allowed: ,5 or .5 or 2, or 2.

Allowed: 0.5 and 2 and 2.0

### 10.2 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

### 10.3 Sign

Numeric data element values shall be regarded as positive. Although conceptually a deduction is negative, it shall be represented by a positive value and such cases shall be indicated in the data elements directory.

If a value is to be indicated as negative, it shall in transmission be immediately preceded by a minus sign, e.g. -112

The minus sign shall not be counted as a character of the value when computing the maximum field length of a data element. However, allowance has to be made for the character in transmission and reception.

## **Annex A**

**(normative)**

### **Definitions**

Some terms in this International Standard have been defined in other ISO standards and have been included for the benefit of the reader. The responsibility for these terms, indicated by the number of the standard, rests with the committee concerned. In cases where the definitions pertinent for this standard are restrictions of such terms with a wider concept, the indication "EDIFACT" is used.

- A.1 alphabetic character set:** A character set that contains letters and may contain control characters and special characters but not digits (ISO 2382/4)
- A.2 alphanumeric character set:** A character set that contains both letters and digits and may contain control characters and special characters (ISO 2382/4)
- A.3 application message type:** A basic message type adapted to suit a certain application area
- A.4 character set:** A finite set of different characters that is considered complete for a given purpose (ISO 2382/4)
- A.5 common access reference:** Key to relate all subsequent transfers of data to the same business file
- A.6 component data element:** A simple data element which is a subordinate portion of a composite data element and in interchange identified by its position within the composite data element
- A.7 component data element separator:** A character used to separate the component data elements in a composite data element
- A.8 composite data element:** A data element containing two or more component data elements
- A.9 conditional:** A statement in a segment or message directory of a condition for the use of a segment, a data element, a composite data element or a component data element (cf. mandatory)
- A.10 connection:** An established link for transmission of data
- A.11 data:** A representation of facts, concepts or instructions in a formalized manner suitable for communication, interpretation or processing by human beings or by automatic means (ISO 2382/1)
- A.12 data element:** A unit of data which in a certain context is considered indivisible (ISO 2382/4)  
EDIFACT: A unit of data for which the identification, description and value representation have been specified
- A.13 data elements directory:** A listing of identified, named and described data element attributes, with specifications as to how the corresponding data element values shall be represented
- A.14 data element name:** One or more words in a natural language identifying a data element concept
- A.15 data element separator:** A character used to separate data elements in a segment
- A.16 data element tag:** A unique identifier for a data element in a data elements directory
- A.17 data element value:** The specific entry of an identified data element represented as specified in a data elements directory
- A.18 functional group:** One or more messages of the same type headed by a functional group header service segment and ending with a functional group trailer service segment
- A.19 functional group header:** The service segment heading and identifying the functional group

- A.20 functional group trailer:** The service segment ending a functional group
- A.21 identifier:** A character or group of characters used to identify or name an item of data and possibly to indicate certain properties of that data (ISO 2382/4)
- A.22 interchange:** Communication between partners in the form of a structured set of messages and service segments starting with an interchange control header and ending with an interchange control trailer
- A.23 interchange control header:** The service segment starting and identifying an interchange
- A.24 interchange control trailer:** The service segment ending an interchange
- A.25 mandatory:** A statement in a segment or message directory which specifies that a segment, a data element, a composite data element or a component data element must be used (cf. conditional)
- A.26 message:** An ordered series of characters intended to convey information (ISO 2382/16)  
EDIFACT: A set of segments in the order specified in a Message directory starting with the Message header and ending with the Message trailer
- A.27 message directory:** A listing of identified, named, described and specified message types
- A.28 message header:** The service segment starting and uniquely identifying a message
- A.29 message trailer:** The service segment ending a message
- A.30 message type:** An identified and structured set of data elements covering the requirements for a specified type of transaction, e.g. invoice.
- A.31 nested segment:** A segment which directly relates to an other segment in an identified and structured group of segments covering the requirements for a specific message type
- A.32 numeric character set:** A character set that contains digits and may contain control characters and special characters but not letters (ISO 2382/4)
- A.33 omission:** Exclusion in an actual message of one or more units of data which are defined as conditional in a message type specification
- A.34 qualifier:** A data element whose value shall be expressed as a code that gives specific meaning to the function of another data element or a segment
- A.35 release character:** A character used to restore to its original meaning any character used as a syntactical separator
- A.36 repeating segment:** A segment which may repeat in a message as specified in the relevant message type specification
- A.37 segment:** A predefined and identified set of functionally related data elements values which are identified by their sequential positions within the set. A segment starts with a segment tag and ends with a segment terminator. It can be a service segment or a user data segment
- A.38 segment code:** A code which uniquely identifies each segment as specified in a segment directory
- A.39 segment tag:** A composite data element, in which the first component data element contains a code which uniquely identifies a segment as specified in the relevant segment directory. Additional component data elements can be conditionally used to indicate the hierarchical level and nesting relation in a message and the incidence of repetition of the segment
- A.40 segment terminator:** A syntax character indicating the end of a segment
- A.41 segments directory:** A listing of identified, named, described and specified segments

**ISO 9735: 1988 (E)**  
**(First edition 1988-07-15)**

- A.42 separator character:** A character used for syntactical separation of data
- A.43 service data element:** A data element used in service segments
- A.44 service segment:** A segment required to service the interchange of user data
- A.45 service string advice:** A character string at the beginning of an interchange defining the syntactically delimiting characters and indicators used in the interchange
- A.46 simple data element:** A data element containing a single value
- A.47 syntax rules:** Rules governing the structure of an interchange and its functional groups, messages, segments and data elements
- A.48 transfer:** A communication from one partner to another
- A.49 user data segment:** A segment containing application data

**ANNEX B**  
(normative)

**Service segments specifications**

The full description of the data elements in the service segments is part of ISO 7372 Trade Data Elements Directory (UNTDED)<sup>\*)</sup>.

Legend:

|         |   |
|---------|---|
| Ref.    | The numeric reference tag for the data element as stated in ISO 7372/UNTDED and, when preceded by S, reference for a composite data element used in service segments  |
| Name    | Name of COMPOSITE DATA ELEMENT in capital letters<br>Name of DATA ELEMENT in capital letters<br>Name of Component data element in small letters   |
| Repr.   | Data value representation:<br>a alphabetic characters<br>n numeric characters<br>an alpha-numeric characters<br>a3 3 alphabetic characters, fixed length<br>n3 3 numeric characters, fixed length<br>an3 3 alpha-numeric characters, fixed length<br>a..3 up to 3 alphabetic characters<br>n..3 up to 3 numeric characters<br>an..3 up to 3 alpha-numeric characters<br><br>M Mandatory element<br>C Conditional element.<br><br>Note that a mandatory component data element in a conditional composite data element must appear when the composite data element is used |
| Remarks | IA Interchange Agreement between interchanging partners   |

---

<sup>\*)</sup> Service data elements is presently an ECE Draft

**ISO 9735: 1988 (E)**  
**(First edition 1988-07-15)**

**UNA, Service String advice**

Function: To define the characters selected for use as delimiters and indicators in the rest of the interchange that follows:

The specifications in the Service string advice take precedence over the specifications for delimiters etc. in segment UNB. See clause 4.

When transmitted, the Service string advice must appear immediately before the Interchange Header (UNB) segment and begin with the upper case characters UNA immediately followed by the six characters selected by the sender to indicate, in sequence, the following functions:

| <u>Repr.</u> |   | <u>Name</u>                      | <u>Remarks</u>                      |
|--------------|---|----------------------------------|-------------------------------------|
| an1          | M | COMPONENT DATA ELEMENT SEPARATOR |                                     |
| an1          | M | DATA ELEMENT SEPARATOR           |                                     |
| an1          | M | DECIMAL NOTATION                 | Comma or full stop                  |
| an1          | M | RELEASE INDICATOR                | If not used, insert space character |
| an1          | M | Reserved for future use          | Insert space character              |
| an1          | M | SEGMENT TERMINATOR               |                                     |

Segment : **UNB, Interchange Header**

Function: To start, identify and specify an interchange

| <u>Ref.</u> | <u>Repr.</u> | <u>Name</u>                                 | <u>Remarks</u>   |
|-------------|--------------|---|--|
| S001        |              | M SYNTAX IDENTIFIER                         |  |
| 0001        | a4           | M Syntax identifier                         | a3, upper case, Controlling Agency (e.g. UNO = UN/ECE) and a1 stating level (e.g. A) (which together give UNOA)                                  |
| 0002        | n1           | M Syntax version number                     | Increments 1 for each new version.   |
| <hr/>       |              |   |  |
| S002        |              | M INTERCHANGE SENDER                        |  |
| 0004        | an..35       | M Sender identification                     | Code or name as specified in IA  |
| 0007        | an..4        | C Identification code qualifier             | Used with sender identification code   |
| 0008        | an..14       | C Address for reverse routing               |  |
| <hr/>       |              |   |  |
| S003        |              | M INTERCHANGE RECIPIENT                     |  |
| 0010        | an..35       | M Recipient Identification                  | Code or name as specified in IA  |
| 0007        | an..4        | C Identification code qualifier             | Used with recipient identification code  |
| 0014        | an..14       | C Routing address                           | If used, normally coded sub-address for onward routing   |
| <hr/>       |              |   |  |
| S004        |              | M DATE/TIME OF PREPARATION                  |  |
| 0017        | n6           | M Date                                      | YYMMDD   |
| 0019        | n4           | M Time                                      | HHMM   |
| <hr/>       |              |   |  |
| 0020        | an..14       | M INTERCHANGE CONTROL REFERENCE             | Unique reference assigned by sender  |
| <hr/>       |              |   |  |
| S005        |              | C RECIPIENTS REFERENCE, PASSWORD            |  |
| 0022        | an..14       | M Recipient's reference/password            | As specified in IA. May be password to recipient's system or to third party network  |
| 0025        | an2          | C Recipient's reference/ password qualifier | If specified in IA   |
| <hr/>       |              |   |  |
| 0026        | an..14       | C APPLICATION REFERENCE                     | Optionally message identification if the interchange contains only one type of message   |
| <hr/>       |              |   |  |
| 0029        | a1           | C PROCESSING PRIORITY CODE                  | Used if specified in IA  |
| <hr/>       |              |   |  |
| 0031        | n1           | C ACKNOWLEDGEMENT REQUEST                   | Set = 1 if sender requests acknowledgement, i.e. UNB and UNZ segments received and identified  |
| <hr/>       |              |   |  |
| 0032        | an..35       | C COMMUNICATIONS AGREEMENT ID               | If used, to identify type of communication agreement controlling the interchange, e.g. Customs or ECE agreement. Code or name as specified in IA |
| <hr/>       |              |   |  |
| 0035        | n1           | C TEST INDICATOR                            | Set = 1 if the interchange is a test. Otherwise not used   |
| <hr/>       |              |   |  |

**ISO 9735: 1988 (E)**  
**(First edition 1988-07-15)**

Segment : **UNZ, Interchange Trailer**

Function: To end and check the completeness of an interchange

| Ref. | Repr.  | Name                            | Remarks  |
|------|--------|---------------------------------|--|
| 0036 | n..6   | M INTERCHANGE CONTROL COUNT     | The count of the number of messages or, if used, the number of functional groups in the interchange. One of these counts shall appear. |
| 0020 | an..14 | M INTERCHANGE CONTROL REFERENCE | Shall be identical to 0020 in UNB  |

Segment : **UNG, Functional Group Header**

Function: To head, identify and specify a Functional Group

| Ref. | Repr.  | Name                                    | Remarks   |
|------|--------|---|---|
| 0038 | an..6  | M FUNCTIONAL GROUP IDENTIFICATION       | Identifies the one message type in the functional group   |
| S006 |        | M APPLICATION SENDER'S IDENTIFICATION   |   |
| 0040 | an..35 | M Sender identification                 | Code or name identifying the division, department etc. within the originating sender's organization                                   |
| 0007 | an..4  | C Sender identification qualifier       | May be used if sender identification is a code  |
| S007 |        | M APPLICATION RECIPIENTS IDENTIFICATION |   |
| 0044 | an..35 | M Recipient's identification            | Code or name identifying the division, department etc. within the recipients organization for which the group of messages is intended |
| 0007 | an..4  | C Recipient's identification qualifier  | May be used if recipient identification is a code   |
| S004 |        | M DATE/TIME OF PREPARATION              |   |
| 0017 | n6     | M Date                                  | YYMMDD  |
| 0019 | n4     | M Time                                  | HHMM  |
| 0048 | an..14 | M FUNCTIONAL GROUP REFERENCE NUMBER     | Unique reference number assigned by sender's division, department etc.  |
| 0051 | an..2  | M CONTROLLING AGENCY                    | Code to identify the agency controlling the specification, maintenance and publication of the message type                            |
| S008 |        | M MESSAGE VERSION                       |   |
| 0052 | n..3   | M Message version number                | Version number of the message type in the functional group  |
| 0054 | n..3   | C Message release number                | Release number within current version number  |
| 0057 | an..6  | C Association assigned Code             | A code assigned by the association responsible for the design and maintenance of the type of message concerned                        |
| 0058 | an..14 | C APPLICATION PASSWORD                  | Password to recipient's division, department or sectional system (if required)  |



**Segment : UNE, Functional Group Trailer**

Function: To end and check the completeness of a Functional Group

| <u>Ref.</u> | <u>Repr.</u> | <u>Name</u>                         | <u>Remarks</u>  |
|-------------|--------------|-------------------------------------|---|
| 0060        | n..6         | M NUMBER OF MESSAGES                | The count of the number of messages in the functional group |
| 0048        | an..14       | M FUNCTIONAL GROUP REFERENCE NUMBER | Shall be identical to 0048 in UNG                           |

**Segment : UNH, Message Header**

Function: To head, identify and specify a Message

| <u>Ref.</u> | <u>Repr.</u> | <u>Name</u>                               | <u>Remarks</u>   |
|-------------|--------------|---|--|
| 0062        | an..14       | M MESSAGE REFERENCE NUMBER                | A sender's unique message reference  |
| S009        |              | M MESSAGE IDENTIFIER                      |  |
| 0065        | an..6        | M Message type                            | Type of message being transmitted  |
| 0052        | n..3         | M Message version number                  | Version number of the message type. If UNG is used, 0052 shall be identical  |
| 0054        | n..3         | C <sup>*)</sup> Message release number    | Release number within current version number   |
| 0051        | an..2        | C <sup>*)</sup> Controlling agency        | Code to identify the agency controlling the specification, maintenance and publication of the message type                                       |
| 0057        | an..6        | C <sup>*)</sup> Association assigned code | A code assigned by the association responsible for the design and maintenance of the message type  |
| 0068        | an..35       | C COMMON ACCESS REFERENCE                 | Key to relate all subsequent transfers of data to the same business case or file. Within the 35 characters the IA may specify component elements |
| S010        |              | C STATUS OF THE TRANSFER                  |  |
| 0070        | n..2         | M Sequence of transfers                   | Starts at 1 and is incremented by 1 for each transfer  |
| 0073        | a1           | C First and last transfer                 | C = Creation, must be present for first transfer if more than one foreseen<br>F = Final, must be present for last transfer                       |

<sup>\*)</sup> Not required if provided in UNG

**ISO 9735: 1988 (E)**  
**(First edition 1988-07-15)**

Segment : **UNT, Message Trailer**

Function: To end and check the completeness of a Message

| <u>Ref.</u> | <u>Repr.</u> | <u>Name</u>                         | <u>Remarks</u>                      |
|-------------|--------------|-------------------------------------|-------------------------------------|
| 0074        | n..6         | M NUMBER OF SEGMENTS IN THE MESSAGE | Control count including UNH and UNT |
| 0062        | an..14       | M MESSAGE REFERENCE NUMBER          | Shall be identical to 0062 in UNH   |

Segment : **TXT, Text**

Function: To give information in addition to that in other segments in the service message, as required

NOTE: Can not be machine processed. Should be avoided if not necessarily required. Normally a conditional segment. It may repeat up to the number of times indicated in the message specification which may not be higher than 5.

| <u>Ref.</u> | <u>Repr.</u> | <u>Name</u>           | <u>Remarks</u>   |
|-------------|--------------|-----------------------|--|
| 0077        | an3          | C TEXT REFERENCE CODE | Qualifies and identifies the purpose and function of the segment if indicated in the message specification |
| 0078        | an..70       | M FREE FORM TEXT      | Not machine-processable information  |

Segment : **UNS, Section Control**

Function: To separate Header, Detail and Summary sections of a message

NOTE: To be used by message designers when required to avoid ambiguities. Mandatory only if specified for the type of message concerned.

| <u>Ref.</u> | <u>Repr.</u> | <u>Name</u>              | <u>Remarks</u>  |
|-------------|--------------|--------------------------|---|
| 0081        | a1           | M SECTION IDENTIFICATION | Separates sections in a message by one of the following codes:<br>D separates the header and detail sections<br>S separates the detail and summary sections |

**Annex C**  
 (informative)

**Order of segments and groups of segments within a message**

The segments used in a message shall appear in the sequence (top to bottom, left to right) specified in the Message Diagram.

Segments are indicated by their codes. The requirement for their inclusion in the message, i.e. their status, is indicated directly below the codes by the letter M for mandatory or C for conditional. The number of times a segment may appear in each instance is indicated directly thereafter. A mandatory segment shall appear at least once but not more times than indicated. A conditional segment may be excluded or appear up to the number of times indicated.

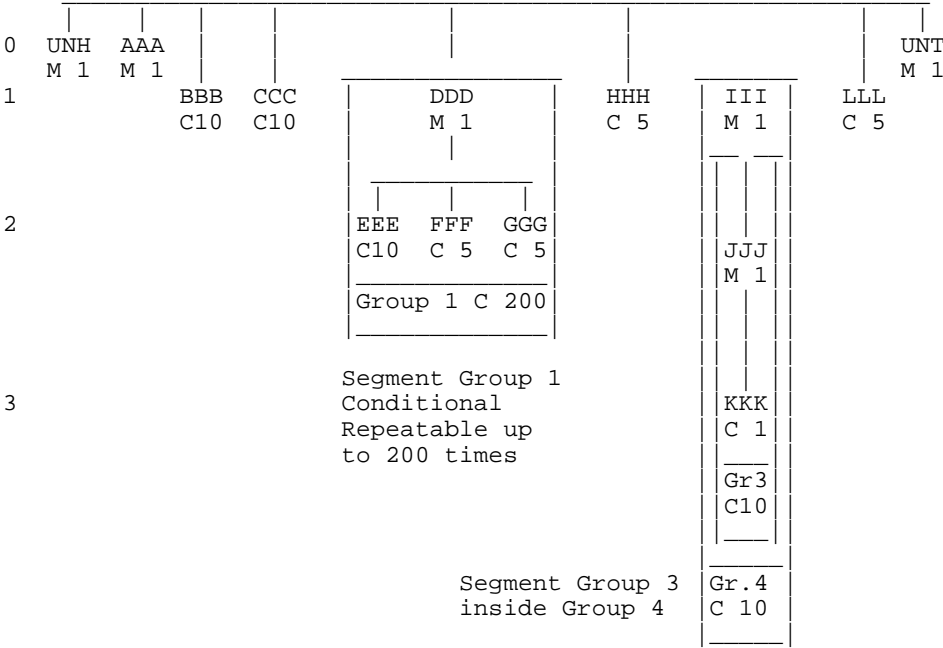
When a segment nests in another segment, it shall be placed on the next lower level in the diagram. Segments in level zero shall not be repeated and shall not contain nested segments.

Two or more segments can be grouped. This is indicated by a box in the diagram. The group and the segments in the box can be mandatory or conditional and can appear up to the number of times indicated. A group can contain another, lower level group or groups (Gr.3 and Gr.4 in the example).

A message shall begin with the message header segment UNH and end with the message trailer segment UNT.

EXAMPLE: Parts of a fictitious message type:

Level



Segments may alternatively be represented as follows:



The processing/sequencing order of the segments is as follows (Group 1 appearing twice, the other groups once and segments not repeated):

UNH,AAA,BBB,CCC,DDD,EEE,FFF,GGG,DDD,EEE,FFF,GGG,HHH,...,III,JJJ,KKK,...,LLL,UNT