
**Joint ISO/TC 154 – UN/CEFACT
Syntax Working Group (JSWG)
publication of ISO 9735-1**

**equivalent to the official ISO publication:
ISO 9735-1** (Second edition 2002-07-01)

**Electronic data interchange for
administration, commerce and transport
(EDIFACT) — Application level syntax rules
(Syntax version number: 4, Syntax release
number: 1) —**

Part 1:
Syntax rules common to all parts

Contents

Page

Foreword.....	v
Introduction.....	vii
1 Scope	1
2 Conformance.....	1
3 Normative references	2
4 Terms and definitions	2
5 Service characters.....	13
5.1 General.....	13
5.2 Default service characters	13
5.3 UNA, service string advice	14
6 Character repertoires	14
7 Syntax structures	15
7.1 General.....	15
7.2 Interchange structure.....	15
7.3 Group structure	15
7.4 Message structure	15
7.5 Segment group structure.....	15
7.6 Segment structure	16
7.7 Segment tag structure.....	16
7.8 Composite data element structure	16
7.9 Simple data element structure	16
7.10 Package structure.....	16
8 Inclusion and exclusion	16
8.1 General.....	16
8.2 Determination of presence	17
8.3 Inclusion of segment groups.....	17
8.4 Exclusion of segment groups	17
8.5 Inclusion of segments.....	17
8.6 Exclusion of segments.....	17
8.7 Inclusion of data elements.....	17
8.8 Exclusion of data elements	18
9 Suppression of characters within data elements.....	19
9.1 General.....	19
9.2 Insignificant characters	19
9.3 Significant zeroes	20
9.4 Significant spaces	20
10 Representation of numeric data element values.....	20
11 Dependency notes	20
11.1 General.....	20
11.2 Dependency notes in the message specification	20
11.3 Dependency notes in the segment specification	21
11.4 Dependency notes in the composite data element specification.....	21
11.5 Notation for dependency notes.....	21
12 Segment collision prevention	22
13 Syntax release identification	22

Annex A (normative) UNA service string advice	23
Annex B (informative) Order of segments and groups of segments within a message	24
Annex C (informative) The use of anti-collision segment group UGH/UGT	27
Annex D (informative) Syntax release identification	30

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO 9735 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 9735-1 was prepared by Technical Committee ISO/TC 154, *Processes, data elements and documents in commerce, industry and administration* in collaboration with UN/CEFACT through the Joint Syntax Working Group (JSWG).

This second edition cancels and replaces the first edition (ISO 9735-1:1998) to which one new feature to the EDIFACT syntax, version 4 has been added: Syntax release identification. However ISO 9735:1988 and its Amendment 1:1992 are provisionally retained for the reasons given in clause 2.

Furthermore, for maintenance reasons the Syntax service directories have been removed from this and all other parts of ISO 9735. They are now consolidated in a new part, ISO 9735-10.

At the time of publication of ISO 9735-1:1998, ISO 9735-10 had been allocated as a part for "Security rules for interactive EDI". This was subsequently withdrawn because of lack of user support, and as a result, all relevant references to the title "Security rules for interactive EDI" have been removed in this second edition of ISO 9735-1.

Definitions from all parts of the ISO 9735 series have been consolidated and included in this part of ISO 9735.

The Introduction has also been updated to summarize the new feature and all other changes.

Together with ISO 9735-2, this part of ISO 9735 is an enhancement of ISO 9735:1988 and its Amendment 1:1992.

ISO 9735 consists of the following parts, under the general title *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1)*:

- *Part 1: Syntax rules common to all parts*
- *Part 2: Syntax rules specific to batch EDI*
- *Part 3: Syntax rules specific to interactive EDI*
- *Part 4: Syntax and service report message for batch EDI (message type — CONTRL)*
- *Part 5: Security rules for batch EDI (authenticity, integrity and non-repudiation of origin)*
- *Part 6: Secure authentication and acknowledgement message (message type — AUTACK)*

ISO 9735-1:2002

- *Part 7: Security rules for batch EDI (confidentiality)*
- *Part 8: Associated data in EDI*
- *Part 9: Security key and certificate management message (message type — KEYMAN)*
- *Part 10: Syntax service directories*

Further parts may be added in the future.

Annex A forms a normative part of this part of ISO 9735. Annexes B, C and D are for information only.

Introduction

This part of ISO 9735 includes the rules at the application level for the structuring of data in the interchange of electronic messages in an open environment, based on the requirements of either batch or interactive processing. These rules have been agreed by the United Nations Economic Commission for Europe (UN/ECE) as syntax rules for Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) and are part of the United Nations Trade Data Interchange Directory (UNTDID) which also includes both batch and interactive Message Design Guidelines.

This part of ISO 9735 may be used in any application, but messages using these rules may only be referred to as EDIFACT messages if they comply with other guidelines, rules and directories in the UNTDID. For UN/EDIFACT messages, the message design rules for batch or interactive usage apply, as appropriate. These rules are maintained in the UNTDID.

Communications specifications and protocols are outside the scope of this part of ISO 9735.

A previous version of ISO 9735 was published in 1988 as a single part. The current version of ISO 9735 consists of multiple parts and incorporates enhancements to extend its application.

This part of ISO 9735 is a re-draft of corresponding sections in the previous version of ISO 9735. It consists of the rules common to all parts of ISO 9735, and includes the definitions for all parts.

The basic syntax rules specified in this part remain unchanged from the previous version, with the exception that the coverage of character repertoires has been extended, and two new techniques have been introduced (the provision for “dependency notes” and the introduction of a service repetition character, to support the capability of permitting multiple occurrences (repeats) of stand-alone and/or composite data elements). Both of these techniques are used in other parts of the current version of ISO 9735, and are available for specification in EDIFACT messages which utilize this International Standard.

In addition, enhancements have been made to the batch interchange; group; and message header segments (UNB; UNG; and UNH).

Character repertoires: Because of the widening use of ISO 9735, it has become necessary to extend its coverage to include all character repertoires covered by ISO 8859, parts 1-9; the code extension techniques covered by ISO 2022 (with certain restrictions on its use within an interchange); and partial use of the techniques covered by ISO/IEC 10646-1.

Dependency notes: These provide a formal notation to express relationships in EDIFACT message, segment and composite data element specifications.

Repeating data elements: The specification of multiple occurrences of a message within a group or within an interchange; a group within an interchange; and a segment group and/or a segment within a message, which existed in the previous version of ISO 9735, has been extended in the current version. The additional capability for the specification of multiple occurrences of a stand-alone data element and/or of a composite data element within a segment has been introduced.

UNB - Interchange header segment: This segment has been enhanced to permit the identification of the service code list directory version number; identification of the character encoding scheme; and internal sub-identification of the sender and recipient. In addition, to conform to year 2000 requirements, the date format in this segment has been extended.

UNG - Group header segment: This segment has been renamed and its function changed to permit one or more message types and/or packages to be contained in the group. As a result, certain data elements, which are now redundant, have been marked for deletion. In addition, to conform to year 2000 requirements, the date format in this segment has been extended.

UNH - Message header segment: This segment has been enhanced to permit the identification of a message subset; of a related message implementation guideline; and of a related scenario.

Segment collision prevention: An addition has been made to permit the prevention of collision, by use of the UGH/UGT segment group. This technique shall be used in a message specification when it is not otherwise possible to ensure unambiguous identification of each message segment upon receipt.

Syntax release identification: An addition has been made to permit the identification of specific releases related to the syntax version number. This will facilitate the publication of minor changes to the standard (if required in the future).

Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) —

Part 1: Syntax rules common to all parts

1 Scope

This part of ISO 9735 specifies common syntax rules for the formatting of batch and interactive messages to be interchanged between computer application systems. It includes the terms and definitions for all parts of ISO 9735.

2 Conformance

Whereas this part shall use a version number of “4” in the mandatory data element 0002 (Syntax version number), and shall use a release number of “01” in the conditional data element 0076 (Syntax release number), each of which appear in the segment UNB (Interchange header), interchanges continuing to use the syntax defined in the earlier published versions shall use the following Syntax version numbers, in order to differentiate them from each other and from this part:

- ISO 9735:1988: *Syntax version number: 1*
- ISO 9735:1988 (amended and reprinted in 1990): *Syntax version number: 2*
- ISO 9735:1988 and its Amendment 1:1992: *Syntax version number: 3*
- ISO 9735:1998: *Syntax version number: 4*

Conformance to a standard means that all of its requirements, including all options, are supported. If all options are not supported, any claim of conformance shall include a statement which identifies those options to which conformance is claimed.

Data that is interchanged is in conformance if the structure and representation of the data conforms to the syntax rules specified in this part of ISO 9735.

Devices supporting this part of ISO 9735 are in conformance when they are capable of creating and/or interpreting the data structured and represented in conformance with the standard.

Conformance shall be based on this part of ISO 9735, on ISO 9735-10, and at least either ISO 9735-2 or ISO 9735-3.

When identified in this part of ISO 9735, provisions defined in related standards shall form part of the conformance criteria.

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 9735. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 9735 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 646:1991, *Information technology — ISO 7-bit coded character set for information interchange*

ISO/IEC 2022:1994, *Information technology — Character code structure and extension techniques*

ISO/IEC 2382-1:1993, *Information technology — Vocabulary — Part 1: Fundamental terms*

ISO/IEC 2382-4:1999, *Information technology — Vocabulary — Part 4: Organization of data*

ISO 6093:1985, *Information processing — Representation of numerical values in character strings for information interchange*

ISO/IEC 6429:1992, *Information technology — Control functions for coded character sets*

ISO/IEC 6523-1:1998, *Information technology — Structure for the identification of organizations and organization parts — Part 1: Identification of organization identification schemes*

ISO 7498-2:1989, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture*

ISO/IEC 9594-8:1998, *Information technology — Open Systems Interconnection — The Directory: Authentication framework*

ISO 9735-2:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 2: Syntax rules specific to batch EDI*

ISO 9735-3:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 3: Syntax rules specific to interactive EDI*

ISO 9735-10:2002, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 10: Syntax service directories*

ISO/IEC 10646-1:2000, *Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane*

ISO/IEC 11770-1:1996, *Information technology — Security techniques — Key management — Part 1: Framework*

ITU-T Recommendation F.400/X.400:1999, *Message handling system and service overview*

4 Terms and definitions

For the purposes of all parts of ISO 9735, the following terms and definitions apply.

NOTE 1 When a word or phrase appears in italics within a definition, this means that a definition for this term is given in this clause.

NOTE 2 The terms are classified alphabetically; an identifier is added at the end of each definition, in square brackets, to facilitate the comparison between different linguistic versions. For example the English term “Alphabetic character set” is called in French “Jeu de caractères alphabétiques”, and will not appear at the same alphabetic place in the two versions of the syntax; the identifier in brackets nevertheless remains “[1]”.

4.1**alphabetic character set**

character set that contains letters and/or *ideograms*, and may contain other *graphic characters* except digits [1]

4.2**alphanumeric character set**

character set that contains letters, digits and/or *ideograms*, and may contain other *graphic characters* [2]

4.3**asymmetric algorithm**

cryptographic algorithm employing a *public key* and a *private key* which together form an asymmetric key set [3]

4.4**attribute**

characteristic of an entity [4]

4.5**authentication**

See *data origin authentication* [5]

4.6**batch EDI**

electronic data interchange in which no strong requirements exist for formalized data exchange using query and response between the parties [6]

4.7**business**

series of processes, each having a clearly understood purpose, involving more than one *organization*, realized through the exchange of information and directed towards some mutually agreed upon goal, extending over a period of time [7]

4.8**certificate**

public key of a user, together with some other information, rendered unforgeable by a signature with the *private key* of the *certification authority* which issued it [8]

[ISO/IEC 9594-8:1998, 3.3.3]

4.9**certification authority**

authority trusted by one or more users to create and assign *certificates* [9]

[ISO/IEC 9594-8:1998, 3.3.8]

4.10**certification path**

ordered sequence of *certificates* of objects in the Directory Information Tree which, together with the *public key* of the initial object in the path, can be processed to obtain that of the final object in the path [10]

[ISO/IEC 9594-8:1998, 3.3.9]

4.11**character**

member of a set of elements used for the organization, control, or representation of *data* [11]

[ISO/IEC 10646-1:2001, 4.6]

4.12

character repertoire

set of *graphic characters* of a *coded character set*, considered independently of its encoding [12]

4.13

code extension

techniques for the *encoding* of *characters* that are not included in the *character repertoire* of a given *coded character set* [13]

4.14

code list

complete set of *data element values* of a coded *simple data element* [14]

4.15

code list directory

listing of identified and specified *code lists* [15]

4.16

coded character set

set of unambiguous rules that establishes a *character set* and the one-to-one relationship between the *characters* of the set and their bit combinations [16]

[ISO/IEC 6429:1992]

4.17

component data element

simple data element used within a *composite data element* [17]

4.18

component data element separator

service character used to separate the *component data elements* within a *composite data element* [18]

4.19

composite data element

identified, named and structured set of functionally related *component data elements*, as described in a *composite data element specification* [19]

NOTE In *transfer*, a composite data element is a specific ordered set of one or more *component data element(s)* in conformance with a *composite data element specification*.

4.20

composite data element directory

listing of identified and named *composite data elements* with their *composite data element specification* [20]

4.21

composite data element specification

description of a *composite data element* in a *composite data element directory*, including the specification of the position and *status* of the *component data elements* constituting the *composite data element* [21]

4.22

conditional

type of *status*, used in a *message specification*, *segment specification*, or *composite data element specification*, to specify that a *segment group*, *segment*, *composite data element*, *stand-alone data element* or *component data element* is used optionally or when the appropriate conditions occur [22]

4.23

confidentiality

property that information is not made available or disclosed to unauthorized individuals, entities or processes [23]

[ISO 7498-2:1989, 3.3.16]

4.24**control character**

a *character* whose purpose is to effect format, to control data transmission, or to perform other control functions [24]

NOTE A control character, although it is not a graphic character, may have a graphic representation.

[ISO/IEC 2382-4:1999, 04.04.01]

4.25**credential**

data that serves to establish the claimed identity of an entity [25]

[ISO 7498-2:1989, 3.3.17]

4.26**cryptography**

discipline which embodies principles, means, and methods for the transformation of data in order to hide its information content, prevent its undetected modification and/or prevent its unauthorized use [26]

[ISO 7498-2:1989, 3.3.20]

4.27**data**

reinterpretable representation of information in a formalized manner suitable for communication, interpretation or processing [27]

[ISO/IEC 2382-1:1993, 01.01.02]

4.28**data element**

unit of *data* described in a *data element specification* [28]

NOTE There are two classes of data element: *simple data elements* and *composite data elements*.

4.29**data element directory**

listing of identified, named and specified *simple data elements* (*simple data element directory*) or *composite data elements* (*composite data element directory*) [29]

4.30**data element separator**

service character used to separate from each:

- non repeating stand-alone data elements; or
- composite data elements in a segment; or
- a set of occurrences of a repeating data element; or
- a null set of occurrences of a repeating data element,
 - where a set of occurrences of a *repeating data element* is a *repeating data element* having one or more of its occurrences (up to a maximum specified number) present in a *transfer*; and
 - where a null set of occurrences of a *repeating data element* is a *repeating data element* for which none of its specified occurrences are present in a *transfer* [30]

4.31

data element specification

specification of a *composite data element* in a *composite data element directory (composite data element specification)*, or of a *simple data element* in a *simple data element directory (simple data element specification)* [31]

4.32

data element value

specific instance of a *simple data element*, represented as specified in a *simple data element specification* and, if the *simple data element* is coded, in a *code list* [32]

4.33

data integrity

property that data has not been altered or destroyed in an unauthorized manner [33]

[ISO 7498-2:1989, 3.3.21]

4.34

data origin authentication

corroboration that the source of data received is as claimed [34]

[ISO 7498-2:1989, 3.3.22]

4.35

data value representation

types of *characters* allowed (e.g. alphabetic, numeric) and conditions of length relating to the *data element values* of a *simple data element* [35]

4.36

decimal mark

character that separates the digits forming the integral part of a number from those forming the fractional part [36]

[ISO 6093:1985, 4.1]

4.37

decipherment

reversal of a corresponding reversible *encipherment* [37]

[ISO 7498-2:1989, 3.3.23]

4.38

decryption

See *decipherment* [38]

[ISO 7498-2:1989, 3.3.24]

4.39

default service characters

set of *characters* used as *service characters* in circumstances where a different set is not defined in the *service string advice* [39]

4.40

dependency identifier

identifier used in a *dependency note* to specify the type of dependency between the entities listed in the *dependency note* [40]

4.41**dependency note**

note used:

- a) in a *message specification* to express relationships between *segment groups* or between *segments*;
- b) in a *segment specification* to express relationships between *data elements*;
- c) in a *composite data element specification* to express relationships between *component data elements* [41]

4.42**dialogue**

two-way conversation between an *initiator* and *responder* within an *I-EDI transaction* [42]

NOTE It is formally composed of a pair of interchanges.

4.43**digital signature**

data appended to, or a cryptographic transformation (see *cryptography*) of, a data unit that allows a recipient of the data unit to prove the source and *integrity* of the data unit and protect against forgery, e.g. by the recipient [43]

[ISO 7498-2:1989, 3.3.26]

4.44**EDI****Electronic Data Interchange**

electronic transfer from computer application to computer application of commercial or administrative transactions using an agreed standard to structure the transaction or message data [44]

4.45**encipherment**

cryptographic transformation of data (see *cryptography*) to produce ciphertext [45]

[ISO 7498-2:1989, 3.3.27]

4.46**encoding**

representation of a *character* as a bit combination [46]

4.47**encryption**

See *encipherment* [47]

[ISO 7498-2:1989, 3.3.28]

4.48**exponent mark**

control character used to indicate that the *character(s)* that follow it are to be interpreted as an exponent [48]

NOTE "E" or "e" is the exponent mark.

4.49**filtering**

process by which octets containing arbitrary bit patterns are converted to octets belonging to the character set which the underlying syntax is capable of supporting [49]

4.50**graphic character**

character, other than a *control character*, that has a visual representation and is normally produced by writing, printing or displaying [50]

[ISO/IEC 2382-4:1999, 04.03.01]

4.51

group

group of *messages* (of one or more *message types*) and/or *packages* (each containing an *object*), headed by a *group header* and ending with a *group trailer* [51]

4.52

group header

service segment heading and identifying a *group* [52]

4.53

group trailer

service segment ending a *group* [53]

4.54

hash function

(mathematical) function which maps values from a large (possibly very large) domain into a smaller range. A “good” hash function is such that the results of applying the function to a (large) set of values in the domain will be evenly distributed (and apparently at random) over the range [54]

[ISO/IEC 9594-8:1998, 3.3.14]

4.55

I-EDI

Interactive EDI

exchange of predefined and structured data within a *dialogue*, in accordance with the syntax specified in ISO 9735-1 and ISO 9735-3 for some *business purpose*, between a pair of co-operating processes, in a timely manner [55]

4.56

I-EDI transaction

instance of a *scenario*, consisting of one or more *dialogues* [56]

4.57

identifier

character or group of *characters* used to identify or name an item of *data* and possibly to indicate certain properties of that *data* [57]

4.58

ideogram

(in a natural language) *graphic character* that represents a concept and associated sound elements [58]

EXAMPLES A Chinese ideogram or a Japanese Kanji.

4.59

initiator

application which starts the *dialogue* and/or *I-EDI transaction* [59]

4.60

integrity

See *data integrity* [60]

4.61

interchange

sequence of *messages* and/or *packages*, of the same or of different types, starting with the *interchange header* (or with the *service string advice* if used), and ending with the *interchange trailer* [61]

4.62

interchange header

service segment starting and uniquely identifying an *interchange* [62]

4.63**interchange trailer**

service segment ending an *interchange* [63]

4.64**key**

sequence of symbols that controls the operations of *encipherment* and *decipherment* [64]

[ISO 7498-2:1989, 3.3.32]

4.65**mandatory**

type of *status*, used in a *message specification*, *segment specification*, or *composite data element specification*, to specify that a *segment group*, *segment*, *composite data element*, *stand-alone data element* or *component data element* shall be used at least one time [65]

4.66**message**

identified, named and structured set of functionally related *segments*, covering the requirements for a specific type of transaction (e.g. invoice), as described in a *message specification*; a message starts with a *message header* and ends with a *message trailer* [66]

NOTE In *transfer*, a message is a specific ordered set of *segments* in conformance with a *message specification*.

4.67**message body**

identified, named and structured set of functionally related *segments*, covering the requirements for a specific type of transaction (e.g. invoice), as described in a *message specification*, excluding the *message header* and the *message trailer* [67]

4.68**message directory**

listing of identified and named *messages* each with its *message specification* [68]

4.69**message header**

service segment starting and uniquely identifying a *message* [69]

4.70**message specification**

description of a *message* in a *message directory*, including the specification of the position, *status* and maximum number of occurrences of the *segments* and *segment groups* constituting the *message* [70]

4.71**message trailer**

service segment ending a *message* [71]

4.72**message type**

code identifying a type of *message* [72]

4.73**non-repudiation of origin**

element of service allowing the originator of a message to provide the recipient(s) of the message irrevocable proof of origin of the message and the integrity of its content. This will protect against any attempt by the originator to subsequently revoke the message or its contents. Non-repudiation of origin is provided to the recipient(s) of a message on a per-message basis using asymmetric encryption techniques [73]

[ITU-T F.400/X.400, Amendment 1]

4.74

numeric character set

character set that contains digits and may contain *control characters* and *special characters* but not letters [74]

[ISO/IEC 2382-4:1999, 04.01.04]

4.75

object

stream of bits grouped in octets (which may be associated with an EDIFACT *message*) [75]

4.76

object header

service segment starting and uniquely identifying an *object* [76]

4.77

object trailer

service segment ending an *object* [77]

4.78

organization

unique framework of authority within which a person or persons act, or are designated to act, towards some purpose [78]

[ISO/IEC 6523-1:1998, 3.1]

4.79

package

object plus its associated header and trailer segments [79]

4.80

parent-child relationship

relationship between two entities, one ("child") being contained within and directly subordinated to the other ("parent") [80]

4.81

position identifier

identifier used in a *dependency note* to identify an entity (*segment group*, *segment*, or *data element*) by its position in the parent entity [81]

4.82

private key

(in a *public key* cryptosystem) that key of a user's key pair which is known only by that user [82]

[ISO/IEC 9594-8:1998, 3.3.19]

4.83

public key

(in a *public key* cryptosystem) that key of a user's key pair which is publicly known [83]

[ISO/IEC 9594-8:1998, 3.3.18]

4.84

qualifier

simple data element whose *data element value*, extracted from a *code list*, gives specific meaning to the function of another *data element* or a *segment* [84]

4.85

release character

character indicating that the *character* immediately following it shall be passed to the application as received [85]

4.86**repeating data element**

composite data element or *stand-alone data element* having a maximum occurrence of greater than one in the *segment specification* [86]

4.87**repetition separator**

service character used to separate adjacent occurrences of a *repeating data element* [87]

4.88**responder**

application replying to an *initiator* [88]

4.89**scenario**

formal specification of a class of *business* activities having the same *business goal* [89]

4.90**secret key**

key used with *symmetric cryptographic* techniques and usable only by a set of specified entities [90]

[ISO/IEC 11770-1:1996, 3.18]

4.91**segment**

identified, named and structured set of functionally related *composite data elements* and/or *stand-alone data elements*, as described in a *segment specification*; a segment starts with the *segment tag* and ends with the *segment terminator* [91]

NOTE In *transfer*, a segment is a specific ordered set of one or more *composite data element(s)* and/or *stand-alone data element(s)* in conformance with a *segment specification* and the syntax rules for *transfer*.

4.92**segment directory**

listing of identified and named *segments* with their *segment specification* [92]

4.93**segment group**

identified hierarchical set of *segments* and/or *segment groups* within a *message* [93]

4.94**segment specification**

description of a *segment* in a *segment directory*, including the specification of the position, *status* and maximum number of occurrences of the *data elements* constituting the *segment* [94]

4.95**segment tag**

simple data element uniquely identifying a *segment*, by reference to a *segment directory* [95]

4.96**segment terminator**

service character indicating the end of a *segment* [96]

4.97**service character**

character reserved for syntactical use; the service characters are the *component data element separator*, the *data element separator*, the *release character*, the *repetition separator* and the *segment terminator* [97]

4.98

service composite data element

composite data element used in *service segments* [98]

NOTE A service composite data element specification contains only service simple data elements.

4.99

service data element

service simple data element or a *service composite data element* [99]

4.100

service message

message used to exchange service information relating to the use of EDIFACT syntax rules or security [100]

NOTE A service message specification contains only service segments.

4.101

service segment

segment used

a) in *service messages*;

b) to control the *transfer of data* [101]

NOTE A service segment specification contains only service composite data elements and/or service simple data elements.

4.102

service simple data element

simple data element used only in *service segments* and/or *service composite data elements* [102]

4.103

service string advice

optional string of *characters* used at the beginning of an *interchange* to specify the *service characters* used in the *interchange* [103]

4.104

simple data element

data element containing a single *data element value* [104]

NOTE There are two uses of a simple data element: within a composite data element (component data element); and within a segment outside a composite data element (stand-alone data element).

4.105

simple data element directory

listing of identified and named *simple data elements* with their *simple data element specification* [105]

4.106

simple data element specification

set of attributes characterizing a *simple data element* in a *simple data element directory* [106]

4.107

special character

graphic character that is not a letter, digit, or blank *character*, and usually not an ideogram [107]

[ISO/IEC 2382-4:1999, 04.03.12]

4.108

stand-alone data element

simple data element used within a *segment* without being in a *composite data element* [108]

4.109**status**

attribute of a segment, a segment group, a composite data element or a simple data element identifying the rules for the presence or absence of the segment/data element in the usage of a message [109]

NOTE The types of status are *conditional* and *mandatory*.

4.110**string**

sequence of elements of the same nature, such as *characters*, considered as a whole [110]

[ISO/IEC 2382-4:1999, 04.05.01]

4.111**symmetric algorithm**

cryptographic algorithm employing the same value of *key* for both enciphering and deciphering or for both *authentication* and validation [111]

4.112**threat**

potential violation of security [112]

[ISO 7498-2:1989, 3.3.55]

4.113**transfer**

communication of information from one partner to another [113]

4.114**trigger segment**

segment starting a segment group [114]

5 Service characters**5.1 General**

The service characters are the component data element separator, data element separator, release character, repetition separator, and segment terminator.

The component data element separator, data element separator, repetition separator, and segment terminator delineate various syntax structures as defined in clause 7.

The purpose of the release character is to allow the use of a character that would otherwise be interpreted as a service character. The character immediately following the release character in an interchange shall not be interpreted as a service character.

When used, the release character is not counted in the length of the data element value.

NOTE Using default service characters given in 5.2, 10?+10=20 appearing in a data transfer is interpreted on receipt as 10+10=20. A question mark in a data element value is represented in transfer as ??.

5.2 Default service characters

The default service characters reserved for use in this International Standard are given in Table 1

Table 1 — Default service characters

Name	Graphic representation	Functionality
Colon	:	component data element separator
Plus sign	+	data element separator
Question mark	?	release character
Asterisk	*	repetition separator
Apostrophe	'	segment terminator

5.3 UNA, service string advice

The conditional service string advice (UNA) provides the capability to specify the service characters used in the interchange (see annex A). The UNA service string advice shall be used if the service characters differ from the defaults (see 5.2). Its use is optional if the default characters are used.

When used, the service string advice shall appear immediately before the interchange header segment.

6 Character repertoires

The character encoding specified in basic code table of ISO/IEC 646 shall be used for the interchange service string advice (if used) and up to and including the composite data element S001 'Syntax identifier' in the interchange header.

The character repertoire used for the characters in an interchange shall be identified from the code value of data element 0001 in S001 'Syntax identifier' in the interchange header (see ISO 9735-10). The character repertoire identified does not apply to objects and/or encrypted data.

The default encoding technique for a particular repertoire shall be the encoding technique defined by its associated character set specification.

If the default option is not used, a code value for the data element 0133 'Character encoding, coded' in the interchange header shall be used.

Code extension technique (ISO/IEC 2022) may only be used in an interchange after the composite data element S001 'Syntax identifier' in the interchange header.

The code extension technique and its target graphic characters shall only be used for:

- plain language (textual) data elements, with a representation of alphabetic or alphanumeric.

The technique shall not be used, for example, for any:

- segment tag; or
- service character; or
- data element with a representation of numeric.

Characters used to indicate code extension shall not be counted in the length of a data element, and shall not be used as service characters.

In calculating data element length, one graphic character shall be counted as one character, irrespective of the number of bytes/octetets required to encode it.

7 Syntax structures

7.1 General

The definitions in this clause specify logical syntax structures. Rules to be applied for their usage are defined in clause 8.

7.2 Interchange structure

An interchange shall be started either by a service string advice or by an interchange header, shall be identified by an interchange header, shall be terminated by an interchange trailer, and shall contain at least one group, or one message or one package. There may be more than one group or message and/or package within an interchange, each identified by its own header and terminated by its own trailer. Messages within an interchange or within a group may comprise one or more message types.

An interchange shall contain only:

- Messages; or
- Packages; or
- Messages and Packages; or
- Groups containing messages; or
- Groups containing packages; or
- Groups containing messages and packages.

7.3 Group structure

A group is a conditional structure which is located between the interchange header and trailer and which comprises one or more messages and/or packages.

A group shall be started and identified by a group header, shall be terminated by a group trailer, and shall contain at least one message or package.

7.4 Message structure

A message comprises an ordered set of segments (see annex B). Segments may be grouped. Each segment's position, status, and maximum number of occurrences shall be stated in the message specification.

A given segment within a message specification shall have a status of mandatory or conditional.

A message specification shall ensure unambiguous identification of each message segment upon receipt. Identification shall be possible on the basis of the segment tag (or the segment tag plus the anti-collision segment group identification in the UGH and UGT segments) and the segment's position in the transferred message. Identification shall not depend on a segment's status or maximum number of occurrences.

A message shall be started and identified by a message header, shall be terminated by a message trailer, and shall contain at least one additional segment.

7.5 Segment group structure

A segment group comprises an ordered set of segments: a trigger segment and at least one more segment or segment group. The trigger segment shall be the first segment in the segment group, shall have a status of mandatory and a maximum number of occurrences of one. Each segment group's position, status, and maximum number of occurrences within the message structure shall be stated in the message specification.

A segment group may contain one or more dependent segment groups. When a segment group is contained within and directly subordinate to another segment group, the subordinate segment group is referred to as the child, and the other segment group is referred to as the parent.

A given segment group within a message specification shall have a status of mandatory or conditional.

7.6 Segment structure

A segment comprises an ordered set of stand-alone data elements and/or composite data elements, each of which are permitted to repeat, if so stated in the segment specification. Each stand-alone or composite data element's position, status and maximum number of occurrences within the segment structure shall be stated in the segment specification. A segment shall be started and identified by a segment tag which references a specific segment specification. A segment shall contain at least one data element in addition to the segment tag.

A given data element within a segment specification shall have a status of mandatory or conditional.

7.7 Segment tag structure

A segment tag is a simple data element.

Segment tags starting with the letter "U" (e.g. UNB, UIH) shall be reserved for service segments.

7.8 Composite data element structure

A composite data element comprises an ordered set of two or more component data elements. Each component data element's position and status within the composite data element structure shall be stated in the composite data element specification.

A given component data element within a composite data element specification shall have a status of mandatory or conditional.

7.9 Simple data element structure

A simple data element contains a single data element value.

A simple data element is used either as a stand-alone data element or as a component data element. A stand-alone data element occurs in a segment outside a composite data element. A component data element occurs within a composite data element.

Each simple data element's data value representation shall be stated in the data element specification.

7.10 Package structure

A package shall be started and identified by an object header, shall be terminated by an object trailer, and shall contain one object.

8 Inclusion and exclusion

8.1 General

The rules in this clause shall be applied when a message is prepared for transfer. Under these rules, in certain circumstances, segment groups, segments, data elements, and characters within a data element value, shall be present, while in other circumstances shall be omitted.

8.2 Determination of presence

A simple data element is considered present if its data element value contains at least one character.

A composite data element is considered present if at least one of its component data elements is present.

A segment is considered present if its segment tag is present.

A segment group is considered present if its trigger segment is present.

8.3 Inclusion of segment groups

A mandatory segment group which is not contained within another segment group shall be present.

A mandatory child segment group shall be present if its parent segment group is present.

A single occurrence of a segment group having a status of mandatory is sufficient to satisfy the mandatory requirement.

8.4 Exclusion of segment groups

If a segment group is omitted, all of its segments and any dependent segment groups contained within it, regardless of their status, shall also be omitted.

8.5 Inclusion of segments

Segments shall appear in the order stated in the message specification.

A segment shall be terminated by a segment terminator.

A mandatory segment which is not in a segment group shall be present.

A mandatory segment contained in a segment group shall be present if the segment group is present.

A single occurrence of a segment having a status of mandatory is sufficient to satisfy the mandatory requirement.

Using a fictitious segment tag of ABC as an example, a mandatory segment defined as containing only conditional data elements for which no data is present at the time of transfer, shall be transferred in the form ABC'.

8.6 Exclusion of segments

A conditional segment for which only the segment tag is present shall be omitted in its entirety.

8.7 Inclusion of data elements

Data elements shall appear in the order stated in the segment specification.

Adjacent non-repeating data elements in the same segment shall be separated by a data element separator.

Adjacent occurrences of the same repeating data element in a segment shall be separated by a repetition separator.

Adjacent component data elements in the same composite data element shall be separated by a component data element separator.

A mandatory stand-alone data element in a segment shall be present if the segment is present.

A mandatory composite data element in a segment shall be present if the segment is present.

A mandatory component data element in a composite data element shall be present if the composite data element is present.

A single occurrence of a repeating data element having a status of mandatory is sufficient to satisfy the mandatory requirement.

8.8 Exclusion of data elements

8.8.1 General

In Figures 1 to 6, “Tag” represents a segment tag, “DE” represents a composite data element or stand-alone data element, and “CE” represents a component data element. The default service characters are used.

8.8.2 Exclusion of composite data elements and stand-alone data elements

If a non-repeating composite data element or stand-alone data element is omitted and is followed by another composite data element or stand-alone data element in the same segment, its position shall be indicated by retention of the data element separator which would normally follow it. This rule also applies if all occurrences of a repeating data element are omitted.

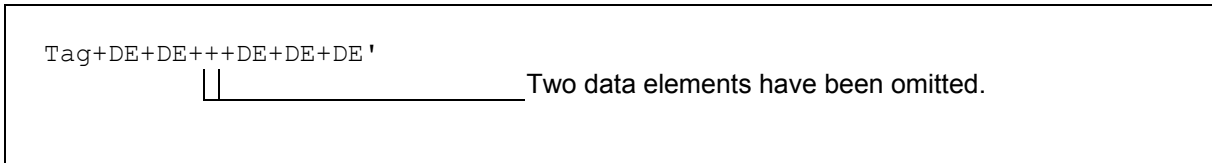


Figure 1 — Exclusion of non-repeating data elements within a segment

If one or more non-repeating composite data elements or stand-alone data elements at the end of a segment are omitted, the data element separators which would normally follow them shall also be omitted.

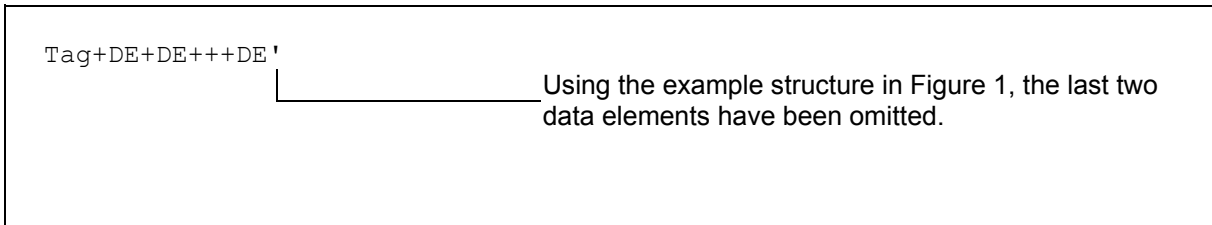


Figure 2 — Exclusion of non-repeating data elements at the end of a segment

8.8.3 Exclusion of component data elements

If a component data element is omitted and is followed by another component data element in the same composite data element, its position shall be indicated by retention of the component data element separator which would normally follow it.

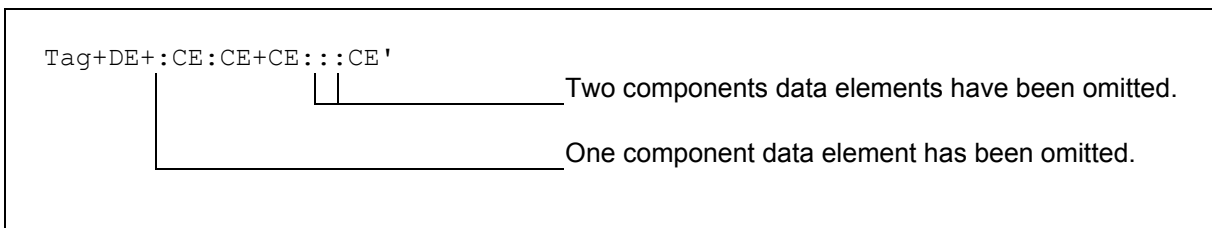


Figure 3 — Exclusion of component data elements within a composite data element

If one or more component data elements at the end of a composite data element are omitted, the component data element separators which would normally follow them shall also be omitted.

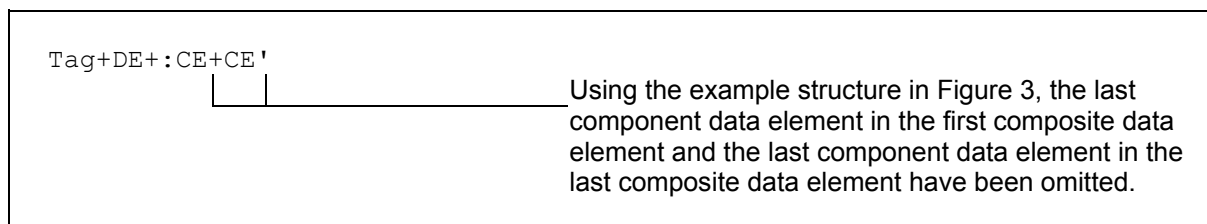


Figure 4 — Exclusion of component data elements at the end of a composite data element

8.8.4 Exclusion of occurrences of repeating data elements

The position of an occurrence of a repeating data element may be significant, for example, to transfer array data.

In such a case, if an occurrence of a repeating data element is omitted and is followed by another occurrence of the same repeating data element, its position shall be indicated by retention of the repetition separator which would normally follow it.

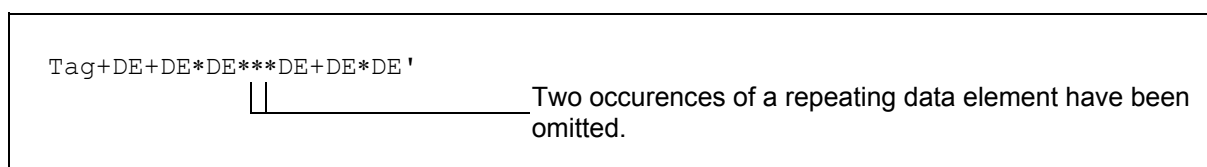


Figure 5 — Exclusion of occurrences within a repeating data element

If one or more occurrences of a repeating data element at the end of a repeating data element are omitted, the repetition separators which would normally follow them shall also be omitted.

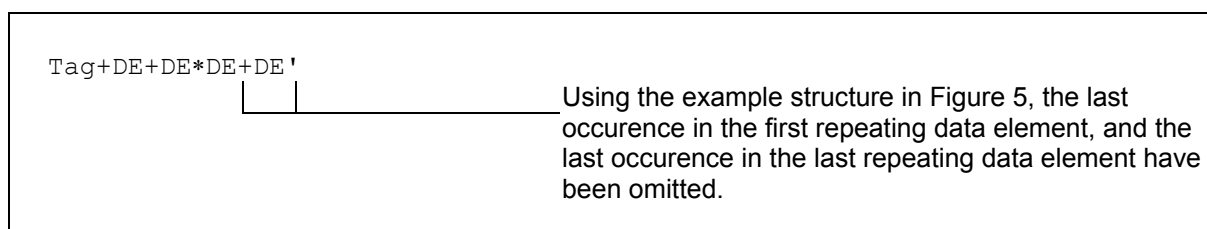


Figure 6 — Exclusion of occurrences at the end of a repeating data element

9 Suppression of characters within data elements

9.1 General

In variable length data elements, insignificant characters shall be suppressed (i.e. omitted from the transfer), while significant characters shall be present.

9.2 Insignificant characters

In variable length numeric data elements, leading zeroes shall be suppressed. Nevertheless, a single zero before a decimal mark is allowed. In variable length alphabetic and alphanumeric data elements, trailing spaces shall be suppressed.

9.3 Significant zeroes

Significant zeroes shall not be suppressed. A single zero may be significant, for example, to indicate a temperature or tax rate. Trailing zeroes following the decimal mark may be significant to indicate precision.

9.4 Significant spaces

Significant spaces shall not be suppressed. Leading and embedded spaces may be significant.

A data element value containing only space(s) shall not be allowed.

10 Representation of numeric data element values

For the purposes of this standard, the representation of numeric data element values shall be any of the representations as specified by ISO 6093 (which excludes the use of triad separators), with the following exceptions:

- The encoding specified in ISO/IEC 646 shall not be required.
- For variable length numeric fields, the rules for suppression apply (see clause 9).
- The space character and plus sign shall not be allowed.
- The length of a numeric data element value shall not include the minus sign (–), the decimal mark (. or ,), or the exponent mark (E or e) and its exponent.
- When a decimal mark is transferred, there shall be at least one digit after the decimal mark.

The full stop or the comma is allowed to represent the decimal mark for a single numerical value.

EXAMPLE Examples using decimal marks:

Allowed (full stop): 2 and 2.00 and 0.5 and .5

Not allowed (full stop): 1. and 0. and .

Allowed (comma): 2 and 2,00 and 0,5 and ,5

Not allowed (comma): 1, and 0, and ,

11 Dependency notes

11.1 General

If required, dependency notes shall be used in the message, segment specification, or composite data element to express relationships.

In a dependency note, a list is defined as two or more entities (where an entity can be a segment group; a segment; a composite data element; a stand-alone data element or a component data element).

Any entity may be subject to more than one dependency note.

11.2 Dependency notes in the message specification

Dependency notes in the message specification are used to describe the relationship between segments, between segment groups, or between segments and segment groups. These entities shall be at the same hierarchical level and within the same parent structure.

11.3 Dependency notes in the segment specification

Dependency notes in the segment specification are used to describe the relationship between stand-alone data elements, between stand-alone data elements and composite data elements, or between composite data elements. These entities shall be in the same segment.

Dependency notes shall not be used to describe a relationship between stand-alone data elements and component data elements, or to describe a relationship between composite data elements and component data elements.

11.4 Dependency notes in the composite data element specification

Dependency notes in the composite data element specification are used to describe the relationship between component data elements. These entities shall be in the same composite data element.

11.5 Notation for dependency notes

The notation for dependency notes comprise a dependency identifier followed by a list, in parenthesis, of position identifiers separated by commas e.g. D3(030, 060, 090). The position identifier identifies an entity by its position number in its parent entity. The dependency identifier identifies the type of dependency between the entities in the list.

A list shall contain at least two position identifiers. The order of position identifiers in a list may be different from that implied by their value.

Dependency identifiers are described below.

D1 ONE AND ONLY ONE

One and only one of the entities in the list shall be present.

D2 ALL OR NONE

If one entity in the list is present, the rest shall be present.

D3 ONE OR MORE

At least one of the entities in the list shall be present.

D4 ONE OR NONE

No more than one entity in the list shall be present.

D5 IF FIRST, THEN ALL

If the first entity in the list is present, then all of the others shall be present. It is permissible that one or more of the entities not specified as the first entity in the list may be present, without requiring the first entity to be present.

D6 IF FIRST, THEN AT LEAST ONE MORE

If the first entity in the list is present, then at least one more shall be present. It is permissible that one or more of the entities not specified as the first entity in the list may be present, without requiring the first entity to be present.

D7 IF FIRST, THEN NONE OF THE OTHERS

If the first entity in the list is present, then none of the others shall be present.

12 Segment collision prevention

The UGH/UGT segment group shall be used in a message specification when it is not otherwise possible to ensure unambiguous identification of each message segment upon receipt through the basis of the segment tag and the segment's position in the transferred message (see annex C).

In this event, the UGH/UGT segment group shall be specified to surround the segment group which otherwise could not be unambiguously identified.

In the UGH/UGT segment group, the UGH segment shall be the first segment, shall be mandatory and shall be specified with a maximum number of occurrences of one. The UGT segment shall be the last segment in the segment group, shall be mandatory and shall be specified with a maximum number of occurrences of one.

The value of data element 0087 'Anti-collision segment group identification' shall be the segment group number of the UGH/UGT segment group as stated in the message specification.

The UGH/UGT segment group shall have a maximum number of occurrences of one and shall be specified with a status of conditional or mandatory, identical to the status of the segment group it surrounds.

When a conditional UGH/UGT service segment group surrounds a conditional group in a message structure which could cause collision, the UGH/UGT service segment group shall only be transferred when data is present for the conditional group which it surrounds.

13 Syntax release identification

In order to identify future releases of the syntax within a syntax version, the Syntax identifier (composite data element S001) within the UNB Interchange header or UIB Interactive Interchange header segments shall be used (see annex D).

Annex A (normative)

UNA service string advice

The service string advice shall begin with the upper case characters UNA immediately followed by six characters in the order shown below. The space character shall not be used in positions 010, 020, 040, 050 or 060. The same character shall not be used in more than one position of the UNA.

POS	REP	S	Name	Remarks
010	an1	M	COMPONENT DATA ELEMENT SEPARATOR	
020	an1	M	DATA ELEMENT SEPARATOR	
030	an1	M	DECIMAL MARK	The character transferred in this position shall be ignored by the recipient. Retained to maintain upward compatibility with earlier versions of the syntax.
040	an1	M	RELEASE CHARACTER	
050	an1	M	REPETITION SEPARATOR	
060	an1	M	SEGMENT TERMINATOR	

Key

POS	The three-digit sequential number of the character in the service string
REP	The representation of the service string character an1 = 1 alphanumeric character
S	The status of the service string character M = Mandatory
Name	Name of the service string advice character
Remarks	Additional remarks

Annex B (informative)

Order of segments and groups of segments within a message

B.1 General

Segments used in a message appear in the sequence (top to bottom) specified in the message segment table.

In the message segment table, segments are indicated by their tags. The requirement for their inclusion in the message, i.e. their status, is indicated by the letter M for mandatory or C for conditional. The number of times a segment may occur in each instance is indicated directly thereafter. This may be followed by any associated dependency note identifier(s).

In the message segment table, segment groups are indicated by their segment group number. The requirement for their inclusion in the message, i.e. their status, is indicated by the letter M for mandatory or C for conditional. The number of times a segment group may occur in each instance is indicated directly thereafter. This may be followed by any associated dependency note identifier(s).

B.2 Segment groups

Two or more segments can be grouped, as shown in Figure B.1. The trigger segment of each segment group appears in the message segment table immediately following the segment group identification (i.e. segment group 1, 2 etc.). All other segments within the segment group follow in sequence, with the last segment in the group identified by the boundary lines specifying the extent of the segment group.

A segment group can contain another dependent segment group or groups (e.g. segment group 2 contains a dependent segment group 3 in the figure), and, as can be seen in the figure, a segment can terminate two (or more) segment groups, as indicated by the segment group boundary lines (segment LLL in the figure).

In the figure, segment group 2 is the parent of segment group 3, and segment group 3 is the parent of segment group 4.

POS	TAG	Name	S	R	Notes
0010	Uxx	Message header	M	1	
0020	AAA	Segment AAA name	M	1	
0030	BBB	Segment BBB name	C	9	
0040	CCC	Segment CCC name	C	9	
0050	-----	Segment group 1 -----	C	999	-----+ 1
0060	DDD	Segment DDD name	M	1	
0070	EEE	Segment EEE name	C	9	
0080	FFF	Segment FFF name	C	9	
0090	GGG	Segment GGG name	C	1	-----+
0100	-----	Segment group 2 -----	C	9	-----+ 1
0110	HHH	Segment HHH name	M	1	
0120	-----	Segment group 3 -----	C	9	-----++
0130	III	Segment III name	M	1	
0140	JJJ	Segment JJJ name	C	9	
0150	-----	Segment group 4 -----	C	9	-----+
0160	KKK	Segment KKK name	M	1	
0170	LLL	Segment LLL name	C	9	-----+++
.					
.					
.					
.					
nnnn	Uxx	Message trailer	M	1	

DEPENDENCY NOTES:

1. D3(0050, 0100) One or more

Key

POS	The sequential position number of the segment or segment group in the message (in steps of 10 to permit later amendment to message structures)
TAG	The tag of the segment in the message
Name	The name of the segment in the message
S	The status (of the segment or segment group) i.e. M = Mandatory, C = Conditional
R	The maximum number of occurrences of the segment or segment group
Notes	The note number

Figure B.1 — Example message segment table

NOTE 1 An example of the processing/sequencing order of the segments (using the segment tags only) is as follows (with segment group 1 appearing twice, the other groups once, and with repeating segments shown as appearing once only):

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

NOTE 2 In the segment table and in the segment string shown above, the first “Uxx Message Header” would be “UNH” for batch EDI, and “UIH” for interactive EDI; the second “Uxx Message trailer” would be “UNT” for batch EDI and “UIT” for interactive EDI.

NOTE 3 As is shown, dependency notes at the message level (note 1 in Figure B.1) can be specified in the message segment table (which in UN/EDIFACT form part of the message specification).

Annex C (informative)

The use of anti-collision segment group UGH/UGT

C.1 Introduction

This annex outlines the use of the segment group UGH/UGT to prevent segment collision. This technique should only be used after alternative approaches have been deemed unsuccessful.

C.2 The problem

EXAMPLE 1:

```

ABC           C 1
DEF           C 5
--- Grp 1 --- C 5 -----+
ABC           M 1      |
JKL           C 5      |
                |
--- Grp 2 --- C 5 ---+  |
ABC           M 1      | |
MNO           C 5      | |
PQR           C 5 ---+--+

```

The example above shows a message structure that uses five different segments in seven positions. Segment ABC is used in three different positions.

The data stream may look like this:

```
ABC+...'DEF+...'ABC+...'JKL+...'ABC+...'MNO+...'PQR+...'ABC+...'
```

The problem of segment collision now exists, since it is not clear if the first ABC segment is the stand-alone segment or the start of Group 1. Further, it is not clear whether the ABC segment after the JKL is the start of Group 2 or the start of Group 1.

The use of the UGH/UGT segment groups will correct this situation. The correct way to place the UGH/UGT group is to envelop the inner-most segment group, of those groups having the problem.

EXAMPLE 2:

```

ABC          C 1
DEF          C 5
--- Grp 1 --- C 1 -----+
UGH          M 1
--- Grp 2 --- C 5 -----+ |
ABC          M 1          | |
JKL          C 5          | |
--- Grp 3 --- C 1 -----+ | |
UGH          M 1          | | |
--- Grp 4 --- C 5 ---+   | | |
ABC          M 1          | | | |
MNO          C 5          | | | |
PQR          C 5 ---+   | | | |
UGT          M 1 -----+---+ |
UGT          M 1 -----+
    
```

An instance of the data stream may look like this:

ABC+...'DEF+...'UGH+1'ABC+...'JKL+...'UGH+3'ABC+...'MNO+...'PQR+...'ABC+...'UGT+3'UGT+1'

In table form, the sample data stream would be as follows:

Data stream value	Description
ABC+...'DEF+...'	Stand-alone segments
UGH+1'	Start of Group 1, signals the start of Group 2
ABC+...'JKL+...'	Group 2
UGH+3'	Start of Group 3, signals the start of Group 4
ABC+...'MNO+...'PQR+...'	Group 4
ABC+...'	Group 4
UGT+3'	End of Group 3, signals the end of Group 4
UGT+1'	End of Group 1, signals the end of Group 2

Now it is clear that the ABC after JKL is the start of Group 4 and the ABC after PQR is the second occurrence of Group 4, since there was no UGT after the PQR.

Considering a more detailed instance of the data stream:

ABC+...'DEF+...'UGH+1'ABC+...'JKL+...'UGH+3'ABC+...'MNO+...'PQR+...'UGT+3'ABC+...'JKL+...'

UGH+3'ABC+...'MNO+...'PQR+...'ABC+...'MNO+...'PQR+...'ABC+...'MNO+...'PQR+...'UGT+3'UGT+1'

In table form, the sample data stream would be as follows:

Data stream value	Description
ABC+...'DEF+...'	Stand-alone segments
UGH+1'	Start of Group 1, signals the start of Group 2
ABC+...'JKL+...'	Group 2
UGH+3'	Start of Group 3, signals the start of Group 4
ABC+...'MNO+...'PQR+...'	Group 4
UGT+3'	End of Group 3, signals the end of Group 4
ABC+...'JKL+...'	Group 2
UGH+3'	Start of Group 3, signals the start of Group 4
ABC+...'MNO+...'PQR+...'	Group 4
ABC+...'MNO+...'PQR+...'	Group 4
ABC+...'MNO+...'PQR+...'	Group 4
UGT+3'	End of Group 3, signals the end of Group 4
UGT+1'	End of Group 1, signals the end of Group 2

In the above example, Group 2 occurred twice. Within its first occurrence, Group 4 occurred once, and within its second occurrence, Group 4 occurred three times.

Annex D (informative)

Syntax release identification

The structure of composite data element S001 is defined as follows:

POS	TAG	Name	S	R	Repr.
010	S001	SYNTAX IDENTIFIER	M	1	
	0001	Syntax identifier	M		a4
	0002	Syntax version number	M		an1
	0080	Service code list directory version number	C		an..6
	0133	Character encoding, coded	C		an..3
	0076	Syntax release number	C		an2

Component data element 0076 provides the means to identify releases within a syntax version number. For example, a technical corrigendum issued against the syntax shall result in the syntax release number (which starts at 00) being incremented by one (e. g. 01, 02).

The bi-annual updating of the service code list directory maintained by UN/CEFACT, and published on the JSWG web site shall result in the last two digits of the service code list directory version number (data element 0080) being incremented by one. The value of the syntax version number (element 0002) and the value of the syntax release number (element 0076) shall also be reflected in the 1st, 2nd and 3rd digits of the service code list directory version number (element 0080) respectively.

Figure D.1 demonstrates this numbering system, which results in a five digit field identification scheme held in element 0080.

0002 Syntax version number	0076 Syntax release number	0080 Service code list directory version number
4	00	40000
	00	40001
	00	40002
	00	40003
	01	40100
	01	40101
	01	40102
	02	40200
	02	40201
	02	40202

	99	49900

Figure D.1 — Identification scheme under syntax version 4

The syntax version number (component data element 0002) would only be increased after a specified number of years, should significant changes or enhancements be required.

Figure D.2 illustrates the effect on the identification scheme when the syntax version number changes to 5.

0002 Syntax version number	0076 Syntax release number	0080 Service code list directory version number
5	00	50000
	00	50001
	00	50002
	01	50100
	01	50101
	01	50102
	01	50103
	02	50200
	02	50201
	02	50202

	99	59900

Figure D.2 — Identification scheme under syntax version 5

It is important to recognize that the syntax version number is the most stable field and the syntax release number field allows for intermediate changes, prior to a decision being taken to update the syntax version number. However, the service code list directory version number is the most dynamic field, because there would be one to two electronic publications of the service code list directories a year.

An example of how the data elements within composite data element S001 would be populated, if the only change to version 4 of the syntax was an addition to the service code list directory, is as follows:

```

S001  SYNTAX IDENTIFIER
0001  Syntax identifier           UNOA
0002  Syntax version number      4
0080  Service code list directory version number  40001
0133  Character encoding, coded
0076  Syntax release number

```

In the UNB segment, this would be transferred as:

```
UNB+UNOA:4:40001+.....
```

where:

UNOA = a3, upper case Controlling Agency (e.g. UNO = UN/ECE) and a1 Stating level (e.g. A) (which together give UNOA);

4 = version 4 of ISO 9735;

40001 = 4 represents the version of the syntax, 00 signifies that the release of the syntax has not changed, and 01 means that there has been a change to the service code list directory.

A subsequent first change to the syntax release number would result in the composite data element S001 being populated as follows:

S001	SYNTAX IDENTIFIER	
0001	Syntax identifier	UNOA
0002	Syntax version number	4
0080	Service code list directory version number	40101
0133	Character encoding, coded	
0076	Syntax release number	01

In the UNB segment, this would be transferred as:

UNB+UNOA:4:40101::01+.....