UN/EDIFACT DRAFT DIRECTORY

PART 4 UNITED NATIONS RULES FOR ELECTRONIC DATA INTERCHANGE
FOR ADMINISTRATION, COMMERCE AND TRANSPORT

CHAPTER 2.3 - UN/EDIFACT Syntax Implementation Guidelines

CONTENTS

SECTION

1. INTRODUCTION

```
The purpose of these guidelines is to assist Electronic Data
Interchange (EDI) users with their implementation of the
ISO-UN/EDIFACT (EDI for Administration, Commerce and
Transport) Syntax Rules, and to expand some of the rules contained
in ISO 9735, supported by examples.

The guidelines are a part of a set of complementary documents
available to users.  The other documents in the series which users
should be aware of are:-

 - UNTDED   : The United Nations Trade Data Elements Directory
              (also published as the International Standard ISO
              7372) and associated code sets

 - ISO 9735 : The EDIFACT Syntax Rules standard
```

- The UN/EDIFACT Message Design Guidelines

- The UN/EDIFACT Directory Set, containing directories for:

  UNEDMD - Internationally agreed UN/EDIFACT Standard Messages (UNSMs)

  UNEDSD - UN/EDIFACT Standard Segments for UNSMs

  UNEDCD - UN/EDIFACT Composite data elements for UNSMs

  UNEDED - UN/EDIFACT Data Elements for UNSMs

  UNEDCL - UN/EDIFACT Code List for UNSMs

UNTDED is published separately by the UN, and maintained jointly with ISO. The remaining documents are compiled in the UN Trade Data Interchange Directory.

In 1987, following the convergence of the UN and US/ANSI syntax proposals, the UN/EDIFACT Syntax Rules were approved as an ISO standard, having been developed within and submitted for approval by the United Nations Economic Commission for Europe's Working Party on the Facilitation of International Trade Procedures (UN/ECE WP.4).

At the same time, WP.4 appointed three Rapporteurs to co-ordinate the continuing work in conjunction with the UN/ECE Secretariat. The Rapporteurs appointed were from Poland (co-ordinating the views and input from CMEA countries), from the United Kingdom (co-ordinating on behalf of the EEC and EFTA countries), and the United States (on behalf of the US and Canada). Additional Rapporteurs may be appointed in the future.

In particular, the UNECE Secretariat and the Rapporteurs, supported by Advisory and Support Teams, will be the focal point for maintenance of the UN/EDIFACT Syntax Rules and for proposals for new UNSMs (or for amendments to existing UNSMs). The agreed maintenance procedures can be found in the Message Design

Guidelines document, as can the contact points for the Rapporteurs.

Under NO circumstances should users, software providers, or network providers make any changes to the UN/EDIFACT Syntax Rules as defined in ISO 9735. Change requests should be registered either direct to the UN/ECE Secretariat, or via one of the Rapporteurs, (or by use of ISO procedures) for international discussion and approval, both at the UN and in the ISO.

From the outset of the development of the UN/EDIFACT techniques, certain important design criteria were adhered to. These were that the techniques should be independent of the computers to be used, the systems using them, the applications using them, the communications methods to be used, and the data to be interchanged. The fact that there are a range of live and pilot applications, using a broad spectrum of mainframe, mini and micro computers, utilizing a range of different computer communications protocols, (such as 2780, 3780, SNA/DNA, packet switching etc.), different systems solutions (including one-to-one direct interchange and mailbox switching), demonstrates that the criteria have been met.

In addition to the above, UN/EDIFACT is designed to have the minimum impact upon the in-house systems using the technique. Many live applications constructing data for transmission of UN/EDIFACT messages, use a technique of creating a simple serial file - often structured to hold records containing data equivalent to that required for segments of data in the messages. This file is then submitted to a formatting routine, which constructs the data according to the UN/EDIFACT syntax.

Experience has shown that for both converting from the in-house file layout into UN/EDIFACT Syntax for transmission, and for converting back into the required in-house layout on receipt of an EDIFACT structured transmission, parameter (or table) driven routines have proven to be very effective. When receiving a transmission for translation, by using such routines, it is possible for the recipient to ignore any data which is of no interest to him for his own in-house needs.

It is stressed that UN/EDIFACT is a user application protocol for use within user systems, which is compatible with the OSI model, in that it presents user data for transmission via the services described by the model.

A common technique is for users to have their own in-house written routines (or a software package), for formatting and de-formatting UN/EDIFACT structured transmission files. All of this is user data, which is then submitted to a proprietary communications protocol (such as 2780, 3780, X25 etc) as defined in the User Interchange Agreement.

```
                           Interchange
              +--------------------------------------+
              |            Agreement                 |
              |                                      |
              |                                      |
   +-----------------------+              +-----------------------+
   | USER 'A' SYSTEM       |              | USER 'B' SYSTEM       |
   |.......................|              |.......................|
   | EDIFACT formatting and|              | EDIFACT formatting and|
   | de-formatting routine |              | de-formatting routine |
   |.......................|              |.......................|
  +|-----------------------|+            +|-----------------------|+
  || Communications        ||            || Communications        ||
  || protocol              ||            || protocol              ||
  |+-----------------------+|            |+-----------------------+|
  |   APPLICATION LAYER     |            |   APPLICATION LAYER     |
  +-------------------------+            +-------------------------+
  |   PRESENTATION LAYER    |            |   PRESENTATION LAYER    |
  +-------------------------+            +-------------------------+
  |   SESSION LAYER         |            |   SESSION LAYER         |
  +-------------------------+            +-------------------------+
  |   TRANSPORT LAYER       |            |   TRANSPORT LAYER       |
  +-------------------------+            +-------------------------+
  |   NETWORK LAYER         |            |   NETWORK LAYER         |
  +-------------------------+            +-------------------------+
  |   DATA LINK LAYER       |            |   DATA LINK LAYER       |
  +-------------------------+            +-------------------------+
  |   PHYSICAL LAYER       |             |   PHYSICAL LAYER       |
  |                        |             |                        |
  +------------------------+             +------------------------+
              |                                      |
              |                                      |
              +--------------------------------------+
```

2. BASIC REQUIREMENTS FOR EDI APPLICATIONS

2.1 Standards

Without strict adherence to the published UN/EDIFACT standards, many of the achievable benefits will be lost.

The UN/EDIFACT Syntax Rules (ISO 9735), set the standards for structuring data into segments, segments into messages, and messages into an interchange.

Data, segment and message standards are equally essential requirements.

The United Nations Trade Data Elements Directory (UNTDED) sets out the standards for administration, commerce and transport data. Where appropriate it also recommends codes for coded representation of data (often internationally maintained codes), and for qualifiers. Since UNTDED is also an ISO standard (ISO 7372), they are maintained jointly by the UN and ISO.

Because of the repetitive nature of information required in each of the sectors for which UN/EDIFACT has been designed, it is possible to assemble logically related data elements into standard segments, which can then be used in several different messages meeting the requirements of related business functions. An example of this can be found by examining United Nations Standard Messages (UNSMs), such as the Commercial Invoice, the Purchase Order and the Dispatch Advice. Such standard segments are published in the UNSM Standard Segments Directory, with other segments specific to certain messages.

Finally, UNSMs are published in the UN/EDIFACT Message Directory.

The procedures for the maintenance of the UN/EDIFACT Syntax and the directories (including how users can propose changes/additions to existing segments/messages, and proposals for new UNSMs) are contained in the Message Design Guidelines.

As far as possible, (providing the required function has been covered) users should try to use existing UNSMs. At first sight, users may find that some UNSMs appear unnecessarily complex. Upon closer examination however, it will be found that many (and in some cases, most) of the segments and groups of segments are defined as being 'Conditional'. This is because the messages have been designed for International and National use, by multi-industry sectors.

Since conditional segments may be completely omitted from a message, a user's requirements can often be met by use of a relatively smaller percentage of the segments specified in a UNSM. Should this prove not to be the case, then the Message Design Guidelines must be studied.

2.2 Interface with the In-house System

Once having settled on the message standard to be used, users must then carry out a careful analysis of their in-house system with respect to the data requirements for the message in question.

This will entail ensuring that all of the data which must be provided for the application in which the message is being used, (which could include data defined as conditional, as well as mandatory data), can be supplied from the in-house system. If not, some way must be found for doing so. (In some cases, certain elements of data can be held on a 'parameter' file--see Section 2.3).

For receipt of EDI messages, clearly users must decide how the data is to be processed. (For example, can a Purchase Order message be taken directly into an existing in-house order processing system, or should some intermediate processing and perhaps re-ordering of data be carried out first?).

For both transmission and receipt of EDI messages, attention should also be paid to any codes specified for use in the messages. These may not equate exactly with those used in the in-house systems, and some form of code conversion may be required. This can be particularly important if data not previously integrated between in-house applications are to be linked.

2.3 Software

Software of one kind or another is needed to format data from in-house systems into the message and syntax structure, and to reverse the process into a form required by in-house systems.

This can either be developed in-house, or proprietary software packages can be obtained. In-house development can be a quick way of getting an application off the ground for one or two messages, but generally will not be cost effective as more EDI message are introduced into the application - mainly because the routines are usually coded to be message dependent. This can cause maintenance difficulties as messages are amended and enhanced over a period of time.

Software packages are available from a variety of sources, ranging from data capture systems to interface translators. Most are message independent, generally by use of table-driven techniques. If message content changes, this means that only the tables have to be amended, not the main modules of the package.

As previously identified, on some occasions users may find that the required data are not always available from the in-house system. For formatting of data for transmission, this can be true, particularly for some of data required for certain of the syntax service segments (e.g. the Interchange Header - UNB; and the Functional Group Header - UNG). It could also be true if for example, the full name and address of the organization originating the data is required in the interchange in order to meet some form of legal requirement.

A useful technique for solving these problems is to hold such constant data on a small parameter file, which can be accessed when required during the message formatting process.

2.4 Communications

Some form of communications carrier is the last essential basic element for implementation of EDI applications.

Some applications still exchange magnetic tapes, but increasingly, telecommunications techniques are being used. Two options are available--either direct communications, or via a third party service.

Direct point-to-point or dial-up techniques may suffice if interchange partners are few in number, and their telecommunications protocols are identical.

However, as the number of interchange partners increases, as the number of different telecommunications protocols have to be catered for, and as scheduling problems become more acute, it may well be found that the services offered by a range of network providers become a possible alternative.

Although the services offered may vary in detail, most offer network communication, the interface to the network, protocol conversion services (i.e. data is provided using a preferred telecommunications protocol and it is a function of the network provider's service to convert to the protocols to those of the interchange partners), and mailbox/clearing-house services.

Users of mailbox/clearing-house services send data to the network provider, who interrogates the header segment of each interchange in order to deposit the data in the mailbox of the appropriate addressee specified in the interchange header segment. Each recipient can then access his own mailbox to retrieve data. This can be a particularly useful facility if scheduling problems are acute, or if data is being exchanged across time zones.

When joining an existing user group, it may be found that the choice of communications techniques has already been made by the group as a whole.

3. INTERCHANGE AGREEMENT

3.1 Introduction

Virtually every live data interchange application operates under an interchange agreement, which normally takes the form of a user manual.

Once created, this has to be maintained in the same way that any computer system user manual has to be.

The type of controlling agency authority created varies from application to application. For example, in Customs interchange applications, it is the Customs authority itself which normally controls and maintains the necessary user documentation. Other examples include trade associations, trade facilitation organizations, and a secretariat created and funded by the trade itself.

3.2 Initial Development and Design

While it is impossible to specify precisely the technique which should be used for the initial development and design of interchange systems, some guidelines can be given, based upon an analysis of the techniques used for existing systems.

A typical technique used is to create a steering committee chosen from a selection of users from the application area. A series of working sub-groups, each charged with a specific task, report to the steering committee as they progress their work. In turn, these sub-groups are formed from users, having the necessary expertise for the task in hand.

The following list of tasks are typical of those which have been carried out by existing user groups. More detailed subjects might need to be included, depending upon the application area, and two or more of the tasks might be undertaken by one sub-group.

A typical list of tasks for a specific application could be:

- to identify the functions - and therefore the types of messages (or transactions) to be interchanged, with reference to the agreed application data element directory and with particular emphasis on the mandatory or conditional status of each data element within each transaction. Since users in different application areas may wish, in the future, to interchange data, wherever possible standardization of message types and structure within each application area will be to everyone's advantage.

- to identify the data elements required, element sizes and formats, element terminology, and to compile a user data element directory. (This would normally be done with reference to the UN Trade Data Element Directory insofar as international data elements are concerned). For national, or industry specific data elements, local agreement would be necessary.

- to identify the functions of user data segments required, making full use of the UNSM Standard Segments Directory, particularly with respect to standard user data segments designed for multi-industry/multi-application use. Should new user data segments need to be designed, the recommendations contained in the UN EDIFACT Message Design Guidelines should be followed, with 'Change Requests' being submitted to the local Rapporteur's Advisory & Support Team as necessary.

- to specify the level of syntax rules to be used by the application, (i.e. Level A or B - see ISO 9735).

- to specify the method(s) to be used for the physical transmission of data within the application, including where relevant, the specification of requirements for magnetic tape transfer, for floppy disks transfer, and for telecommunications protocols.

- to identify any legal and security problems related to the transfer of information, which might require resolution. (It should be noted that the UN/ICC UNCID recommendations - 'Uniform Rules of Conduct for the Interchange of Trade Data' - address all legal questions which need to be considered. UNCID is included in the UN Trade Data Interchange Directory).

- to identify and recommend common coding techniques to be implemented by all participants in the user group.

- to identify and recommend encryption techniques (if required).

- to recommend the form and period of a trial phase of testing, prior to full implementation.

3.3 User Manual

Taking account of the above subjects, it is recommended that the user manual should at least include:

- a description of the level of EDIFACT syntax to be used

- a description of the agreed character set to be used

- a full and detailed description of the structure of message types to be used. (As far as possible, it is stressed that UNSMs should be used - or if not exactly suitable - that the procedures for amendment set out in the Message Design Guidelines should be followed).

- the user data element directory (utilizing UNTDED as far as possible)

- the user segment directory (utilizing the standard segments defined in EDSD as far as possible)

- the user message directory

- a specification of legal/security requirements to be observed.

- a description of the communications service(s) to be utilized.

- a specification of the transmission record length to be used (which must be standard within the application area)

- a indication of the techniques to be used for error correction, acknowledgements, etc

- an indication whether or not Functional Grouping is to be used.

- an indication of the type of encryption to be used (if any).

- an indication of the type of password facility to be used (if any).

3.4 Check List for the Interchange Agreement

An Interchange Agreement (normally in the form of a User Manual) governs all of the participants subscribing to the application interchange which it describes.

Separate bi-lateral agreements between participants in such an interchange application are not recommended, since they only serve to defeat the purpose of the standards specified for all users in the Interchange Agreement.

For certain data fields in the Service segments which form the syntax, the Interchange Agreement must specify the code sets, lists of qualifiers, etc to be used. The fields and the criteria to be addressed are listed below, with the service data element reference number and the segment in which it appears, in brackets after the field name.

INTERCHANGE SENDER (S002 UNB)

The Interchange Agreement (IA) must specify whether a name or code should be used to identify the organizations sending data. If a code, the various code sets must be identifiable by use of qualifiers, which must be specified.

INTERCHANGE RECIPIENT (S003 UNB)

The IA must specify whether a name or code should be used to identify recipients. If a code, the various code sets must be identifiable by use of qualifiers, which must be specified.

RECIPIENT'S REFERENCE/PASSWORD (S005 UNB)

The IA must state if the field is to be used. If so, either a list of passwords must be maintained, or (more likely), senders must ascertain from their various partners what reference or password is to be provided.

APPLICATION REFERENCE (0026 UNB)

The IA must state whether the field is to be used, if so, it must state what information should be carried in the field.

PROCESSING PRIORITY CODE (0029 UNB)

The IA must state whether the field is to be used, if so, a list of codes and meanings must be provided.

COMMUNICATIONS AGREEMENT IDENTIFICATION (0032 UNB)

The IA must state whether the field is to be used, if so, whether a name or code should appear. If a code, its value must be provided.

APPLICATION SENDER'S IDENTIFICATION (S006 UNG) & RECIPIENT'S IDENTIFICATION (S007 UNG)

The IA must either inform users either that it is their own responsibility to maintain code lists (plus qualifiers if necessary) for their partners, or it should publish and maintain agreed lists for all participants.

CONTROLLING AGENCY (0051 UNG)

The IA must provide the list of codes which could be used (although within one interchange application, it is most likely that only one code would be used. For example, if UNSMs are being used, the code would have the value UN).

MESSAGE VERSION NUMBER (0008 UNG)

If UNSMs are being used, the current version number (and if necessary, the release number) of the message in question should be specified. If the application is using other than UNSMs, then the IA must publish the version numbers (and release numbers if necessary) - see Section 7 Identification and Control of messages.

MESSAGE IDENTIFIER (S009 UNH)

The message identifier field has 5 component data elements. The value of each of these must be specified as necessary per message type in the IA, according to the procedures set out in Section 7, dealing with the identification and control of messages.

3.5 Interchange maintenance authority

It is strongly recommended that some form of interchange maintenance authority be created. Such an authority would be responsible for the control and maintenance of the interchange agreement, with particular responsibility for the production and circulation of amendments to the user manual, and for control of change-over to new versions of messages.

4. TERMINOLOGY

For the definitions of the terminology used in this document, please see Annex A of the ISO 9735 standard.

5. CHARACTER SET FOR INTERCHANGE

In order to cover the range of user preference, two levels of syntax are identified with respect to use of character sets. These levels are defined in the Interchange Header (UNB) segment (in the data element S001 Syntax Identifiers) as UNOA (for the basic level A), and UNOB for level B.

The full character set for both levels is specified in ISO 9735 EDIFACT Syntax Rules.

Level B only, may use a higher level character set taken from ISO 646 IRV, including the use of three of the IS 1-4 non-printable separator characters in place of the printable separator characters used in Level A, as defined in ISO 9735. However, it should be clearly understood that users of Level B must revert to Level A if this is necessary to meet the capability and wishes of the recipient.

Care should also be taken regarding the use of the IS 1-4 separator characters with respect to certain communications protocols. (For example, if the 2780 protocol is used, certain of the IS characters are not passed through to the application level process. In this case, use of the Level A set will overcome the problem).

If not otherwise agreed between interchange partners, the 'bit' representation of the recommended character set for computer-to-computer interchange for both Level A and B is the seven-bit representation specified in the basic ISO 646 standard.

Binary coded decimal or other forms of hardware/software dependent character representation (for examples EBCDIC) must not be used for interchange (other than by prior agreement between interchange partners), as these features are not available, or are not dealt with in the same way, in all makes of computers.

International Reference Version (IRV) ISO 646-1983 (E)

```
            +------------------------------------------------+
            |b7    |  0 |  0 | 0  | 0  |  1  |  1 |  1  | 1  |
            |------|----|----|----|----|-----|----|-----|----|
            |b6    |  0 |  0 |  1 |  1 |  0  |  0 |  1  | 1  |
            |------|----|----|----|----|-----|----|-----|----|
            |b5    |  0 |  1 |  0 |  1 |  0  |  1 |  0  | 1  |
            +------+----+----+----+----+-----+----+-----+----+
+--------------|   |    |    |    |    |     |    |     |    |
|b4| b3| b2| b1|   |  0 |  1 |  2 |  3 |  4  |  5 |  6  | 7  |
|--------------+---+----+----+----+----+-----+----+-----*---D*
|0 | 0 | 0 | 0 | 0 | NUL| DLE|  SP|  O |  @  |  P |  `  | p  |
|--------------+---+----|----|----|----|-----|----*-----*----*
|0 | 0 | 0 | 1 | 1 | SOH| DC1|   !|  1 |  A  |  Q |  a  | q  |
|--------------+---.----.*---*--------------------*---------D*
|0 | 0 | 1 | 0 | 2 | STX| DC2|   "|  2 |  B  |  R |  b  | r  |
|--------------+---.----.*---*--------------------*---------D*
|0 | 0 | 1 | 1 | 3 | ETX| DC3|   #|  3 |  C  |  S |  c  | s  |
|--------------+---.----.*---*--------------------*----------*
|0 | 1 | 0 | 0 | 4 | EOT| DC4| $  |  4 |  D  |  T |  d  | t  |
|--------------+---.---------.--------------------*----------*
|0 | 1 | 0 | 1 | 5 | ENQ| NAK|   %|  5 |  E  |  U |  e  | u  |
|--------------+---.---------.--------------------*----------*
|0 | 1 | 1 | 0 | 6 | ACK| SYN|   &|  6 |  F  |  V |  f  | v  |
|--------------+---.---------.--------------------*----------*
|0 | 1 | 1 | 1 | 7 | BEL| ETB|   '|  7 |  G  |  W |  g  | w  3
|--------------+---.---------.--------------------*----------*
```

```
|1 | 0 | 0 | 0 | 8 | BS | CAN|   (|  8 |  H  |  X |  h  | x  |
|--------------+---.---------.--------------------*----------*
|1 | 0 | 0 | 1 | 9 | HT | EM |   )|  9 |  I  |  Y |  i  | y  |
|--------------+---.---------.--------------------*----------*
|1 | 0 | 1 | 0 | 10| LF | SUB|   *|  : |  J  |  Z |  j  | z  |
|--------------+---.---------.--------------------*-----*---D*
|1 | 0 | 1 | 1 | 11| VT | ESC|   +|  ; |  K  |  [ |  k  | {  |
|--------------+---.---------.*---*---------------*-----*----|
|1 | 1 | 0 | 0 | 12| FF | IS4|  , |  < |  L  |  \ |  l  |  | |
|--------------+---.---------.*---*---------------*-----*----|
|1 | 1 | 0 | 1 | 13| CR | IS3|   -|  = |  M  |  ] |  m  |  } |
|--------------+---.---------.--------------------*-----*----|
|1 | 1 | 1 | 0 | 14| SO | IS2|   .|  > |  N  |  ^ |  n  | ~  |
|--------------+---.---------.---------------*----*-----*----|
|1 | 1 | 1 | 1 | 15| SI | IS1|   /|  ? |  O  |  _ |  o  | DEL|
+--------------+---.-D--.----.---------------*----*-----*---D+
```

Specified by the standard and widely implemented

```
------------
|  2 |  B  |
------------
|  3 |  C  |
------------
```

Specified by the standard, but not implemented by all manufacturers

```
.----.
| STX|
.----.
| ETX|
.----.----
| EOT| DC4|
.---------.
| ENQ| NAK|
.---------.
```

Specified by the standard, but implemented with varying bit patterns

```
      *---D*
      | p  |
*-----*----*
|  a  | q  |
*---------D*
|  b  | r  |
*---------D*
```

6. TRANSMISSION COMPONENTS

An interchange of data in the context of EDIFACT, is composed of one or more messages containing segments which in turn are made up of data elements.

6.1 Data Elements

(NOTE: It is stressed that all the examples of data which follow in this paper are examples. UNTDED should be studied to obtain the current formats, code set and qualifier values, etc).

A data element may consist of a single data item, e.g. '2310 Delivery month' in which case it is called a simple data element, or it may consist of several data items, e.g. the composite data element 'C198 PRODUCT IDENTIFICATION' which consists of two data elements, 7020 Article Number and 7823 Article Number Qualifier. In this case it is called a composite data element; each data item within a composite data element is called a component data element.

A component is identified by its position within a data element. For example, if a data element was required to express the cost of insurance, it would be defined as a composite data element with two component data elements. In the first position would be '5486 Insurance Cost', accompanied by '6345 Currency

Code' as the second component.

The data elements referred to in the EDIFACT standard are either user data elements or service data elements.

User data elements contain the substantive data which are to be transmitted. They are outside the scope of the standard and should be defined and agreed (preferably based on the UN trade data element directory - TDED) between interchange partners, and specified in a user data element directory.

Service data elements contain data required for structuring the transmission. A list of the service data elements provided in this standard and their detailed descriptions are given in the UN Trade Data Element Directory (TDED), in the 'S' and the '000' series. They will also be found in ISO 9735.

Data elements can only be transmitted within a segment.

Each data element in TDED is allocated a unique 4-digit numeric tag. In addition, each data element is allocated a unique and mnemonic four-character data element identifier which must be alphabetic. These identifiers can be used in internal systems, e.g. for system and programming documentation.

6.2 Segments

There are two types of segments: User Data segments and Service segments. Conditional Segments containing no data must be omitted from the message in which they would normally appear.

User data segments contain data elements such as amounts, values, names, places and other data to be transmitted. The contents of user data segments are outside the scope of the UN/EDIFACT Syntax standard. User data segments must not be created with the first two letters of the tag 'UN', as these are reserved for use in service segments.

Service segments contain service data elements such as sender of the transmission, syntax rules type and level, date of the preparation of the transmission, priority type, etc. and/or other specific data which are required for the transmission. All service segment tags start with the two letters 'UN' which are reserved for this purpose. On no account should users change service segments. A 'Change Request' procedure is described in the Message Design Guidelines for the purpose of proposing amendments. The following categories of service segments are provided in the UN/EDIFACT syntax standard:

Transmission structuring syntax segments, which are used to assemble transmissions in a standard way, e.g. to start and end each transmission, to start and end each message within a transmission, and to start and end functional groups of messages within a transmission (if this facility is required).

Segments used in the Service messages 'CONTRL&q' & 'APPLIC', which respectively are used for acknowledgements requests, for correction of syntax errors and rejections, and for confirmation of receipt or requests for correction of application errors. (The latter - APPLIC - is under development).

A segment used in the General Purpose Message 'GENRAL', which is used to indicate the type, title and references for the message.

Segments are identified by a code which uniquely identifies each segment as specified in a segment directory.

A list of the service segments provided in the EDIFACT syntax standard, with detailed descriptions, is given in Annex B of ISO 9735.

6.3 Messages

A Message consists of a number of segments structured in accordance with the syntax rules. It must begin with the service segment 'UNH -Message header' and end with the service segment 'UNT - Message trailer'. It must contain at least one user data segment, containing at least one user data element.

There are two classes of messages: User messages and EDIFACT Service messages.

User messages contain the user data segments required for the message in addition to the 'Message header' and 'Message trailer' segments.

There is an option to transfer a message progressively. At the time of the first transmission, it generally would not contain all of the information as defined in the interchange application message specifications, other than the data defined as being mandatory within the message. In this case, the originator may transmit a selection of the data elements at first, followed by a second (or successive) transmission(s), adding to or updating the data previously transmitted, the data being related by means of a structured, unique key.

(An example might be a Booking message, where the transport operator needs a rough estimate of the space required for the shipment as early as possible to facilitate his planning. The precise details may be supplied by the originator later as they become available, until finally the transport operator has sufficient data to create a bill of lading).

The use of the progressive message transfer technique is explained in more detail in Section 8.3.9 of this guide.

UN/EDIFACT Service messages contain service segments for error correction, either at the syntax protocol level, or at the application level, and service segments for general free text. Their use is explained in Section 10 of this guide.

Messages are uniquely identified by a message identifier field consisting of five component data elements, used to effect identification and control of messages, as explained in the next Section.

7. Identification & Control of UN/EDIFACT Messages

Note: The information in this section and, in particular that on version/release, no longer conforms to the existing Syntax implementation guidelines which need revision in light of the decision at the March 1994 session of WP.4 to implement Standard and Draft directories. Paragraphs where changes have been made are marked at the beginning with an *

7.1 Definition of a UNSM

A United Nations Standard Message (UNSM) is one which:-

i) has been registered, published, and which is maintained by the United Nations Economic Commission for Europe;

ii) has the values contained in the Controlling Agency, Message Type, Message Version Number and Message Release Number fields (the requirements for the use of which are specified in ISO 9735), allocated and controlled by the UN/ECE;

iii) always has the code value 'UN' in the Controlling Agency field.

7.2 Definition of a Sub-set of a UNSM

A Sub-set of a UNSM is a message which is directly derived from an approved UNSM, has the same function as the UNSM from which it is derived, and which:

i) contains all of the groups and segments defined as having a mandatory status within the message, and the mandatory data elements within them. There shall be no change to the status, order or content of the groups, segments, and composite data elements and data elements contained within the segments.

(It should be noted, however, that although many UNSMs contain Conditional Groups of segments which may contain one or more mandatory segments, providing the complete conditional group is omitted from the sub-set, this does not break the rule regarding the inclusion of mandatory segments);

ii) does not change the status, order or content of the segments, composite data elements and data elements in the conditional segments chosen for use from the UNSM;

iii) does not add any segments, composite data elements or data elements to the message;

iv) contains the identical values specified for use in the Message Type, Controlling Agency, Message Version Number and Message Release Number fields, as are specified for the UNSM from which the sub-set is derived.

7.3 UN/EDIFACT Directory Set Issue Numbers

It is essential that messages should be capable of being identified in relation to the current version of the directory from which they are derived, (i.e. the code lists, data elements, composite data elements and segments).

* Whenever a new Standard directory set is published it contains

the message specifications for all registered UNSMs (Status 2 messages) and their supporting segments, composites, data elements and codes.

*   Draft directory sets contain all Status 1 (Draft Recommendation) messages and all Status 2 (UNSM) messages in their latest form and the supporting segments, composites, data elements and codes.

*   A directory will be identified by an issue number, allocated and controlled under UN/EDIFACT procedures. The issue number will be a single Character indicating if the directory contains Draft or Standard material (S or D) followed by a period separator, followed by the last two digits of the year in which the directory is agreed, followed by a sequential alpha character assigned by the UN/ECE. This sequential alpha character begins with A at the start of each year and is incremented if more than one directory of the same type (S or D) is published during the same year.

7.4 Message Version & Release Numbers for UNSMs and for Subsets of UNSMs

*   When a document related to a UNSM under development reaches a Status of 2 (i.e. the status of 'Recommendation'), and the UNSM is agreed and published in a new issue of UNTDID, i.e. in the Standard Directory, the values in the following fields of the UNH/UNG segments used for the message (or a subset of the message) will be:

    i)   Controlling Agency (data element 0051) - always the two characters 'UN';

  * ii) Message Version Number (data element 0052) - Always 'S' when published in a Standard directory, always 'D' when published in a Draft directory.

  * iii) Message Release Number (data element 0054) - The last

two digits in the year of agreement followed by a single, sequential alpha character assigned by the UN that starts with A at the beginning of each year and is incremented if more than one directory of the same type (S or D) is published in the same year.

*7.5 Message Version/Release Numbers for Status 1,Draft Standard

Messages and Sub-sets of Draft Standard Messages

*   When a document related to a UNSM under development reaches a Status of 1 (i.e. the status of 'Draft Recommendation'), and is agreed and published in a new issue of the Draft directory, the values in the following fields of the UNH/UNG segments used for the message (or a subset of the message) will be:

  * i)   Controlling Agency (data element 0051) - always the two characters 'UN';

  * ii) Message Version Number (data element 0052) - Always 'D'.

  * iii) Message Release Number (data element 0054) - The last two digits in the year of agreement followed by a single, sequential alpha character assigned by the UN that starts with A at the beginning of each year and is incremented if more than one directory of the same type (S or D) is published in the same year.

7.6 Message Version/Release Numbers for pre-draft Messages

If users wish to test messages (or sub-sets of messages) which have not yet reached the 'Draft for formal trial' stage (i.e. messages under development which have a document status of 'O' or 'P'), a different procedure must be followed.

The full procedures for the identification of documents containing messages under development are contained in the UN paper WP.4/GE.1/R.785. Such documents will have a Status of 'O', plus a 'Revision' number controlled by the Rapporteur's Team where the request for the new UNSM originated.

Users wishing to test such messages must always use a Message Version number of zero, a Message Release number equivalent to the Revision number of the document revision upon which they are basing their test, and a 'Controlling agency' code of 'RT'. (Users should not test such messages until they have been passed by the relevant Technical Assessment Group of the RT where the request for the message originated. Further, users are strongly recommended to delay testing of messages under development until the point where the development is fairly stable, and even then, they must be aware that the message content may well change before it reaches the status of 'Draft for formal trial').

Example: (Document Status 'O' or 'P')

```
  S009 MESSAGE IDENTIFIER
* 0065   Message type               :     NEWMSG
  0052   Message version number     :     O
  0054   Message release number     :    'n'
  0051   Controlling agency         :     RT
  0057   Association assigned code  :    (not used)
```

Where 'n' in the release number field is equal to the Revision number of the development document used as the basis of the message being tested.

7.7 User Conventions for the Implementation of Sub-sets of UNSMs

United Nations Standard Messages are structured in such a way that they can be used by companies and organizations in many different industries. For example, the invoice UNSM contains data elements and segments which are in common use in the majority of invoicing applications. Other data elements and segments specified for use in the message have a more restricted application, and will probably be required by only a few industry applications. Thus, in the vast majority of cases, industries will choose and become responsible for specific industry related sub-sets from the total message structure. (The definition of a true sub-set defined above).

However, still abiding by this principle, users may wish to go beyond the standard sub-set definition, and for reasons of integrity, agree between a group of participants that certain data elements and/or segments which are conditional in a UNSM should always be required in their application. In choosing to go beyond the true sub-set definition set out in paragraph 2 above, users must bear in mind that to comply with the spirit of sub-sets, any sub-set must always be more restrictive than its parent UNSM.

To provide a unique identification for any particular sub-set of a UNSM, users may wish to assign a code for use in the 'Association assigned code' field of the UNH and/or UNG segments. Further, if it is considered that there may be a problem in assigning a code which may be duplicated by another group of users, it is suggested that the local Rapporteur's Team be consulted regarding the allocation of a code value.

7.8 User Conventions for the Implementation of UNSMs under Amendment

As UNSMs are used by more industries, it is quite likely that some messages will be found not to cover some of the specific requirements of those industries. To provide these requirements so that the messages can be used, immediate changes to (or additions of) segments and elements may be necessary by use of the official UN/EDIFACT 'Change Request' procedures.

Since the standards maintenance time-scales may delay the implementation of the required modifications to the UNSM for some time, users may wish to implement the changes immediately so that the message can be used in their application.

In order to identify the amended message (which then is NOT a UNSM) during the interim period, the user group concerned should consider including an appropriate code in the 'Association assigned code' field of the UNH and/or UNG segments. Where it is thought that there may be problems of duplicated Association assigned code values, the local Rapporteur's Team should be consulted regarding the allocation of a code value.

As an alternative, in order to identify the group of users requesting the amendments to the UNSM, in the interim period of use of the message, the 'Controlling Agency' data element should be used for this purpose.

CHAPTER 2.2 - General Information: UN/EDIFACT Application Syntax Rules (ISO 9735) Issue 94.2, March 1994

The standard ISO 9735, Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) - Application level syntax rules, has received an Amendment 1 dated 1992-12-01.

This Amendment 1 is as follows:

Clause 2 (Normative references)
===============================

Add a footnote reference '1)' after the title of ISO 8859.

Add a footnote as follows:

```
-------------------
1) See also annex B.

Clause 4 (Syntax levels)
========================

Add the words ', see annex D' at the end of the second sentence in
the first paragraph.

Add the words 'and annex D' after the words 'clause 5' in the third
paragraph.

Annex D
=======

In summary, this annex contains the codes for data elements 0001,
syntax identifier, in the interchange header UNB and describes the
corresponding character sets.

               -------------------------------------

Note: The earlier versions of ISO 9735 (1988-07-15 and 1990-11-01)
are still usable for the interchange of EDIFACT messages. The 1988
version is identified by a value of 1, and the 1990 version by a
value of 2 in the mandatory data element 0002 (Syntax version
number) in the segment UNB (Interchange header).

When this 1992 version is used the value in data element 0002 shall
be 3.
```

8. Basic UN/EDIFACT Syntax Rules

```
The UN/EDIFACT syntax rules apply only to the data to be interchanged.

They are independent of the type of computer to be used for the
interchange application, whether mainframe, mini or micro.
```

They are also independent of the data to be interchanged and the data standards in use; new data elements or data segments can be introduced and existing data elements and data segments changed (by use of the maintenance procedures) without affecting the rules.

The syntax rules are independent of both the type of application (i.e. commercial, governmental, transport etc.), and of the telecommunications protocols or interchange media to be used. For example, a range of different applications are successfully using them on packet switched services, SNA, 2780, 3780, magnetic tape etc. Therefore it can be seen that the data to be interchanged sits inside the 'envelope' provided by the data communication transport service.

8.1 Interchange Structure

As previously defined, in the syntax rules data elements are carried in user data segments, while service segments contain service data elements which form the structure of the syntax rules protocol.

In OSI terms, a connection could include one or more EDIFACT interchanges, each separated from the other by control service segments which identify the start and end of each interchange. Within each interchange, there is then a hierarchical structure which allows for both control and identification of data for processing. This structure is shown in Section 6 of ISO 9735.

The syntax deals with the representation of the syntax protocol and user data only, and not with the requirements of the technical transmission aspects of telecommunications protocols, media labels, etc.

Further, it should be pointed out that UN/EDIFACT in no way encroaches upon the OSI concept. In an UN/EDIFACT interchange, everything from the first character of the Interchange Control Header segment to the last character of the Interchange Control

Trailer segment, is user data, emanating from one computer system in a structure agreed by the users, for receipt into another computer system at the application level.

8.2 Use of the character set

In general, all of the characters specified in the User Interchange Agreement, (see section 3), can be used in data.

8.2.1 Relationship to syntax control characters

If Level A syntax is being used,, it is recommended that the following four characters of the recommended character set, namely:

- the plus sign (+)
- the colon (:)
- the apostrophe (')
- the question mark (?)

should not deliberately be chosen for use in data elements, as they are reserved for use within the EDIFACT syntax rules rules as syntax characters for Level A.

However, it should be pointed out that in practice in live applications, this causes few problems, because they rarely appear in genuine data. If they do, it is a relatively trivial task for the program which formats the data into the syntax structure to insert a release character where appropriate, and for the program which de-formats the data to remove the release character - thus permitting that character to be passed on to the recipient's system as a character which is part of the user data.

8.2.2 Syntax separators, terminator and release character

```
+---------------------------------+---------------+------------¿
```

Name	Separator	Separator
Function of Separator	in Level A	in Level B
Segment terminator (To terminate all segments. A data element separator must not be used after the last data element in a segment.)	' (apostrophe)	ISO646 IS4
Segment tag and data element separator (To separate all segment tag service data elements from the first user data element in the segment, and to separate simple and/or composite data elements in a segment from each other.)	+ (plus sign)	ISO646 IS3
Component data element separator (To separate all component elements in a composite data element from each other.)	: (colon)	ISO646 IS1
Release character (To release any of the characters + ; ' ? appearing in user data in Level A syntax. It MUST immediately precede the character in question and signifies that the NEXT single character is not to be interpreted as a syntax separator, terminator, or release character.)	? (question mark)	NOT USED

Examples

Level A syntax separators

```
NAD+BY+ABC CO:26 MAIN STREET:LONDON:TW17 9NW'
     where: NAD is the unique segment code for a Name and Address
          segment
          BY is a qualifier meaning 'Buyer', thus identifying
          the function of the NAD segment
          ABC to 9NW is a composite data element
```

Level A release character

```
SEG+75?+73?+ABC+HOW MANY PACKAGES??'

     Where:  In the users system, the first data element appears as
          75+73+ABC and the second data element appears as HOW
          MANY PACKAGES?
```

The release character is not counted as part of the length of any data element or component data within which it is transmitted. Release characters can be inserted by program so that data can be input and output without any special manual requirements.

Level B syntax separators

Note: The information separators IS1; IS3 and IS4 are described and their coded representation specified in clause 4 of ISO 646. They are control characters and thus non-printable. However, in the example below they are shown in brackets thus (IS1) to illustrate their use.

```
NAD(IS3)BY(IS3)ABC CO(IS1)26 MAIN STREET(IS1)LONDON(IS1)TW17 9NW(IS4)
```

8.3 Interchange formatting rules

8.3.1 Interchange structure

As previously defined, service segments contain service data elements which are required at the application level for interchange or syntax control.

A set of user interchange data must start with a 'UNA Service String Advice' if for any reason neither the Level A or B the syntax separators, as defined in the standard, are used in the interchange which follows.

If it is possible that the application may have to process interchanges preceded by the UNA string, you must ensure that your de-formatting software can process the data, which effectively is using 'non-standard' separator characters. One technique for achieving this is to have a routine which, when the UNA string is detected, dynamically updates the module which processes and checks the syntax separator characters.

The set of user interchange data following the service string advice (if used) must start with the service syntax segment 'Interchange Control Header' which has the segment code UNB.

The set of user data must be terminated with the the 'Interchange Control Trailer' which has the segment code UNZ.

If several sets of user interchange data are included in one transmission, (i.e. there are several pairs of UNB and UNZ segments), each UNB segment must be preceded by a delimiter string advice if neither Level A or B separators are being used.

With the exception of these service segments used to delimit a transmission (and two other service segments which are used to identify functional groups within a transmission), all data in a transmission must be interchanged within a message.

A transmission may contain one message or any number of messages.

A transmission in general form can therefore be depicted as:

```
     +--- UNB segment
     |
     |        message(s)
     |
     +--- UNZ segment
```

The syntax rules do not specify the order in which messages of different types should be transmitted within an interchange. The sender can forward messages in an order of his choosing, unless partners in an interchange application agree otherwise, or unless otherwise stated in the Interchange Agreement.

8.3.2 Service string advice - UNA

The string has a mandatory fixed length of 9 characters. The first three are UNA, immediately followed by the 6 characters as defined in ISO 9735.

NOTE: ONLY in very special circumstance, (for example, at level A Syntax, if a user application group were interchanging only data related to mathematical equations, or, at level B syntax, if it were found that any IS character from ISO646 had been utilized for a specific function by a network or by a communications protocol), should any other delimiters than those detailed for Levels A and B be used.

8.3.3 Interchange control header segment - UNB (See ISO 9735 Annex B for detailed specification)

In addition to the segment code UNB, the following mandatory service data elements must be included in the following order:

- the syntax identifier and version number;
- the interchange sender;
- the interchange recipient, plus an optional (normally coded) sub-address for onward routing;
- the date and time of the transmission; and
- the interchange control reference.

Some, or all of the following conditional service data elements can be included in the segment, if specified for use in the Interchange Agreement. If included, they must be in the following order:

- the recipient's transmission reference (or password to be provided by the sender);
- the application reference;
- the processing priority code;

. an acknowledgement request;
. the communications agreement identification; and
. a test indicator.

(The recipient's transmission reference field may optionally be used to contain a password instead of a transmission reference). Clearly, such use must be specified in the Interchange Agreement. It may be required for example, where a group of users are interchanging their data via a bureau mailbox service, or where a password is needed to gain access to the recipient's system. If functional grouping is being used in the interchange, the facility exists to include a password in each group header. To gain access to the recipient's sub-systems, if required.

As identified in Section 2, some of the above data may not be held in the in-house system. If not, it can be provided at run-time via a parameter file.

The last field in the segment, the 'Test Indicator', is used during the run-up phase to live working, since its use enables the recipient to identify all the data contained in the interchange as for trial use only. Clearly, care must be taken to set the field to zero as soon as live interchanges are started. An example UNB segment containing all of the conditional data elements and using level A syntax would be transmitted as:

```
UNB+UNOA:1+123:AB:PO168+3572:DN:B1342+771206:121500+A143+B26AZ+
DELINS+X+1+CANDE+1'
```

where:

UNB   is the segment tag code.

UNOA:1 identifies version 1, level A of the syntax rules and Controlling Agency UNO. (For level B, the field would contain UNOB:1).

The purpose of the version number is to allow for maintenance of the standard. Each future amendment, or groups of amendments to the syntax, will cause the version number to be incremented by 1.

123:AB:PO168 identifies the sender of the transmission in code with a qualifier of AB to identify the code set being used, followed by a code representing a reverse routing address to which the recipient should address any reply.

3572:DN:B1342 identifies the recipient of the transmission in code (qualified by DN), plus a sub-address code. The sub-address code for onward routing may be used if the functional grouping facility, (which also provides for sub-addressing), is not used.

771206:1215 771206 is the date and 1215 is the time of the preparation of the transmission. This is the date/time that the interchange is assembled for transmission.

A143 is the unique interchange control reference for this transmission, allocated by the sender of the interchange.

B26AZ is recipient's reference, or a password. (This can be accompanied by a qualifier component element, if so specified in the Interchange Agreement) the field is left blank if not used

DELINS is an example of an application reference. A common usage of this field is to keep all messages of the same type within one unique transmission, and to carry the appropriate message identifier in this field. Such usage allows a transmission of a particular message type to be retrieved by the recipient from a mailbox service in advance of transmissions containing a different message type. The technique would not be used if either Functional Grouping or an interchange containing a mixture of different message types were being used, in which case it would be left blank.

X is a processing priority code, using a code defined in the Interchange Agreement (or left blank if not used)

1 indicates that the sender is requesting an acknowledgement for the interchange, but only that the recipient has successfully received and identified the header and trailer segments (UNB & UNZ) for the interchange. The recipient would reply, using a 'CONTRL' message (see Section 10). Such an acknowledgement does not mean that the contents of the interchange have been processed correctly and are acceptable to the recipient. The field is set to zero if an acknowledgement is not required.

CANDE is an example of a code specified in the Interchange Agreement, which identifies the type of communications agreement under which the interchange is controlled, (or left blank if not used).

1 indicates that this is a test transmission. The field is set to zero for transmission of live data

8.3.4 Interchange control trailer segment - UNZ (See ISO 9735 Annex B for detailed specification

In addition to the segment code UNZ, this control segment contains two mandatory service data elements. The first, the interchange control count, contains either a count of the number of messages in the interchange, or a count of the number of functional groups in the interchange if that facility has been used (see Section 8.3.5).

The second data element, the interchange control reference, contains the identical reference to that carried in the same field of the UNB interchange header segment for the same interchange. Checking that the two fields are identical ensures that the set of interchange data has been successfully received.

A UNZ segment which indicates a transmission with an interchange control reference of A143, containing 7 functional groups, would be transmitted as: UNZ+7+A143'

For a transmission with the same reference where functional grouping has not been used, and which contains 2500 messages, the UNZ segment would be transmitted as: UNZ+2500+A143'

8.3.5 Functional group structure

Messages in a transmission can be assembled into one or more functional groups. For interchanges of data to/from North America, the use of functional grouping is mandatory. For other interchange applications, its use is optional, but should be specified in the interchange agreement.

It is not permitted to mix the use of functional groups with messages not contained within functional groups in the same interchange.

Each functional group must begin with the control service segment 'Functional Group Header' which has the segment code UNG. Each functional group must end with the control service segment 'Functional Group Trailer' which has the segment code UNE.

An interchange consisting of a single functional group of 'n' messages can therefore by depicted as:

```
+--- UNB segment
|
|    +- UNG segment
|    |
|    |
|    |  first message
|    |  .......
|    |  'n-th' message
|    |
|    +- UNE segment
|
+--- UNZ segment
```

An interchange consisting of 'n' functional groups, can be depicted as:

```
+--- UNB segment
|
|    +--- UNG header of the 1st functional group
|    |
|    |       messages
|    |
|    +--- UNE trailer of the 1st functional group
|
|
|    +--- UNG header of the 2nd functional group
|    |
|    |       messages
|    |
|    +--- UNE trailer of the 2nd functional group
|
|        ...
|
|    +--- UNG header of the 'n-th' functional group
|    |
|    |       messages
|    |
|    +-- UNE trailer of the 'n-th' functional group
|
+--- UNZ segment
```

8.3.6 Functional Group header - UNG (See ISO 9735 Annex B for detailed specification)

The main benefit of the use of functional grouping is that it permits large organizations which have several functional processes, (e.g. purchasing, invoicing etc) or data processing centres (for example at a divisional or departmental level), to create their own identifiable application level envelopes, which can also be addressed from the originating department to a department in the recipient's system.

An example functional group, (which has the segment code UNG), could be transmitted as:

```
  UNG+INVOIC+15623+23457+860606:183500+CD1352+UN+89:1+A3P52'
```

where:

UNG is the segment tag code

INVOIC is the functional identification, used to identify the one message type contained within the functional group

15623 is the Application Sender's Identification, which is a code identifying a specific division, department, section, etc., which has originated (or is responsible for) the messages contained in the functional group. If necessary, the data element can contain a second component of a qualifier, to identify the code set being used.

23457 is the Application Recipients Identification, a code identifying a specific division, department, section, etc., to which the messages in the functional group are finally destined. If necessary, it too can be qualified by a second component to identify the code set being used.

860606:1835 is the date and time that the functional group of messages was assembled. (NOTE that the time, and perhaps the date, will often be earlier than the date and time carried in the UNB interchange header segment).

CD1352 is a unique reference number for the functional group, assigned by the division, etc., which originated the group of messages

UN is the controlling agency code (see Section 7), which identifies the agency responsible for the production and maintenance of the message standards for the message type contained in the group

89:1 is the version and release number of all of the messages in the group, which must all be the same message type. This composite data element could contain one additional component data element for an association assigned code. It should be noted that if conditions also demand the presence of a number in the association assigned code field, the same data for the composite need not be repeated in the equivalent fields of the Message Header (UNH) service segment preceding each message in the Functional Group.

Usage is fully explained in Section 7.

A3P52 is an application password, and is the only conditional data element in the segment, all the rest being mandatory. The password is only used if specified in the interchange agreement (or if agreed bi-laterally) and could, for example, be used to gain entry to the recipient's sub-system which will process the functional group

The example above shows the typical use of the functional group facility, however, it should be noted that the facility for grouping messages of the same type together may still be used without using the internal system addressing capabilities from and/or to sub-systems in the sender and/or recipient organizations. In this case, the second, third, and fourth data elements in the group (Application Sender's Identification; Application Recipient's Identification and Date/Time of Transmission) could contain identical data to that contained in the comparable fields in the UNB interchange header segment (i.e. Transmission Sender; Transmission Recipient and Date/Time of preparation). For this type of usage the last data element in the functional group header (Application Password) would be omitted. Thus, the whole interchange and the functional groups contained within it, can be addressed point to point.

8.3.7 Functional Group trailer segment - UNE (See ISO 9735 Annex B for detailed specification)

In addition to the segment code UNE, this service segment contains two mandatory service data elements.

The first, 'Number of messages' contains a count of the total number of messages in the functional group.

The second, 'Functional group reference' contains the identical reference to that carried in the same field of the UNG header segment for the functional group. Checking the two fields to be identical ensures that the functional group has been successfully received.

A group trailer for a group of 72 messages with a group reference of CD1352, would be transmitted as:

```
  UNE+72+CD1352'
```

8.3.8 Message structure

A message must begin with the service segment 'Message header' which has the segment code UNH. A message must end with the service segment 'Message trailer' which has the segment code UNT. In addition to these two delimiting segments, a message may contain one or more user data segments required for the message.

A message containing one user data segment to be transmitted can therefore be depicted as:

```
 . UNH segment
   |
   |   data segment
   |
 . UNT segment
```

A message containing ' n ' segments to be transmitted can be depicted as:

```
 . UNH segment
   |
   |   first data segment
   |   ...
   |   n-th data segment
   |
 . UNT segment
```

The syntax rules do not specify the order in which these user data segments should be transmitted within a message. This is a function of message design. The Interchange Agreement must contain specifications for all interchange messages and their segments as required by the user group.

8.3.9 Message header control segment - UNH (See ISO 9735 Annex B for detailed specification)

This segment, used for both data and service messages, has the segment code of UNH. It contains two mandatory service data elements:

- the message reference number;
- the message identifier;

and two conditional data elements for use with progressive message transfers only:

- the common access reference
- the status of the transfer

The message reference can be either:

- a program generated reference number starting at 1 for the first message in the interchange, and incremented by 1 for every successive message in the interchange; when the Functional Grouping technique is not being used; or

- when Functional Grouping is being used, a program generated number starting at 1 for the first message in each functional group, and incremented by 1 for every successive message in each functional group; or

- a meaningful reference number provided from the originator's in-house system.

The two techniques of program generated or meaningful reference numbers must not be mixed within an interchange.

The preferred technique should be specified in the Interchange Agreement. Most live systems use the former techniques, with the numbers being program generated.

The message identifier is a composite data element, having five component data elements:

- the type of message (mandatory)
- the version number of the message (mandatory)
- the message release number (conditional)*
- the controlling agency (conditional)*
- the association assigned code (conditional)*

* (NOTE: If Functional Grouping (See Section 8.3.6) is not being used, the controlling agency field becomes mandatory in the UNH. If conditions demand the presence of data in the release number, and/or association assigned code fields, (see Section 7) then these too must be provided in the UNH if Functional Grouping is not being used.

The use of message release number, controlling agency and association assigned code has been fully explained in Section 7.

The type of message identifier is of variable length 6 alphanumeric, e.g. INVOIC for invoice messages, or 850 for a purchase order transaction set (a non-UNSM message).

The version and release numbers associated with the type of message are both variable length, 3 numeric. The rules for the control of version and release numbers are explained in Section 7.

The common access reference (CAR) is a variable length alpha-numeric field, with a maximum of 35 characters. Within this maximum, any combination of sub-elements may also be specified according to user preference. Clearly however, all users in an interchange group must use the same format, which must be specified as part of the group's interchange agreement.

The purpose of the reference is to serve as a key to which all subsequent transfers of data for the same business file, can be related. Therefore, the same reference must be included in the UNH of all related transfers.

The status of the transfer has two component data element the first a variable length numeric field, with a maximum of 2 characters, which can have the values:

- 1 to 99 (indicating the sequence of transfers of data starting at 1 for the first transfer incremented by one for each additional transfer),

The second sub-element is a fixed length field of one alphabetic character, which can have the following values:

- C this must be present for the first transfer of data related to the common access reference if more than one transfer is foreseen, (i.e. indicating that a file is to be created, using the CAR as its key)

- F this must be present for the final transfer of data related to earlier transfers for the same CAR key.

The message reference, message identifier, common access reference, and the status of the transfer are all used for progressive transfer messages, related to a business file.

The concept of progressive message transfer is that after the initial transmission of data related to a business file, successive transmissions can be used, either to upgrade or amend previously transmitted data related to the same business case, or to add new items of data. The decision as to what data is transmitted at any given time is under the control of the originator.

Each transmission related to the same business file is linked by means of a unique common access reference (CAR), carried in the

MHD segment. This reference is imposed by the originator of data, (in practice, the principal), and is held as a key by all other partners with whom the principal is in communication.

The trade file thus created, covers the whole set of data necessary to carry out the task(s) proper to a specific party concerned with a trade transaction. This party receives into his own application the information made available by the sender, which may be data useful at a given time to allow initial processing to take place. An example could be an initial booking for transport, where the full detail of the goods to be carried is not yet available.

The common access reference, unique to every business file, is the key by which the various transmissions of progressive transfer messages coming from or going to any specific trade operator are linked together. Within an exchange between partners, the CAR permits linkage of the different elements of

required information to a series of operations connected with

the same business file:

Example

```
  Operation 1 - Order
  Operation 2 - Dispatch advice
  Operation 3 - Delivery note
  Operation 4 - Insurance Certificate
  Operation 5 - Invoice
```

From this, it can be seen that any data common to two or more of these operations between two parties, need only be transmitted once.

The progressive message transfer technique permits the transmission of data which is available at a given time, sufficient to allow the recipient to act upon the information.

8.3.10 Message trailer control segment - UNT (See ISO 9735 Annex B for detailed specification)

In addition to the segment code UNT, this segment contains two mandatory service data elements.

The first, 'Number of segments in a message' contains a count of the total number of segments in the message, including the UNH and UNT segments.

The second, 'Message Reference Number' contains the identical reference to that carried in the same field of the UNH message header segment for the same message.

8.3.11 Section control segment - UNS (See ISO 9735 Annex B for detailed specification)

Some UN Standard Messages (UNSM's) have been designed having three distinct logical sections, the first containing what may be termed user header data for the message, the second containing detail information (which will often repeat within the message), and the third containing summary information related to the contents of the message.

In this type of structure, the situation may arise where identical user data segment(s) may be required in more than one of the sections, but containing different values for their data elements, (or in some cases overriding the data carried in the identical segment in the header section). In order to permit the precise identification and control of this situation, a control segment is provided to delimit the three sections.

An example of the function of overriding data carried in an earlier segment could be as follows. A purchase order message could specify in the header section a delivery address to which the (majority of) the order should be delivered. The same segment containing this information could be repeated in the detail section, related to a specific line item of the order. In this case, the delivery address related to the line item only, overrides the address given in the header section (which could well apply to all of the other line items in the detail section).

The section control segment has a segment tag code of UNS and contains one data element. When used to delimit the header section from the detail section, it contains the value 'D', and when used to delimit the detail section from the trailer section, it contains the value 'S'.

When used, UNS segments must be specified in their correct positions within the relevant message(s) in the message directory. If messages are designed having a different structure - for example, where it is only necessary to separate a summary section from the rest of the message - then only the relevant UNS segment should be used. No more than two UNS segments should be used.

An example of the use of the UNS segment is shown below:-

```
UNH+........data..........'       ) Message Header
     AAA+..........data............' ]
     BBB+..........data............' ]User data segments
     CCC+..........data............' ]forming the header section.
     UNS+D' - Section control segment separating the header and
          detail sections
     BBB+..........data............' ]User data segments
     FFF+..........data............' ](including an identically
     GGG+..........data............' ]described BBB segment)
     HHH+..........data............' ]forming the detail section.
     UNS+S' - Section control segment separating the detail and
          summary sections
     III+..........data............' ] user data segments
     JJJ+..........data............' ] forming the summary
     KKK+..........data............' ] section.
     UNT+.........data...........'
```

9. SEGMENT CONSTRUCTION AND TRANSMISSION RULES

9.1 Segment Composition

A segment must start with a tag, which is a mandatory composite service data element.

The first component data of the tag is mandatory, and contains a unique code to identify the segment. Service segment codes are defined in the EDIFACT standard and must not be changed.

User data segment codes are allocated by the Controlling Agency responsible for the message and must be unique within the application(s) in which they are used.

The remaining component data elements of the tag are conditional. They are only used if the segment can repeat within a message, and when it has been stated in the message specification that control numbers to indicate the level at which the segment in question is repeating within the message are to be transmitted, (explicit representation). Section 9.5.3 details the use of this technique.

The structure of all segments therefore, is:-

```
      SEG+........data elements.......'
```

where: SEG is the code for the segment tag.

Segments may contain one, or several related user data elements, which may be either simple and/or composite, as previously defined. For a segment to be transmitted, it must contain data for at least one data element.

The order of the data elements within the segment must be specified in a pre-defined sequence. For data segments, the order and content must be specified in the interchange application segment directory, along with the data segment code.

At the segment level, every segment must be defined as being either mandatory or conditional within a message. Mandatory segments must be transmitted if the message is present, whereas conditional segments may be transmitted depending upon the pre-agreed conditions defined in the message specification.

Data elements within segments, and component data elements within composite data elements, must also be defined as being either mandatory or conditional. If a data element or component data element is mandatory within a message, then the segment in which it appears must also be defined as being mandatory. Conditional data elements or component data elements within such a segment may be transmitted depending upon the pre-agreed and specified conditions.

A conditional segment in a message may, however, contain one or more data elements or component data elements which are mandatory when the segment is used in the message. In this case, such data elements are mandatory at the segment level, rather than at the message level, and such component data elements are mandatory at the data element level.

The syntax separators used will depend upon whether level A or level B syntax is being used.

9.2 Absence of Data

9.2.1 Absence of a segment

Where no data needs to be transmitted for a conditional segment, the complete segment (including the segment tag) must be omitted.

9.2.2 Absence of data within a segment

As described earlier, the order of data elements within a segment must be specified in a pre-defined sequence. This allows the receiving system to identify and process each individual data element. It follows therefore, that if data is omitted at the beginning, or the middle of a segment--followed by data which is present, some means is necessary of recognizing the fact. In the examples which follow, the component data elements of the segment tag, (which can be used to indicated explicitly, the level at which the segment repeats), have been suppressed, and therefore do not appear.

The absence of data for one or more conditional data elements preceding other data in the segment which is present, is indicated by the data element separator + for each missing data element:

```
Example:  SEG+DE1+DE2+DE3+DE4+DE5'   A segment with all data
                                     elements present.

          SEG++DE2+DE3+DE4+DE5'      A segment with DE1 omitted.

          SEG+DE1+DE2+++DE5'         A segment with DE3 and DE4
                                     omitted.
```

Where no data is required for one or more conditional data elements at the end of a segment, the preferred technique is to use the segment terminator to truncate the segment following the last data element for which data is required:

```
Example:  SEG+DE1+DE2'          A truncated segment with DE3,
                                DE4 and DE5 omitted.
```

Alternatively, data element separators can be used to indicate positively the absence of data for each, or some, of the data elements not required at the end of a segment.

```
Example:  SEG+DE1+DE2+++'       A segment with DE3, DE4
                                and DE5 omitted (no + permitted
                                after DE5, since it is the last
                                element in the segment)
```

It should be appreciated however, that if unwanted data element separators at the end of segments are not truncated and there are a high percentage of such separators, an interchange of many thousands of characters could take appreciably longer to process.

A further possible difficulty if there are certain types of telex links in the interchange chain, is that a string of 4 consecutive plus-signs can cause some automatic telex systems to abort the transmission.

The absence of data for one or more conditional component data elements within a composite data element is indicated using similar principles to those described above. A data element separator must be inserted following the last component element for which data is required within a composite element. The absence of data for one or more component elements preceding another component element which is present, is indicated by the component element separator : for each missing component data element:

```
Example:  +CE1:CE2:CE3:CE4+     A composite data element with
                                four component elements, all
                                present.

          +:CE2:CE3:CE4+        A composite data element with
                                CE1 omitted.

          +CE1:CE2::CE4+        A composite data element with
                                CE3 omitted.

          +CE1:CE2+             A truncated composite data
                                element with CE3 and CE4
                                omitted.

          ++                    A truncated composite data
                                element with CE1, CE2, CE3 and
                                CE4 omitted.

          SEG+:CE2:CE3:CE4+     A composite data element
                                following a segment tag with CE1
                                omitted.
```

9.3 Suppression of Non-significant Characters

To increase efficiency of transmission and to limit processing overheads at reception, it is recommended that only the significant characters of data elements and component data elements defined as being variable length for the interchange application, should be transmitted.

If in an in-house file, a fixed numeric field of 10 digits is defined as variable length for the interchange segment and has only two significant digits '12', the leading zeros can be suppressed:

Example:

the 0000000012 form in the in-house system becomes the +12+ form when suppressed.

If in an in-house file, an alphanumeric/alphabetic field of 10 characters is defined as variable length for the interchange segment and has only two significant characters 'AB', the trailing space characters can be suppressed,

e.g. ....AB spaces........ becomes

+AB+ when suppressed.

Leading spaces must not be suppressed.

e.g. .....spaces AB spaces..... in the in-house system becomes

+spaces AB+ when suppressed.

This can be summarized as variable length numeric fields being right justified and variable length alphanumeric and alphabetic fields being left justified.

It should be pointed out however, that recipients must have the capability of accepting either suppressed or non-suppressed variable length data, accepting however that processing overheads will be increased if suppression is not used.

Any further types of compression, for example those offered by some proprietary telecommunications protocols, are outside the scope of EDIFACT.

9.4 Negative Values

Where negative values are required, a leading minus sign (a hyphen), should be transmitted before the numeric value.

```
 e.g.  inhouse form        1352|negative

      becomes           -1352 for transmission
```

Numeric fields transmitted without a sign are assumed to be positive, including where the data element description has an implied negative value (e.g. 'Debit'). It is the function of the in-house process after receipt to take account of this type of data.

9.5 Repeated and nested segments

A common requirement for many types of message is the need to repeat segments. For example, an invoice could contain a number of items, each item containing a product code, quantity, price, etc.

Sometimes several repeats of a segment can occur within an already repeating segment, for example, there could be several goods items in a container and several containers within a consignment. The data elements for a goods item are grouped together in one repeating data segment while the details for each container are grouped in another repeating segment, at a higher level.

There are two types of structure which can occur in such repeating or repeating and nested segments. These can be defined as being vertical or horizontal. To meet some more complex requirements, the two structures can be combined within a specific message.

For example, a message containing no repeating segments, only non-repeating data segments AAA, BBB and CCC, can be expressed diagrammatically as follows:

```
                  _____________________________
                  |      |      |      |      |
      Level 0    UNH    AAA    BBB    CCC    UNT
```

Each segment (including the service segments UNH and UNT) appear at a hierarchical level of zero, and in the transmission, each segment will appear only once in each message of the same type in character string form as:-

```
UNH+...data...'AAA+...data...'BBB+...data...'CCC+...data...'UNT+5'
```

For easier legibility in examples, this can also be shown as:

```
      UNH+....data....'
      AAA+....data....'
      BBB+....data....'
      CCC+....data....'
      UNT+....data....'
```

9.5.1 Repeating segments

A message may also contain one (or more) segments which may repeat in the message.

```
                 ___________________________
                 |     |    |    |    |    |
e.g. Level 0    UNH   AAA   |   BBB   |   UNT
                            |         |

     Level 1               GDS       FTX
```

where data segments AAA and BBB are non-repeating segments at level zero, whereas GDS and FTX are repeating segments at level 1.

Segments at Level 0 are non-repeating segments having a status of either M 1 or C 1.

Segments at Level 1 are either:-

i) a segment which can repeat 2 or more times and which does not start a group of segments, or

ii) a segment shown as having one or more occurences, which starts a group.

Segments at Levels 2 to 'n' are those which appear consecutively below those at Levels 1 to n-1, having a hierarchical relationship to the segment(s) above.

How many times they can repeat within the message is something which is dictated by the application for which they are specified, and which must be stated in the relevant message specification.

A repeating segment can be defined as being 'Conditional', repeating up to a maximum of 'n' times. If there is no data for such a segment, it would not appear at all in the message.

Alternatively, a repeating segment can be defined as being 'Mandatory' repeating up to a maximum of 'n' times. In this case there must be at least one occurrence of the segment in the message.

Diagrams greatly assist users in understanding message structures. This is particularly true with regard to the level and status of each segment, and the number of times each segment can appear in the message.

The conventions which should be followed are that each segment in the diagram, (or groups of segments as will be seen later), should be 'boxed' with the status shown in the bottom left corner of the box, and the maximum number of occurrences shown in the bottom right corner of the box.

Using this convention, the above diagram could become:-

```
             ___________________________
             |     |    |    |    |    |
           +---+ +---+  |  +---+  |  +---+
Level 0    |UNH| |AAA|  |  |BBB|  |  |UNT|
           |---| |---|  |  |---|  |  |---|
           |M|1| |M|1|  |  |C|1|  |  |M|1|
           +---+ +---+  |  +---+  |  +---+
                        |         |
                        |         |
                      +----+    +---+
Level 1               |GDS |    |FTX|
                      |----|    |---|
                      |M|50|    |C|5|
                      +----+    +---+
```

This implies the following structure:-

All of the segments, if they appear, must appear in the order shown

UNH is mandatory and must appear once in the message

AAA is mandatory and must appear once in the message

GDS is a repeating segment at Level 1, it must appear at least once in the message, and can repeat consecutively up to a maximum of 50 times

BBB is a conditional segment which may be omitted if it contains no data, or may appear only once in the message

FTX is a repeating segment at Level 1, which may be omitted if there is no data for the segment, or can repeat concurrently up to a maximum of 5 times

UNT is mandatory and must appear once in the message.

The standard specification for all segments is that the segment tag comprises 10 component data elements. The first is mandatory and contains the unique code to identify the segment. The remainder are conditional and are available to carry control numbers for use when required with repeating segments.

When a segment appears at level zero in a message (i.e. it is a simple non-repeating segment), following the normal rules for truncation, all of the unused component separators following the segment codes are replaced by the segment tag separator.

Any segment which can repeat within a message, can be constructed for output in one of two ways:

- without the use of control numbers associated with the segment code (subsequently referred to in this guide as implicit representation of repeating segments), in which case, as for non-repeating segments, the unwanted component element separators are truncated and replaced by the segment tag separator, or

- with the use of control numbers associated with the segment code, to specify the level at which the segment repeats, (subsequently referred to in this guide as explicit representation of repeating segments). Use of control numbers is detailed in Section 9.5.3.

It is stressed that it is the responsibility of message designers to decide and to state in each message specification, what form of representation is to be used for segments in the message which can repeat. However, it is not permitted to mix the two techniques within a message.

In current UNSM's implicit representation is specified, which is also the preferred technique for messages in use in and to North America. Some pre-EDIFACT applications in Europe still use explicit representation.

9.5.2 Comparison of implicit and explicit representation

Using a message with the following structure:

```
                  ___________________________
                   |     |    |    |    |    |
                 +---+ +---+  |  +---+  |  +---+
     Level 0     |UNH| |AAA|  |  |BBB|  |  |UNT|
                 |---| |---|  |  |---|  |  |---|
                 |M|1| |M|1|  |  |M|1|  |  |M|1|
                 +---+ +---+  |  +---+  |  +---+
                              |         |
                              |         |
                            +----+    +---+
     Level 1                |GDS |    |FTX|
                            |----|    |---|
                            |M|50|    |C|5|
                            +----+    +---+
```

Assuming that there are 3 occurences of the GDS segment and one occurrence of the FTX segment, the message would be transmitted as follows, (although in a character string, rather than the layout used for the example):

```
          Implicit                          Explicit

UNH+........data..........'         UNH+........data..........'
AAA+........data..........'         AAA+........data..........'
     GDS+..........data..........'      GDS:1+........data.......'
     GDS+..........data..........'      GDS:2+........data.......'
     GDS+..........data..........'      GDS:3+........data.......'
BBB+........data...........'        BBB+........data...........'
     FTX+..........data..........'      FTX:1+........data.......'
UNT+........data...........'        UNT+........data...........'
```

9.5.3 Repeating groups of segments

Within a message, two or more segments can be specified as a group of segments which can repeat as a group, with the group itself having either a Mandatory or Conditional status. There may be any number of groups within a message, with a group or groups also appearing within another group.

As for conditional segments, a conditional group of segments may be completely omitted from the message if there is no data for the group, (even though the group may contain one or more segments within it which have a mandatory status).

If a conditional group is transmitted, the group must contain the segment(s) within it having a mandatory status.

```
9.5.4 Message branching diagrams

      Message specifications must include a branching diagram showing
      the structure of the message.  The conventions used are shown in
      the following fictitious message.

      Level


             _______________________________________________.....___
             |      |     |        |       |       |       |       |
           __|__  __|__   |        |       |       |       |     __|__
      0    |UNH|  |AAA|   |        |       |       |       |     |UNT|
           |---|  |---|   |        |       |       |       |     |---|
           |M|1|  |M|1|   |        |       |       |       |     |M|1|
           +---+  +---+   |        |     +---+     |     +---+   +---+
                          |        |     |Gr1|     |     |Gr2|
                          |        |     |---|     |     |---|
                          |        |     |C|5|     |     |M|9|
                       +----+   +----+   |---|   +---+   |---|
      1                |RFF |   |CTA |   |NAD|   |CUX|   |TRD|
                       |----|   |----|   |---|   |---|   |---|
                       |C|10|   |C|10|   |M|1|   |C|5|   |M|1|
                       +----+   +----+   +---+   +---+   +---+
                                  |                        |
                                  |                      __|__
                           _______|_______               |Gr3|
                           |      |      |               |---|
                           |      |      |               |C|5|
                         +---+  +---+  +---+             |---|
      2                  |RFF|  |CTA|  |BNK|             |NAD|
                         |---|  |---|  |---|             |---|
                         |C|6|  |C|5|  |C|5|             |M|1|
                         +---+  +---+  +---+             +---+
                           |
                         __|__
      3                  |DTM|
                         |---|
                         |C|1|
                         +---+
```

In the above example, Group 1 is Conditional and the whole group (containing NAD, RFF, CTA and BNK) could either by omitted completely, or could repeat up to a maximum of 5 times. Each occurrence of the group must contain at least the NAD segment.

Group 2 (containing TRD and the segments in Group 3) is Mandatory and must appear at least once. It could appear up to a maximum of 9 times.

Within each occurrence of Group 2, Group 3 could be omitted completely if there were no data for the segments within the group, or alternatively could repeat up to a maximum of 5 times.

For each occurrence of Group 3, the NAD segment must appear, while DTM may or may not appear, depending upon data being present for the segment.

As can be seen, virtually any combination of horizontal and vertical structures for segments can be combined.

The order in which segments must be processed is first from left to right, and then if necessary, from top to bottom when a vertical structure is encountered. Left to right processing is then followed again if horizontal and vertical structures are combined.

Using Group 1 in the above diagram as an example, after having processed the segments to the left of the group in sequence, the first occurrence of the NAD segment (if present), would be followed by up to 6 occurrences of RFF, followed by up to 5 CTAs and up to 5 BNKs, before returning to the top of the loop.

Once all of the occurrences of NAD had been exhausted, processing would continue with the next segment to the right of the Group 1 on the diagram (CUX).

It follows from this therefore, than when formatting a message, the user's in-house system must ensure that subordinate and repetitive records used to construct segments are sorted into the correct sequence for processing.

It can also be seen from the diagram that a group must be entered via a single segment. In software terms, this segment is sometimes referred to as a 'trigger' segment, i.e. the in-house record containing the data for the segment is used by the software as a 'trigger' to start another occurrence of the group. It should be born in mind however that:-

i) two or more different loops cannot begin at the same segment, at the same position in a message; and

ii) an inner loop must be completed before returning to a further occurrence of the outer loop within which it appears.

The last point to be borne in mind, is that the levels at which segments appear must be shown on the left of the branching diagram, with the segments opposite the level number being kept in a horizontal line. This become particularly important for understanding and checking purposes, if the message is a complex one with parts of the diagram being 'exploded' onto another page of the branching diagram.

10. EDIFACT SERVICE & CONTROL MESSAGES

[NOTE: This Section is being submitted initially as a separate paper, which after approval, will be inserted].

11. MIGRATION TO EDIFACT

Any user seeking advice on migration from any other interchange standard to EDIFACT, should contact their local EDIFACT Rapporteur's Advisory Team.

11.1 Rapporteur Advisory & Support Teams (RT's)

It is via the RT's that maintenance procedures for the EDIFACT Syntax Rules, Data Element Directory, Segment Directory, and Message Directory will be co-ordinated, in conjunction with the UN/ECE Secretariat.