SECTION VI -
EDIFACT message formatting

1. Introduction

UN/EDIFACT is a standard for representation of data during transmission between parties. Version 3 of this standard is foreseen for NCTS.

UN/EDIFACT foresees a number of standard messages for various business purposes. Within Transit, it is foreseen to use the standard directory D96B. Within this directory, the usage of the following standard EDIFACT messages is foreseen: CUSDEC and CUSRES. In addition, it is foreseen to use the CONTRL message.

Every EDIFACT message is built according to a number of conventions:

- At the bottom, there are a number of *data elements*. These data elements have a predefined name, and a type. For some data elements, the standard is foreseeing the usage of predefined *code lists*. The collection of data elements is common to a directory definition (all EDIFACT messages are built upon a common collection of data elements).
- *Composite data elements* are built as a sequence of individual data elements, and EDIFACT *segments* are built as a sequence of composite data elements and single data elements. The collection of segments is common to all messages that are part of a directory (all EDIFACT messages are built with the same segments). A segment has a name, an optionality (Mandatory or Conditional), and a maximum repeat count. Within a segment, the sequence of composite data elements and data elements is fixed. A composite data element is either Mandatory or Conditional; an individual data element is also Mandatory or Conditional. Within a composite data element, it is common to use *qualifier* values in order to denote the meaning of a particular data element (one data element then contains the qualifier, while another data element contains the actual value). The standard also foresees a number of predefined qualifier values.
- EDIFACT messages are built as structures of segment groups or individual segments. A segment group is a sequence of segments. Within the structure of an EDIFACT message, there can be hierarchic levels. Within a hierarchy, every segment will have a predefined position, with an associated optionality and repeat count.

When a system such as NCTS is using the EDIFACT standard, it is common to define an *Interchange Agreement*. This agreement specifies how the different standards are to be applied, and which common conventions need to be maintained. This section is acting as the EDIFACT interchange agreement for NCTS.

To be noted is that changes can be applied to the various items that are part of the EDIFACT standard:

- Overall EDIFACT message structure;
- Optionality and repeat count of the segments;
- Structure of the segments (data elements and composite data elements), optionality of these elements;
- Data types;
- Usage of code lists;
- Usage of qualifier values

This section therefore first specifies a number of common conventions (such as common message header structure), that apply to all IEs that are exchanged using UN/EDIFACT. Next, it defines which IEs are mapped upon which EDIFACT messages. It then defines in detail which changes have been applied to the EDIFACT standard. Finally, it defines the mapping rules (correlation between IEs and EDIFACT messages).

2. EDIFACT conventions for NCTS

2.1. UN/EDIFACT choices

This section lists a number of choices made with respect to UN/EDIFACT syntax options. These choices are identical to the ones made in the Single Administrative Message (SAM) Mapping Guide. They are:

1. A UN/EDIFACT interchange starts with an interchange header segment (UNB). **The UNA segment is not used**.
2. One UN/EDIFACT interchange contains only one message. One IE will correspond to one EDIFACT interchange, which will correspond to one EDIFACT message. Conceptually, EDIFACT enables to transmit several messages in one interchange. Within the Common Domain, this will be restricted to one message per interchange only.
3. The following separator set is used:
   - ‘ segment separator
   - + data element separator
   - : composite data element separator
   - ? release character
4. The decimal notation is a ‘.’ (point)
5. Functional grouping is not used (UNG/UNE segments).
6. Nesting indicators are not used.
7. This guideline specifies the technical aspects of the NCTS Interchange Agreement. Separate reference to this agreement is not required, because other mechanisms like technical message structures and queue naming conventions are provided.

2.1. Common Message Header Structure

IEs are mapped to UN/EDIFACT UNSMs, as specified in this section. The common specification for the use of the interchange service segment UNB (present in every EDIFACT message for Transit) is according the SAM Mapping Guide:

UNB, INTERCHANGE HEADER, M, 1 x				functional
S001	SYNTAX IDENTIFIER	M		
0001	**Syntax identifier**	M	a4	**UNOC**
0002	**Syntax version number**	M	n1	**3**
S002	INTERCHANGE SENDER	M		
0004	**Sender identification**	M	an..35	Message sender (an..14)
0007	Identification code qualifier	C	an..4	Identification code qualifier (an..4)
0008	Address for reverse routing	C	an..14	Sending application (an..14)
S003	INTERCHANGE RECIPIENT	M		
0010	**Recipient identification**	M	an..35	Message recipient (an..14)
0007	Identification code qualifier	C	an..4	Identification code qualifier (an..4)
0014	Routing address	C	an..14	Receiving application (an..14)
S004	DATE/TIME OF PREPARATION	M		
0017	**Date**	M	n6	Date of preparation (n6)
0019	**Time**	M	n4	Time of preparation (n4)
0020	INTERCHANGE CONTROL REFERENCE	M	an..14	Interchange control reference (an..14)
S005	RECIPIENTS REFERENCE, PASSWORD	C		
0022	Recipient's reference/password	M	an..14	Recipient’s reference/password (an..14)
0025	Recipient's reference/password qualifier	C	an2	Recipient’s reference/password qualifier (an2)

0026	APPLICATION REFERENCE	C	an..14	Application reference (an..14)
0029	PROCESSING PRIORITY CODE	C	a1	Priority (a1)
0031	ACKNOWLEDGEMENT REQUEST	C	n1	Acknowledgement request (n1)
0032	COMMUNICATIONS AGREEMENT ID.	C	an..35	Communications agreement id. (an..35)
0035	TEST INDICATOR	C	n1	Test indicator (n1)

Table 17 - Common message header structure

The left part of this table shows the EDIFACT definition of the segment. The right part shows the corresponding Transit data items. All these data items belong to the data group MESSAGE, as specified for every IE.

The different items of the UNB segment are now discussed in detail. Mandatory EDIFACT data elements are the following:

Syntax Identifier: this data element is specifying the character set used in the message. This should be equal to 'UNOC". Within an EDIFACT message, the UNOC character set will be used except for some free text fields (these may be encoded in a different character set).

Syntax version number is the current version of the EDIFACT standard. This is always equal to '3'.

In the Common Domain MESSAGE data group, the data items "Message sender" and "Sending application" (mapped to **'Address for reverse routing'** in EDIFACT) are considered as synonymous, as are "Message recipient" and "Receiving application" (mapped to **'Routing address'** in EDIFACT).

Date and **Time** are also required, being the date and time when the IE was put in an EDIFACT representation. As syntax version 3 is used, the date format in data element 0017 is limited to n6.

The **Test Indicator** requires a value '1' if the interchange contains a test message. Otherwise, its value is '0'. When it is not present, this should also be considered as an operational message.

Interchange Control Reference needs to be unique for every EDIFACT interchange for the same MRN. Any EDIFACT message relating to the same MRN (even if it was the same IE sent twice) should contain a unique interchange reference identification.

All other elements of the UNB segment are completely optional.

2.3. UNH segment

Every EDIFACT exchange will contain a UNH segment. Within this segment, the only mandatory data element is the message type string. The message type is a short string denoting the IE type, and the Domain in which it is interchanged. Message type strings are defined in the next chapter.

2.4. Segment conventions

Some clarifications on EDIFACT standards:

- Within every message, the last but one segment will be a UNT segment, denoting the count of the number of segments in the message. This count should include every segment in the EDIFACT message, including UNH and UNT itself, but not UNB and UNZ (last segment) from the EDIFACT interchange.
- Some EDIFACT messages need to point to other segments in an EDIFACT message. In this case, segment number one is the UNH segment.
- Some segments require the presence of a segment at a higher level. This higher-level segment must always be present (even if it does not contain any data at all).

2.5. Amendments to UNSMs

The detailed structure of the UNSMs to be used for NCTS is documented as follows:

The overall message structure is defined in appendix G. This appendix defines the structure and the hierarchy of the UNSMs, and the exact location of the various segments in the UNSMs.

The detailed specification of the segments is included in appendix H (left-hand side).

The following paragraphs define how UNSMs have been modified in order to meet the needs of NCTS.

It is assumed that the NCTS IEs need to be supported by an EDIFACT UNSM. Directory D96B is used for mapping the templates to UNSMs. If no UNSM supports the FMS requirements, an existing UNSM is adjusted to support the FMS.

In various cases, the repeat factor of segments UNSMs is insufficient. The repeat factor has been raised to meet the requirements whenever necessary. Requirements of repeat factors higher than offered by the UNSMs, is due to two reasons:

- **proper mapping, repeat count too low**
The first reason is the correct mapping of repeating data groups or data items to a segment with a repeat count that is too low.

- **stuffing**
The second reason is a semantically incorrect mapping of data items to segments or misusing data elements to qualify a segment, because otherwise the required FMS hierarchy and repeating of data groups would be violated.

These types of errors can not be solved by simply reducing the repeat count of data groups in the FMS. It may imply deleting data items if repeat counts of UNSMs need to be adhered to. Therefore, the repeat factors of UNSMs have been adjusted.

The detailed list of all changes that have been made to the UNSM is included in Appendix H for consistency reasons. This appendix defines the mapping of the TMS to the UNSM. This appendix therefore starts by providing the reader the full list of changes to the UNSM standard.

3. Mapping of IEs on EDIFACT UNSMs

This chapter contains the structure of all NCTS messages, and tables listing which IEs are mapped to the EDIFACT Customs declaration, Customs response (respectively named CUSDEC and CUSRES, all part of the D96B-directory).

3.1. Mapping overview

The following tables give the list of the mapping of the IEs to UN/EDIFACT UNSMs and their code used in that particular UNSM. In general, an IE is mapped to CUSDEC D96B when it is used to exchange declaration data. If an IE is a response to a received IE and is not used to exchange declaration data, it is mapped to CUSRES D96B. The other UNSM serve specific purposes.

The correlation between the IEs and the EDIFACT UNSMs can be found in Appendix I.

The tables below also define the message type string (to be included in segment UNH) for the various IEs. The following remark is given to the column Message type string: the ISO Country Code of the NA that exchanges this particular IE with a Trader will replace CC, e.g. FI015A identifies an E_DEC_DAT in the External Domain of Finland. CD identifies the exchange across the Common Domain. The 'A' represents the version of the FMS structure ('A' being the first version implemented).

3.2. CUSDEC (CORE) Correlation Table

The following table lists the IEs that are mapped to CUSDEC D96B. For every IE, the table shows the IE number, name and reference name as well as the used Message Type string for the IE (mapped to the UNH[1].S009.0057 (Association assigned code) element in EDIFACT).

IE	IE name	Reference	Message Type string
4	Amendment acceptance	E_AMD_ACC	CC004A
7	Arrival notification	E_ARR_NOT	CC007A
13	Declaration amendment	E_DEC_AMD	CC013A
14	Declaration cancellation request	E_DEC_CAN	CC014A
15	Declaration data	E_DEC_DAT	CC015A
19	Discrepancies	E_DIS_SND	CC019A
21	Diversion rejection notification	E_DIV_NOT	CC021A
23	Enquiry initiation notification	E_ENQ_NOT	CC023A
29	Release for Transit	E_REL_TRA	CC029A
43	Unloading permission	E_ULD_PER	CC043A
44	Unloading remarks	E_ULD_REM	CC044A
45	Write-off notification	E_WRT_NOT	CC045A
51	No release for Transit	E_REL_NOT	CC051A
54	Request of release	E_REQ_REL	CC054A
100	Ask for documents	E_ASK_DOC	CC100A

Table 18 - IEs mapped to CUSDEC D96B

3.3. CUSRES (Core) Correlation Table

The following table lists the IEs that are mapped to CUSRES D96B. For every IE, the table shows the IE number, name and reference name as well as the used Message Type string for the IE (mapped to the UNH[1].S009.0057 (Association assigned code) element in EDIFACT).

IE	IE name	Reference	Message Type string
9	Cancellation decision	E_CAN_DEC	CC009A
25	Goods release notification	E_GDS_REL	CC025A
28	MRN allocated	E_MRN_ALL	CC028A
60	Control decision notification	E_CTR_DEC	CC060A

Table 19 - IEs mapped to CUSRES D96B

All rejection messages, serving to reject previously-sent information, as well as the functional error message IE906, are mapped to CUSRES as well. These rejection messages (in total) now use the same technique as used for the notification of a functional error via IE906. The data group FUNCTIONAL ERROR has been added for this purpose.

In Appendix I, containing the correlation tables for the different UNSMs, the correlation between these messages and CUSRES is shown under the label FUNACK to distinguish the core messages mapped to CUSRES from the functional error messages mapped to CUSRES.

IE	Name	Reference	Message Type string
05	Amendment Rejection	E_AMD_REJ	CC005A
08	Arrival Notification Rejection	E_ARR_REJ	CC008A
16	Declaration Rejection	E_DEC_REJ	CC016A
58	Unloading Remarks Rejection	E_ULD_REJ	CC058A
62	Release Request Rejection	E_REQ_REJ	CC062A
906	Functional NACK	C_FUN_NCK	CD906A

Table 20 - Rejection and functional error messages

A minimal requirement for the C_FUN_NCK is to exchange the first error detected in a message that has been received. Any other detected errors in the same message can optionally be exchanged in the same C_FUN_NCK. It is not allowed to exchange more than one C_FUN_NCK to report several functional errors detected in one message.

The E_AMD_REJ, E_ARR_REJ, E_DEC_REJ, E_ULD_REJ and E_REQ_REJ have to contain all errors detected in an E_DEC_AMD, E_ARR_NOT, E_DEC_DAT, E_ULD_REM and E_REQ_REL respectively.

The C_FUN_NCK is a high-priority message.

3.4. CONTRL Correlation table

The C_EDI_NCK (EDIFACT NACK) is the only message mapped upon CONTRL. Its message type string is equal to CD906A. It is a high-priority message.

4. Message Hierarchies

The following paragraph contains the message hierarchies for the UNSMs. These message hierarchies have been used in order to define mapping of IEs on UNSMs. Basically, a data group of an IE is looked up in this hierarchy. It is then mapped upon items belonging to this hierarchy. The detailed hierarchy definition is included in appendix H.

This guideline contains three hierarchies covering all Technical Messages Structures, that are mapped to the following EDIFACT messages:

- D96B CUSDEC;
- D96B CUSRES;
- ISO 9735 version 3 (syntax) CONTRL.

Hierarchical nesting is shown by indentation. Each level in the hierarchy has a repeat factor and a status (R: Required, O: Optional, D: Dependent).

4.1. CUSDEC (CORE) Hierarchy

MESSAGE	1 x, R
HEADER	1 x, R
(PRINCIPAL) TRADER	1 x, D
(CONSIGNOR) TRADER	1 x, O
(CONSIGNEE) TRADER	1 x, O
(AUTHORISED CONSIGNEE) TRADER	1 x, O
(DESTINATION) TRADER	1 x, R
GUARANTOR	1 x, O
(DEPARTURE) CUSTOMS OFFICE	1 x, D
(PRESENTATION OFFICE) CUSTOMS OFFICE	1 x, O
(TRANSIT) CUSTOMS OFFICE	9 x, O
(DESTINATION) CUSTOMS OFFICE	1 x, O
(RETURN COPIES) CUSTOMS OFFICE	1 x, O
CTL_CONTROL	1 x, O
CONTROL RESULT	1 x, O
UNLOADING REMARK	1 x, R
RESULTS OF CONTROL	9 x, O
REPRESENTATIVE	1 x, O
SEALS INFO	1 x, O
SEALS ID	99 x, O
GUARANTEE	9 x, R
GUARANTEE REFERENCE	99 x, D
VALIDITY LIMITATION EC	1 x, O
VALIDITY LIMITATION NON EC	99 x, O
EN ROUTE EVENT	9 x, O
CTL_CONTROL	1 x, R
INCIDENT	1 x, O
SEALS INFO	1 x, O
SEALS ID	99 x, R
TRANSHIPMENT	1 x, O
CONTAINERS	99 x, O
GOODS ITEM	999 x, O
PREVIOUS ADMINISTRATIVE REFERENCES	9 x, O
PRODUCED DOCUMENTS/CERTIFICATES	99 x, O
SPECIAL MENTIONS	99 x, O
RESULTS OF CONTROL	199 x, O
(CONSIGNOR) TRADER	1 x, O
(CONSIGNEE) TRADER	1 x, O
CONTAINERS	99 x, O
PACKAGES	99 x, O
SGI CODES	9 x, O

4.2. CUSRES (CORE) Hierarchy

MESSAGE	1 x, R
HEADER	1 x, R
(PRINCIPAL) TRADER	1 x, O
(DESTINATION) TRADER	1 x, O
(DEPARTURE) CUSTOMS OFFICE	1 x, O
(PRESENTATION OFFICE) CUSTOMS OFFICE	1 x, O
FUNCTIONAL ERROR	999 x, O

4.3. CONTRL hierarchy

MESSAGE	1 x, R
INTERCHANGE ERRORS	1 x, R
MESSAGE ERRORS	99 x, D
SEGMENT ERRORS	999 x, D
DATA ELEMENT ERRORS	99 x, D

5. Correlation tables

The correlation tables list the correlation between the Message Hierarchies (which are each mapped to one particular UNSM) and the IEs. All correlation tables, as well for Core Business messages as for Exception Handling, can be found in **Appendix I.**

Thus, only IEs that need to be sent in EDIFACT-format will be found in these correlation tables.

The right-hand side of appendix H defines the direct correlation between UNSM segments and the NCTS data items.

Code lists are included in appendix C.

5.1. Correlation Tables

Appendix I documents the correlation between the Technical Message Structure (and all its composites), and the data elements of the EDIFACT messages. For each mapping of a NCTS message to an EDIFACT message, a Correlation Table is given. Many NCTS IEs can be mapped to the same EDIFACT messages. The following mapping categories have therefore been defined:

- Core Business IEs mapped to D96B CUSDEC UNSM;
- Core Business IEs mapped to D96B CUSRES UNSM;
- Functional error messages mapped to D96B CUSRES UNSM;
- UN/EDIFACT error reporting message mapped to CONTRL UNSM.

These correlation tables contain the following columns:

- **SAD Box**

It specifies the box used in the SAD (Single Administrative Document). It is only given for those data items for which a SAD box has been identified in the SAM project. This column is only applicable to the mapping of the Core IEs.

- **Name**

It specifies the name of the box used in the SAD (Single Administrative Document). It is only given for those data items for which a SAD box has been identified in the SAM project. This column is only applicable to the mapping of the Core IEs.

- **Elements**

They specify:

Hierarchy level specifying the origin of the information to map into the EDIFACT element. The higher level data groups are separated from the (lower level) data groups by a '-'. An example is 'GOODS ITEM – PACKAGES' where 'GOODS ITEM' is the higher level data group and 'PACKAGES' is the concerned data group. The information in this column refers to the *Message* Hierarchy;

Data item (after the full stop) specifies the actual name of the application data item to map into the EDIFACT data element.

- **Data-type**

It describes the type and the length of the data item. When a data-type includes a decimal, the maximum number of decimals after the decimal sign is included in the length of the data-type. For instance, the format n..11,3 can have 3 decimals included in its maximum total length of 11 numeric digits. Neither the decimal point nor the sign are included in the length of a data-type.

- **Status (numbered columns)**

It specifies if the data item is required [R], dependent [D] or optional [O] per Functional Message Structure. The status given in a column needs to be read in conjunction with the status of the data group specified by the message hierarchy. For instance, the status in a column may read R (required), whereas the status of the related data group in the message hierarchy reads O (optional, see appendix Q). In those cases, the entry in the column is required only when the related data group is used.

- **Pos**

  It identifies the EDIFACT segment according to its position in the standard message. The position refers to the branching diagram. The UNB segment is shown as position 0 as it is the interchange header and not part of the message.

- **EDIFACT mapping**

  It gives the mapping information referring to one particular data element in a segment of a particular EDIFACT message, possibly with reference to all applicable qualifier values. Furthermore, the position number of the segment is given. An example is given by FTX[11](4451=ABL).C108.4440, which is the mapping to data element 4440 of composite C108 with qualifier value 'ABL' for element 4451 of the FTX segment at position 11 of the EDIFACT message. In some occasions, a data element is not uniquely identified within a composite or a segment. In those cases, the mapping information is followed by '#' and the sequence number of that particular data element. For instance, FTX[11](4451=ABL).C108.4440#2, is a mapping to the second free text data element of the composite identified by C108 in the segment FTX.

  In case no qualifier value is required, e.g. a mapping is constructed to a qualified segment only for one data item, this is indicated by a '-' as the value for a qualifier (the qualifier data element is conditional in those occasions). In case all qualifier values are allowed for a particular mapping, this is indicated by a '*' as the value for a qualifier.

  It may appear occasionally that, although this is not mentioned, the number of occurrences of an EDIFACT segment is higher than the maximum allowed by the CUSDEC. However, the number of occurrences always needs to be read in conjunction with the status given by one of the columns.

  In some instances, the Correlation Table does not have an entry for data elements of the service segments. This is applicable to those data elements that have a fixed value; e.g. data element 0002 always has the value '3' identifying the syntax version. Those UN/EDIFACT service elements that cannot be found in the Correlation Table are available in the segment description.

  Mappings can be constructed to repeating composite data elements within one segment. UN/EDIFACT prescribes that the first composites of a segment need to be filled, which implies that the filling of a composite is independent of its position in a segment. However, if the composite does not contain a composite qualifier and the mapping of a data item is fixed by a composite's position in a segment, it implies that data for such an data item is always mapped to the same position in a segment independent of empty composites before that position.

- **Code List**

  If a code list can and should be applied to the data item, the reference number of this code list is listed in this column. The applicable set of code lists can be found in Appendix C.

6. The structure and use of error messages

hError messages are exchanged at both functional and UN/EDIFACT level. Both error messages are specified in this section.

6.1. Functional error messages

6.1.1. Functional error data group

To be able to exchange functional errors, a data group 'FUNCTIONAL ERROR' is introduced. This data group is the technical implementation of Rule 123 in the FMS of IE05 (E_AMD_REJ), IE08 (E_ARR_REJ), IE16 (E_DEC_REJ), IE58 (E_ULD_REJ) and IE62 (E_REQ_REJ), as specified in FTSS, Appendix B. The data group consists of the following data items:

Data Item	Content	Status	Format
Error type	Values taken from the first column of table ….	Required	n2
Error pointer	This data item points to the data item or data group that caused the error by listing the hierarchy of that data item and its occurrence in the hierarchy. In case of error type 90, 91 or 93, the error pointer points to the MRN. In case of error type 92, the error pointer points to the Message Type in UNH. The syntax for the value of the error pointer is as follows: (data group code '(' [(occurrence)] ')' '.')$^+$ [(data item name)]	Required	an..210
Error reason code	This data item contains the identification of the condition or rule in case error type '15' is detected due to an error related to a condition or a rule or a technical rule (for example 'C99' to denote a violation of condition 99, and 'TR01' to denote a violation of technical rule 1).	Dependent	an..4
Original data item value	This data item is used to exchange the original value in case sequencing of data groups is changed at reception of a message.	Optional	an..50

Table 21 - Data Items for Functional Error

Notes to the functional error data group:

- The data group codes used for the error pointer are listed below:

HEA	HEADER	R x 1
EPT	(DEPARTURE) CUSTOMS OFFICE	O x 1
PC1	(PRINCIPAL) TRADER	O x 1
ER1	FUNCTIONAL ERROR	R x 999

- The notation used for specifying the pointer is as follows:

Pattern	Semantics	Example
A B	A followed by B	(data group code) (data item name)
[A]	A or nothing	[occurrence]
A$^+$	One or more occurrences of A	(data group code)$^+$

(expression)	Expression is treated as unit and may be combined as described in this list	(data group code)
'string'	A literal string	'(' or '.'

Table 22 - Notation of error pointer

- Occurrence is a sequence number for a data group. An occurrence is only given for repeatable or erroneously repeated data groups and is therefore optional. An occurrence relates to the sequence in which a message is received. This sequence is not necessarily equal to the sending sequence.

- The data item names of the FMS are listed in appendix Q.

Examples of the error pointer are as follows:

Error pointer value	Semantics
HEA.containerised indicator	Pointer to 'containerised indicator' of the Header data group.
GUA(3).REF(5).Access code	Pointer to 'access code' of the fifth Guarantee reference data group within the third Guarantee data group.
GDS(3).GS2(4).Kind of packages	Pointer to 'Kind of packages' of the fourth Package data group within the third goods item.
CE1	Pointer to '(CONSIGNEE) TRADER' data group.

Table 23 - Examples of error pointer

6.1.2. Functional error CUSRES Hierarchy

The data group 'FUNCTIONAL ERROR' points to a certain data item in an FMS.

The E_ARR_REJ can only refer to a previously exchanged E_ARR_NOT, of which only one is exchanged. The same rule is applicable to the E_DEC_REJ in its relation to an E_DEC_DAT and an E_ULD_REJ in its relation to an E_ULD_REM. Also, the E_AMD_REJ (rejection of E_DEC_AMD) and E_REQ_REJ (rejection of E_REQ_REL) are within the category of functional errors.

Functional errors are exchanged by D96B CUSRES. The data group 'FUNCTIONAL ERROR' is mapped to an FTX segment.

The hierarchy is given as follows:

MESSAGE	1 x, R
HEADER	1 x, R
FUNCTIONAL ERROR	999 x, D

6.1.3. Correlation table of functional error messages

To meet CUSRES requirements, the E_DEC_REJ contains an extra data item that is not specified in FTSS, Appendix B. The data item HEADER.Declaration type is required to be able to map to the mandatory UN/EDIFACT BGM segment and to be in line with all other FMS mapped to CUSRES.

The correlation tables can be found in Appendix I.

6.2. UN/EDIFACT CONTRL MESSAGE

6.2.1. General

The UN/EDIFACT CONTRL message structure is used to exchange errors detected in a received interchange. The minimal requirement is to report the first error detected. All other detected errors should be reported if possible.

The structure of CONTRL is based on four segments: UCI (Interchange Level), UCM (Message Level), UCS (Segment Level) and UCD (Data Element Level), each containing a reference to a part of the subject interchange. The parts of the subject interchange are:

- the UNB and UNZ segments, referenced in the UCI segment. UCI refers to the original UN/EDIFACT interchange in which errors have been detected, by copying the sender identification, recipient identification, and interchange reference of that erroneous interchange.
- the UNH and UNT segments, referenced in the UCM segment. UCM refers to the original UN/EDIFACT message in which errors have been detected, by copying the message reference and the message type/version/release number/controlling agency/association assigned code of that erroneous message. The action taken by the recipient of the erroneous message as well as the specific error information (error code – message segment – position in this segment) is transmitted as well.
- a segment in a message, referenced in the UCS segment. UCS refers to a position of a segment for which an error has been detected in the original UN/EDIFACT message, by means of a segment position. The segment position is a sequence number of the erroneous segment in the UN/EDIFACT message. It starts with, and includes, the UNH segment as segment number '1'. To report a missing segment, this is the numerical count of the last segment that was processed before the position where the missing segment was expected to be. A missing segment group is denoted by identifying the first segment in the group as missing.
- a simple, composite or component data element, referenced in the UCD segment. UCD refers to a position of a data element in a segment for which an error has been detected in the original UN/EDIFACT message. EITHER the data element position is a counter of all fields starting at '1' for the segment tag OR, if the information can be supplied by the EDI-converter, the data element position is a counter of simple and composite fields starting at '1' for the segment tag and the data component position is the position of the component within the composite field.

On receipt of a CONTRL message, it must be possible to display and/or print the position of an error, regardless of whether a component position is present or not

The UCI segment can only report one error. If more than one error is detected at interchange level, the receiver of the interchange is free to choose which error to report.

It is not allowed to exchange more than one CONTRL message to report several errors in the same interchange.

The list of allowed error codes is given by the column with heading 'UN/EDIFACT' in ….

The structure of the CONTRL is based on the assumption of one message per UN/EDIFACT interchange.

The message type as exchanged in the association assigned code of the UN/EDIFACT message header (UNH.S009.0057) of the UN/EDIFACT CONTRL is CD907A.

Should an error be detected in a CD907A, no further message is exchanged but as much data as possible provided for manual intervention.

6.2.2. CONTRL correlation table

The correlation table can be found in Appendix I .

6.2.3. CONTRL rules

At every one of the levels UCI, UCM, UCS or UCD an error, detected at the corresponding level (UNB, UNH, segment, data element) in the subject interchange, can be specified. Logically, only the error at the right level (and only up to that level) needs to be specified.

Note that the data item specifying the error at UCI level is called 'Syntax Error' because at this level, it will always concern an error against EDIFACT syntax rules.

This leads to two Technical Rules for IE907 which are documented in Appendix Q.