CONTRL - SHIPPING INSTRUCTIONS

Messaging User Guide (XML)

Technical Guide on XML format messages for the acknowledgment (CONTRL) message, in valenciaport's Shipping Instructions Service

CONTENTS

1 // Introduction

1.1 // Track changes

The following table details the chapters in which changes have been made compared to the document's previous version.

The specific changes made are shown in red throughout the document.

Version	Parts that change	Change description
04th Nov 2015	All	Original version
21th Oct 2021	InterchangeHeader, InterchangeSender, MessageResponseGroup	New aggregator CARGOSMART added

1.2 // Purpose

The object of this document is to define the user guide for the XML message for the notification sent by the PCS platform about the correct reception or error of a Shipping Instruction. This guide specifies the format and semantics of the CONTRL acknowledgment message.

This document is aimed at companies that wish to integrate their information systems with ***valenciaportpcs.net***'s messaging system which offers a Shipping Instructions service.It is also directed at Project Managers, in charge of managing and monitoring the project, as well as staff involved in developing it.

1.3 // Scope

The user guide described in this document is part of the project to migrate the Shipping Instructions service to the ***valenciaportpcs.net*** transactional platform.

The Shipping Instructions message is part of a set of messages previous to the export process between a contracting party (sender of the Shipping Instructions) and a transport provider (recipient of the Shipping Instructions). This message is used so that the contracting party can create or change a Shipping Instruction which can then be sent to the transport provider, be it a carrier, through an intermediary or a shipping agent. The acknowledgment message defined in this document corresponds to each Shipping Instruction sent to the ***valenciaportpcs.net*** platform.

Therefore, the CONTRL message is used so that an information system can confirm that a Shipping Instruction has been received correctly or report the existence of errors in this instruction.

The message is based on the EDIFACT CONTRL message.

1.4 // Reference documents

The following references have been used to develop this document.

Document name	Source
CONTRL Syntax and Service Report Message for Batch EDI (from INTTRA to Trading Partner)	INTTRA

Document name	Source
Acknowledgment XSD Schema	GT Nexus
PR04049-MN03v1.02__CONTRL User Guide	Valenciaport

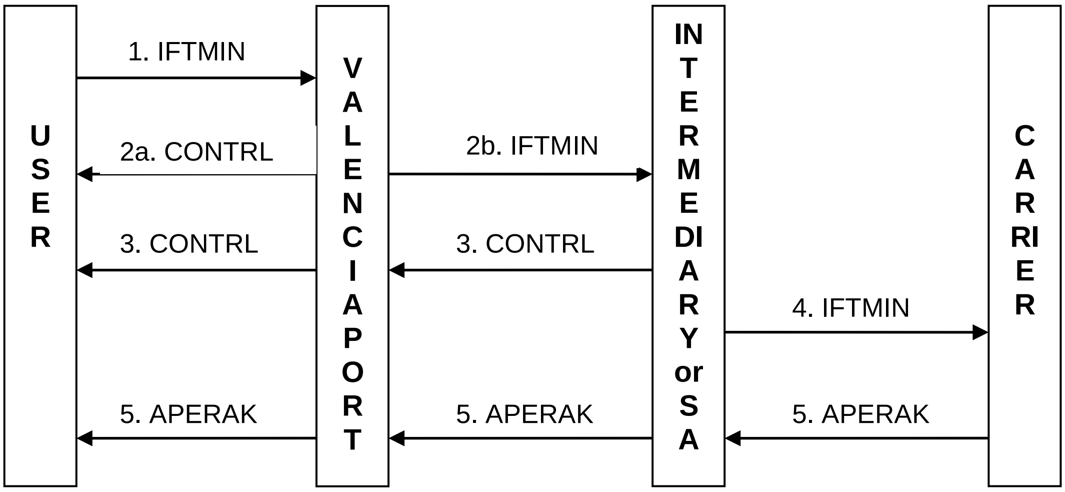
1.5 // Abbreviations and acronyms

Term	Meaning
SA	Shipping Agent
ISO	International Organization for Standardization
M	Mandatory
O	Optional
PCS	valenciaportpcs.net
UN/EDIFACT	United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport

2 // Business context and associated messaging

2.1 // Message flow

Sending a Shipping Instruction to the end recipient (transport provider) and the different responses that the sender can receive follow the message flow shown below.



1. The user sends the Shipping Instruction message to valenciaportpcs.net.
2. If the message contains any errors, ***valenciaportpcs.net*** sends the user a CONTRL (2a) rejection message, defined in this guide. If the message does not contain any errors, it is forwarded (2b) to an intermediary or shipping agent, depending on who the carrier receiving the message works with.
3. The intermediary or shipping agent processes the message and responds with a CONTRL acceptance message, which is forwarded to the user.

   N.B.: Therefore, irrespective of whether the Shipping Instruction sent by the user contains errors or not, the user will always receive a CONTRL message, detailing whether the instruction has been correctly processed.
4. If the message is sent to an intermediary, they will in turn send it to the receiving carrier.
5. The receiving carrier or shipping agent can send a message accepting or rejecting the contents of the Shipping Instruction sent, i.e. an APERAK message, defined in its own particular guide. ***valenciaportpcs.net*** forwards this message to the user.

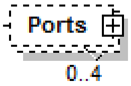
   N.B.: The user will always receive a CONTRL message for each Shipping Instruction they have sent. However, the reception of an APERAK message depends on whether the recipient of the Shipping Instruction sends it or not.

3 // Special considerations: format and content

3.1 // Mandatory nature, cardinalities, sizes and types

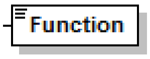
The specifications of the schema for the XML message have been drawn up on the basis of the following premises and considerations:

- The cardinalities of each element, the maximum sizes and the types of data have been established according to the operational needs of valenciaportpcs.net's users.
- Each element's cardinality is indicated as follows:



- The types of data and maximum sizes are shown in each element table (the different types used and their meaning is described in the following chapter):

- The mandatory data is marked:
    - As a schema: represented as a solid line

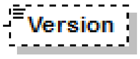


    - In the element table: marked with an "M"

Function	Message function Accepted values: • **ORIGINAL:** original • **REPLACE:** replacement • **CANCELLATION:** cancellation	M	an..35

N.B.: details may not be mandatory according to the schema, but may subsequently be defined as mandatory (in a detailed validation in the remarks section) and may thus be marked in the element table.

- The optional data is marked:
    - As a schema: represented as a broken line



    - In the element table: marked with an "O"

Version	Message version Accepted values: • ***1.0***	O	an..5

- The conditional data is marked:
    - In the element table: marked with a "C"

LoadingVesselDetails	Group of elements which contains details about the vessel loading the containers.	C	G

  - Details which appear according to whether certain rules are complied with or other message elements are included. Normally, they are associated with business rules which appear in the “comments” section of the data group in question.

- The data groups (elements composed in XML, which also contain an ordered sequence of elements) are marked in the element table with a “G” in the “Type” field:

LoadingVesselDetails	Group of elements which contains details about the vessel loading the containers.	C	G

- This guide includes the business rules that complement the message schema specification.

3.2 // Data formats and types

3.2.1. Character sets supported

The accepted character code format is UTF-8 or UTF-16, in line with Unicode characteristics and ISO-10646.

3.2.2. Alphanumeric type (<xs:string>)

The XML alphanumeric type is represented in this guide as “an..NNN”, when NNN indicates the maximum size accepted in the field.

3.2.3. Boolean type (<xs:boolean>)

The XML boolean type is represented in this guide as “boolean”.

The accepted values for this type of data are “true” or “1" and "false" or "0".

3.2.4. Numeric types (<xs:positiveInteger> and <xs:decimal>)

The XML numeric type is represented in this guide as “int” for positive whole numbers and “decimal” for real numbers. There is no whole data that accepts negative numbers.

Decimals

- Decimal values should be represented using the dot (‘.’).
  - Example: 10455.12 or 45.8735
- Group separators should not be used.
  - Example: 10,455.125 is not valid.
- If the value is logical according to the data (for example, for temperatures), negative numbers can be indicated (by placing a minus sign ‘-‘ in front of them).

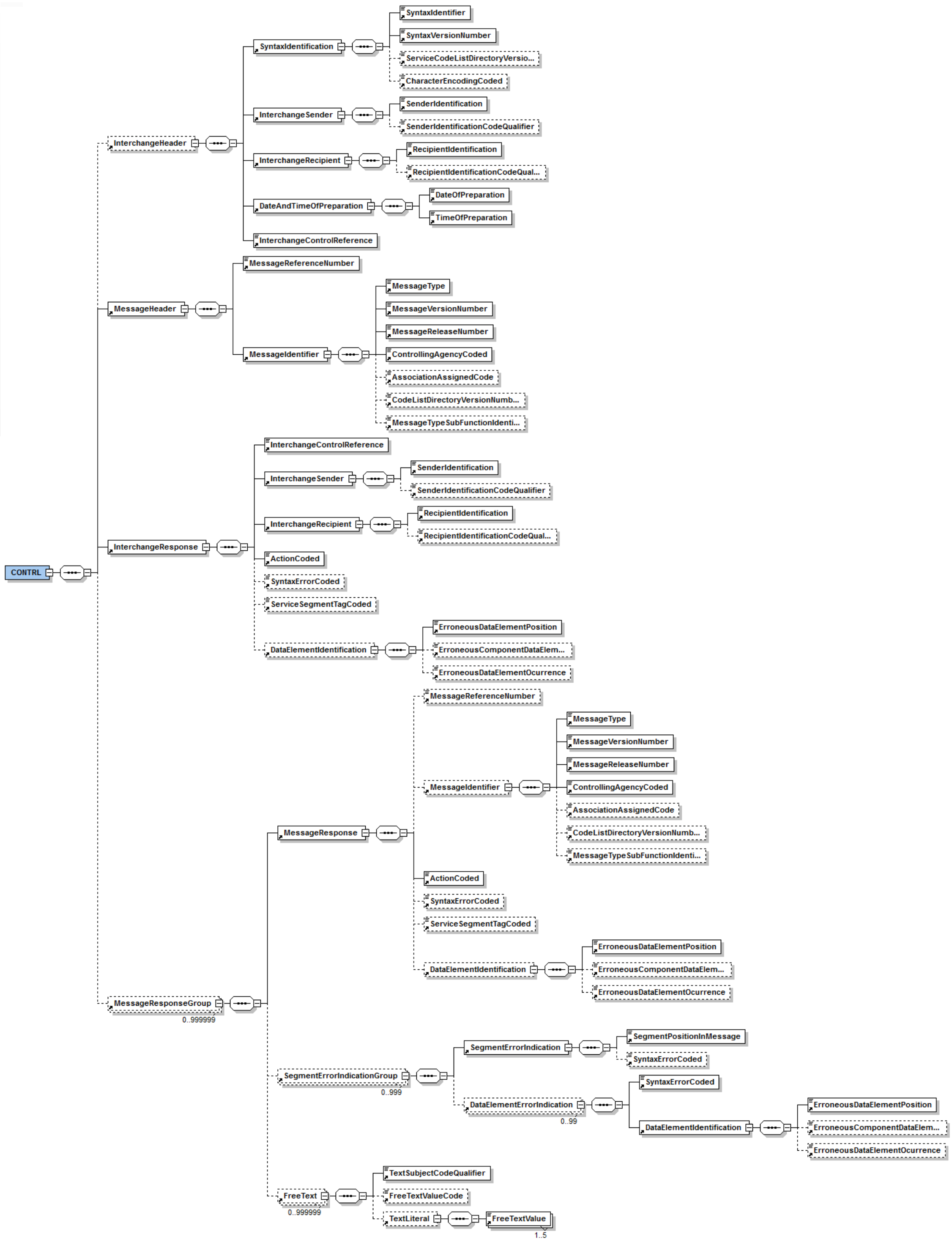
3.2.5. Types for date and date/time (<xs:date> and <xs:dateTime>)

- The XML date and time type is represented in this guide as “dateTime” and just the date as “date”.

- Both the date and the date/time must follow the standard XML format:
  - “YYYY-MM-DD” for the date
  - “YYYY-MM-DDThh:mm:ss” for the date/time, where T is a fixed character separator for the date and time fields.

4 // Message structure

As this is an XML message, it must contain the specified header in the XML syntax: <?xmlversion=”1.0” encoding=”UTF-8”?>, followed by the rest of the message. The only accepted encoding for the message is UTF (either UTF-8 or UTF-16).



4.1 // Issues to be considered

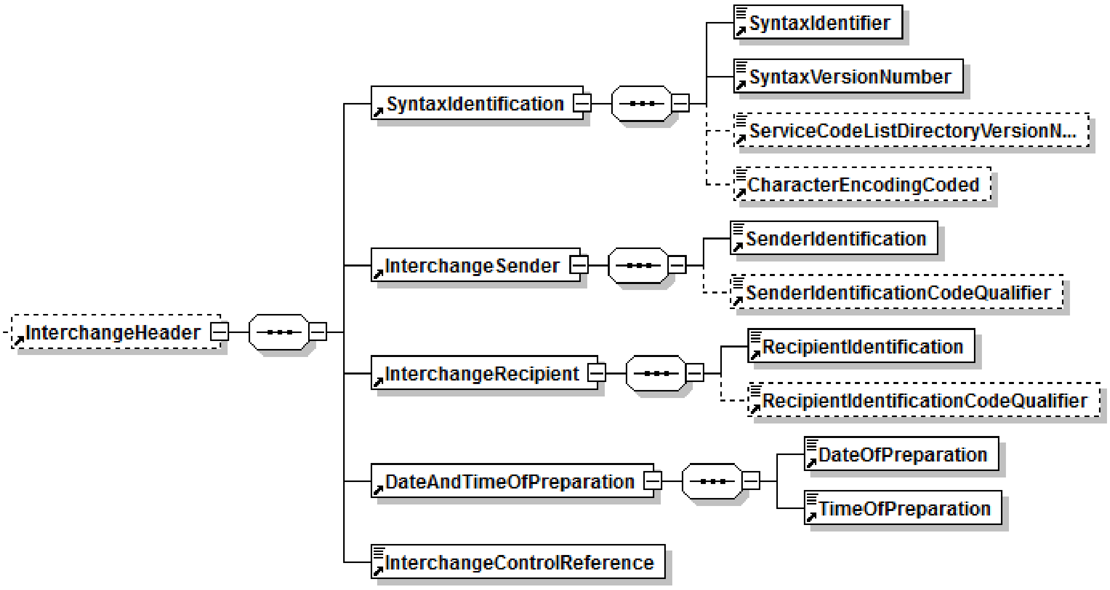
The specifications of the schema for the CONTRL XML message have been drawn up on the basis of the following premises and considerations:

1. It has been defined using the same structure, conditions, cardinalities and lengths as those defined by UN/EDIFACT. These characteristics have been adapted to the message requirements specified by INTTRA in [CONTRL-INTTRA] and by GT Nexus in [ACK-GT Nexus].
2. This guide includes the business rules specified by INTTRA in [CONTRL-INTTRA] that complement the message schema specification.

5 // CONTRL message elements

5.1 // CONTRL\InterchangeHeader

Level	1
Usage	O (Optional)
Max. Use	1



5.1.1. Purpose

The *InterchangeHeader* group of elements is used to identify and specify the interchange of messages.

5.1.2. Comments

This group of elements is required by ***valenciaportpcs.net.***

The ServiceCodeListDirectoryVersionNumber and CharacterEncodingCoded elements are not used by **valenciaportpcs.net**.

5.1.3. Elements

Name	*Purpose*	*M/O*	*Type*
InterchangeHeader		*M*	
SyntaxIdentification	Group of elements which identifies the agency and the syntax used in the message	M	G
InterchangeSender	Group of elements which identifies the message sender	M	G
InterchangeRecipient	Group of elements which	M	G

	identifies the message recipient		
DateAndTimeOfOperation	Group of elements which identifies the day and time of the message	M	G
InterchangeControlReference	Message identifier	M	String 1..14
InterchangeHeader\SyntaxIdentification			
SyntaxIdentifier	Code which identifies the agency responsible for the syntax Accepted values: **UNOC**: *UN/ECE Level C*	M	String 4
SyntaxVersionNumber	Syntax number Accepted values: **2** (ISO 9735:1990)	M	String 1
~~**ServiceCodeListDirectoryVersionNumber**~~	Service code directory version number	O	String 1..6
~~**CharecterEncodingCoded**~~	Code which identifies the code used in the interchange	O	String 1..3
InterchangeHeader\InterchangeSender			
SenderIdentification	Code which identifies the message sender Accepted values: **VALENCIAPORT, INTTRA, GT Nexus, CARGOSMART or Valenciaport code for the logistics operator** (according to the scenario)	M	String 1..35
SenderIdentificationCodeQualifier	Code which identifies the source of the code used to identify the sender Accepted values: **ZZZ**: Mutually Defined	O	String 1..4
InterchangeHeader\InterchangeRecipient			
RecipientIdentification	Code which identifies the message recipient Accepted values: **VALENCIAPORT or Valenciaport code for the logistics operator** (according to the scenario)	M	String 1..35
RecipientIdentificationCodeQualifier	Code which identifies the source of the code used to identify the recipient Accepted values: **ZZZ**: Mutually Defined	O	String 1..4
InterchangeHeader\DateAndTimeOfPreparation			
DateOfPreparation	Date the message is prepared (YYMMDD format)	M	Decimal 6
TimeOfPreparation	Time the message is prepared (HHMM format)	M	Decimals 4

5.1.4. Scenario 1 XML example

Message received by valenciaportpcs.net and sent by the carrier

```
<InterchangeHeader>
    <SyntaxIdentification>
        <SyntaxIdentifier>UNOC</SyntaxIdentifier>
        <SyntaxVersionNumber>2</SyntaxVersionNumber>
    </SyntaxIdentification>
    <InterchangeSender>
        <SenderIdentification>INTTRA</SenderIdentification>
        <SenderIdentificationCodeQualifier>ZZZ</SenderIdentificationCodeQualifier>
    </InterchangeSender>
    <InterchangeRecipient>
        <RecipientIdentification>VALENCIAPORT</RecipientIdentification>
        <RecipientIdentificationCodeQualifier>ZZZ</RecipientIdentificationCodeQualifier>
    </InterchangeRecipient>
    <DateAndTimeOfPreparation>
        <DateOfPreparation>151224</DateOfPreparation>
        <TimeOfPreparation>1215</TimeOfPreparation>
    </DateAndTimeOfPreparation>
    <InterchangeControlReference>1</InterchangeControlReference>
</InterchangeHeader>
```

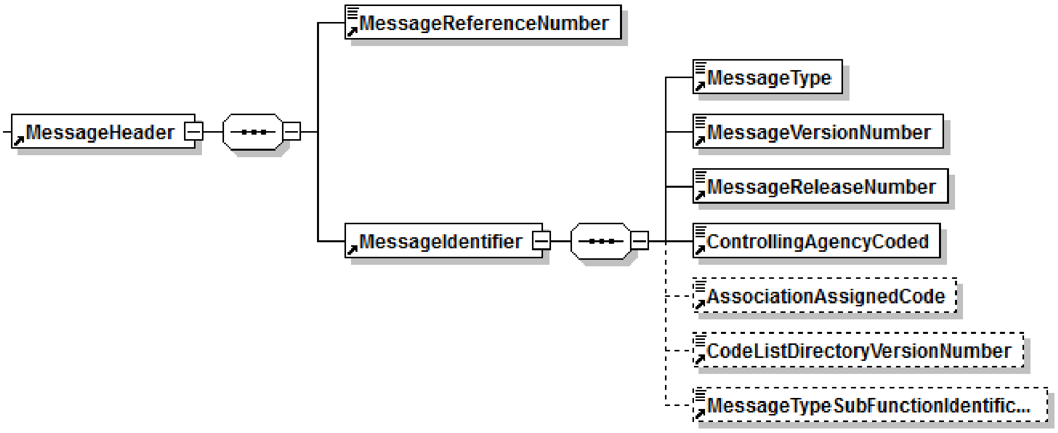
5.1.5. Scenario 2 XML example

Message sent by valenciaportpcs.net and received by the logistics operator.

```
<InterchangeHeader>
    <SyntaxIdentification>
        <SyntaxIdentifier>UNOC</SyntaxIdentifier>
        <SyntaxVersionNumber>2</SyntaxVersionNumber>
    </SyntaxIdentification>
    <InterchangeSender>
        <SenderIdentification>VALENCIAPORT</SenderIdentification>
        <SenderIdentificationCodeQualifier>ZZZ</SenderIdentificationCodeQualifier>
    </InterchangeSender>
    <InterchangeRecipient>
        <RecipientIdentification>CODE</RecipientIdentification>
        <RecipientIdentificationCodeQualifier>ZZZ</RecipientIdentificationCodeQualifier>
    </InterchangeRecipient>
    <DateAndTimeOfPreparation>
        <DateOfPreparation>20031224</DateOfPreparation>
        <TimeOfPreparation>1215</TimeOfPreparation>
    </DateAndTimeOfPreparation>
    <InterchangeControlReference>1</InterchangeControlReference>
</InterchangeHeader>
```

5.2 // CONTRL\MessageHeader

Level	**1**
Usage	M (Mandatory)
Max. Use	1



5.2.1. Purpose

The *MessageHeader* group of elements is used to identify the information in the message header.

5.2.2. Comments

This group of elements is required by ***valenciaportpcs.net.***

The *AssociationAssignedCode*, *CodeListDirectoryVersionNumber* and *MessageTypeSubFunctionIdentifier* elements are not used by **valenciaportpcs.net**.

5.2.3. Elements

Name	*Purpose*	*M/O*	*Type*
MessageHeader		*M*	
MessageReferenceNumber	Unique reference assigned by the message sender which identifies the message	M	String 1..14
MessageIdentifier	Group of elements which identifies the type, version and other information in the interchanged message	M	G
MessageHeader\MessageIdentifier			
MessageType	Code which identifies the type of message Accepted values: • **CONTRL**	M	String 1..6
MessageVersionNumber	Message version number Accepted values: • **4**	M	String 1..3
MessageReleaseNumber	Message version release number Accepted values: • **1**	M	String 1..3
ControllingAgencyCoded	Identification code of the controlling	M	String

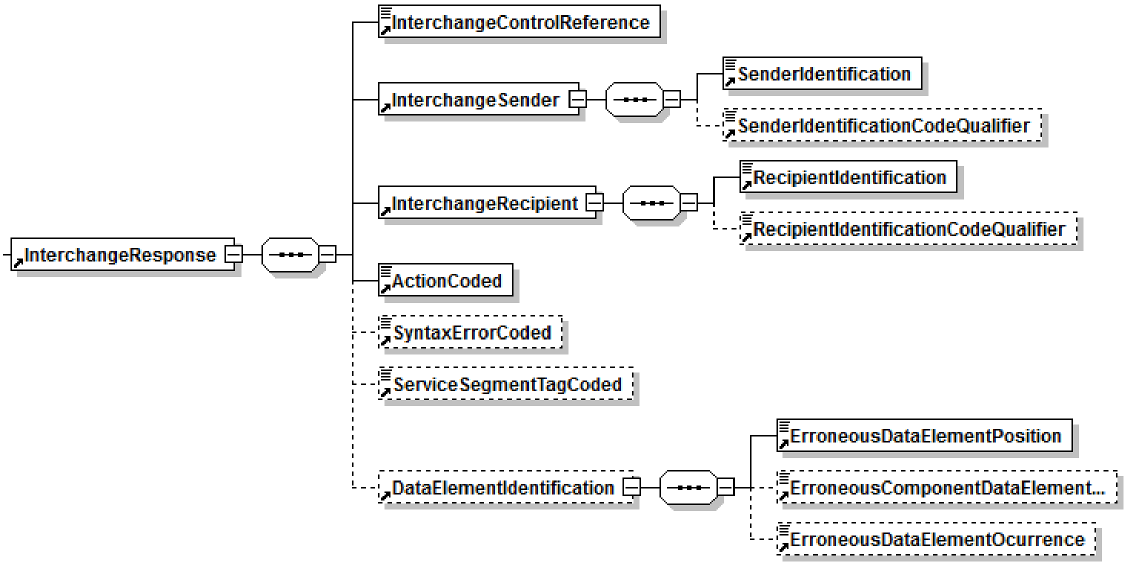
	agency Accepted values: • **UN**		1..2
~~AssociationAssignedCode~~	Code which identifies the message, assigned by the association responsible for its design and maintenance	O	String 1..6
~~CodeListDirectoryVersionNumber~~	Code list directory version number	O	String 1..6
~~MessageTypeSubFunctionIdentifier~~	Code which identifies the subfunction of the type of message	O	String 1..6

5.2.4. XML example

```
<MessageHeader>
    <MessageReferenceNumber> VPP12003102301</MessageReferenceNumber>
    <MessageIdentifier>
        <MessageType>CONTRL</ MessageType >
        <MessageVersionNumber>4</ MessageVersionNumber >
        <MessageReleaseNumber>1</ MessageReleaseNumber >
        <ControllingAgencyCoded>UN</ ControllingAgencyCoded >
    </MessageIdentifier>
</MessageHeader>
```

5.3 // CONTRL\InterchangeResponse

Level	**1**
Usage	M (Mandatory)
Max. Use	1


InterchangeResponse
InterchangeControlReference
InterchangeSender
SenderIdentification
SenderIdentificationCodeQualifier
InterchangeRecipient
RecipientIdentification
RecipientIdentificationCodeQualifier
ActionCoded
SyntaxErrorCoded
ServiceSegmentTagCoded
DataElementIdentification
ErroneousDataElementPosition
ErroneousComponentDataElement...
ErroneousDataElementOcurrence

5.3.1. Purpose

The *InterchangeResponse* group of elements is used to identify the interchange being responded to, as well as indicating whether an error has occurred.

5.3.2. Comments

This group of elements is required by ***valenciaportpcs.net.***

The *InterchangeSender*, *InterchangeRecipient* and *InterchangeControlReference* elements correspond to the message being responded to.

The *SyntaxErrorCoded* and *ServiceSegmentTagCoded* elements and the *DataElementIdentification* group of elements are not used by the carriers that work with GT Nexus.

5.3.3. Elements

Name	***Purpose***	***M/O***	***Type***
InterchangeResponse		*M*	
InterchangeControlReference	Identifier of the original message being responded to	M	String 1..14
InterchangeSender	Group of elements which identifies the message sender	M	G
InterchangeRecipient	Group of elements which identifies the message recipient	M	G
ActionCoded	Code which identifies the acceptance of the message or errors in it Accepted values: • **4**: Interchange rejected • **7**: Interchange accepted and document accepted, if the opposite is not specifically detailed in a segment or element	M	String 1..3
SyntaxErrorCoded	Error code detected	O	String 1..3
ServiceSegmentTagCoded	Code which identifies the segment	O	String 1..3
DataElementIdentification	Group of elements which identifies the position of an erroneous data element	O	String 1..3
InterchangeResponse\InterchangeSender			
SenderIdentification	Code which identifies the sender of the message being responded to Accepted values: **VALENCIAPORT, INTTRA, GT Nexus, CARGOSMART or Valenciaport code for the logistics operator**	M	String 1..35

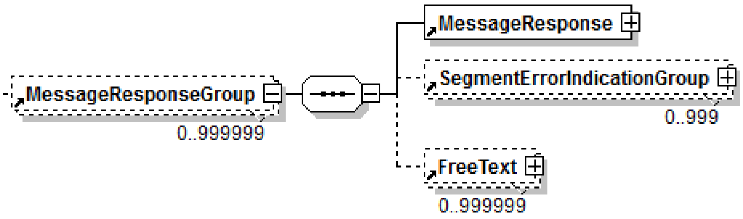
	(according to the scenario)		
SenderIdentificationCodeQualifier	Code which identifies the source of the code used to identify the sender Accepted values: **ZZZ**: Mutually Defined	O	String 1..4
InterchangeResponse\InterchangeRecipient			
RecipientIdentification	Code which identifies the recipient of the message being responded to Accepted values: **VALENCIAPORT, INTTRA, GT Nexus, CARGOSMART or Valenciaport code for the original message recipient** (according to the scenario)	M	String 1..35
RecipientIdentificationCodeQualifier	Code which identifies the source of the code used to identify the recipient Accepted values: **ZZZ**: Mutually Defined	O	String 1..4
InterchangeResponse\DataElementIdentification			
ErroneousDataElementPosition	Position of the element which contains the erroneous data	M	Decimal 3
ErroneousComponentDataElementPosition	Position of the erroneous component data element	O	Decimal 3
ErroneousComponentDataElementOccurrence	Number of the occurrence or repetition of the erroneous data element	O	Decimal 6

5.3.4. XML example

```
<InterchangeResponse>
    <InterchangeControlReference>123</InterchangeControlReference>
    <InterchangeSender>
        <SenderIdentification>VALENCIAPORT</SenderIdentification>
        <SenderIdentificationCodeQualifier>ZZZ</SenderIdentificationCodeQualifier>
    </InterchangeSender>
    <InterchangeRecipient>
        <RecipientIdentification>INTTRA</RecipientIdentification>
        <RecipientIdentificationCodeQualifier>ZZZ</RecipientIdentificationCodeQualifier>
    </InterchangeRecipient>
    <CodedAction>4</CodedAction>
    <DataElementIdentification>
        <ErroneousDataElementPosition>2</ErroneousDataElementPosition>
    </DataElementIdentification>
</InterchangeResponse>
```

5.4 // CONTRL\MessageResponseGroup

Level	**1**
Usage	O (Optional)
Max. Use	999999



5.4.1. Purpose

The *MessageResponseGroup* group of elements is used to identify the message being responded to, as well as indicating whether an error has occurred.

5.4.2. Comments

The *FreeText* group of elements is not used by the carriers that work with INTTRA.

The *SegmentErrorIndicationGroup* group of elements is not used by the carriers that work with GT Nexus or CARGOSMART.

5.4.3. Elements

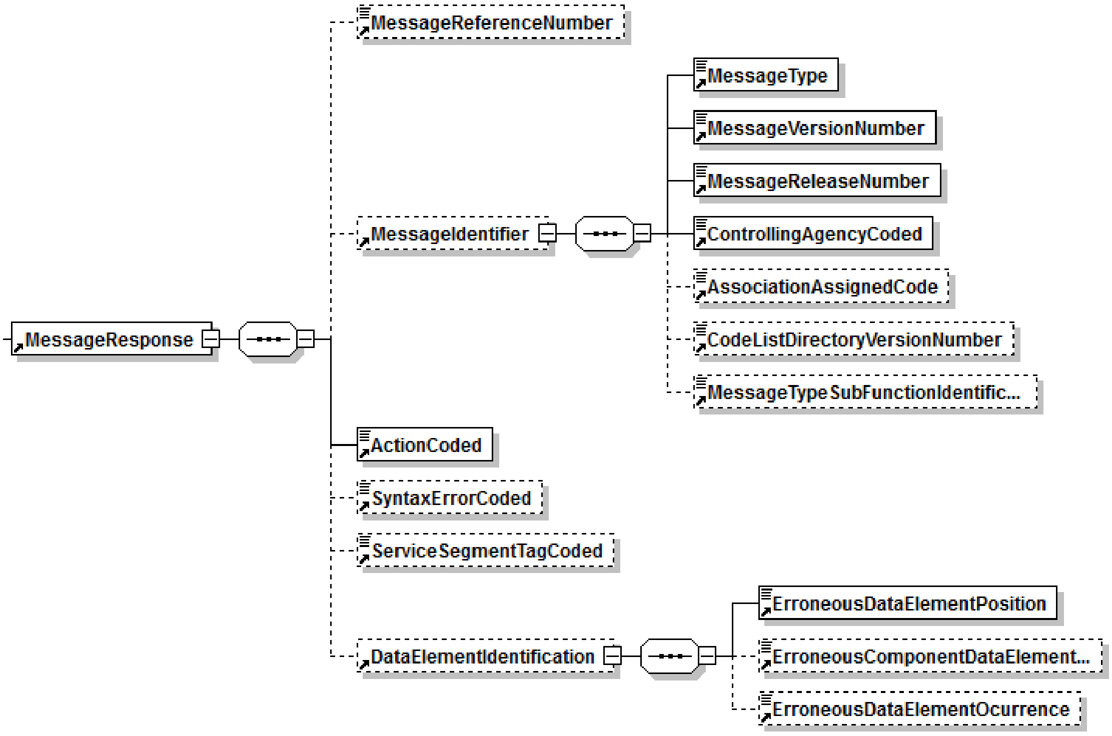
Name	*Purpose*	*M/O*	*Type*
MessageResponseGroup		*M*	
MessageResponse	Group of elements which identifies the message being responded to, indicating whether it contains any errors.	M	G
SegmentErrorIndicationGroup	Group of elements which identifies the segments and/or elements which contain errors	O	G
FreeText	Group of elements which describes whether the message contains any errors	O	G

5.4.4. XML example

```
<MessageResponseGroup>
    <MessageResponse>
        …
    </MessageResponse>
    <SegmentErrorIndicationGroup>
        …
    </SegmentErrorIndicationGroup>
    <FreeText>
        …
    </FreeText>
</MessageResponseGroup>
```

5.5 // CONTRL\MessageResponseGroup\MessageResponse

Level	2
Usage	M (Mandatory)
Max. Use	1
Group	MessageResponseGroup



5.5.1. Purpose

The *MessageResponseGroup* group of elements is used to identify the message being responded to, as well as indicating whether an error has occurred.

5.5.2. Comments

The *MessageIdentifier* and *MessageReferenceNumber* elements correspond to those of the message being responded to.

The *AssociationAssignedCode*, *CodeListDirectoryVersionNumber* and *MessageTypeSubFunctionIdentifier* elements in the *MessageIdentifier* group, and the *SyntaxErrorCoded* and *ServiceSegmentTagCoded* elements in the *DataElementIdentification* group are not used by the carriers that work with GT Nexus.

5.5.3. Elements

Name	*Purpose*	*M/O*	*Type*
MessageResponse		*M*	

MessageReferenceNumber	Reference number of the message being responded to	O	String 1..14
MessageIdentifier	Group of elements which identifies the type of message being responded to	O	G
ActionCoded	Code which indicates the acceptance of the message or errors in it Accepted values: • **4**: Document rejected • **7**: Document accepted • **8**: Interchange received	M	String 1..3
SyntaxErrorCoded	Code which indicates the type of error detected Accepted values: • **2**: *Syntax version or level not supported* • **7**: *Interchange recipient not actual recipient* • **12**: *Invalid value* • **13**: *Missing* • **14**: *Value not supported in this position* • **15**: *Not supported in this position* • **16**: *Too many constituents* • **17**: *No agreement* • **18**: *Unspecified error* • **20**: *Character invalid as service character* • ***21***: *Invalid character(s)* • ***22***: *Invalid service character(s)* • ***23***: *Unknown Interchange sender* • ***24***: *Too old* • ***25***: *Test indicator not supported* • ***26***: *Duplicate detected* • ***28***: *References do not match* • ***29***: *Control count does not match number of instances received* • ***30***: *Groups and messages/packages mixed* • ***32***: *Lower level empty* • ***33***: *Invalid occurrence outside message, package or group*	O	String 1..3

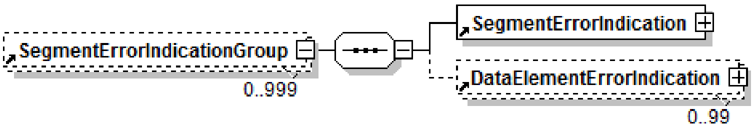
	- ***35***: *Too many repetitions* - ***36***: *Too many segment group repetitions* - ***37***: *Invalid type of character(s)* - ***39***: *Data element too long* - ***40***: *Data element too short* - ***44***: *Trailing separator* - ***45***: *Character set not supported* - ***46***: *Envelope functionality not supported*		
ServiceSegmentTagCoded	Identification code of a service segment	O	String 1..3
DataElementIdentification	Group of elements which identifies the position of an erroneous element	O	G
MessageResponse\MessageIdentifier			
MessageType	Code which identifies the type of message	M	String 1..6
MessageVersionNumber	Message version number	M	String 1..3
MessageReleaseNumber	Message version release number	M	String 1..3
ControllingAgencyCoded	Identification code of the controlling agency	M	String 1..2
AssociationAssignedCode	Code which identifies the message, assigned by the association responsible for its design and maintenance	O	String 1..6
CodeListDirectoryVersionNumber	Code list directory version number	O	String 1..6
MessageTypeSubFunctionIdentifier	Code which identifies the subfunction of the type of message	O	String 1..6
MessageResponse\DataElementIdentification			
ErroneousDataElementPosition	Position of the element which contains the erroneous data	M	Decimal 3
ErroneousComponentDataElementPosition	Position of the erroneous component data element	O	Decimal 3
ErroneousComponentDataElementOccurrence	Number of the occurrence or repetition of the erroneous data element	O	Decimal 6

5.5.4. XML example

```
<MessageResponse>
    <MessageReferenceNumber>318</MessageReferenceNumber>
    <MessageIdentifier>
        <MessageType>IFTMIN</MessageType>
        <MessageVersionNumber>D</MessageVersionNumber>
        <MessageReleaseNumber>99B</MessageReleaseNumber>
        <ControllingAgencyCoded>UN</ControllingAgencyCoded>
        <AssociationAssignedCode>String</AssociationAssignedCode>
        <CodeListDirectoryVersionNumber>String</CodeListDirectoryVersionNumber>
        <MessageTypeSubFunctionIdentification>String</MessageTypeSubFunctionIdentification>
    </MessageIdentifier>
    <ActionCoded>4</ActionCoded>
    <SyntaxErrorCoded>12</SyntaxErrorCoded>
    <ServiceSegmentTagCoded>Str</ServiceSegmentTagCoded>
    <DataElementIdentification>
        <ErroneousDataElementPosition>3</ErroneousDataElementPosition>
        <ErroneousComponentDataElementPosition></ErroneousComponentDataElementPosition>
        <ErroneousDataElementOccurrence></ErroneousDataElementOccurrence>
    </DataElementIdentification>
</MessageResponse>
```

5.6 // CONTRL\MessageResponseGroup\SegmentErrorIndicationGroup

Level	**2**
Usage	O (Optional)
Max. Use	999
Group	MessageResponseGroup



5.6.1. Purpose

The *SegmentErrorIndicationGroup* group of elements is sent in response to segments which contain errors and are part of the message identified in the *MessageResponse* group.

5.6.2. Comments

This group of elements is not used by carriers that work with GT Nexus or CARGOSMART.

5.6.3. Elements

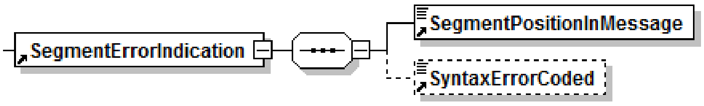
Name	***Purpose***	***M/O***	***Type***
SegmentErrorIndicationGroup			
SegmentErrorIndication	Group of elements which identifies the segment and the error it contains	M	G
DataElementErrorIndication	Group of elements which identifies an erroneous data element	O	G

5.6.4. XML example

```
<SegmentErrorIndicationGroup>
    <SegmentErrorIndication>
        …
    </SegmentErrorIndication>
    <DataElementErrorIndication>
        …
    </DataElementErrorIndication>
</SegmentErrorIndicationGroup>
```

5.7 // CONTRL\MessageResponseGroup\SegmentErrorIndicationGroup\SegmentErrorIndication

Level	**3**
Usage	M (Mandatory)
Max. Use	1
Grupo	MessageResponseGroup\SegmentErrorIndicationGroup



5.7.1. Purpose

The *SegmentErrorIndication* group of elements identifies a segment which contains errors and the type of error detected.

5.7.2. Comments

Not applicable.

5.7.3. Elements

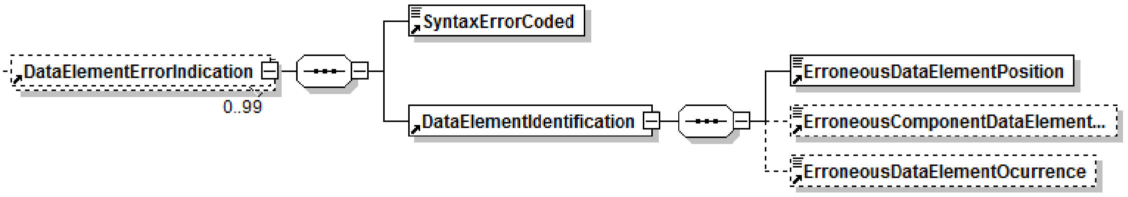
Name	***Purpose***	***M/O***	***Type***
SegmentErrorIndication		*M*	
SegmentPositionMessage	Segment position	M	Decimal 1..6
SyntaxErrorCoded	Code of the type of error detected	O	String 1..3

5.7.4. XML example

```
<SegmentErrorIndication>
    <SegmentPositionInMessage>4</SegmentPositionInMessage>
    <SyntaxErrorCoded>12</SyntaxErrorCoded>
</SegmentErrorIndication>
```

5.8 // CONTRL\MessageResponseGroup\SegmentErrorIndication\DataElementErrorIndication

Level	**3**
Usage	O (Optional)
Max. Use	99
Group	MessageResponseGroup\SegmentErrorIndicationGroup


DataElementErrorIndication
0..99
SyntaxErrorCoded
DataElementIdentification
ErroneousDataElementPosition
ErroneousComponentDataElement...
ErroneousDataElementOcurrence

5.8.1. Purpose

The *DataElementErrorIndication* group of elements identifies a data element in the segment identified in the *SegmentErrorIndication* group which contains errors and the type of error detected.

5.8.2. Comments

Not applicable.

5.8.3. Elements

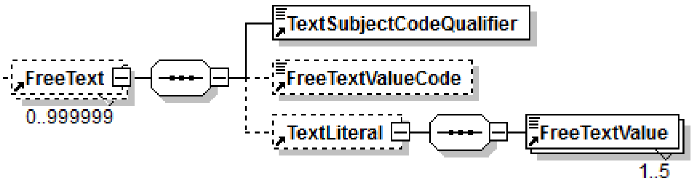
Name	*Purpose*	*M/O*	*Type*
DataElementErrorIndication			
SyntaxErrorCoded	Code of the type of error detected	M	String 1..3
MessageIdentifier	Group of elements which identifies the position of an erroneous element	M	G
DataElementErrorIndication\DataElementIdentification			
ErroneousDataElementPosition	Position of the element which contains the erroneous data	M	Decimal 3
ErroneousComponentDataElementPosition	Position of the erroneous component data element	O	Decimal 3
ErroneousComponentDataElementOccurrence	Number of the occurrence or repetition of the erroneous data element	O	Decimal 6

5.8.4. XML example

```
<DataElementErrorIndication>
    <SyntaxErrorCoded>12</SyntaxErrorCoded>
    <DataElementIdentification>
        <ErroneousDataElementPosition>3</ErroneousDataElementPosition>
        <ErroneousComponentDataElementPosition>1</ErroneousComponentDataElementPosition>
        <ErroneousDataElementOcurrence>1</ErroneousDataElementOcurrence>
    </DataElementIdentification>
</DataElementErrorIndication>
```

5.9 // CONTRL\MessageResponseGroup\FreeText

Level	**2**
Usage	O (Optional)
Max. Use	999999
Group	MessageResponseGroup



5.9.1. Purpose

The *FreeText* group of elements is used to provide additional information about whether there are any errors or not.

5.9.2. Comments

The FreeTextValueCode element is not used by **valenciaportpcs.net.**

The *TextLiteral* group of elements is required by ***valenciaportpcs.net*** and must contain only one instance of the *FreeTextValue* element.

5.9.3. Elements

Name	*Purpose*	*M/O*	*Type*
FreeText			
TextSubjectCodeQualifier	Code which specifies the purpose of the text **Accepted values:** **AAI:** General description	M	String 1..3
~~FreeTextValueCode~~	Code which specifies the text	O	String 1..17
TextLiteral	Free text group of elements	O	G
FreeText\TextLiteral			
FreeTextValue	Free text	M	String 1..512

5.9.4. XML example

```
<FreeText>
    <TextSubjectCodeQualifier>AAI</TextSubjectCodeQualifier>
    <TextLiteral>
        <FreeTextValue>Mensaje recibido y procesado sin errores</FreeTextValue>
    </TextLiteral>
</FreeText>
```

6 // CONTRL XML example

The following example aims to serve as a reference for sending or receiving a CONTRL message. Logically, the organization codes are not valid codes. Imaginary codes have been used for demonstration purposes. We have not aimed to use logical message content from a business point of view. Sometimes details which would never be used in real life have been used, but once again, the idea is to demonstrate all the possible message elements that may exist.

```
<?xml version="1.0" encoding="UTF-8"?>
<CONTRL>
    <InterchangeHeader>
        <SyntaxIdentification>
            <SyntaxIdentifier>UNOC</SyntaxIdentifier>
            <SyntaxVersionNumber>2</SyntaxVersionNumber>
        </SyntaxIdentification>
        <InterchangeSender>
            <SenderIdentification>INTTRA</SenderIdentification>
            <SenderIdentificationCodeQualifier>ZZZ</SenderIdentificationCodeQualifier>
        </InterchangeSender>
        <InterchangeRecipient>
            <RecipientIdentification>VALENCIAPORT</RecipientIdentification>
            <RecipientIdentificationCodeQualifier>ZZZ</RecipientIdentificationCodeQualifier>
        </InterchangeRecipient>
        <DateAndTimeOfPreparation>
            <DateOfPreparation>151224</DateOfPreparation>
            <TimeOfPreparation>1215</TimeOfPreparation>
        </DateAndTimeOfPreparation>
        <InterchangeControlReference>1</InterchangeControlReference>
    </InterchangeHeader>
    <MessageHeader>
        <MessageReferenceNumber>VPP12003102301</MessageReferenceNumber>
        <MessageIdentifier>
            <MessageType>CONTRL</MessageType>
            <MessageVersionNumber>4</MessageVersionNumber>
            <MessageReleaseNumber>1</MessageReleaseNumber>
            <ControllingAgencyCoded>UN</ControllingAgencyCoded>
        </MessageIdentifier>
    </MessageHeader>
    <InterchangeResponse>
        <InterchangeControlReference>123</InterchangeControlReference>
        <InterchangeSender>
            <SenderIdentification>VALENCIAPORT</SenderIdentification>
            <SenderIdentificationCodeQualifier>ZZZ</SenderIdentificationCodeQualifier>
        </InterchangeSender>
        <InterchangeRecipient>
            <RecipientIdentification>INTTRA</RecipientIdentification>
            <RecipientIdentificationCodeQualifier>ZZZ</RecipientIdentificationCodeQualifier>
        </InterchangeRecipient>
        <ActionCoded>4</ActionCoded>
        <DataElementIdentification>
            <ErroneousDataElementPosition>2</ErroneousDataElementPosition>
        </DataElementIdentification>
    </InterchangeResponse>
    <MessageResponseGroup>
        <MessageResponse>
            <MessageReferenceNumber>318</MessageReferenceNumber>
            <MessageIdentifier>
                <MessageType>IFTMIN</MessageType>
                <MessageVersionNumber>D</MessageVersionNumber>
                <MessageReleaseNumber>99B</MessageReleaseNumber>
                <ControllingAgencyCoded>UN</ControllingAgencyCoded>
                <AssociationAssignedCode>String</AssociationAssignedCode>
                <CodeListDirectoryVersionNumber>String</CodeListDirectoryVersionNumber>
                <MessageTypeSubFunctionIdentification>String</MessageTypeSubFunctionIdentification>
            </MessageIdentifier>
            <ActionCoded>74</ActionCoded>
            <SyntaxErrorCoded>12</SyntaxErrorCoded>
            <ServiceSegmentTagCoded>Str</ServiceSegmentTagCoded>
            <DataElementIdentification>
                <ErroneousDataElementPosition>3</ErroneousDataElementPosition>
                <ErroneousComponentDataElementPosition>1</ErroneousComponentDataElementPosition>
                <ErroneousDataElementOcurrence>2</ErroneousDataElementOcurrence>
            </DataElementIdentification>
        </MessageResponse>
        <SegmentErrorIndicationGroup>
            <SegmentErrorIndication>
                <SegmentPositionInMessage>4</SegmentPositionInMessage>
                <SyntaxErrorCoded>12</SyntaxErrorCoded>
            </SegmentErrorIndication>
            <DataElementErrorIndication>
                <SyntaxErrorCoded>12</SyntaxErrorCoded>
                <DataElementIdentification>
                    <ErroneousDataElementPosition>3</ErroneousDataElementPosition>
                    <ErroneousComponentDataElementPosition>1</ErroneousComponentDataElementPosition>
                    <ErroneousDataElementOcurrence>1</ErroneousDataElementOcurrence>
                </DataElementIdentification>
            </DataElementErrorIndication>
        </SegmentErrorIndicationGroup>
        <FreeText>
            <TextSubjectCodeQualifier>AAI</TextSubjectCodeQualifier>
            <FreeTextValueCode>Error de esquema</FreeTextValueCode>
            <TextLiteral>
                <FreeTextValue>Error: El tipo de BL es obligatorio. </FreeTextValue>
            </TextLiteral>
        </FreeText>
        <FreeText>
            <TextSubjectCodeQualifier>AAI</TextSubjectCodeQualifier>
            <TextLiteral>
                <FreeTextValue>Mensaje recibido y procesado sin errores</FreeTextValue>
            </TextLiteral>
        </FreeText>
    </MessageResponseGroup>
</CONTRL>
```

valenciaport

valenciaport pcs.net
Port Community System

User Service Desk
Avenida Muelle del Turia, s/n
46024 Valencia
Tel. No.: 902 88 44 24
R.C.I.: 10 001
www.valenciaportpcs.net
cau@valenciaportpcs.net