



Customs Department under the
Ministry of Finance of the
Republic of Lithuania

CUSTOMS CONTROL SYSTEM FOR PRESENTATION OF GOODS

Interface Specification for customs clients

Reference PPMKS-CLIENT-IS-EN

Version 1.00

Volume I Introduction

Document control

Project	PPMKS	
Product	Interface Specification for customs clients	
Description	The specification defines interface between PPMKS system and trader's applications.	
Files pages	PPMKS-CLIENT-IS-EN_v1.00_vol_I.docx	47
	PPMKS-CLIENT-IS-EN_v0.05_vol_II.docx	32
	PPMKS-CLIENT-IS-EN_v0.05_vol_II.docx	59

Revisions

Ver.	Date	Description	Actions*	Chapters
1.00	2022-02-10	First official version	I	All

* Actions: I = Insert, C = Change, D = Delete

Table of contents

1	Introduction	5
1.1	Goal and structure.....	5
1.2	Reference and applicable documents	6
1.3	Terminology	6
1.4	Message structure definition	7
1.4.1	Layout	7
1.4.2	Rules explanation.....	9
1.4.3	Data types	10
1.4.4	Optionality of data items and data groups	11
1.4.5	Technical structure of the XML document	11
1.4.6	Allowed character set	12
1.4.7	Indicating data groups and data items.....	12
1.4.8	“Empty” values	13
1.4.9	Digital signature.....	14
1.5	Metadata	15
2	Message flows	16
2.1	Receiving message from trader	16
2.2	Sending message to trader	18
2.3	Receiving messages by trader	19
3	Messages.....	20
3.1	ACK.....	21
3.2	NACK	23
3.3	Rules	26
4	Reference data	27
4.1	ReferenceData	28
4.2	Rules	43
4.3	Code lists	44

List of figures

Figure 1 Receiving message from trader 16

Figure 2 Sending message to trader 18

Figure 3 Receiving messages by trader 19

1 Introduction

1.1 Goal and structure

The purpose of this specification is to define interface between PPMKS system and trader's applications. Whole communication is going through EM-VARTAI system. Most of technical details for trader's application and EM-VARTAI communication are defined in separate specification [A1].

This specification is addressed to the developers who construct trader's application which will exchange information with PPMKS system.

The specification is divided on volumes. Each volume contains the following main chapters:

- **Introduction** that defines the goal of volume.
- **Message flows** that show the sequence of messages exchanged between trader's application and PPMKS.
- **Messages** that define structure of all the exchanged messages.

The specification is delivered as "zip" file with the following directory structure:

- **text**: consists of text of all specification's volumes (and optionally additional documents e.g., list of changes made in comparison to the previous version).
- **xsd**: messages' schema definitions.

The goal of this first volume is to define common information for all other volumes. It means:

- **Applicable documents**.
- **Terminology**.
- **Message structure definition**.
- **Used Metadata**.
- General message flow scenarios between trader's application and PPMKS through EM-VARTAI system. See **Message flows**.
- General purpose Acknowledgement (ACK) and Negative acknowledgement (NACK) messages structure. See **Messages**.
- **Reference data** structures definition and list of possible reference data (list of code lists).

1.2 Reference and applicable documents

Id	Title	Reference	Version
A1	ESB modulio EM-VARTAI žiniatinklio paslaugos, skirtos išorinių sistemų komunikacijai, specifikacija		3.00

Table 1 Reference and applicable documents

1.3 Terminology

Acronym/Term	Description
EM-VARTAI	System for communication of external systems with customs subsystems.
EU	European Commission
EUCDM	European Commission Customs Data Model. Defined by UCC-DA and UCC-IA.
ENS	Entry Summary Declaration
IE	Information Exchange.
IS	Interface Specification.
LRN	Local Reference Number. Business identifier assigned to declaration by trader.
MRN	Master Reference Number. The registration number allocated by competent customs authority to declarations or notifications.
PPMKS	CUSTOMS CONTROL SYSTEM FOR PRESENTATION OF GOODS (Prekių Pateikimo Muitinės Kontrolės Sistema)
PPMKS Portal	Portal for traders' to support Presentation Notification communication and tool for ENSes searching.
Trader	Client i.e., any person who sends/receives messages defined by this specification.
UCC	Union Customs Code. Regulation (EU) No 952/2013 with later amendments
UCC-DA	UCC Delegated Act. Commission Delegated Regulation No 2015/2446 with later amendments.
UCC-IA	UCC Implementing Act. Commission Implementing Regulation No 2015/2447 with later amendments.
XML	The Extensible Markup Language (XML) is the universal format for structured documents and data on the Web.
XML Schema	XML Schema, published as a W3C recommendation in May 2001, is one of several XML schema languages.

Table 2 Terminology

1.4 Message structure definition

Each message (technically XML document) is defined in separate subchapter of **Messages** chapter. The last additional **Rules** subchapter collects all messages rules.

1.4.1 Layout

Message definition consists of Structure and Contents sections as follows (fragmentary examples):

Structure

a	b	c
IEG347	1	-
- Notification	1	-
- - Declaration	1..9999	-
- - - Consignment	1	-
- - - - TransportDocumentMasterLevel	0..1	R803LT

where:

- a** Column defines the message structure – a tree of data groups (technically XML elements). The number of “-” characters reflects the nesting level of the data group in the tree.
- b** Column defines data groups cardinalities. Possible values are “*n*” and “*n..k*”, where *n*, *k* are integer values. Value “*n*” means that the element must occur exactly “*n*” times. Value “*n..k*” means that element must occur at least “*n*” and maximum “*k*” times. Typical cases:
 - “1” element must occur only once,
 - “0..1” element may occur but only once,
 - “1..9” element must occur at least once and maximum 9 times,
 - “0..9” element may occur, but maximum 9 times.
- c** Column contains rule identifiers assigned to the data groups (see **Rules explanation** chapter for details). If there are no rules assigned “-” character is used.

Contents

a	IEG347.Notification.Declaration.Consignment. TransportDocumentMasterLevel					
Transport document	b			c	d	
	12 05 000 000			0..1	R803LT	
<hr/>						
e	f	g	h	i	j	
referenceNumber	12 05 001 000	an..70	-	R	R780LT	
Reference number						
type	12 05 002 000	an4	329COM	R	-	
Type						

where:

- a** Dot notation path to the specified data group. The XML name of this data group is bolded. Second line is business name, subsequent lines (if any) contain optional commentary text.
- b** EUCDM "Data class/sub-class number". "-" if not relevant.
- c** Data group cardinality (see also **Structure** section).
- d** Rule identifiers assigned to the data group (see **Rules explanation** chapter). "-" if not relevant.

Data items contained inside data group:

- e** Data item XML name (bolded) followed by business name (second line) and optional commentary text (subsequent lines).
- f** EUCDM "Data element number". "-" if not relevant.
- g** Data item type (see **Data types** chapter).
- h** Identifier of reference data (e.g., code list of country codes). "-" if not relevant.
- i** Possible values: „R” - required, „O” – optional.
- j** Rule identifiers assigned to the data item (see **Rules explanation** chapter). "-" if not relevant.

1.4.2 Rules explanation

All rules are defined in **Rules** chapter and described as follows:

a	b
R794LT	#SchemaValidated
containerIdentificationNumber may contain only the following characters: '0-9', 'A-Z', 'a-z'.	
c	
<hr/> <i>Container number may only contain numbers from '0' to '9', capital letters from 'A' to 'Z' and lowercase letters from 'a' to 'z'.</i>	
d	

where:

- a Rule's identifier.
- b Optional tag(s) of rule. Possible values:
 - #SchemaValidated. The rule is validated by XML Schema files of this specification.
 - #Contextual. The rule cannot be checked without real sending the message to the PPMKS.
- c Formal text of rule.
- d Optional informal text of rule, useful for end users when formal text contains a lot of technical details.

1.4.3 Data types

Alphanumeric type is described by „ann” or „an..n”, where “n” is integer value. This type defines alphanumeric text of an exactly specified length - „ann” or text of allowed maximum length - „an..n”.

This type is also applicable to data items composed of digits only which are not interpreted as numbers in mathematical meaning but as identifiers (e.g., mode of transport, customs procedure number). Examples:

an2 LT

an..35Trade-Service Enterprise

an2 40

Decimal number type is described by „n..k,l”, where “k” is a number, defining the maximum number of digits including a fractal part and “l” stands for maximum number of fractal digits. This type defines decimal numbers in mathematical meaning. The separator character for fractal parts is “.” (full stop) mark. This mark is not considered in “k”, “l” numbers. Examples:

n..3,2 0.22, 3.22; 33.2, 333

Integer number type is described by „n..k”, where “k” is a number, defining the maximum number of digits. This type defines integer numbers in mathematical meaning. Examples:

n..3 3, 33; 333

Date type is described as „date” and defines date. Values of this type should be entered in the following format: yyyy-mm-dd, where: yyyy stands for a year, mm stands for a month, dd stands for a day. A separator mark is “-”. The number of a month and day must be preceded by zero if applicable. Example:

date 2020-07-17

Date and time type is described as „dateTime” and is used to define date and time. Values of this type follow date type rules and are followed by the “T” character and time value using format: hh:mm:ss, where hh stands for hours, mm stands for minutes, ss stands for seconds. A separator sign is “:” (colon). Hours, minutes, and seconds must be preceded by zero if applicable. The time is defined in 24-hour system. Example:

dateTime 2020-07-17T10:05:55

Boolean type is described by „boolean”. It defines two values „1” – yes/true and „0” – no/false. Examples:

boolean 1; 0

Base64 binary type described as “base64”. Binary data encoded with base64 algorithm. It is useful to include any format file inside message e.g., file with scan of document. Example:

base64 U29tZSBCYXNINjQgdmFsdWU

1.4.4 Optionality of data items and data groups

Optional data items (defined by “O”) and optional data groups (defined by “0..n”) should be treated as optional only from data structure point of view.

In this specification data items/data groups may have additionally assigned rules which may change the optionality to mandatory or prohibited.

Outside this specification the additional rules or explanations may exist which precise data items/data groups meaning and rules of filling.

1.4.5 Technical structure of the XML document

XML document, compliant with this specification should consist of:

- XML declaration that defines version of XML standard (attribute version) and encoding (attribute encoding) used to encode the entire XML document. The used version of XML standard is 1.0.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

- Body of the document compliant with definitions of this specification and included XML schemas. Fragmentary example:

```
<n1:IEG347 xmlns:n1="http://lrmuitine.lt/ppmks/IEG347.xsd"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  <messageSender>/IEG347/Notification/PersonPresentingGoods/identificationNumber</messageSender>
  <preparationDateAndTime>2001-12-17T09:30:47</preparationDateAndTime>
  <messageIdentification>03904b97-b5f1-49f6-a718-f403d787b804</messageIdentification>
  <Notification>
    <lrm>SK-220421-01</lrm>
    <declarationDate>2001-12-17T09:30:00</declarationDate>
    ...
  </Notification>
  <ds:Signature Id="sig00">
    ...
  </ds:Signature>
</n1:IEG347>
```

Remarks:

Schemas defined by this specification forces elements to be unqualified – it means that root element, **and only root element**, must be prefixed (“n1:IEG347” on above example). This rule is not relevant for digital signature part (see “Digital signature” chapter).

The prefixes’ names can be anything (here “n1” and “ds”).

1.4.6 Allowed character set

Allowed character set is defined by the encoding included in the XML document declaration part as follows:

```
<?xml version="1.0" encoding="UTF-8"?>.
```

Only “UTF-8” encoding is allowed (XML documents with other encoding are not accepted).

Some characters, regardless of the encoding, are reserved for the XML standard. These characters when needed to be used in values must be replaced by the following sequences of characters:

- "&" must be replaced with "&"
- quotation mark ' ' ' must be replaced with """
- apostrophe " ' " must be replaced with "'"
- lower than "<" must be replaced with "<"
- greater than ">" must be replaced with ">"

For example, text: "ABC" Trade-Service Enterprise, Smith & Sons Civil Association

Must be replaced by "ABC" Trade-Service Enterprise, Smith & Sons Civil Association.

1.4.7 Indicating data groups and data items

To indicate data group precisely a full dot path notation is used. The expression begins with name of root data group and the “indicated” element is placed right at the end.

For example, to indicate the data group „PersonPresentingGoods” in the data group „**Notification**”, which is in „**IEG347**”, the following expression must be used:

IEG347.Notification.PersonPresentingGoods

To indicate the number of repeatable data group “[*number*]” expression is used.

For example, to point second „**Declaration**”, the following expression must be used:

IEG347.Notification.Declaration[2]

To indicate a data item precisely, the same expressions are used as for indicating data groups.

For example, to indicate data item „mrn” of the second element „**Declaration**”, the following expression should be used:

IEG347.Notification.Declaration[2].mrn

In the rules descriptions shorter useful notation is used very often. Path of parent’s data groups is not fully expressed. For example:

Notification.Irn instead **IEG347.Notification.Irn**

1.4.8 “Empty” values

It is not allowed to use “empty” values or only white spaces (e.g., spaces, tabulators) for data item (XML element).

For example, if phone number is not used then, instead:

```
<ContactPerson>
  <name>Vardenis Pavardenis</name>
  <phoneNumber></phoneNumber>
  <eMailAddress>vardenis.pavardenis@domain.lt</eMailAddress>
</ContactPerson>
```

or

```
<ContactPerson>
  <name>Vardenis Pavardenis</name>
  <phoneNumber> </phoneNumber>
  <eMailAddress>vardenis.pavardenis@domain.lt</eMailAddress>
</ContactPerson>
```

or

```
<ContactPerson>
  <name>Vardenis Pavardenis</name>
  </phoneNumber>
  <eMailAddress>vardenis.pavardenis@domain.lt</eMailAddress>
</ContactPerson>
```

<phoneNumber> should be skipped at all as in the following example:

```
<ContactPerson>
  <name>Vardenis Pavardenis</name>
  <eMailAddress>vardenis.pavardenis@domain.lt</eMailAddress>
</ContactPerson>
```

1.4.9 Digital signature

The digital signature included in XML documents must comply with the following requirements:

- Digital signature should be enclosed in an additional element „**<ds:Signature ...**”
- Element „ds:Signature” should be the last sub-element of the main element.
- Element „ds:Signature” comes from namespace "http://www.w3.org/2000/09/xmldsig#" and has assumed for it prefix „ds” (but can be any other) thus this element (and its sub-elements) is proceeded by “ds.” - „ds:Signature”. Declaration of namespace and prefix are realised by notation: **xmlns:ds="http://www.w3.org/2000/09/xmldsig#"**. In example here, it is placed on root element but can be set on ds:Signature element as well.
- The content of ds:Signature element must comply with „XML-Signature Syntax and Processing, W3C Recommendation 12 February 2002” specification. Xsd schema for elements and attributes representing digital signature part of a document (it means ds:Signature element) is enclosed to the current specification in „xmldsig-core-schema.xsd” file.
- Document must be signed as a whole - it is not allowed to sign chosen parts of the document only.

Fragmentary example of a document signed electronically is presented below (digital signature elements are highlighted with bold font):

```
<n1:IEG347 xmlns:n1="http://lrmutine.lt/ppmks/IEG347.xsd"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <messageSender>/IEG347/Notification/PersonPresentingGoods/identificationNumber</messageSender>
  <preparationDateAndTime>2001-12-17T09:30:47</preparationDateAndTime>
  <messageIdentification>03904b97-b5f1-49f6-a718-f403d787b804</messageIdentification>
  <Notification>
    <lrn>SK-220421-01</lrn>
    ...
  </Notification>
  <ds:Signature Id="sig00">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>CaVrgVSJFL+uz7yUdMlpq5wfIG0=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>S03DvAqStrStt5qi4nxEQNGNoXwEe7lOQXj0lqU0Ga+M1c471TwoNSamnk96bnitgC
SPEYuOXESb
AsTHBKAMP02yzYv3sSCH2x15vBEJTnv9E/yLXdzp0do3E4ALhYyz10Q1BTUHjSy0HKRG+5whRirk
REUkM8MtLZkhfV350H4=</ds:SignatureValue>
    <ds:KeyInfo>
      <ds:KeyName>d67cce00f84c364eb608e7351de5711a8b33d650</ds:KeyName>
    </ds:KeyInfo>
  </ds:Signature>
</n1:IEG347>
```

1.5 Metadata

To each send/received message an additional data called "metadata" are assigned. They allow to search and to correlate messages.

For details how to use metadata in communication and searching see [A1] specification, chapter "Išorinės sistemos pranešimų siuntimas muitinės sistemoms" (methods: *SubmitMessage*, *GetMessageList*, *GetMessage*).

The following metadata are used by PPMKS.

Basic:

- **DocID.** Unique technical identifier of the message in customs administration.
- **Time.** Date and time of reception/sending.
- **Name.** Name of message (typically name of XML root element) e.g., "IEG347".
- **Format.** MIME type format of the message (typically "text/xml").

Sender/Recipient:

- **Sender.domain.** Possible values: "TRAD" (sender is trader), "PPMKS" (sender is PPMKS).
- **Sender.identifier.** Possible values: EORI number of trader (sender is trader), "PPMKS" (sender is PPMKS).
- **Recipient.domain.** Possible values: "TRAD" (recipient is trader), "PPMKS" (recipient is PPMKS).
- **Recipient.identifier.** Possible values: EORI number of trader (recipient is trader), "PPMKS" (recipient is PPMKS).

Other metadata (as pair name and value):

- **messageReference.** Value extracted from *.messageIdentification element of the message.
- **localReference.** When relevant, LRN number assigned to notification/declaration.
- **customsReference.** When relevant, MRN number assigned to notification/declaration.
- **responseTo.** Value of "DocID" metadata of the other message to which message is direct response.
- **customsIS.** Value of customs information system which is used for communication (Here always value "PPMKS").
- **direction.** Possible values "IN" and "OUT", representing message communication direction from PPMKS point of view (IN – PPMKS receiving message, OUT – PPMKS sending message).
- **TIN.** Person Presenting Goods EORI number of the notification/declaration to which the message refers. Placed automatically by EM-VARTAI, when operation *SubmitMessage* is called.
- **SID.** Unique identifier of the system that sent the message. Placed automatically by EM-VARTAI when operation *SubmitMessage* is called.

2 Message flows

This chapter defines how using EM-VARTAI interface:

- PPMKS receives any message from Trader's application.
- Any message is sent from PPMKS to trader's application

In the subsequent volumes of this specification details of each above communication cases are intentionally hidden - there is only presented business point of view: Trader's application, PPMKS and business messages passed between them (business protocol).

2.1 Receiving message from trader

Each message sent from trader (Trader's application) to PPMKS, before it is handled by a specific business process, is initially validated according very limited but crucial rules.

The result of this validation is sent back immediately and synchronously to the Trader's application as an ACK or NACK message.

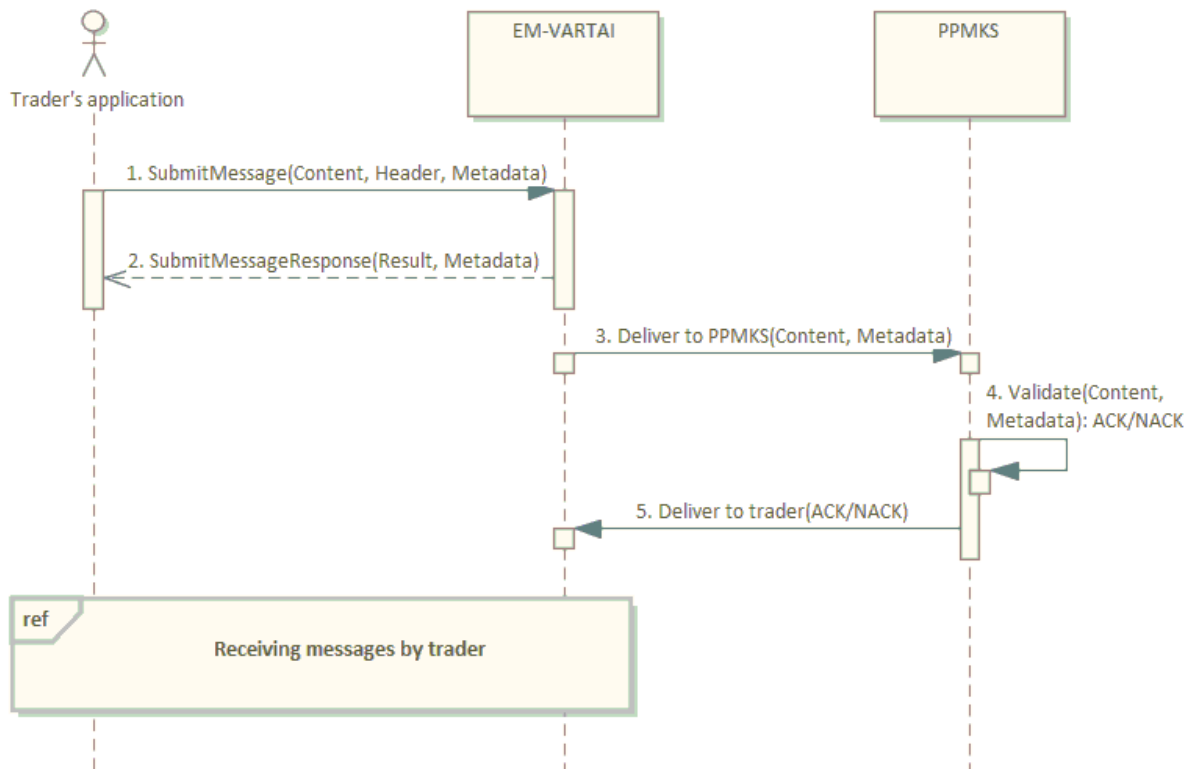


Figure 1 Receiving message from trader

1. Trader's representative using Trader's application prepares and sends to PPMKS using EM-VARTAI interface some business message (e.g., IEG347).

For list and details of business messages which can be sent/receive to/from PPMKS see subsequent volumes of this specification.

For technical details how to send any message to any customs system (e.g., PPMKS) see [A1] specification - in particular "Išorinės sistemos pranešimų siuntimas muitinės sistemoms" chapter about "SubmitMessage" method.

The following data must be set for "SubmitMessage" method to pass message to PPMKS correctly:

- SubmitMessage.Content (message content)
- Header.ticket (current ticket)
- Header.tin (as stated in the message PersonPresentingGoods .identificationNumber)
- Header.typeOfIdentifier ("EORI" value)
- Metadata.Time (current date and time)
- Metadata.Name (root name of the business message structure e.g., "IEG347"),
- Metadata.Format = 'text/xml'
- Sender.domain = 'TRAD',
- Sender.identifier (identifier of business sender - see 'messageSender' data items description of messages in subsequent volumes),
- Recipient.domain = 'PPMKS',
- Recipient.identifier = 'PPMKS'.

2. EM-VARTAI responses synchronously.

For details see also [A1] "SubmitMessage" chapter (SubmitMessageResponse method).

In case of error (SubmitMessageResponse.Result.resultCode <> "1") the scenario finishes here (not depicted on diagram intentionally).

In case of positive scenario in the response additional metadata "DocID" is returned.

3. EM-VARTAI delivers to PPMKS business message content and its metadata.

4. PPMKS makes initial validation. The validation scope is limited to message structure, metadata, and message uniqueness (see rules R783LT, R784LT, R785LT, R786LT and R787LT in subsequent volumes of this specification). As a result, PPMKS prepares ACK - acknowledge response message (success - no errors) or NACK - negative acknowledgement response message (errors found).

5. PPMKS passes to EM-VARTAI response (ACK/NACK message content and its metadata).

Finally, the message is delivered to the trader - see "Receiving messages by trader" chapter.

2.2 Sending message to trader

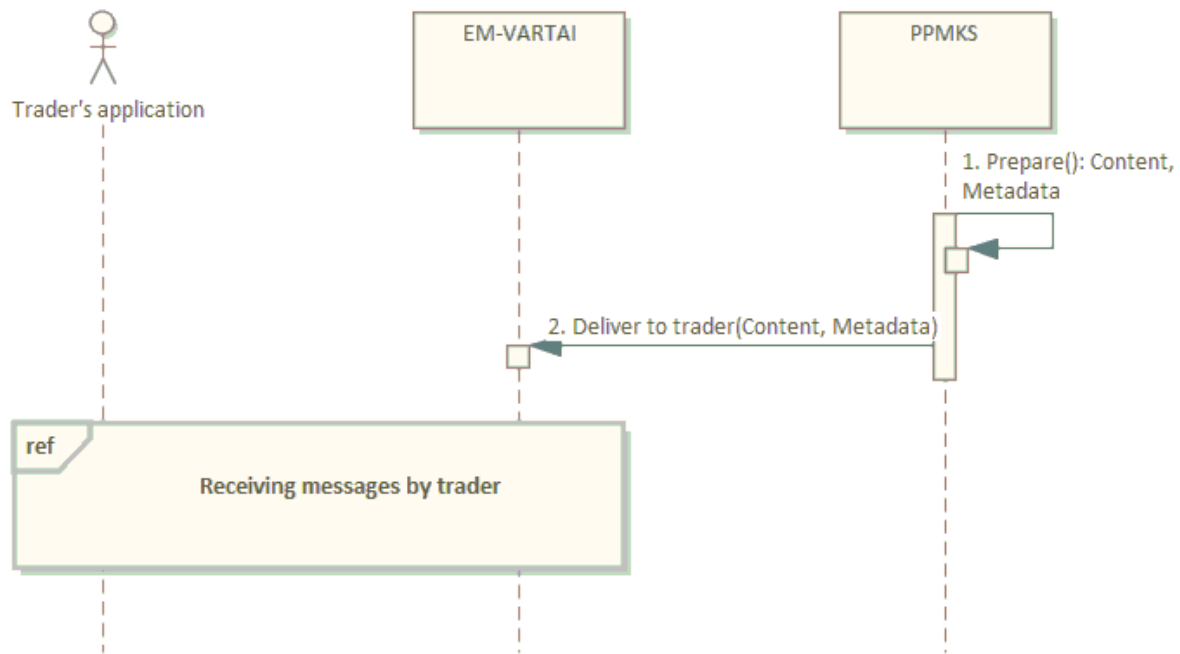


Figure 2 Sending message to trader

1. PPMKS prepares some business message to trader (e.g., IEG348). Together with message content the metadata are prepared.

2. PPMKS passed to EM-VARTAI prepared message content and metadata.

Finally, the message is delivered to the trader - see "Receiving messages by trader" chapter.

2.3 Receiving messages by trader

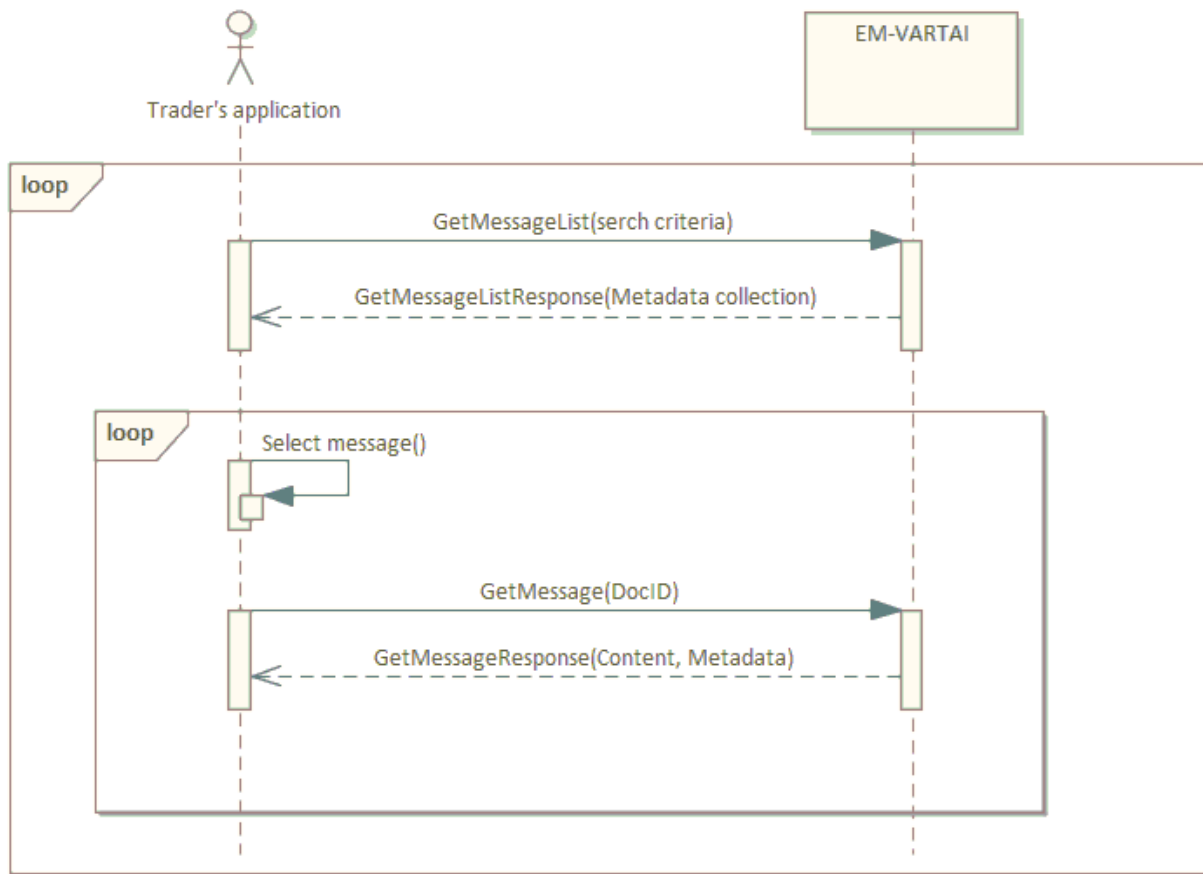


Figure 3 Receiving messages by trader

Receiving messages by the trader is based on polling method of asking EM-VARTAI for messages list according to criteria (GetMessageList) and retrieving their content and metadata by additional request (GetMessage).

For details see [A1] chapter "Pranešimų paieška".

For example, to check and get messages related to declaration where LRN or MRN is known GetMessageList with localReference or customsReference criteria is set to known LRN/MRN value.

Additionally, includeRelated option can be set to true to receive also response messages to fulfilled criteria messages.

As a result, list of messages (collection of metadata) will be returned (if such messages exist at all).

Next Trader's application must select new message(s) (from its point of view) and ask by GetMessage for its content setting as parameter its DocID metadata.

Remark

Header data of the GetMessageList and GetMessage should be the same as defined for SubmitMessage (see chapter Receiving message from trader)

3 Messages

This page is intentionally empty (each message definition starts from new page).

3.1 ACK

ACKNOWLEDGEMENT

General purpose technical acknowledgement message. It points that received message (see correlationIdentifier) passed preliminary validation with success and is directed to final business processing.

Structure

ACK	1	-
– Signature	0..1	-

Contents

ACK

Root element of the message	-			1	-
<hr/>					
preparationDateAndTime	-	dateTime	-	R	-
Preparation date and time of the message					
messageIdentification	-	an..40	-	R	R783LT
Message identification					
Unique identifier of the message assigned by sender.					
correlationIdentifier	-	an..40	-	R	R783LT
Correlation Identifier					
Value of messageIdentification data item of the message to which this message is direct response.					
ACK.Signature					
Digital signature	-			0..1	-
See chapter "Digital signature".					

3.2 NACK

NEGATIVE ACKNOWLEDGEMENT

General purpose technical rejection message. It points that received message (see correlationIdentifier) is wrong (see Error) and its processing is stopped.

Structure

NACK	1	-
– Error	1..999	-
– Signature	0..1	-

Contents

NACK

Root element of the message	-			1	-
<hr/>					
preparationDateAndTime	-	dateTime	-	R	-
Preparation date and time of the message					
messageIdentification	-	an..40	-	R	R783LT
Message identification					
Unique identifier of the message assigned by sender.					
correlationIdentifier	-	an..40	-	O	R783LT
Correlation Identifier					
Value of messageIdentification data item of the message to which this message is direct response.					
NACK. Error					
Error	-			1..999	-
<hr/>					
sequenceNumber	-	n..5	-	R	R789LT
Sequence number					
pointer	-	an..512	-	O	R780LT
Pointer					
Pointer to the place where error occurred.					
code	-	an..2	45LT_49	R	-
Code					
Code of error					
reason	-	an..10	-	O	R780LT
Reason					
Identifier of rule which produced error.					
originalAttribute	-	an..512	-	O	R780LT
Original attribute					
Original value the error concerns.					

description	-	an..1024	-	O	R780LT
Description					
Error description.					
source	-	an..20	243NOV	R	-
Source					
Code of error source (i.e., code of system or process in the system).					
NACK. Signature					
Digital signature	-			0..1	-
See chapter "Digital signature".					

3.3 Rules

R780LT

#SchemaValidated

Value of this data item (free text item) cannot:

- be empty,
- contain only white spaces (like spaces, tabulators),
- contain white spaces at the beginning or at the end of the text.

Value of this data item cannot be empty.

R783LT

#SchemaValidated

messageIdentification data item cannot be empty and may contain only the following characters:

'0-9', 'A-Z', 'a-z', '- '.

Message identification data item cannot be empty and may contain only the following characters:

'0-9', 'A-Z', 'a-z', '- '.

R789LT

Each sequenceNumber is unique for the Data group it belongs to. The sequence numbers shall be sequential, starting from '1' for the first iteration of the Data group and increasing by '1' for each iteration.

Sequence numbers must be unique sequential numbers in collection starting from 1.

4 Reference data

Reference data (also called as "Code lists") are published in XML files. The structure of this files is defined here (ReferenceData.xsd). Conventions used here are the same as for messages (see "Message structure definition" chapter).

Additionally in the subchapter "Code lists" all used reference data are enumerated pointing:

- "Code" (ReferenceData.id),
- name
- "Item type" (ReferenceData.itemType),
- when relevant "Item code type" (RerferenceData.itemCodeType),
- when useful description.

4.1 ReferenceData

REFERENCE DATA

Structure

ReferenceData	1	-
- NameAndDescription	1..*	-
- SimpleItem	0..*	-
- - CodeDescription	0..*	-
- CustomsOffice	0..*	-
- - CustomsOfficeDedicatedTrader	0..99	-
- - CustomsOfficeSpecificNotes	0..99	-
- - CustomsOfficeLSD	1..9	-
- - CustomsOfficeTimetable	0..9	-
- - - CustomsOfficeTimetableLine	1..99	-
- - - - CustomsOfficeRoleTraffic	1..99	-
- CountryRegion	0..*	-
- - Holiday	0..99	-
- - - HolidayLSD	1..9	-
- - CountryRegionLSD	1..9	-
- CountryHoliday	0..*	-
- - Holiday	1..99	-
- - - HolidayLSD	1..9	-
- Country	0..*	-
- - CodeDescription	0..*	-
- DocumentType	0..*	-
- - CodeDescription	0..*	-
- UN-LOCODE	0..*	-
- CommodityCode	0..*	-
- - CodeDescription	0..*	-
- Correlation	0..*	-
- ExchangeRate	0..*	-

– Location	0..*	-
– ValueAdjustement	0..*	-
– – CodeDescription	0..*	-
– – Apportionment	1..*	-

Contents

ReferenceData

Root element	-			1	-
id	-	an..20	-	R	-
Identifier of the particular reference data type.					
itemType	-	an..35	-	R	-
Name of the structure which carries values of particular reference data type.					
itemCodeType	-	an..6	-	O	-
Item type code.					
lastChange	-	date	-	R	-
Date of last change.					
ReferenceData.NameAndDescription					
Name and description of the reference data type in particular-language.				1..*	-
languageCode	-	an..2	-	R	-
Language code of description (see "12" reference data list).					
name	-	an..70	-	R	-
Name.					
description	-	an..4000	-	O	-
Description.					
ReferenceData.SimpleItem					
The structure which carries values of simple code lists.	-			0..*	-
code	-	an..20	-	R	-
Code.					
national	-	boolean	-	O	-
Nationally defined code flag (only value "true" is used, lack of this value means "false").					

validFrom	-	date	-	O	-
Valid from.					

validTo	-	date	-	O	-
Valid to.					

ReferenceData.SimpleItem.CodeDescription

This represents description of the code in particular language.	-			0..*	-
---	---	--	--	------	---

description	-	an..3072	-	R	-
Description.					

languageCode	-	an..2	-	R	-
Language code of description (see "12" reference data list).					

ReferenceData.CustomsOffice

Customs Office information.	-			0..*	-
-----------------------------	---	--	--	------	---

referenceNumber	-	an..8	-	R	-
Customs office reference number.					

referenceNumberMainOffice	-	an..8	-	O	-
Reference number of the main customs office.					

referenceNumberHigherAuthority	-	an..8	-	O	-
Reference number of the higher authority customs office.					

referenceNumberTakeOver	-	an..8	-	O	-
Reference number of the take-over customs office.					

referenceNumberAuthorityOfEnquiry	-	an..8	-	O	-
Reference number of the authority of enquiry customs office.					

referenceNumberAuthorityOfRecovery	-	an..8	-	O	-
Reference number of the authority of recovery customs office.					

countryCode	-	an..2	-	R	-
Country code.					

unLocodeId	-	an..3	-	O	-
Location code identifier.					

regionCode	-	an..3	-	O	-
Region code.					
nctsEntryDate	-	date	-	O	-
nearestOffice	-	an..175	-	O	-
Nearest office information.					
nearestOfficeLNG	-	an..2	-	O	-
Nearest office information - language code.					
postalCode	-	an..9	-	R	-
Postal code.					
phoneNumber	-	an..35	-	O	-
Phone number.					
faxNumber	-	an..35	-	O	-
Fax number.					
telexNumber	-	an..35	-	O	-
Telex number.					
emailAddress	-	an..70	-	O	-
Email address.					
geoInfoCode	-	an..6	-	O	-
Geo info code.					
traderDedicated	-	boolean	-	R	-
Trader dedicated flag.					
validFrom	-	date	-	O	-
Valid from.					
validTo	-	date	-	O	-
Valid to.					

ReferenceData.CustomsOffice.CustomsOfficeDedicatedTrader

Dedicated trader's data.	-			0..99	-
--------------------------	---	--	--	-------	---

name	-	an..35	-	R	-
Name.					

tin	-	an..17	-	O	-
------------	---	--------	---	---	---

Trader Identification Number.

languageCode	-	an..2	-	R	-
---------------------	---	-------	---	---	---

Trader's data - language code.

ReferenceData.CustomsOffice.**CustomsOfficeSpecificNotes**

Customs Office - specific notes.	-			0..99	-
----------------------------------	---	--	--	-------	---

specificNotesCode	-	an..6	-	R	-
--------------------------	---	-------	---	---	---

Specific notes code.

ReferenceData.CustomsOffice.**CustomsOfficeLSD**

Customs Office Local Specific Data.	-			1..9	-
-------------------------------------	---	--	--	------	---

languageCode	-	an..2	-	R	-
---------------------	---	-------	---	---	---

Customs Office Local Specific Data - language code.

usualName	-	an..35	-	R	-
------------------	---	--------	---	---	---

Usual name.

streetAndNumber	-	an..35	-	R	-
------------------------	---	--------	---	---	---

Street and number.

city	-	an..35	-	R	-
-------------	---	--------	---	---	---

City.

prefixSuffixLevel	-	an..1	-	O	-
--------------------------	---	-------	---	---	---

Prefix-suffix level.

prefixSuffixFlag	-	boolean	-	R	-
-------------------------	---	---------	---	---	---

Prefix-suffix flag.

prefixSuffixName	-	an..35	-	O	-
-------------------------	---	--------	---	---	---

Prefix-suffix name.

spaceToAdd	-	boolean	-	R	-
-------------------	---	---------	---	---	---

Space to add flag.

ReferenceData.CustomsOffice.**CustomsOfficeTimetable**

Customs office - timetable.	-			0..9	-
seasonCode	-	an..1	-	R	-
Season code.					
seasonName	-	an..35	-	O	-
Season name.					
seasonNameLNG	-	an..2	-	O	-
Season name - language code.					
seasonStartDate	-	date	-	R	-
seasonEndDate	-	date	-	R	-

ReferenceData.CustomsOffice.CustomsOfficeTimetable.**CustomsOfficeTimetableLine**

Customs office - timetable line.	-			1..99	-
beginDay	-	an..1	-	R	-
Day in the week (begin day).					
hourFrom	-	an..4	-	R	-
Opening hours time first period from.					
hourTo	-	an..4	-	R	-
Opening hours time first period to.					
endDay	-	an..1	-	R	-
Day in the week (end day).					
secondHourFrom	-	an..4	-	O	-
Opening hours time second period from.					
secondHourTo	-	an..4	-	O	-
Opening hours time second period to.					

ReferenceData.CustomsOffice.CustomsOfficeTimetable.CustomsOfficeTimetableLine
.CustomsOfficeRoleTraffic

Customs Office - role and traffic competence.	-			1..99	-
role	-	an..3	-	R	-
The role of customs office.					
trafficType	-	an..3	-	R	-
Traffic type handled in customs office.					

ReferenceData.**CountryRegion**

Country - regions and holidays data.	-			0..*	-
countryCode	-	an..2	-	R	-
Country code.					
countryRegionCode	-	an..3	-	R	-
Region code.					
validFrom	-	date	-	O	-
validTo	-	date	-	O	-
Valid to.					

ReferenceData.CountryRegion.**Holiday**

Holiday data.	-			0..99	-
variable	-	boolean	-	R	-
Variable holiday flag.					
day	-	an..2	-	R	-
Day.					
month	-	an..2	-	R	-
Month.					
year	-	n..4,0	-	O	-
Year.					

ReferenceData.CountryRegion.Holiday.HolidayLSD

Holiday - Local Specific Data.	-			1..9	-
languageCode	-	an..2	-	R	-
Holiday - Local Specific Data - language code.					
name	-	an..35	-	R	-
Public holiday name.					

ReferenceData.CountryRegion.CountryRegionLSD

Country - region Local Specific Data.	-			1..9	-
languageCode	-	an..2	-	R	-
Country region Local Specific Data - language code.					
countryRegionName	-	an..35	-	R	-
Region name.					

ReferenceData.CountryHoliday

Country holiday data.	-			0..*	-
countryCode	-	an..2	-	R	-
Country code.					
validFrom	-	date	-	O	-
Valid from.					
validTo	-	date	-	O	-
Valid to.					

ReferenceData.CountryHoliday.Holiday

Holiday data.	-			1..99	-
variable	-	boolean	-	R	-
Variable holiday flag.					
day	-	an..2	-	R	-
Day.					

month	-	an..2	-	R	-
Month.					

year	-	n..4,0	-	O	-
Year.					

ReferenceData.CountryHoliday.Holiday.**HolidayLSD**

Holiday - Local Specific Data.	-			1..9	-
--------------------------------	---	--	--	------	---

languageCode	-	an..2	-	R	-
Holiday - Local Specific Data - language code.					

name	-	an..35	-	R	-
Public holiday name.					

ReferenceData.**Country**

Country codes.	-			0..*	-
----------------	---	--	--	------	---

countryCode	-	an..2	-	R	-
Country code.					

tccEntryDate	-	date	-	R	-
---------------------	---	------	---	---	---

nctsEntryDate	-	date	-	R	-
----------------------	---	------	---	---	---

geoNomenclatureCode	-	an..3	-	O	-
Geo-nomenclature code.					

countryRegimeCode	-	an..3	-	R	-
Country regime code.					

validFrom	-	date	-	O	-
------------------	---	------	---	---	---

validTo	-	date	-	O	-
Valid to.					

ReferenceData.Country.CodeDescription

This represents description of the code in particular language.	-			0..*	-
description	-	an..3072	-	R	-
Description.					
languageCode	-	an..2	-	R	-
Language code of description (see "12" reference data list).					

ReferenceData.DocumentType

Document types.	-			0..*	-
documentType	-	an..4	-	R	-
Document type code.					
transportDocument	-	boolean	-	R	-
Transport document flag.					
national	-	boolean	-	O	-
Nationally defined document flag (only value "true" is used, lack of this value means "false").					
validFrom	-	date	-	O	-
Valid from.					
validTo	-	date	-	O	-
Valid to.					

ReferenceData.DocumentType.CodeDescription

This represents description of the code in particular language.	-			0..*	-
description	-	an..3072	-	R	-
Description.					
languageCode	-	an..2	-	R	-
Language code of description (see "12" reference data list).					

ReferenceData.UN-LOCODE

United Nations Codes for Trade and Transport Locations.	-			0..*	-
countryCode	-	an..2	-	R	-
Country code.					
unLocodeId	-	an..3	-	R	-
Location code identifier.					
unLocodeName	-	an..29	-	R	-
Location name.					
validFrom	-	date	-	O	-
validTo	-	date	-	O	-
Valid to.					

ReferenceData.CommodityCode

Goods nomenclature commodity codes.	-			0..*	-
code	-	an10	-	R	-
Commodity code.					
indents	-	n..2,0	-	R	-
Number of indents.					
suffix	-	an2	-	R	-
Suffix of commodity code.					
import	-	boolean	-	R	-
Information whether commodity code can be used in import declarations (1 - yes, 0 - no).					
export	-	boolean	-	R	-
Information whether commodity code can be used in export declarations (1 - yes, 0 - no).					
excise	-	boolean	-	R	-
Excise object flag.					
validFrom	-	date	-	O	-
Valid from.					

validTo	-	date	-	O	-
Valid to.					

ReferenceData.CommodityCode.CodeDescription

This represents description of the code in particular language.	-			0..*	-
---	---	--	--	------	---

description	-	an..3072	-	R	-
Description.					

languageCode	-	an..2	-	R	-
Language code of description (see "12" reference data list).					

ReferenceData.Correlation

Correlation table - unified structure used for defining possible-combination of two codes from any reference data lists (e.g. requested and previous procedures).				0..*	-
---	--	--	--	------	---

code1	-	an..20	-	R	-
First code.					

code2	-	an..20	-	R	-
Second code.					

code3	-	an..20	-	O	-
Third code.					

code4	-	an..20	-	O	-
Fourth code.					

code5	-	an..20	-	O	-
Fifth code.					

code6	-	an..20	-	O	-
Sixth code.					

validFrom	-	date	-	O	-
Valid from.					

validTo	-	date	-	O	-
Valid to.					

ReferenceData.ExchangeRate

Exchange rate table.	-			0..*	-
currency	-	an..3	-	R	-
Currency code (see "18LT" reference data list);					
multiplier	-	n..6,0	-	O	-
Multiplier.					
rateLTL	-	n..9,4	-	O	-
LTL exchange rate.					
rateEUR	-	n..20,7	-	R	-
EUR exchange rate.					
validFrom	-	date	-	R	-
Valid from.					
validTo	-	date	-	O	-
Valid to.					

ReferenceData.Location

Location's data. Unified structure used for storing information- about locations like warehouses, authorized locations etc.				0..*	-
referenceNumber	-	an..17	-	R	-
Reference number.					
name	-	an..35	-	R	-
Name.					
streetAndNumber	-	an..35	-	R	-
Street name and number.					
postalCode	-	an..9	-	R	-
Postal code.					
city	-	an..35	-	R	-
City name.					
countryCode	-	an..2	-	R	-
Country code.					

validFrom	-	date	-	O	-
------------------	---	------	---	---	---

validTo	-	date	-	O	-
----------------	---	------	---	---	---

Valid to.

ReferenceData.ValueAdjustement

Value adjustment codes definitions.	-			0..*	-
-------------------------------------	---	--	--	------	---

code	-	an..4	-	R	-
-------------	---	-------	---	---	---

Value adjustment code

additionalCosts	-	an..1	-	R	-
------------------------	---	-------	---	---	---

Additional costs (1 - plus, 0 - minus)

validFrom	-	date	-	O	-
------------------	---	------	---	---	---

Valid from.

validTo	-	date	-	O	-
----------------	---	------	---	---	---

Valid to.

ReferenceData.ValueAdjustement.CodeDescription

This represents description of the code in particular language.	-			0..*	-
---	---	--	--	------	---

description	-	an..3072	-	R	-
--------------------	---	----------	---	---	---

Description.

languageCode	-	an..2	-	R	-
---------------------	---	-------	---	---	---

Language code of description (see "12" reference data list).

ReferenceData.ValueAdjustement.Apportionment

Apportionment data.	-			1..*	-
---------------------	---	--	--	------	---

apportionmentMode	-	an..1	-	R	-
--------------------------	---	-------	---	---	---

Apportionment mode. Possible values: 1 - apportionment by value, 2 - apportionment by weight, 3 - apportionment not applicable.

apportionment	-	an..1	-	R	-
----------------------	---	-------	---	---	---

Apportionment. Possible values: 1 - by value, 2 - by weight, 3 - apportionment not applicable.

4.2 Rules

4.3 Code lists

225NOV Declaration rejection codes

Item type: SimpleItem

Item code type: an..3

241NOV House consignment states

Item type: SimpleItem

Item code type: an2

242NOV Entry Summary Declaration states

Item type: SimpleItem

Item code type: an..20

243NOV Sources of errors/warnings in PPMKS system

Item type: SimpleItem

Item code type: an..20

276COM Qualifier of identification

Item type: SimpleItem

Item code type: an1

302COM Type of locations

Item type: SimpleItem

Item code type: an1

328COM Control result codes

Item type: SimpleItem

Item code type: an..2

329COM Transport document type

Item type: SimpleItem

Item code type: an4

351COM Document type

Item type: DocumentType

Item code type: an4

352COM Package type

Item type: SimpleItem

Item code type: an2

353COM UNLOCODE

Item type: UN-LOCODE

Item code type: an..17

354COM Mode of transport at the border

Item type: SimpleItem

Item code type: an1

356COM Country code (ISO 3166)

Item type: Country

Item code type: an2

357COM AEO Status

Item type: SimpleItem

Item code type: an1

358COM Additional information code

Item type: SimpleItem

Item code type: an5

360COM Additional Supply Chain Actor Role Code

Item type: SimpleItem

Item code type: an..3

361COM Communication means type

Item type: SimpleItem

Item code type: an..3

363COM Container packed status

Item type: SimpleItem

Item code type: an..3

364COM Container size and type identification

Item type: SimpleItem

Item code type: an..10

365COM Container supplier type code

Item type: SimpleItem

Item code type: an..3

372COM Specific circumstance indicator

Item type: SimpleItem

Item code type: an3

374COM Type of person

Item type: SimpleItem

Item code type: an1

387COM Type of Identification of Means of Transport

Item type: SimpleItem

Item code type: an2

388COM Type of Goods

Item type: SimpleItem

Item code type: an..3

390COM Representative Status

Item type: SimpleItem

Item code type: an1

393COM Currency code

Item type: SimpleItem

Item code type: an3

395COM Transport charges method of payment

Item type: SimpleItem

Item code type: an1

396COM Type of Means of Transport

Item type: SimpleItem

Item code type: an..4

398COM Additional fiscal references role

Item type: SimpleItem

Item code type: an3

400COM United Nations Dangerous Goods code

Item type: SimpleItem

Item code type: an4

45LT_49 Functional error/warning types (national)

Item type: SimpleItem

Item code type: an..20

56 Roles of Lithuanian customs offices

Item type: SimpleItem

Item code type: an3

94LT Relationship between Lithuanian customs offices and their roles

Item type: Correlation

Item code type:

COL Customs offices

Item type: CustomsOffice

Item code type: an8