Message Implementation Guide

PackagingInformation Message

April 13th, 2016

Auteur	Jaco Oskam
Tel	06-27068307
Rapport	**Message Implementation Guide** PackagingInformation Message
Versie	2.1

Version control

Version	Date	Author	Description
0.1	23-2-2015	Jaco Oskam	Initial version
0.9	11-3-2015	Jaco Oskam	Rework after internal review
0.91	4-5-2015	Jaco Oskam	Removed IssueID from interface
0.92	22-5-2015	Jaco Oskam	Updated description of master data
1.0	29-5-2015	Jaco Oskam	1.0 version without changes
1.1	3-11-2015	Manon Havinga	Explanation XSD versus MIG added Decimal separator added
1.2	1-2-2016	Manon Havinga	Message structure error codes added Functional error codes added
2.0	21-3-2016	Manon Havinga	• Textual adjustments to align all MIGs • Message structure code E012 combined with E009 • Packaging labels will be presented in a ZIP file. The ZIP file will contain one or more PDF files.
2.1	13-4-2016	Manon Havinga	• Improved applicability: In the event that a value is supplied for an element which is “N/A”, than the value will be ignored instead of generating an error message. • Updated diagrams in paragraph 2.1 • Added status InProgress

Distribution

Who	Function
Ben Damhuis	Business Product Owner
Antoine Houben	Project Manager
Manon Havinga	Information Analyst
Peter Por	Architect
Roeland Kok	Architect
Wiebo Boer	Application Consultant
Bob Sotthewes	Test Manager

This document is based on MIG Template version 1.0, March 1st, 2013

Table of Contents

1 Introduction

This document provides the definition of the PostNL PackagingInformation message to be used in the electronic data conversation with the PostNL MijnPost B2B system. This Message Implementation Guidelines document is based on the PackagingInformation.xsd.

The associated XSD with this MIG contains more elements/fields than described in this MIG. This is not an error but caused by the fact that we use standard elements which (can) contain more sub elements and/or fields than strictly necessary for this specific service. The reason for the use of a generic XSD is that future enhancements and changes can be easily implemented.

The associated XSD with this MIG contains elements/fields that will not be used. However it is required that all elements/fields described in this MIG are implemented in the messages exchanged between the different parties.

The current definition of the message, as described in this guideline, reflects its use in PostNL MijnPost. It does not preclude the use of this message between other organisational units or parties than those indicated in this guideline. The criteria for the use of the message should be its functionality rather than the organisational units or parties involved.

1.1 Functional definition

The B2B module handles several messages including the PackagingInformation message.
After an address file, containing delivery addresses of a sorted or unsorted bulk mail consignment, has been validated or validated and sorted, all packaging information can be retrieved using the PackagingInformation message. Packaging information consists of:

- List of all bundles
- List of equipment
- Equipment labels
- A report containing validation (and sorting) result and details

1.2 Principles

The PackagingInformation message is exchanged to retrieve packaging information related to a specific issue and address file validation (and sort) action.

1.3 Field of application

The PackagingInformation message is used by the system of a customer to retrieve packaging information that can be used to print labels and to consult a summary of the packaging into bundles and roller cages or pallets. The message is communicated from customer systems to the MijnPost B2B module. See paragraph 2.1 for details about the message flow.

1.4 PostNL canonical model

The structure of the PackagingInformation message is based on the definition of terms and conditions as agreed by PostNL. The definition of this messages is based on existing canonical messages.

The PackagingInformation message consists always of two components:

- A verb part, described in chapter 4 ‘Packaging Information processing’
- A noun part, described in chapter 5 ‘Information Model for Packaging Information’

1.5 Conventions

The following conventions apply in this document.

Character type:

a	:	string with alphabetic characters
an	:	string with alphabetic and/or numeric characters
n		string with numeric characters
i	:	integer
d	:	decimal with decimal point as separator
boolean	:	“true” or “false”
dt	:	date time
guid	:	global unique identifier

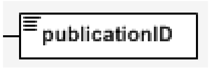
Size:

Fixed	:	all positions must be used
Variable	:	positions may be used up to a specified maximum

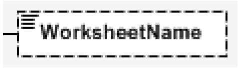
Examples:

a3	:	3 alphabetic characters, fixed length
n3	:	3 numeric characters, fixed length
d4,3	:	decimal, fixed length with 4 numeric characters left of the decimal separator and 3 numeric characters right of the decimal separator.
a..3	:	up to 3 alphabetic characters
n..3	:	up to 3 numeric characters
dt	:	2011-01-20T13:35:32Z
guid	:	55f1ebee-5d37-41be-b260-b07b1929a1f7

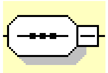
The following conventions apply when parts of the message are presented as an XSD model.



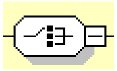
Box with a full line is a mandatory message element



Box with a dotted line is an optional message element



The component can contain a number of mandatory and optional elements. The elements must appear in the sequence mentioned



The component contains a number of elements. Only one of the possible elements may be present (choice).

Each message element is elaborated in this document using the format below.

Action	Description
Element	Name of the element
Description	Description of the element
Type and size	Type and size of the element. See the text above for the conventions
Formatting	Format of the element. For example a year has the format “YYYY”
Default value	Standard value of the element
Applicability	A meaning indicates how the message is handled by the MijnPost B2B module. For example the meaning “Show”. For each meaning is described whether the element is: • Mandatory: The element must have an allowed value. If a correct value is supplied, then MijnPost B2B module will check and register the supplied value. If no value or an incorrect value is supplied, then MijnPost B2B module will generate an error message. • Optional: The element doesn’t have to have a value. If a value is supplied, then MijnPost B2B module will check and register the supplied value. If no value is supplied, then MijnPost B2B module doesn’t perform any action. • Not applicable (N/A): The element shouldn’t have a value. If a value is supplied, then MijnPost B2B module will ignore the supplied value and this value is not returned in the response. If no value is supplied, then MijnPost B2B module doesn’t perform any action.
Dependence requirements	Explanation of dependencies between this element and other elements

1.6 References

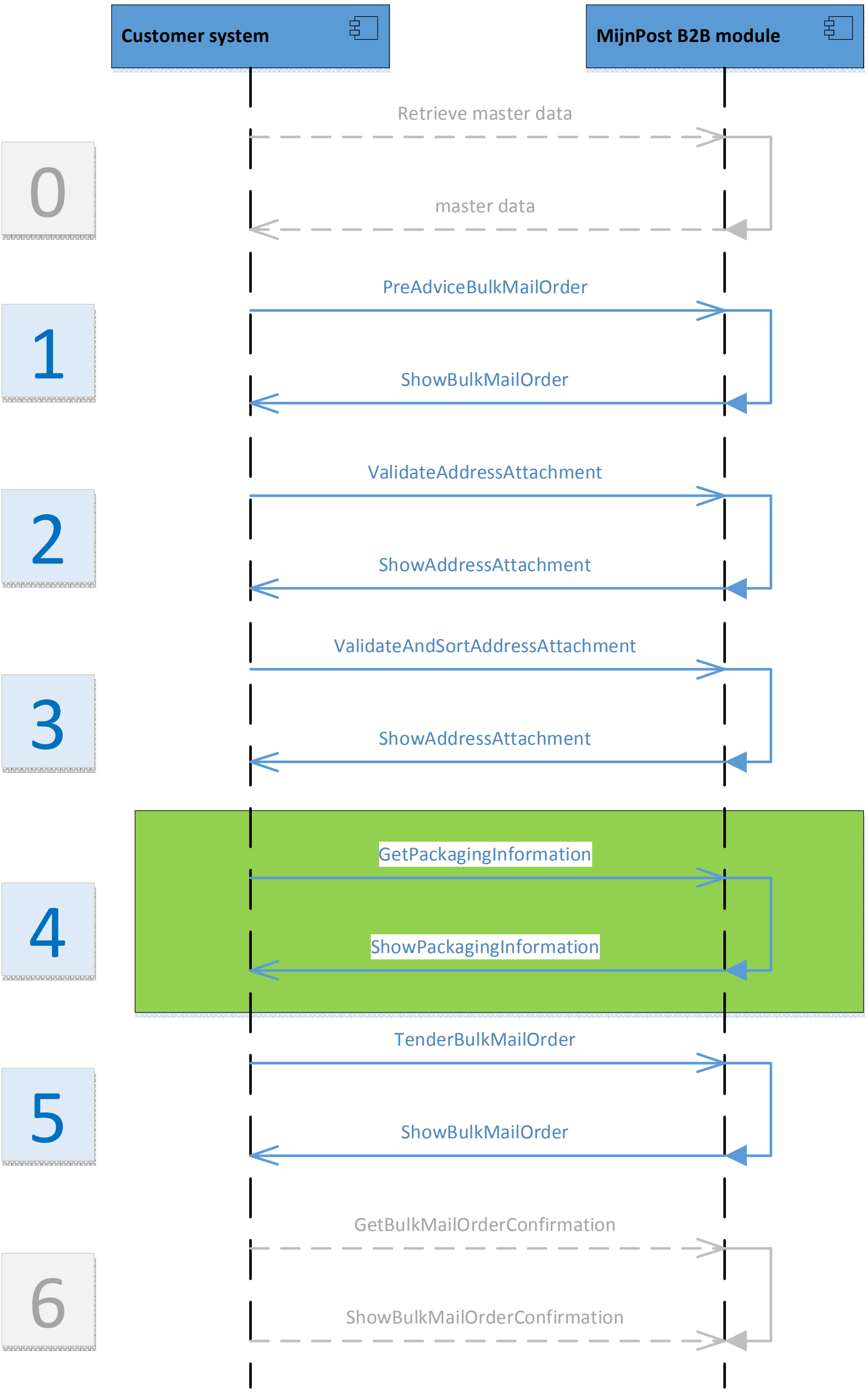
Reference	Details
[ref1]	MIG BulkMailOrder message MIG BulkMailOrder message Jaco Oskam Version 2.9 May 22nd, 2015
[ref2]	MIG AddressAttachment message Jaco Oskam Version 0.93 May 22nd, 2015
[ref3]	MIG OrderConfirmation message <TBD>

2 Interaction specification

This chapter describes the interactions with the MijnPost B2B module using an interaction diagram. In addition, technical aspects like protocol and channel are described.

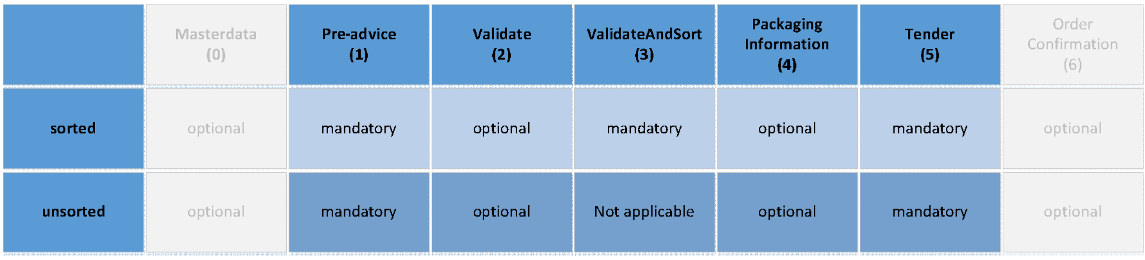
2.1 Interaction diagram

The next diagram shows the functional flows between customer systems and the MijnPost B2B module. Only the communications within the green box are in scope of this MIG.



2 Interaction specification

A typical customer flow for creating and tendering an ordered bulk mail order consists of 6 steps. When creating and tendering a bulk mail order for an unordered product, specific steps are optional. The next figure indicates which steps are mandatory and optional for both sorted and unsorted products.



	Masterdata (0)	Pre-advice (1)	Validate (2)	ValidateAndSort (3)	Packaging Information (4)	Tender (5)	Order Confirmation (6)
sorted	optional	mandatory	optional	mandatory	optional	mandatory	optional
unsorted	optional	mandatory	optional	Not applicable	optional	mandatory	optional

Before a bulk mail order is created the applicable master data like geo data and location data can be downloaded from the MijnPost web portal. The B2B webservice does not offer any master data since the relevant master data is all relatively static and the same for all customers. When, in the future, more dynamic master data becomes relevant, the B2B webservice might be updated to supply these data.

Step 1 is to pre-advice the bulk mail order. Within that step a consignment is created. Multiple consignments can be pre advised within a single invocation. As a result, unique ConsignmentId(s) and unique IssueId(s) are returned of which ConsignmentId(s) must be used for further processing. The PreAdviceBulkMailOrder method can be invoked multiple times. When an existing ConsignmentId is used, an existing consignment gets updated. Otherwise a new consignment is created. See [ref1] for details about the *BulkMailOrder* message.

Bulk mail orders are sent to many addressee. When a customer has an address file containing these addressee, the addresses within this file can be validated in step 2. Validation is mandatory for sorted products and optional for unsorted products. Validation can be executed separately or in combination with sorting (step 3). When ConsignmentId is present, addresses can be validated. Validation includes categorizing addresses in domestic, foreign and unrecognized. Validation and sorting is only supported on a single address file at a time.

After validating addresses, the addresses can be sorted. This applies to bulk mail orders with sorted products only. During sorting in step 3, each individual address is also enriched with a sorting code. See [ref2] for details about the *AddressAttachment* message.

Bulk mail orders are packaged into bundles and roller cages or pallets. Based on the sorting result, a set of packaging forms (that can be printed to label the packaging materials) can be retrieved in step 4. This step requires the unique ConsignmentID that was returned in step 1 and is mandatory for sorted products and optional for unsorted products. The packaging forms can be retrieved multiple times (in case the forms get misplaced for example).

It is important to notice that the B2B module is realized as a webservice so all interactions exist of request and responses. The customer system is always the requesting party and the B2B module replies. This is summarized in the next table for the packaging functionality.

Request	Response
GetPackagingInformation	ShowPackagingInformation

The bulk mail order is tendered in step 5. ConsignmentId is mandatory. Bulk mail orders can only be tendered after pre advising the bulk mail order first. Bulk mail orders with a sorted product can only be tendered after validating and sorting the related address file.
The result of a tender is a customer order number and a P1700 form that must be used when physically tendering the bulk mail order at a PostNL location. See [ref1] for details about the *BulkMailOrder* message.

After physically tendering the bulk mail order a confirmation of the order can be retrieved in step 6. This is the final step of the functional flow. Getting the order confirmation is optional. See [ref3] for details about the *OrderConfirmation* message.
This message hasn't been implemented yet. The B2B webservice will be updated in future to supply also these data.

The remainder of this document only describes the highlighted part of the interaction diagram.

2.2 Channel, protocol and security

The MijnPost B2B module offers its functionality as a SOAP webservice. It is consumed over an HTTPS connection. The calling party (customer system) identifies itself using a client certificate.

3 Operational interface management

3.1 Frequency and volume

Bulk mail orders are created throughout the day (24*7 support) and packaging information can also be retrieved throughout the day.

3.2 Validation and processing

Before the message is processed it will be validated by the receiving party (MijnPost B2B module) on multiple levels:

1. Structure and field definitions
2. Contents and syntax
3. Meaning and semantics

Result of validation is either OK or not OK.
When an error is detected, the complete message is rejected. The sending party must repair the message based on the supplied error code(s) and resend the message.

3.3 Scheduling

Not applicable.

3.4 Interface monitoring

Monitoring takes place on a daily basis and involves:

- Tracking the total number of received messages
- Detect abnormalities
- Investigate abnormalities
- Verify that the webservice is up and running

3.5 Restart and recovery procedures

This will be according to the standard system maintenance of MijnPost.NET modules.

4 Packaging Information processing

The PackagingInformation message supports a single meaning: Get.
A meaning indicates how the PackagingInformation message is handled by the MijnPost B2B module.

4.1 Getting Packaging Information

Packaging information can only be retrieved after an address file has been validated or validated and sorted. For issues with sorted products, validation and sorting is mandatory. For issues with unsorted products, validation is optional and sorting is not possible.
When Packaging Information is retrieved for an issue with a sorted product an error is returned when the related address file has not yet been validated and sorted.

For issues based on a sorted product, the packaging information consists of the following items:

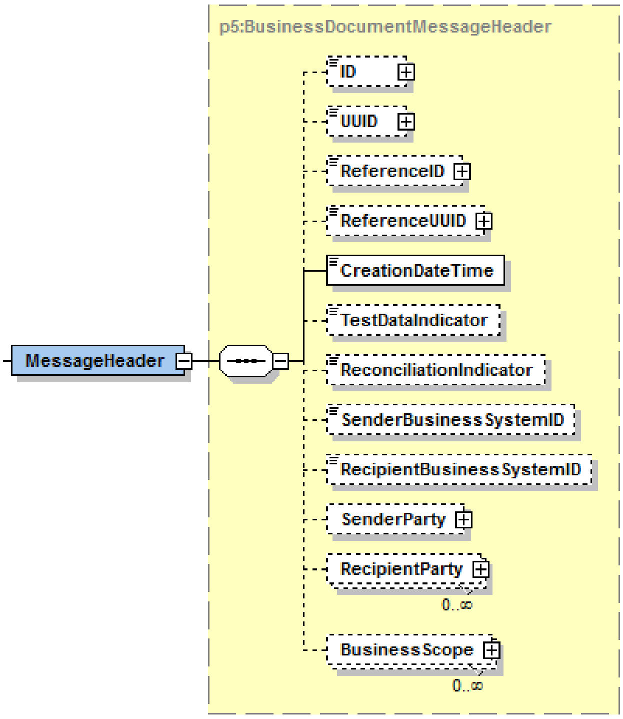
- A text file that contains a list of all bundles and details for each bundle.
- A text file that contains a list of all roller cages or pallets and details for each roller cage or pallet.
- A PDF file that contains the labels for the roller cages or pallets. Each page in the PDF file contains a single label.
- A text file that contains a summary of the consignment and statistics of the address validation and sorting.

For issues based on an unsorted product, the packaging information consists of only a PDF file that contains a single page that can be used as a label for all equipment types.

The scope of the retrieval of packaging information is limited to a single issue and the validation (and sorting) action on a related address file.

4.2 Common elements for the different PackagingInformation meanings

The meanings described above share some common elements which are required in the header of the message. Only the relevant items within the header are described in the next table. Items that are not listed are not applicable, will be ignored and do not have to be present in the message.



Action	Description
Element	CreationDateTime
Description	Timestamp when the message was created.
Type and size	dt
Formatting	[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}(.[0-9]{1,7})?Z
Default value	N/A
Applicability	
GetPackagingInformation	Mandatory
Show	Mandatory
Dependence requirements	None

Action	Description
Element	TestDataIndicator
Description	Indicates whether or not this message should be - fully processed (false) - only checked (true)
Type and size	boolean
Formatting	true or false

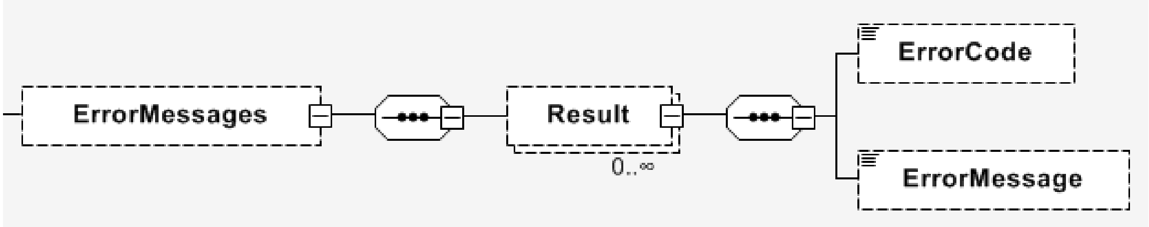
4 Packaging Information processing

Action	Description
Default value	false
Applicability	
GetPackagingInformation	Optional
Show	Optional
Dependence requirements	When the TestDataIndicator is true, no actual result is returned. Only the structure of the messages is checked including the check on mandatory message elements and message element types and values.

This default header contents is extended with an error messages section. Within this section all error messages regarding the structure of the message will be logged. A list of these messages can be found in section 6.1.
Functional error messages regarding the state of the PackagingInformation object itself are part of the message body described in chapter 5.

4.2.1 ErrorMessages

The ErrorMessages structure depicted below is added to the header of the message.



Both ErrorMessages and Result are not data elements themselves but components that hold data elements. The ErrorMessages component can hold zero to many Result components. Each Result component holds one specific ErrorCode element and one ErrorMessage element.

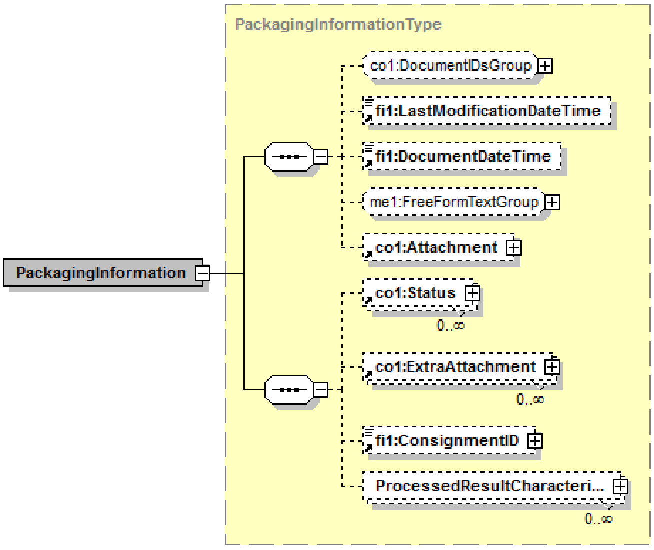
Action	Description
Element	ErrorCode
Description	A code that uniquely identifies the error that occurred when packaging information was retrieved for a specific issue. A specific error code is returned when no error occurred. See chapter 6 for all supported error codes.
Type and size	a1 or an3
Formatting	Success: S Error: Enn, for example “E09”

Action	Description
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Mandatory
Dependence requirements	ErrorCode and ErrorMessage relate to each other.

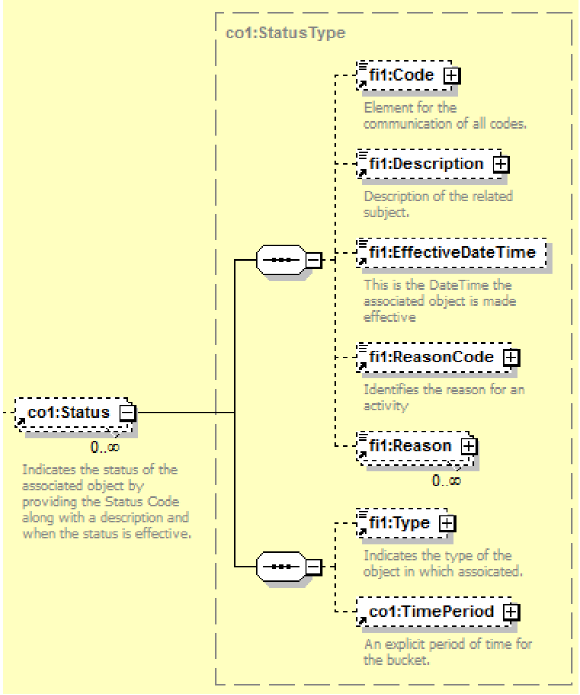
Action	Description
Element	ErrorMessage
Description	A text message that explains the error that occurred.
Type and size	a
Formatting	Text “Success” when no error occurred. See section 6.1 for all valid error messages.
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Mandatory
Dependence requirements	ErrorCode and ErrorMessage relate to each other.

5 Information model for Packaging Information

All relevant items within the message body are described in this chapter. Items that are not listed in this chapter but are present in the message structure (XSD) are not applicable for this appliance, will be ignored, and do not have to be present in the message.



5.1 Status


co1:StatusType
fi1:Code
Element for the communication of all codes.
fi1:Description
Description of the related subject.
fi1:EffectiveDateTime
This is the DateTime the associated object is made effective
fi1:ReasonCode
Identifies the reason for an activity
fi1:Reason
0..∞
fi1:Type
Indicates the type of the object in which assoicated.
co1:TimePeriod
An explicit period of time for the bucket.
co1:Status
0..∞
Indicates the status of the associated object by providing the Status Code along with a description and when the status is effective.

The Status section is used to supply status information about the PackagingInformation object. Two statuses can be distinguished:

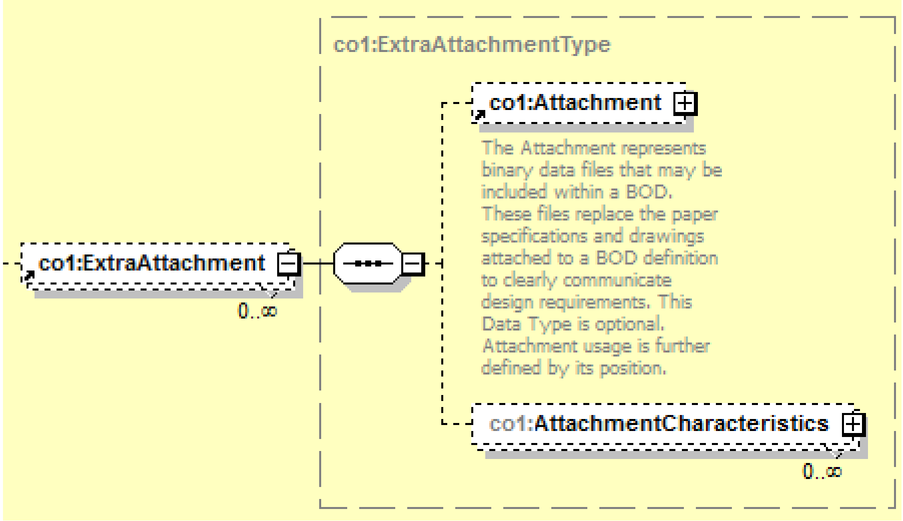
- **Success**, action completed without errors.
- **InProgress,** action is currently being executed.
- **Error** when an error occurred during message processing.

Action	Description
Element	Code
Description	A code that uniquely identifies success, in progress or error relating to the current action that is executed on a PackagingInformation object. See section 6.2 for all supported functional error codes.
Type and size	Success: a1 InProgress: a1 Error: an3
Formatting	Success: “S” Error: <Enn>, for example “E09”
Default value	N/A
Applicability	
GetPackagingInformation	N/A

Action	Description
Show	Mandatory
Dependence requirements	Code and Description relate to each other.

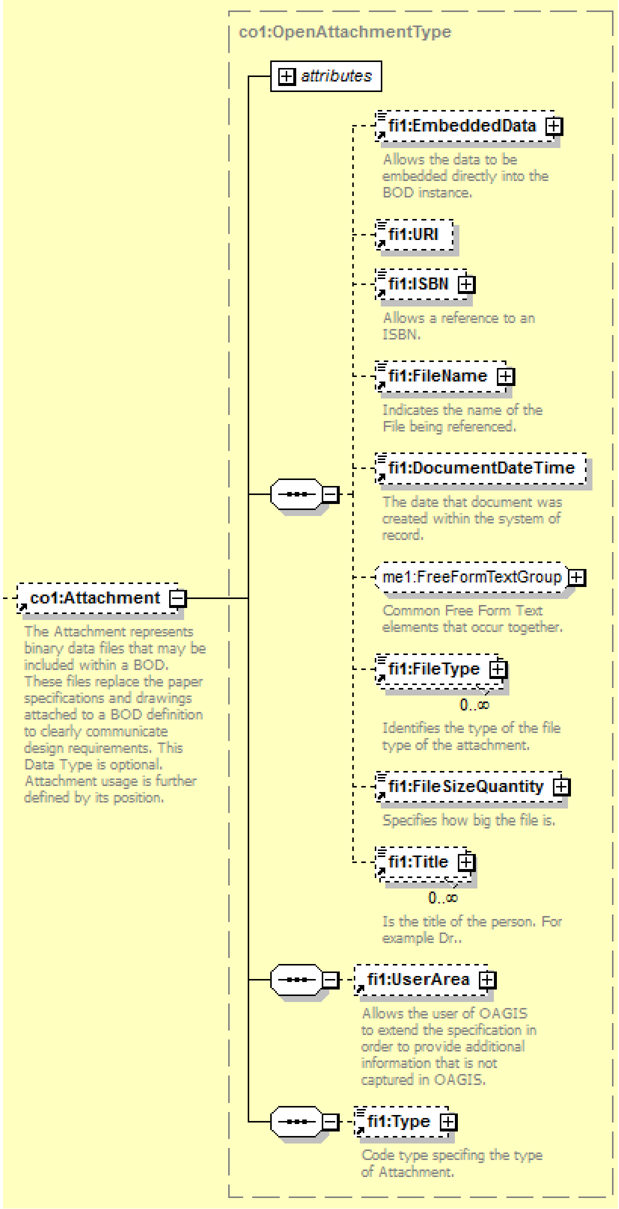
Action	Description
Element	Description
Description	A text message that explains the Code.
Type and size	a
Formatting	Success: “Success”. Error: a text explaining the error. See section 6.2 for all supported error messages.
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Mandatory (success, inprogress, error)
Dependence requirements	Code and Description relate to each other.

5.2 ExtraAttachment


co1:ExtraAttachmentType
co1:ExtraAttachment
0..∞
co1:Attachment
The Attachment represents binary data files that may be included within a BOD. These files replace the paper specifications and drawings attached to a BOD definition to clearly communicate design requirements. This Data Type is optional. Attachment usage is further defined by its position.
co1:AttachmentCharacteristics
0..∞

The packaging information files are placed in the Attachment component that is part of the ExtraAttachment component.

5.2.1 Attachment


co1:OpenAttachmentType
attributes
fi1:EmbeddedData
Allows the data to be embedded directly into the BOD instance.
fi1:URI
fi1:ISBN
Allows a reference to an ISBN.
fi1:FileName
Indicates the name of the File being referenced.
fi1:DocumentDateTime
The date that document was created within the system of record.
me1:FreeFormTextGroup
Common Free Form Text elements that occur together.
fi1:FileType
0..∞
Identifies the type of the file type of the attachment.
fi1:FileSizeQuantity
Specifies how big the file is.
fi1:Title
0..∞
Is the title of the person. For example Dr..
fi1:UserArea
Allows the user of OAGIS to extend the specification in order to provide additional information that is not captured in OAGIS.
fi1:Type
Code type specifing the type of Attachment.
co1:Attachment
The Attachment represents binary data files that may be included within a BOD. These files replace the paper specifications and drawings attached to a BOD definition to clearly communicate design requirements. This Data Type is optional. Attachment usage is further defined by its position.

Action	Description
Element	EmbeddedData
Description	A ZIP archive that contains the set of files that contain the packaging information.
Type and size	an
Formatting	base64 encoded file

Action	Description
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Depends on Status (section 5.1). Mandatory when Status equals *Success*. Not present when Status equals *Error*.
Dependence requirements	In case of an issue based on a sorted product, the ZIP archive that is returned contains: 1. A text file that contains a list of all bundles. Filename: "<input filename> - <DD-MM-YYYY HH.MM> - Bundellijst.txt". 2. A ZIP file that contains all packaging labels. Filename: "<input filename> - <DD-MM-YYYY HH.MM> - EmballageKaarten.zip". The ZIP file contains one or several PDF files. To accelerate performance an additional PDF file is generated each time the number of roller cages or pallets exceeds a specific maximum number. E.g. the maximum number of packaging labels is set to 200 per PDF file. So in case of a consignment with 300 pallets there will be two separate PDF files. Filename: "<input filename> - <DD-MM-YYYY HH.MM> - EmballageKaarten (<Transportequipment number from> - <Transportequipment number to>).pdf". 3. A text file that contains a report of the sorting action. Filename: "<input filename> - <DD-MM-YYYY HH.MM> - Rapportage.txt". 4. A text file that contains a list of all roller cages or pallets. Filename when roller cages are used: "<input filename> - <DD-MM-YYYY HH.MM> - Rolcontainerlijst.txt". Filename when pallets are used: "<input filename> - <DD-MM-YYYY HH.MM> - Palletlijst.txt". In case of an issue based on an unsorted product, the ZIP archive that is returned contains:

Action	Description
	1. A PDF file that contains a single page that represents a fixed packaging label that can be used on all equipment. Filename: “<consignmentID>.pdf”.

Action	Description
Element	FileName
Description	Name of the ZIP archive.
Type and size	a
Formatting	“<input filename> - <DD-MM-YYYY HH.MM>.zip”
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Optional. Mandatory when a file is returned.
Dependence requirements	None

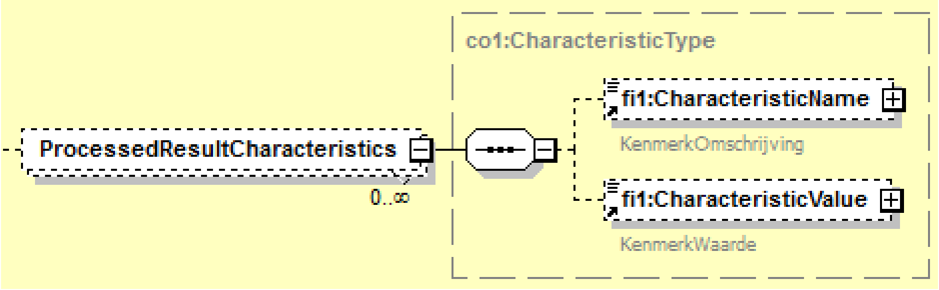
Action	Description
Element	FileType
Description	Output file type is always a ZIP archive.
Type and size	a3
Formatting	“ZIP”
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Optional. Mandatory when a file is returned.
Dependence requirements	None

5.3 ConsignmentID

Action	Description
Element	ConsignmentID
Description	A unique identification of a consignment for which addresses have been validated or validated and sorted and for which packaging information is retrieved. This ConsignmentID is returned without modification.
Type and size	n12
Formatting	N/A

Action	Description
Default value	N/A
Applicability	
GetPackagingInformation	Mandatory
Show	Mandatory
Dependence requirements	None

5.4 ProcessedResultCharacteristics



Characteristics defined in this section of the PackagingInformation message acts as output parameters for the result of a packaging information retrieval action. The data elements that are returned this way (as metadata) are also present in the files inside the attached zip archive. Each characteristic is described by its name and a value. The next table lists all supported characteristics for the result.

Characteristic name	Action	Description
NumberOfTransportEquipmentUnits	ShowPackagingInformation	Number of roller cages or pallets that are used.
NumberOfBundles	ShowPackagingInformation	Number of bundles that are used.
MinimumNumberInABundle	ShowPackagingInformation	Actual minimum value that was used during the sorting process.
MaximumNumberInABundle	ShowPackagingInformation	Actual maximum value that was used during the sorting process.
MaximumWeigthTransportEquipment	ShowPackagingInformation	Actual maximum value that was used during the sorting process.

For each characteristic, the characteristic name and value must be supplied. All characteristic names must be supplied as described in the next table.

Action	Description
Element	CharacteristicName
Description	Name of the characteristic that relates to the attachment.
Type and size	a
Formatting	One of the values listed in the table above must be used.
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Mandatory
Dependence requirements	None

Characteristic names are always strings without further restrictions where characteristic values can have different restrictions. All characteristic values are described in the next tables.

Action	Description (CharacteristicValue)
Element	NumberOfTransportEquipmentUnits
Description	Number of roller cages or pallets that are used.
Type and size	i
Formatting	N/A
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Mandatory for issues based on a sorted product. Not applicable for issues based on an unsorted product.
Dependence requirements	

Action	Description (CharacteristicValue)
Element	NumberOfBundles
Description	Number of bundles that are used.
Type and size	i
Formatting	N/A
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Mandatory for issues based on a sorted product. Not applicable for issues based on an unsorted product.

Action	Description (CharacteristicValue)
Dependence requirements	

Action	Description (CharacteristicValue)
Element	MinimumNumberInABundle
Description	Minimum number of mail piece in a single bundle.
Type and size	i
Formatting	N/A
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Mandatory for issues based on a sorted product. Not applicable for issues based on an unsorted product.
Dependence requirements	This element is the minimum value that was actually used during the sorting process and may differ from the value provided when invoking the validating and sorting action.

Action	Description (CharacteristicValue)
Element	MaximumNumberInABundle
Description	Maximum number of mail piece in a single bundle.
Type and size	i
Formatting	N/A
Default value	N/A
Applicability	
GetPackagingInformation	N/A
Show	Mandatory for issues based on a sorted product. Not applicable for issues based on an unsorted product.
Dependence requirements	This element is the maximum value that was actually used during the sorting process and may differ from the value provided when invoking the validating and sorting action.

Action	Description (CharacteristicValue)
Element	MaximumWeigthTransportEquipment
Description	Maximum net weight of a single transport equipment unit in kg.
Type and size	i
Formatting	N/A
Default value	N/A
Applicability	
GetPackagingInformation	N/A

Action	Description (CharacteristicValue)
Show	Mandatory for issues based on a sorted product. Not applicable for issues based on an unsorted product.
Dependence requirements	This element is the maximum value that was actually used during the sorting process and may differ from the value provided when invoking the validating and sorting action.

6 Error handling

Remark: The list of error codes is not yet complete and will be extended when the functionality of the B2B module is realized.

There are 2 components reserved for holding status and error codes and descriptions as described in sections 4.2 and 5.1. The header part of the message (section 4.2) contains error codes and error messages regarding errors on message (structure) level. These are described in section 6.1.

The body part of the message (section 5.1) contains status codes and status messages regarding:

- State of the address attachment object. An example is the progress state when validation or sorting is being executed.
- Functional errors that occurred while validating the content of the message. An example of a functional error is when a non-existing ConsignmentId is provided.

These errors are described in section 6.2.

When an AddressAttachment message could be processed correctly the specific success code and status code "0" with message "Success" are returned.
When an error is encountered, a specific error code and error message are returned.

6.1 Success

Error code 0 and message "Success" is returned.

Code	Message
0	Success

6.2 Message structure error codes and messages

The range "E0001 – E0999" is reserved for message structure errors.

Code	Message
E0001	The element '{0}' cannot be null
E0002	The parameter '{0}' cannot be null
E0003	The parameter '{0}' does not contain elements
E0004	The value of parameter '{0}' cannot be null
E0005	Exactly one '{0}' should be specified
E0006	Calling party is not authorized to use B2B-service.
E0007	The user is not authorized or does not exist. Verify the provided Id in '{0}'
E0008	A unknown error has occurred
E0009	A backend system was not available to process your request. Please try again. Keep an eye on the website of MijnPost in case it is a general failure of the PostNL systems.
E0010	User validation failed. Please try again.

Code	Message
E0011	Consignment retrieving failed. Please try again.

6.1 Validation and functional error codes and messages

The range “E3000 – E3999” is reserved for validation and functional errors.

Code	Message
E3000	Invalid or non-existing ConsignmentId was used: {0}.
E3001	No packaging information found. An address file has not yet been offered for validation.
E3002	No packaging information found. An address file has not yet been offered for validation and sorting.
E3003	No packaging information found. Packaging forms have expired and have been deleted.