bpost business

Data Exchange

Technical Guide

January 2023

bpost business

Table of Contents

PART I: General Information

1. Introduction

Why this guide?

Some products offered by bpost need exchanges of data via electronic file between bpost and the customer. The goal of this guide is to give all relevant information to allow the implementation of these exchanges.

The different sorts of products covered by this guide are listed and briefly described below.

Deposit

It is possible to announce MassPost deposits via electronic files, and so to receive electronically the authorization of deposit.

This automated process is available for most of the MassPost products (Mail ID, Round & Sequence, and non-Mail ID products).

MAIL ID deposit

With a MAIL ID deposit, a barcode is printed on each mail piece to uniquely identify it. It allows bpost to efficiently sort these mail pieces, on the basis of these barcodes and the linked data sent electronically.

Prior to the deposit of the physical mail pieces, the customer must send electronically to bpost a mailing file that holds all the MAIL ID barcode identifiers for a given deposit and the corresponding delivery addresses. There is a possibility for the customer to ask bpost to generate the MAIL ID barcode identifiers.

Round & Sequence deposit

Round & Sequence is used only for Large format.

A Round & Sequence deposit is similar to a Mail ID deposit in the sense that address data will be sent electronically by the customer to bpost. However, for Round & Sequence deposit, a sequence reference will be printed on the envelope in addition to the barcode. This sequence information is provided by bpost in the response file, and will allow the customer to pre-sort the deposit following these sequence references.

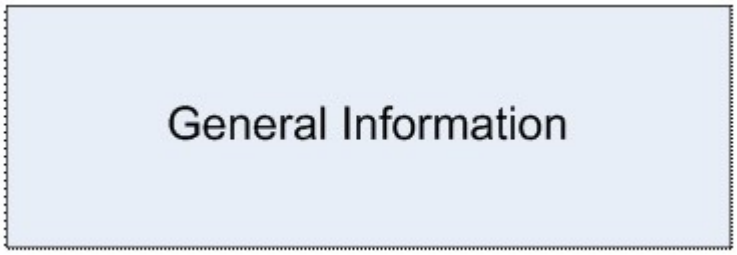
bpost business

OptiAddress

OptiAddress is a tool used for address validation. Authorized mailers can submit addresses to the MAIL ID system for verification, independently of any deposit, and bpost will send back a useful feedback on these addresses, with potential corrections.

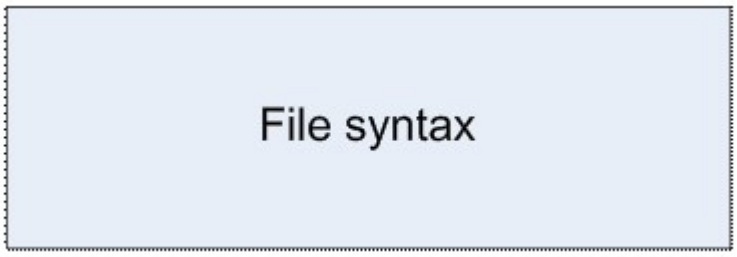
What is in this guide?

This guide gives an overview of the different options for exchanging data with bpost. The guide is divided in 4 parts. Part 1 provides general information independent of the product for which the data exchange occurs. In part 2, a more detailed product overview is given with information product by product. The third part contains the file syntax details. This guide concludes with an annexe part that gives additional information.

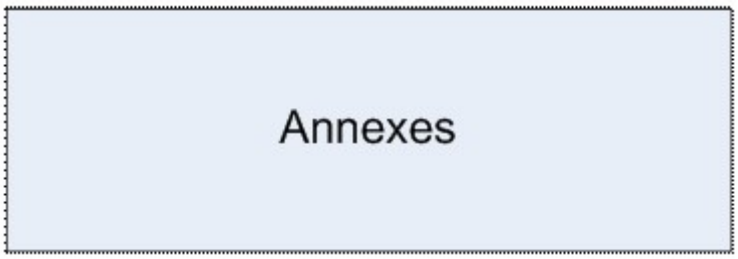


Information that is applicable to all customers wanting to implement data exchange with bpost.

Overview of the different products that may require data exchange. For each product we refer to the relevant File Syntax sections.



File syntax for the different files that are subject to data exchange, i.e. Deposit Files and Mailing Files.



Specific (additional) information provided to illustrate, clarify or debug.

Each part contains a number of chapters.

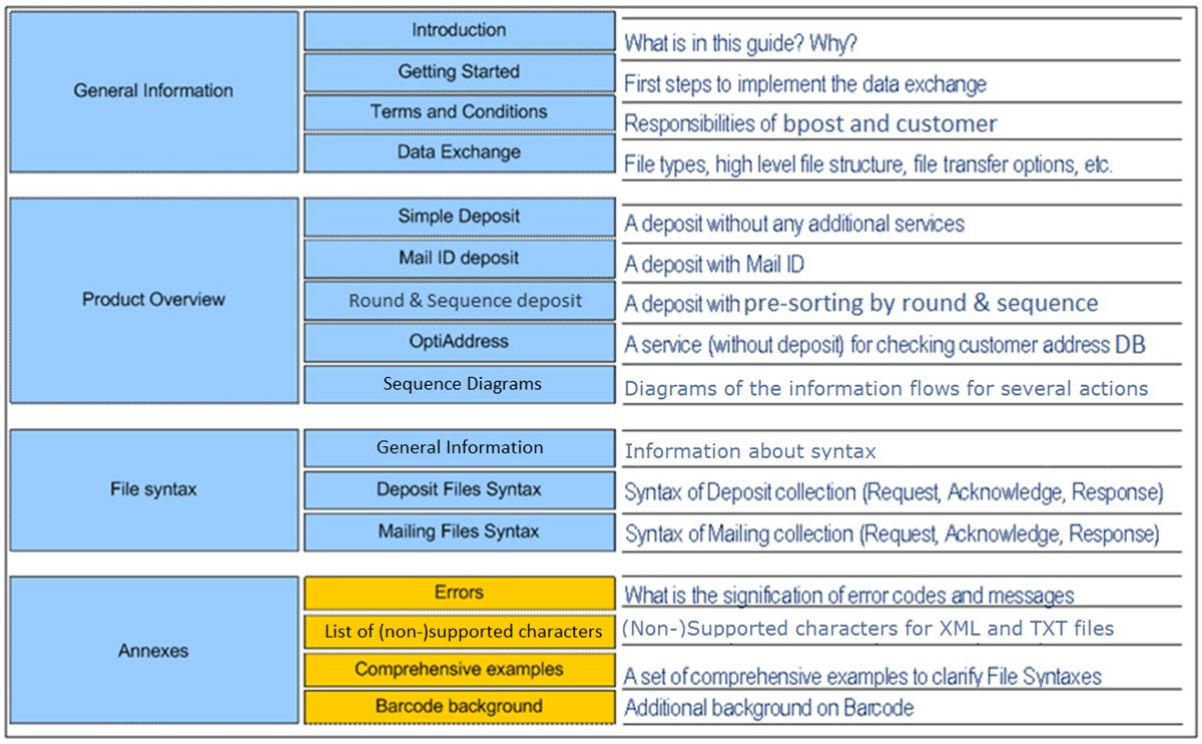


Figure 1: Overview of the chapters

2. Getting started

This chapter describes the steps needed to start using the data exchange options. The first section describes the process to start using data exchange. The second section provides contact details.

2.1. Process

The overall process to implement data exchange with bpost is represented graphically as follows:

Figure 2: Steps of the overall process

Get access

To get access to structured data exchange for one of the products and services listed in part II, the customer must contact the Business contact centre or his Account Manager to start the process.

Implement technical guide

After access has been granted, a technical specialist will be available to answer questions the customer may have with the implementation of the technical requirements described in this document.

Manage account

In addition to setting up the communication and implementing the file syntax, the customer should also manage his account online[1]. This includes the following actions:

1. Create and manage users, including assigning specific user rights, etc.,
2. Create and manage models.

Unless users and models have been created, the customer will not be able to use the data exchange, as the information created in this step is needed to create the files and/or start the communication with bpost.

The e-MassPost user guide[2] provides additional information on how to perform these tasks. Contact your Technical Specialist if there is any issue.

Once all technical requirements are implemented and users with the right user access have been created, the customer can proceed to the test phase. When data exchange is done via FTP, the test phase consists of two parts, i.e. a connection and security test followed by a file syntax test. When the data exchange is done via HTTPS, the connection and security tests are not needed, leaving the file syntax test as the only test to be performed.

Connection and security test

This test is only needed for data exchange using the ftp transfer method.

Before actually testing the content and syntax of the files, a test will be done to verify whether communication is possible. In other words, this test will verify the configuration of all parameters having an influence on the ftp communication between the cutomer and bpost.

The customer must contact his technical specialist to inform him that he is ready to test the connection, to be guided through the process.

File syntax test

This test is needed for all methods of data exchanges.

Now that communication is possible, it is needed to test whether the content sent matches the technical requirements, to insure proper processing by the bpost's systems.

The customer constructs the needed file syntaxes for the product or service he wants to use and sends them to bpost via the selected transfer method (ftp, http). The communication mode will be "T", the code for test. The communication mode is included in the sent files and is explained in Part III, File Syntax.

An acknowledgement file is generated upon reception of the Request File by bpost. When the system has processed the Request, the system will send a Response file containing feedback and

[1] Regardless of the transfer options, all users need to manage their account online, i.e. via the e-MassPost website.

[2] e-MassPost user guide : NL; FR; EN

errors (if any). A detailed description of the Response file can be found in Part III, File Syntax. In case of errors, the technical specialist can assist the customer in resolving the issues.

Certification and non-compliancy

In this final step before using data exchange in "production" mode, a certification process is required. The objective of the certification is to ensure that the end-to-end process runs smoothly. The process will be tested end-to-end, from generating the files to sorting the mail on the machine. Because of evolution of systems and services, the certification could be repeated in order to guarantee that the configuration of customer's system is still in line with the system requirements. This kind of 'maintenance' ensures the smooth process over time.

In case the customer's printing processs and/or data exchange process is changed, bpost strongly recommends to renew certification(s).

bpost reserves the right to require re-certification of a customer in case of major technical and/or process non-compliancy with specifications described in this document.

Although all functionalities of the system are available in certification mode, the mail used in this certification phase will not be distributed. A subset of this mail sample will be kept for reporting purposes; the remainder will be destroyed after processing.

The certification phase consists of four steps:

1. Creation of a physical sample,
2. Announcement of the deposit,
3. Deposit of the physical sample
4. Processing of the sample.

If all steps are successful, the customer can be certified.

Create a physical certification sample

In the first step, a physical mailing sample must be created. It is composed of at least 1000 barcoded mail pieces. The barcode on each mail piece needs to correspond to the MAIL ID number and address in the mailing file.

Announcing the certification sample deposit (Data exchange)

In the second step the customer announces the deposit. The announcement of the deposit is done via one of the methods described in this guide, but always includes deposit data (day, content of deposit) as well as mailing data (addresses). The deposit announcement can be done either via the webform or via a structured file. The mailing data, however, always needs to be in the form of a structured file. Hence, a Mailing Request File needs to be generated, containing at least 1000 addresses.[3]. The communication mode will be "C", the code for Certification. The communication mode is included in the data files, and is explained in Part III, File Syntax.

[3] The composition of this file will be detailed in the remainder of this guide.

Feedback about the data exchange will be provided in structured files[4]. This is explained in the remainder of this document.

Deposit of the certification sample

In the third step the customer needs to perform the physical deposit of the sample. Therefore a number of actions need to be carried out.

First, the customer needs to inform his technical specialist over the HyperMassPost center chosen for the certification deposit. Note that he can only deposit the certification sample in a HyperMassPost center and NOT in a MassPostCenter. He must follow the normal procedure for announcing his deposit[5], but he must indicate that it is a MAIL ID certification deposit sample.

Second, he must inform his technical specialist about the certification deposit (date, hour and deposit number).

Third, he needs to condition the barcoded mail sample in the following way:

The mail must be put in blue trays (+/- 4 trays for small format, +/- 10 trays for large format). The certification sample should contain at least 1000 pieces.

Each tray needs to be provided with three A4 papers, clearly containing the single word "CERTIFICATION". On each of the two long sides of the tray one should be taped and the third one needs to be put into the tray, on top of the mail.

If the trays are placed in a container together with other regular mail, these trays must be clearly separated from the other trays.

Finally the certification sample must be deposited in the HyperMassPost center on the agreed upon date and time.

Processing certification sample

When the certification sample arrives at the HyperMassPost center, the mail will be sorted on the machine for validation. From all the letters that were read correctly by the sorting machines, about ten are kept in customer's file as sample. The remainder of the sample will be destroyed. The letters that were not correctly read will be analysed by the technical specialist in order to understand the reason why the machine could not correctly sort the letters[6].

The technical specialist will give the customer feedback on any errors encountered during the processing of the mail. If needed, the technical specialist will ask to redo the certification.

When no issue is encountered during the processing of the mail, the Customer Reference Data will be updated, enabling the customer to use the Mail ID system in production mode. However, as a final check, the technical specialist will verify the first use in production mode. Therefore, the customer should inform his technical specialist of this first 'real' deposit (date, hour, deposit number). The technical specialist will get back with feedback.

---

[4] These are the Acknowledgement and Response files. The acknowledgement file indicates the reception of a file, the response file contains information relevant to the processing of the file (including errors)

[5] In short, notify 48 hours beforehand the date and time of your deposit to the HMPC where you will deposit the certification sample.

[6] e.g. The barcode is not visible, the barcode is distorted (badly printed), etc.

bpost business

Production

After all tests have been performed and certification has been succesfully completed, the customer may start using the data exchange in production mode. The communication mode will be "P", the code for Production. The communication mode is included in the sent files and is explained in Part III, File Syntax.

2.2. Contact List

The customer can reach bpost in a number of different ways.

Website

Does the customer need a list of MassPost centers near him? Does he need a manual on how to use e-MassPost? The masspost website contains answers to standard questions:

http://www.bpost.be/masspost

e-mail

customer.operations@bpost.be is the designated email address for all MAIL ID related questions.

Telephone

For all questions related to data exchange, the Service Center is available every weekday from 8h30 to 17h30

Tel number: 022 011111

3. Terms and conditions

Participant Agreement

When the customer starts with the MAIL ID program, a participant agreement is made between the customer and bpost. This agreement outlines the terms and conditions of participation in the MAIL ID program and provides a number of parameters necessary for proper configuration.

Site Administration

The mailer is responsible for maintenance of his e-MassPost user rights after the start up of the MAIL ID program.

Technical Specifications

In the MAIL ID program, the customer's responsibility is engaged regarding the exchange of virus-free information between his IT infrastructure and the MAIL ID system. The customer must follow the guidelines defined in this document in order to be compliant.

Pre-Sorting Codes

As pre-sorting codes change over time, for customers sending pre-sorted mail, it is strongly recommended to use the latest current pre-sorting codes of bpost.

If for some reason the mailers wish to use a different presorting codes set than the one currently active (for example, a planned presorting code update will take place between declaration and actual deposit), it should be indicated using the "PresortingCodeFile" tag (see further). The mailers will first have to ensure availability of the file on the servers of bpost.

Printing of barcodes

The mailers need to ensure that the barcodes on the physical mail pieces are correct, in line with the corresponding data file, and readable by the sorting machines of bpost.

Certification Program

Every applicant using MAIL ID has to successfully pass through a certification program to finalize the application process. The certification program has an electronic and a physical part. After a successful certification the customer will gain access to the MAIL ID system.

MAIL ID number uniqueness

To insure proper sorting of the customer's mail, it is indispensable that the requirement of uniqueness of the MAIL ID number is satisfied for 30 days, at least.

Accuracy of the files transmitted

To insure proper sorting of the customer's mail, it is indispensable that the data in the files transmitted to bpost are an exact representation of the information printed on the envelopes.

Malicious software

Malicious software includes viruses and other destructive programs, such as Trojan horses and network worms. Customers need to clean Request files submitted to bpost from any malicious software.

Processing

bpost cannot guarantee that the customer's files will be processed correctly if they don't fit the requirements. In return, bpost takes the engagement that the files will be processed in time, given compliance with all requirements.

Process Time

The table underneath gives an indication of the expected processing time needed to process a file. The processing time depends, amongst other parameters, on the number of addresses in the file and on the type of action (MailingCreate for MAIL ID and Round & Sequence, or MailingCheck for OptiAddress). For reasons of simplicity only these factors are taken into account here, as they are the most important factors determining the processing time needed for a specific file. Other factors that could impact performance involve number of concurrent users, required product (e.g. MAIL ID or Round & Sequence), exceptional errors, etc. Therefore we advise to include some lead time.

Number of Addresses	Expected Processing Time MAIL ID	Lead Time Advised MAIL ID	Expected Processing Time OptiAddress	Lead Time Advised OptiAddress
1.000	< 10 min	30 min	< 2 hours	12 hours
10.000	< 13 min	30 min	< 2 hours	12 hours
50.000	< 15 min	30 min	< 2 hours	12 hours
150.000	< 30 min	45 min	< 2 hours	12 hours
600.000	< 90 min	2h	< 2 hours	12 hours

Table 1: Indication of processing time

Data Protection

bpost values the confidentiality of its customer's data. The data will not be used for other purposes than for the sorting and distribution of letters.

The customer's data will not be given or sold to any third party and will be periodically removed from our systems, when no longer needed. For the proper processing described herein, it is required to transmit some data to a closed-loop system of a sub-contractor of bpost.

Only statistical information will be retained for a period of time needed for proper management of the program and the customer's account.

In case of corrupted, tampered or damaged file, the responsibility of bpost cannot be engaged.

General information for convention customers can be found here:

Fr: Document "Conditions Générales Envois Adressés (National)" on the page of bpost website

Nl: Document "Algemene Voorwaarden Geadresseerde Zendingen (Nationaal)" on the page of bpost website

Software

Customers will need common software to handle the data exchange with bpost. The software may include[7] text editor or XML editor, XSLT mappers, browser, Zip/Unzip software, FTP client (be carrefull Gzip is not supported).

bpost does not provide any software.

Initiation of communication

All communications with bpost are initiated by the Customer: both when customers initiate communication by uploading Request files to be processed by bpost, and when bpost provides Response files, that customers need to download from the FTP server.

Folder management

The receiving party is responsible for cleaning up files after processing. After a customer uploads a Request file, it is the responsibility of bpost to remove the Request file from the \requests directory of the customer after loading it. Similarly, after bpost provides an Acknowledgement or a Response file in the \responses directory of a customer, it is the responsibility of the customer to delete the file after downloading it.

Evolution

bpost will regularly review and update these specifications and reserves the right to change it at any time. Every effort will be made to make new specifications backward compatible, and every effort will be made to give mailers as much time as practically possible to get ready for new specifications.

[7] depending on the options chosen for file format and file transfer,

4. Data exchange

This chapter indicates how data is exchanged between the Customer and bpost. The introduction gives a generic overview on data exchange. Section 2 describes the different file formats and section 3 elaborates on how the data exchange occurs, indicating the different file transfer possibilities. Section 4 finishes with important conventions and standards that must be observed.

4.1. Introduction

Generic File flow: Request, Acknowledgement, Response

All data exchanges between the Customer and bpost are initiated by the customer, through the transmission of a Request File.

bpost acknowledges the receipt of a file with an Acknowledgement File. The only objective of this file is to report that the file was received.

However, the acknowledgement file does not report on the syntax and content correctness of the Request file. This is done through the Response File. The objective of the response file is: (1) to indicate whether the treatment of the file was successful (including the reason if unsuccessful), (2) to offer feedback on the content of the file, like a price quote in case of a Deposit Request File.

File availability

Files generated by bpost (Acknowledgement File, Response File) are made available to the customer. The customer needs to initiate the data exchange to retrieve these files. However, customers can sign up for email notifications indicating that a Response File will be ready for download in a few minutes.

The good processing of the Request file is not guaranted before the Response File is available.

Underneath there is an illustration of the generic file flow (Request, Acknowledgement, Response) explained above.

The file flow principle is very simple:

1. The customer sends a **Request File** to the system.
2. At reception of the request file, the system generates an **Acknowledgement File** available for the customer.
3. After having processed the request file, the system creates a **Response File** available for the customer.

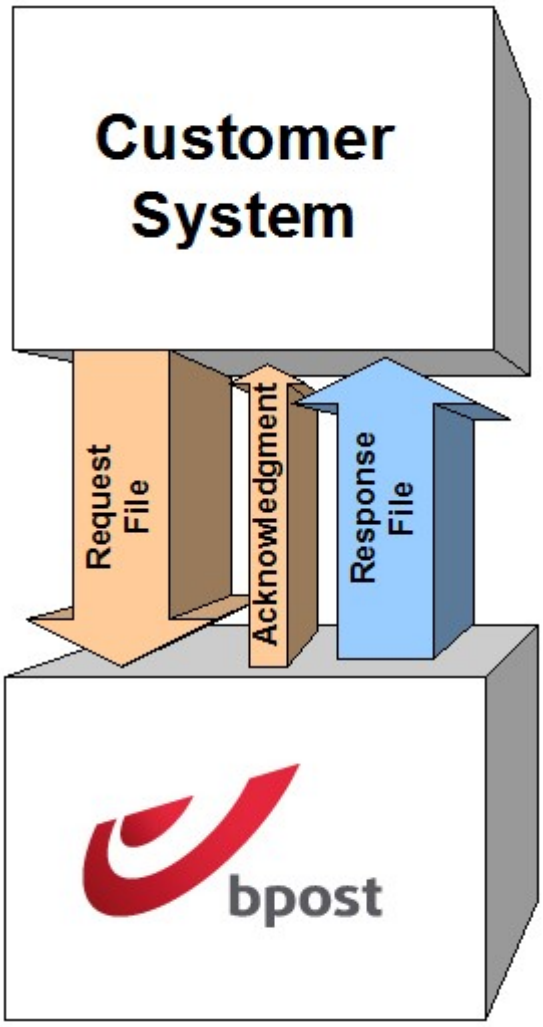


Figure 3: Generic File Flow (Request, Acknowledgement, Response)

Generic File structure

In this section, the high-level structure of the files that are part of the Generic File Flow is discussed. There are subsequently a description of the Request file, the Acknowledgement File and the Response file.

Request File

The request file, created by the customer, has the following structure[8].

Each **Request File** must contain:

- One **Context** section
- One **Header** section
- One or more **Action** section(s)

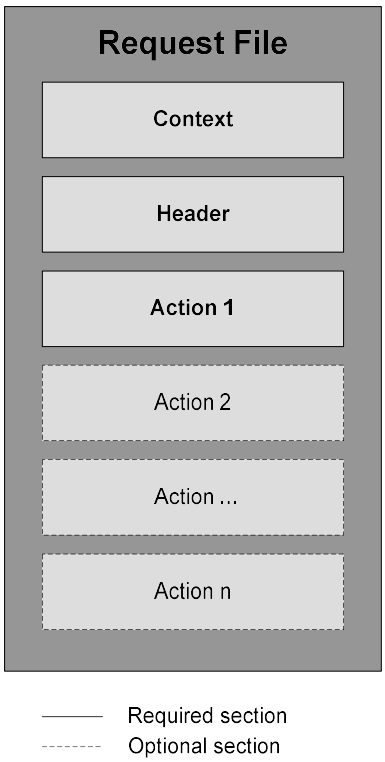


Figure 4: Request File Structure

Mailing Request file can contain three types of actions: (1) Mailing create, to create a new mailing, (2) Mailing delete, to delete an existing, previously created mailing, (3) Mailing check to verify addresses with the OptiAddress option and independently of a physical mail deposit. More information can be found later on in the document.

Acknowledgement file

The acknowledgement file is very simple and straightforward. A high-level representation of this file would not make any sense. See relevant sections in Part 3 File syntax.

[8] This structure is further detailed later on in the document.

Response File

The **Response File,** created by bpost, has the following structure[9].

Each **Response File** must contain:

- One **Context** section
- One **Header** section
- Optional **Replies** in case of errors and/or messages for the Context and Header sections
- One or more **Action** section(s), (one section for each action from the corresponding Request File), including optional **Replies** in case of errors and/or messages for the **Action** section

Figure 5: Response File Structure

4.2. Supported file formats

This part handles three file formats that can be used for implementing the data exchange, i.e. XML, TXT, and XLS/XLSX. The first point provides additional information concerning the XML file format. More information on the TXT format is given in the secont point. The third point presents the possibility to use in some cases the XLS/XLSX file type. The fourth point, file naming conventions, deals with the generic file naming convention that applies for the different file formats. The last point concludes with a list of auxiliary tools and methods (related to file format) that can be used before transmitting files.

[9] This structure is further detailed later on in the document.

Software

An XML editor and XSLT mappers are needed to generate XML files, or equivalent capabilities native to software applications.

Standards

For XML format, the standards followed by bpost are:

- XML:
  Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04th February 2004
  More information can be found on http://www.w3.org/TR/2004/REC-xml-20040204
- XML Schema:
  XML Schema Part 1: Structures / Part 2: Datatypes (Second Edition), W3C Recommendation 28 October 2004
  More information can be found on http://www.w3.org/TR/2004/REC-xmlschema-1-20041028, and on http://www.w3.org/TR/2004/REC-xmlschema-2-20041028
- XPath:
  XML Path Language (XPath) Version 1.0, W3C Recommendation 16 November 1999
  More information can be found on http://www.w3.org/TR/1999/REC-xpath-19991116
- codepage[10]:
  The codepage used by bpost is LATIN-1 (ISO-8859-1). Consequently, for XML files, the XML header must contain this encoding information:

  <?xml version="1.0" encoding="ISO-8859-1"?>

Notation

Although the customer can choose between the XML and TXT file format, in the remainder of this document the file content of the data exchange is described with the XML format in mind. The file content for the TXT file format can then be easily deducted from the XML format using some simple rules. Therefore, underneath we first describe the notation used for XML file, followed by a description of how to apply this to the TXT file format.

XML tag types

Hierarchy: There are a number of different mandatory and optional 'tags' sorted in a number of different levels. All tags from a certain level belong to a certain tag from the previous level. All tags from the first level belong to the same single tag, which is refered to as the root tag, e.g. the root tag name for a deposit request File is <DepositRequest>.

Obligation: There are mandatory and optional tags. Mandatory tags need to be present when their previous level tag is present. Optional tags may or may not be present. In the file structure

[10] A codepage is used by the system to encode and interpret strings of characters. Codepage formats are not the same for all languages. Some languages such as Japanese and Hindi have multi-byte characters while others like English only need one byte to represent each character.

diagrams, mandatory tags are represented by a full line, while optional tags are represented by a dotted line.

Action Tags: Action tags indicate the actual actions requested from the system. This is best explained with an example. The possible action tags for the Deposit Request File are: DepositCreate, DepositUpdate, DepositDelete and DepositValidate.

XML notation

Tag and attribute notation

Tags: the first letter of each word in the tag is uppercase, including the first letter of the tag. All other letters are in lowercase.

Example: SomeInterestingTag

Attributes: the first letter of each word in the attribute is uppercase, except for the very first letter of the attribute, which is lowercase. All other letters will be in lowercase,

Example: someInterestingAttribute

Table notation

The XML structure is described in tables (e.g. Table « DepositRequest Context Tag – XML Structure »). The column "Tag Name" contains the name of the tag.

- Root tags have a simple name

  Example: Book

- Child tags have a compounded name, made of the name of the parent, a '/' as separator, then name of the child. Note that if the parent tag is also a child tag, the parent name itself is a compound name.

  Example: Book/Chapter, Book/Chapter/Paragraph

- When a tag can occur more than once, the tag name is followed by (#N)
- The *Mandatory* column indicates if a tag is required or not.

XML to TXT

To apply the XML structure into the TXT structure:

- Tags level 1 are all present
- Tags level 2 and after are present if they have attributes or direct content
- First column/field in the text-format file is always a tag name, and cannot be changed
- There is no correspondance in the TXT format for the XML tags used for aggregation, so they are omitted

TXT

Software

Text editor to create the ASCII text formatted files or equivalent capabilities native to software applications.

Standards

Separator

For the TXT format, the character pipe ('|' - ASCII 124) is chosen as delimiter. If customers need to embed the pipe character in a TXT file, the backslash ('\' - ASCII 92) must be used as an escape character. So, the character sequence "\|" is the pipe character itself and not the separator.

Notation

Please note that all File syntax is described primarily with XML notation in mind. Please refer to the XML paragraph for details on XML notation and for how to transform the XML notation to TXT.

XLS

It is also possible to use the XLS(X) or CSV file type for the mailing files (Mailing Request and Mailing Response) only (i.e. not for the deposit files). For this, it is necessary to use the Address File Tool (AFT), available on e-MassPost interface.

The structure of the file is simplified with this file format. Especially, some data are encoded via webform on the e-MassPost website before the upload of the file containing the addresses.

A dedicated guide exists for this tool. This "Address File Tool" guide can be obtain on request from Customer Operations team (customer.operations@bpost.be).

Software

XLS(X) or CSV editor to create the files.

Standards

Separator

For the CSV format, the character pipe ('|' - ASCII 124) is chosen as delimiter.

File Naming conventions

Files transmitted to bpost need to fulfill strict naming requirements, as outlined below. Subsequently we will handle the general syntax of the file name, an explanation of the different elements of this syntax, a few examples and finally the occasion when the tmp extension must be used.

Generic File Name

XML and TXT

The file name consists of a number of fields (all **UPPERCASE**) separated by underscores and terminated by a file extension:

AAA_VVVV_CCCCCCCC_ NNNNNNNNNN_YYMMDDHHMMSS_SSS.XXX

Where:

AAA is a 3-character alphanumeric code identifying the application responsible for the management of this specific data stream within bpost.

For deposit files, the code "*EMP*" (for "eMassPost") is required.

For mailing list files, the code "*MID*" (for "MAIL ID" or OptiAddres) is required.

VVVV is a 4-digit code identifying the version of the request. Currently, the version is *0100* for EMP-files and 0100, 0102 or 0200 for MID-files (code for versions 1.00, 1.02 and 2.00). This version code is provided by bpost.

CCCCCCCC is a numeric identifier (8 digits max) provided by bpost, uniquely identifying the sender. This identifier is known as the PRS-ID of the sender of the file. If a mail handler sends transactions on behalf of his customer, the PRS-ID of the mail handler needs to be used. The PRS-ID of the router's customer, on the other hand, will be referenced within the file content (more information are available on the Customer Reference Data Sheet).

NNNNNNNNNN is a customer-assigned 10-characters alphanumeric code, uniquely identifying the file. This field can be used for a file unique serial number, or an application code, or an internal customer, or combination thereof. bpost will be including this field in filenames of acknowledgments and responses.

YYMMDDHHMMSS is a timestamp of when the file is generated. The presence of this field is necessary to identify multiple transactions possibly generated for the same NNNNNNNNNN file.

SSS is a 3-character alphanumeric code identifying the communication step:

"0RQ" for Request Files

"1AK" for Acknowledgement Files

"2RS" for Response Files

XXX is the file extension identifying the file format (XML or TXT). This extension **must** use capital letters (all uppercase).

Examples

1. Deposit Request file (version 1.00) ABCD123456 for customer 12345:
EMP_0100_12345_ABCD123456_120214150334_0RQ.XML

2. Corresponding Acknowledgement file:
EMP_0100_12345_ABCD123456_120214150445_1AK.XML

3. Corresponding Response file: EMP_0100_12345_ABCD123456_120214151235_2RS.XML

XLS

With the XLS(X) and CSV file formats, no specific rule is applied on the file naming.

TMP extension

When a file is uploaded using the FTP protocol (see furtheron), it can be named with the extension ".TMP" (in place of the extension ".XML" or ".TXT") during the uploading. This indicates that the file is currently in the process of being transmitted and ensures that bpost never processes a partial file.

Once the uploading is completed, the file needs to be renamed to the appropriate extension.

Pre-sorting codes file

For the pre-sorting of the mail, it is possible to use a file containing all the Belgian postal codes linked to their associated pre-sorting code (e.g. "B-W1-L1"). This file is available on the publisher FTP account (see subchapter 4.3 "Supported File Transfer Options", section "FTP Publisher"), and its file naming convention is a bit different from the generic file name.

The naming convention for such publication is the following:

MID_FFFF_PSCVVVVVVV_YYMMDDHHMMSS_3PR.XXX

Where:

FFFF is the version of the data/structure format. Actually only format 0100 is supported.

VVVVVVV is the version of the presorting code file on 7 digits (with leading zeros)

YYMMDDHHMMSS is a timestamp of when the file is generated

3PR is the alphanumeric code identifying the communication step reserved for the technical documents

XXX is the file extension identifying the file format (.TXT or .XML)

Example:

MID_0100_PSC0000107_120214143743_3PR.XML

This file contains presorting codes with format 0100 and version 0000107. That file has been created the 14/02/2012 at 14:37:43.

Optional tools and methods

Compression (optional)

The file can be compressed using the Zip algorithm (and no other compression methods) prior to transmission. More information concerning this compression algorithm can be found at http://www.info-zip.org/.

The file can be compressed manually using a zip tool or by a zip library. The customer must verify that the compressed file can be opened with a tool such as WinZip.

If compressed, the file name must finish with the mention ".ZIP" (in upper cases) after the file format (XML ot TXT) (see Table below in the sub-chapter "Summary of data exchange standards" for examples of name of compressed files).

Validating XML files (optional)

Before sending an XML file to the MAIL ID system, the customer can validate the structure of the file. To do this, bpost proposes a schema with which the customer can validate his XML file. The XML schema cannot be used to validate a text delimited ASCII format file.

These schemas are available on the e-massPost website, under the tab "Information", or via request to customer.operations@bpost.be.

4.3. Supported file transfer options

This section deals with the *two options* that are available for customers to exchange files with bpost.

Data Transfer Options

Customers transmitting data to bpost can choose to:

- Use the **e-Masspost website** to communicate with bpost. This is called "***interactive***" communication, making use of the HTTP protocol. Data can be entered in the webforms or can be uploaded with a structured file via a webpage[11];
- Use a **FTP client** to communicate with bpost. This is called "***unattented***" communication, making use of the FTP protocol[12]. The data is transmitted to bpost in a structured file. The files are stored in a Request folder, containing all files[13] sent by the customer, and a Response folder, containing all files[14] sent by bpost.

In both cases customers need to provide their e-MassPost login and password at the initiation of the data exchange.

The situation is summarized in table below:

Communication	Protocol	Data Entry
Interactive	HTTP	Web form or Structured file
Unattended	FTP	Structured file

Table 2: File Transfer Options

HTTP protocol

The File Transfer via the HTTP protocol is very straightforward. The technology uses a simple browser.

Interactive mode uses the HTTP(S) protocol to communicate through port 80 or port 443. Firewall settings must allow communication through these ports.

---

[11] Depending on the service that the customer wants to use, he will have to exchange different data with bpost. Refer to Part II, Products and services for more information. Nonetheless, the 'data entry via webform' in interactive mode is only applicable to Deposit Request Files.

[12] This communication mode is not available for the AFT (XLS(X) and CSV file formats). The customers using AFT must so use the HTTP protocol via the e-MassPost website.

[13] These are Request files. Note that the request folder only shows the items that the logged-on user has sent, whereas the response folder containes all files, regardless of which user has sent the corresponding request file.

[14] These are Acknowledgement files, response files and Deposit Authorizations.

To perform the transfer in a secure way, the web server and the browser will encrypt the session using SSL. No user setup is required for this process except to install the SSL certificate, following instructions of the browser.

FTP

The FTP protocol is used for transmission between the Customer and bpost in "Unattended Mode", only for TXT and XML format.

During the transmission of a file between the customer network and the FTP depository, the extension of the file can be ".TMP". Once the file is completely transferred on the depository, the extension must be switched to ".XML" or ".TXT".

General information

FTP host details

Customers are advised to use the DNS host name for the FTP server of the bpost rather than the IP address, as this address may change over time. Refer to next paragraph (technical details FTP transfer) for DNS host name to be used.

FTP client configuration

The FTP server of bpost is configured in passive mode. Firewall settings must allow communication through the ports used for FTP communication. Refer to next paragraph (technical details FTP transfer) for an overview of which ports need to be open.

In order to allow compression and encryption, the FTP data format (type) must be "binary" and the data exchange (transmission) mode must be "stream".

Customers must provide a fixed IP address for connecting FTP client[15]. This prevents unknown IP addresses from gaining access to the FTP server.

A maximum limit of 1 connection initiation per 5 minutes for bpost outbound transfers (initiated by partner) and that polling of the bpost file transfer system should be in line with the agreed upon use case.

Security

FTPS

Secure FTP (i.e. FTPS, FTP over SSL) is the security offered on FTP communication. If customers choose this mode, any data is transmitted encrypted over the Internet.

Non-secure FTP

If customers choose this mode, username and password information is transmitted "in clear" over the Internet, and any data transmitted is not encrypted.

[15] Or an IP address range.

Folder management

As mentioned in the "Terms & Conditions" part, the customer is responsible to ensure that the free space on the \responses directory is sufficient.

Technical details FTP transfer

	FTP	FTPS
FTP host	filetransfer.bpost.be	filetransfer.bpost.be
Ports used	Tcp/30000-40000 and tcp/21	Tcp/30000-40000 and tcp/21

Table 3: FTP Transfer Details

FTP Publisher

To provide some documents to customers, bpost uses a specific FTP account. The username is MID_Publisher and the password will be communicated to MAIL ID customers.

All documents are posted in the 'response' sub-directory as for any FTP account and will have a specific naming identifying the content and the version of that content.

The FTP Publisher contains the structured files with the pre-sorting codes. These files can be used to automate the update of pre-sorting codes. At any time, two versions of the pre-sorting codes are available in this folder : one for the current pre-sorting codes, and another for future pre-sorting codes. These files exists as well in TXT than in XML.

See subchapter 4.2 "Supported File Formats", section "File Naming conventions", subsection "Pre-sorting Codes File" for more information about the nature and the naming convention of this file.

4.4. Summary of data exchange standards

Transfer options

Transfer type	Security	Host name	Firewall settings (open ports)
HTTPS	security is enabled via SSL	www.bpost.be/emasspost	TCP/80 TCP/ 443
FTP	no security	filetransfer.bpost.be	Tcp/30000-40000 and tcp/21
FTPS	security is enabled via SSL	filetransfer.bpost.be	Tcp/30000-40000 and tcp/21

Table 4: Summary of Transfer Options

bpost business

File Formats

File naming convention (All uppercases)

AAA_VVVV_CCCCCCCC_ NNNNNNNNNN_YYMMDDHHMMSS_SSS.XXX

AAA the Application Code,

VVVV the Version Code,

CCCCCCCC the Sender Identification,

NNNNNNNNNN the Customer File Reference,

YYMMDDHHMMSS the Time Stamp,

SSS the Communication Step, and

XXX the file extension.

File extensions

Different file formats are supported, that are XML, TXT and XLS(X) (or CSV). Hereunder a concise overview of the relevant technical information for both formats.

Extension	Summary technical details
.XML	Software: XML editor, XSLT mappers Standards: XML 1.0 (third edition), W3C recommendation 04/02/2004 XML Schema Part 1: Structures / Part 2: Datatypes (Second edition), W3C recommendation 28/02/2004 XML Path Language (Xpath) version 1.0, W3C Recommendation 16/11/1999 Codepage LATIN-1 (ISO-8859-1)
.TXT	Software: text editor for ASCII text formatted files Standards: delimiter is pipe character ('|' - ASCII 124)
.XLS .XLSX	Software: XLS or XLSX editor Standards: Excel 97 and later
.CSV	Software: CSV editor Standards: delimiter is pipe character ('|' - ASCII 124)

Table 5: Overview of the supported file formats

Apart from that, there are two other file extensions that can occur for XML and TXT files, i.e. the TMP extension and the ZIP extension.

.TMP	Must be used when using FTP while uploading a file. The extension should be changed to TXT or XML when the upload is completed.
.ZIP	XML and TXT files with the corresponding extension may be zipped to speed up the data exchange.

Table 6: Additional file extensions

Therefore both .XML.ZIP and .TXT.ZIP are possible, as the following table indicates.

Type	Extension	Comment
Zipped XML	.XML.ZIP	The name of the data file contained in the zip file will match exactly the name of the zip file, minus the .ZIP extension. Example: the customer sends: EMP_0100_12345678_ABCD123456_120214150334_0RQ.XML.ZIP The only accepted file name in this ZIP file is: EMP_0100_12345678_ABCD123456_120214150334_0RQ.XML
Zipped TXT	.TXT.ZIP	The name of the data file contained in the zip file will match exactly the name of the zip file, minus the .ZIP extension. Example: the customer sends: EMP_0100_12345678_ABCD123456_120214150334_0RQ.TXT.ZIP The only accepted file name in this ZIP file is: EMP_0100_12345678_ABCD123456_120214150334_0RQ.TXT

Table 7: Zipped extensions

Part II: Product & technical information

This chapter describes different products that require date exchange between bpost and the customer.

1. Deposit

1.1. Introduction

It is possible to announce MassPost deposits via electronic files, and so to receive electronically the authorization of deposit.

This automated process is available for most of the MassPost products (Mail ID, Round & Sequence, and non-Mail ID products).

1.2. Data Exchange Options

Recall from Part I that data exchange for this product is possible in two distinct ways, by webform or by structured file.

Webform

The deposit can be created online via web interface, i.e. the e-MassPost deposit announcement module. The website will guide the customer through this process.

As such, this method of communication is not treated within this document. Refer to the e-MassPost user guide (fr) or e-MassPost user guide (nl) for more information.

Structured file

To use the structured file, the customer needs to implement all requirements stated in Part I.

Finally, he will need to adapt his ICT infrastructure to enable the generation of structured files, whose syntax is detailed in Part III, chapter 2 "Deposit Files Syntax".

For deposits:

- To create a deposit, the customer will need the DepositCreate action tag. Refer to the DepositCreate section.
- To update a deposit (i.e modify deposit features), he will need the DepositUpdate action tag. Refer to the DepositCreate section and replace all DepositCreate action tags with DepositUpdate action tags.
- To delete a deposit, he will need the DepositDelete action tag. Refer to the DepositDelete section.
- To validate a deposit (deposit authorization), he will need the DepositValidate action tag. Refer to the DepositValidate section and replace all depositUpdate action tags with DepositValidate action tags.

It is important to remember:

- That multiple actions (DepositCreate, DepositUpdate, DepositDelete and DepositValidate) can be used within the same Deposit Request File.
- That each DepositCreate needs to be followed by a DepositValidate, unless the customer uses autovalidation in the DepositCreate action. As described furtheron, this can be done by setting the attribute autoValidate to 'Y'. After completion of the DepositValidate, a deposit authorization (in PDF format) is provided that needs to be presented at the MassPost dock when delivering the deposit.

2. MAIL ID deposit

The first point of this chapter gives an overview of what a MAIL ID deposit is. The second point is an overview of how data exchange for this product should be implemented.

The information about the barcodes are given in the chapter "4. Barcode".

2.1. Introduction

MAIL ID is an innovation of bpost. MAIL ID is a barcode generated by the customer or by bpost (on request). The barcode itself holds no routing or sorting information. Its only purpose is to identify the mail piece in a unique way.

When customers prepare a MAIL ID deposit, a unique (during 30 days at least) MAIL ID barcode for each mail piece will be generated by customer or by bpost (if demanded in the Request file). The customer will also generate a mailing file that holds all the MAIL ID barcodes for a given deposit and the corresponding delivery addresses and (facultatively) recipients. This file is sent to bpost prior to the delivery of the physical mail pieces.

bpost will interpret the addresses in the mailing file and match them to the correct sorting information. The unique MAIL ID barcode and the obtained sorting information are then stored in the sorting machines. When the mail pieces are processed on the machine they will be sorted based on the MAIL ID barcode and the reference data it points to.

This schema repeats the concept and the process steps:

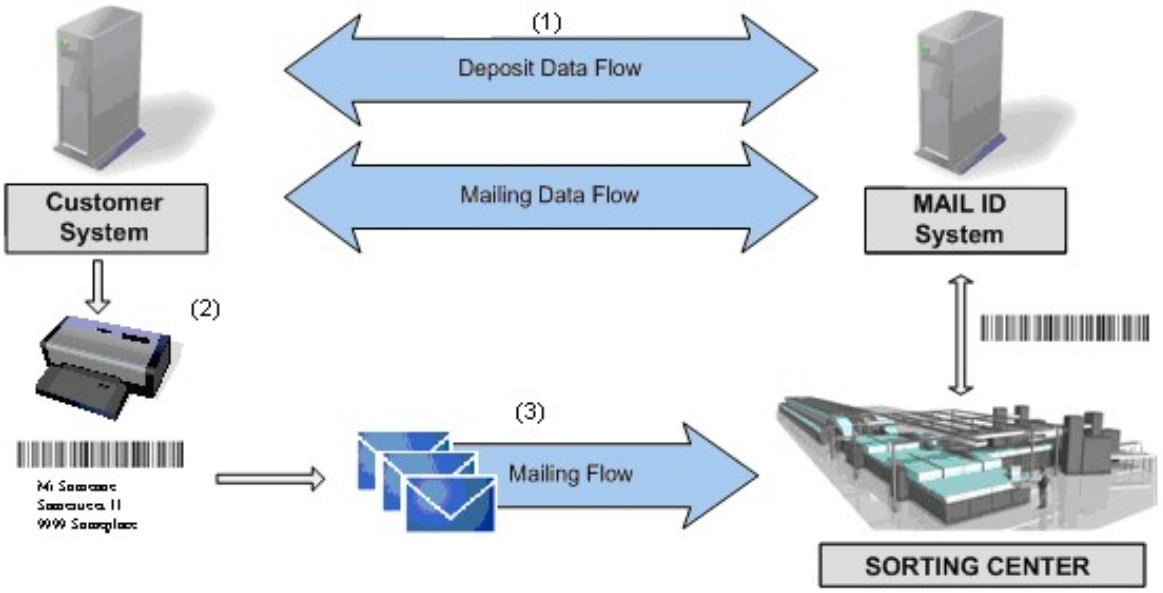


Figure 6: MAIL ID flows Schema

There are two data flows, the Deposit Data flow and the Mailing data flow (1). Each requires sending a Request file by the customer and the generation of both an Acknowledgement and a Response files by bpost. These will be handled in Section 2, Data Exchange options.

The customer also needs to print on each mailpiece the barcode (2) just above the associated address, before delivering them to the MassPostCenter (3). Strict guidelines and rules apply to the barcode. These are explained in Section 3, Barcode[16].

Finally, the mailpieces will be sorted in the sorting center using the barcodes to uniquely identify each of them. The machine communicates with the MAIL ID system to retrieve the address information needed to sort the mail pieces.

A typical[17] sequence of actions would be:

1. The mailer creates an e-MassPost Deposit[18] selecting a MAIL ID product;
2. The mailer sends a structured Mailing file with the destination addresses for his deposit. A unique MAIL ID number identifies each address record;
3. The MAIL ID system interprets[19] each destination address to retrieve the correct sorting information;
4. The MAIL ID system returns error feedback, and the number of address records that could be interpreted. The number of interpreted addresses is called the compliance rate;
5. The mailer evaluates his compliance rate as acceptable and proceeds with the mailing. He validates his deposit announcement and he receives a deposit authorization;
6. The mailer makes his physical deposit in a MassPost centre;
7. The mail pieces are sorted by bpost using the reference in the MAIL ID barcode to retrieve the sorting information obtained during the interpretation.

Note: customers can request that bpost generates the unique MAIL ID numbers, with the disadvantage that they will have to wait for the feedback from bpost before any of the mail pieces can be printed.

2.2. Data Exchange Options

When using the MAIL ID system, there will be two file flows, the EMP (for e-MassPost) file flow for the deposit itself, and the MID (for MAIL ID) file flow for the mailing list (the list of addresses linked to the barcode numbers). The EMP file flow is described in section 1 of part II : Deposit. It

---

[16] The customer can start printing the addresses and barcodes before or after he receives feedback from bpost.

[17] In fact the system allows great flexibility in this respect. It is possible to send the Mailing Request first. In that case the mailing Request is master, refer to section 2, subsection about linking mailing and deposit files.

[18] See for e-MassPost the e-MassPost user guide (fr) or e-MassPost user guide (nl)

[19] A set of sophisticated algorithms are used to recognize addresses, comparing them to a reference database.

can be done via a webform or a structured file. The MID file flow is described in paragraph 1 (just below). These 2 flows need to be connected. This is explained in paragraph 2.

MID Flow

As described in Part I, data exchange for MAIL ID data (addresses) can only occur via structured files.

When the customer wants to use the structured file, he needs to implement all requirements stated in Part I.

It is needed to adapt the ICT infrastructure to enable the generation of structured files, whose syntax is detailed in Part III, chapter 2 "Mailing Files Syntax", subchapter "Mailing Request File".

- To create a mailing, he will need the MailingCreate action tag. Refer to the MailingCreate subsection.
- To delete a mailing, he will need the MailingDelete action tag. Refer to the MailingDelete subsection.
- To update the addresses of the mailing, he will need to delete the mailing, using the MailingDelete action tag, and recreate it by using the MailingCreate action tag. In this case, the mailingRef tag must be changed (and the fileRef tag too, if possible). If the customer does not submit his own barcodes but ask bpost to create them, it is important to notice that the barcodes sent by bpost at the second MailingCreate will be different than the first ones. The last barcodes sent by bpost are the good ones, which must be printed on the mail.

It is important to remember that the customer can use multiple actions (MailingCreate and MailingDelete) within the same Mailing Request File.

Linking Mailing and Deposit Files

Connecting deposit and mailing files

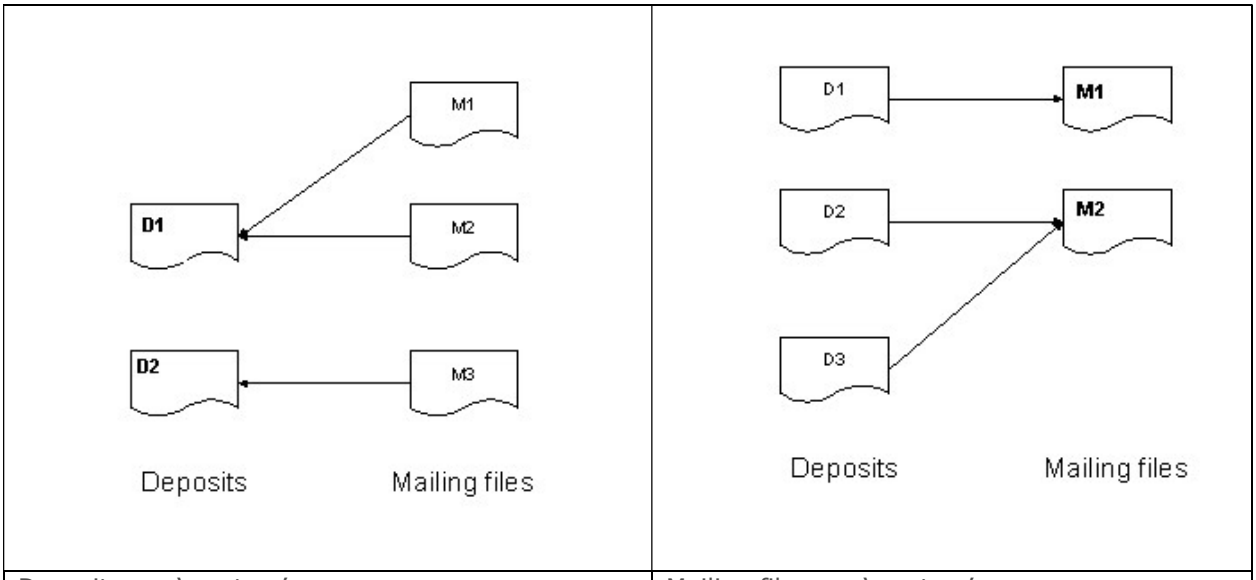
When the customer plans to deposit MAIL ID product, he is required to create a deposit and send a selection of destination addresses. But the sequence of events may vary depending on preference.

In order to successfully manage the deposit and mail flow transactions, a link is needed between the two types of transactions. Certain rules need to be followed to create correct relationships between deposits and mailing files.

Deposits and mailing files can be related to each other in one of two ways:

Deposit is the master: one or more mailing lists are linked to the same deposit (1 deposit <-- 1 to N mailing lists)

Mailing is the master: one or more deposits are linked to the same destination address list (1 to N deposits --> 1 mailing list)

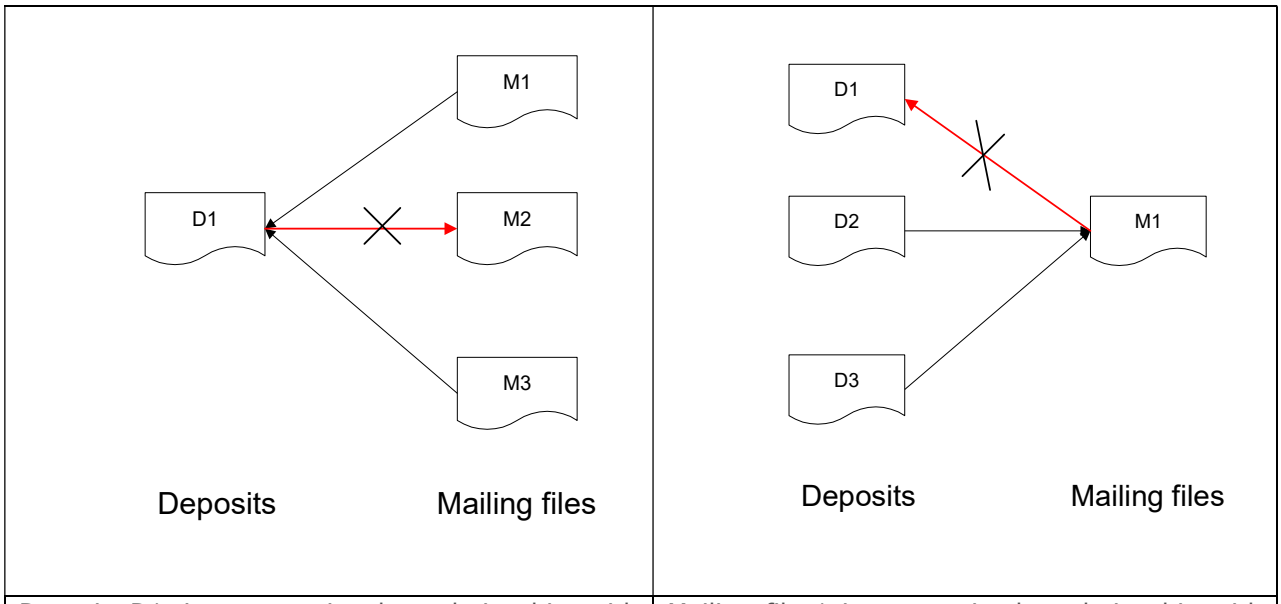


Deposits are 'masters'	Mailing files are 'masters'

Figure 7: Master – Slave Relationship

In any relationship between deposits and mailing lists, the following rule must also apply:

Once an item (deposit or mailing) is the master in a relationship: it may be linked to 1 to n slave items, while 1 slave item can be linked to only 1 master item.



Deposit D1 is master in the relationship with Mailing files M1 and M3, so it cannot become slave in a relationship with Mailing file M2.	Mailing file 1 is master in the relationship with Deposits D2 and D3, so it cannot become slave in a relationship with any Deposit.

Figure 8: Master – Slave Rules

Certain rules are embedded in the MAIL ID system for the validation of MAIL ID deposits.

Deposit is the Master

The diagram below shows the typical steps of the data exchange when the deposit is the master.

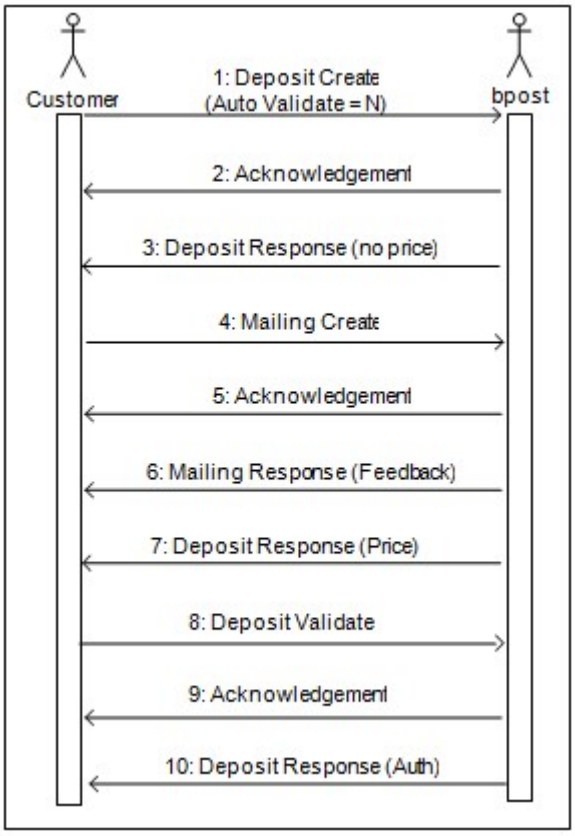


Figure 9: Deposit master

Business rules

- The master must be sent first, before sending the slaves. In this case the deposit is sent first (step 1 in the diagram above), so the related mailing files may only be sent (step 4) when the Deposit Request file has been processed, so when both the Deposit Acknowledgement and Deposit Response files have been generated by bpost (step 2 and 3).
- Each mail piece of the deposit must have a corresponding address in one of the mailing files. So the total number of addresses present in all the related mailing files must be at least equal to the number of announced mail pieces in the deposit.
- The MAIL ID deposit pricing takes the compliance rate into account to calculate the discount for each deposit. A price cannot be determined :
    - while no mailing file has been received.
    - when the total number of address records in the mailing files attached to the deposit is inferior to the number of announced mail pieces in the deposit.
- Every change made by the customer (update of the deposit, delete of mailing file, ...) can have an impact on the price and the price determination.
- A delete of the deposit results in the delete of all attached mailing files, but a delete of one or more of the mailing files does not delete the deposit.

- When a price is communicated for the deposit (step 7), the customer may validate that deposit (step 8). After validation:
    - the customer can no longer change or delete the deposit or related mailing file(s).
    - a deposit authorization (in PDF format) is provided (step 10), which needs to be presented at the MassPost dock when making the deposit.

Technical details linking

If the customer wishes to link mailing file(s) to one deposit he must first indicate that the deposit is the master by creating a unique reference for the deposit. When attaching mailing files to the master deposit, the unique deposit reference is repeated thus establishing the relationship of the mailing file(s) to the deposit:

- Step 1: when the customer creates a deposit:
    - the depositRef attribute must have a value (the unique customer reference for the deposit).
    - the mailingRef attribute must be empty.
- Subsequent step(s): when the customer creates mailing files:
    - the mailingRef attribute must have a value (the unique customer reference for the mailing).
    - the depositRef attribute must have a value (the ***same*** value as the one specified for depositRef in step 1).
- The system will store one deposit and one or more mailing files, all linked to the deposit.

There are 2 differents way to reference a deposit:

- The deposit reference (contaigned in the deposit request in step 1).
- The temporary number generated by the application (given by bpost in step 3).

Mailing File is Master

The diagram below shows the typical steps of the data exchange when the mailing file is the master.

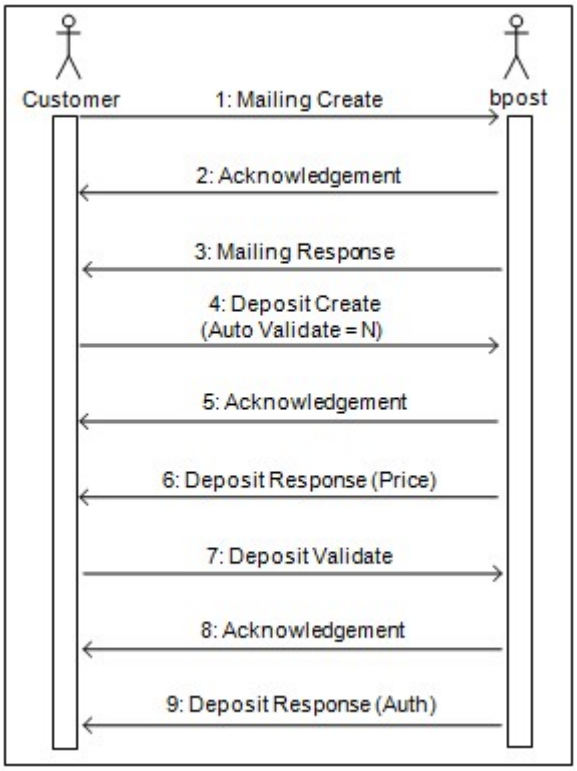


Figure 10: Mailing file master

Business rules

- The master must be sent first, before sending the slaves. In this case the Mailing Request file is sent first (step 1 in the diagram above), so the related Deposit Request file(s) may only be sent (step 4) when the Mailing Request file has been processed, so when both the Mailing Acknowledgement file and the Mailing Response file have been generated by bpost (step 2 and 3).
- Each address of the mailing file can be linked to only one physical piece of the related deposit(s), as the barcode of this address must be unique. So, the total number of addresses of the mailing file must be equal or greater than the total number of related deposit file(s).
- The MAIL ID deposit pricing takes the compliance rate into account to calculate the discount for each deposit. A price cannot be determined for a deposit when the total number of announced mail pieces in the deposit(s) attached to the mailing file, is higher than the number of address records in the mailing file.
- Every change that the customer makes (update of the deposit, delete of mailing file, ...) can have an impact on the price and the price determination.
- A delete of the mailing file results in the delete of all attached deposits but a delete of the deposit does not impact the mailing file (all addresses available).
- When a price is communicated for a deposit (step 6), the customer may validate that deposit (step 7). After validation:
    - the customer can no longer change or delete the deposit or mailing file.
    - a deposit authorization (in PDF format) is provided (step 9), which needs to be presented at the MassPost dock when making the deposit.

Technical details linking

If the customer wishes to link deposit(s) to one mailing file he must first indicate that the mailing file is the master by creating a unique reference for the mailing file. When attaching deposits to the master mailing file, the unique mailing reference is repeated thus establishing the relationship of the deposit(s) to the mailing file:

- Step 1: when the customer creates a mailing file:
    - the mailingRef attribute must have a value (the unique customer reference for the mailing)
    - the depositRef attribute must be empty
- Subsequent step(s): when the customer creates deposit(s):
    - the depositRef attribute must have a value (the unique customer reference for the deposit)
    - the mailingRef attribute must have a value (the ***same*** value as that specified for mailingRef in step 1)
- The system will store one mailing file and one or more deposits, all linked to the mailing file.

Deposit Response (Price)

Anytime an action has an influence on the price, the system sends a DepositResponse with the calculated price (e.g.: changing the characteristics of a deposit, updating the announced number of pieces, a MailingCreate or MailingDelete, etc.).

3. Round & Sequence deposit

This chapter starts with an overview of what a Round & Sequence deposit is, followed by an overview of how data exchange for this product should be implemented and at the end, by an overview of the sequence reference.

The information about the barcodes are given in the next chapter.

3.1. Introduction

Round & Sequence is used only for "Large" format. Basically, a Round & Sequence deposit is the same than a Mail ID deposit as for both deposits address data will be provided by the customer, but for Round & Sequence deposit, in addition to printing a barcode, a sequence reference will be printed on the envelope. This sequence is provided by bpost in the response file, and will allow the customer to pre-sort the deposit following these sequence references.

The system works in the following way:

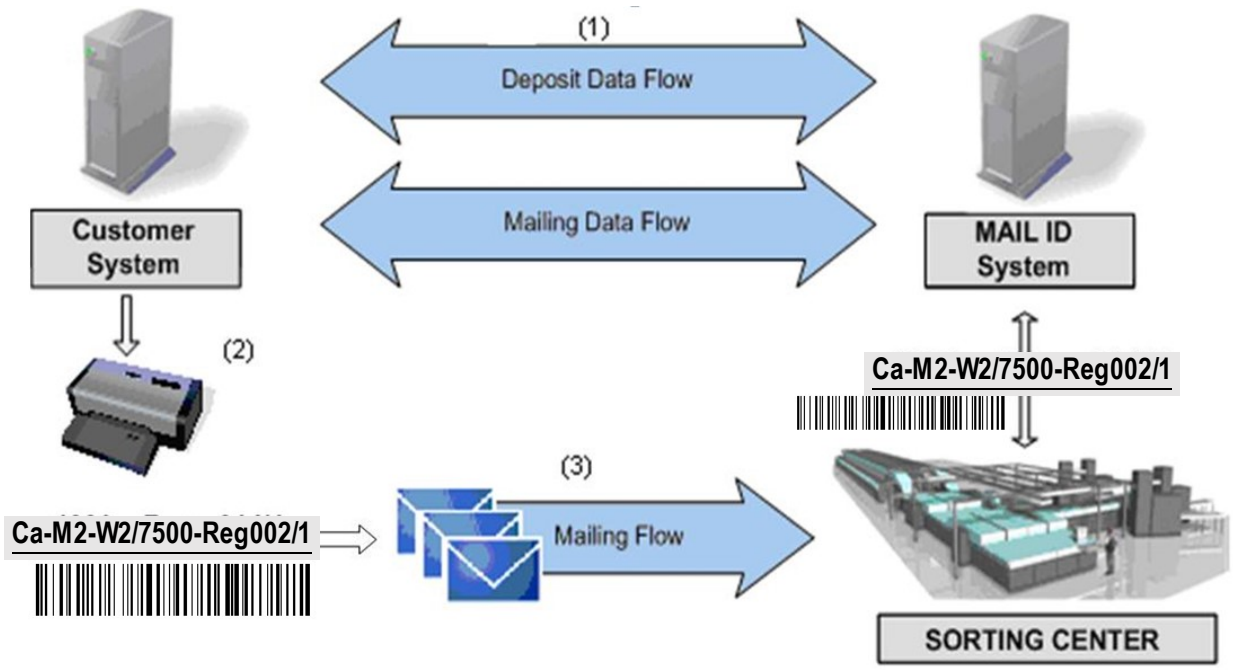


Figure 11: Round & Sequence Flows Schema

3.2. Data exchange options

The data exchange options for the Round & Sequence deposits are the same than those described in MAIL ID chapter, point "Data Exchange Options", given that 'MAIL ID deposits' and 'Round & Sequence deposits' need the same file flow.

However it is important to keep in mind that:

- Barcodes are needed for each address.
- The type of treatment needed must be specified in the FileInfo tag of the Mailing Request file as "RS3" (for Round & Sequence deposit), and not as "MID2" (for MAIL ID deposit). There is in this case an extra attribute at the Format tag, the "sortingMode" attribute, with two possible values : "PO" and "CU". This attribute determines the order of the mail pieces in the Response file sent by bpost. If the sortingMode is set at "PO" (for "Print Order") by the customer, the mail pieces are ordered in the Response file following the print order requested for the presorting. If the sortingMode is set at "CU" (for "Customer way"), the mail pieces are ordered following the same order than in the Request file.
- For using "Round & Sequence", the customer needs to overrun a specific certification.

3.3. Sequence reference

General information

The sorting information is based on MAIL ID technology. However, the uploaded MAIL ID file (Request file) contains tags (**Format** and **FileInfo**) with other data than for the classic Mail ID (see above: "Large" in place of "Small" and "RS3" in place of "MID2"). Refer to the subchapter "MailingCreate tag" of the chapter "Mailing files syntax" to see the syntax of the Request file.

A response will be sent by bpost with, for each mail piece, information about the distribution order and information to ease the printing and conditioning of the mailing. Here is an example of the information than we can find for a mail piece in a XML Response file :

```
<Item prtOrder="3" seq="3" fieldToPrint1="La-M3-W5" fieldToPrint2="4020-Res-173"
fieldToPrint3="549" icti="Begin_End" izon="Begin_End" imac="Begin_End" iwav="Begin_End"
ioff="Begin_End"/>
```

- The two first fields give the particular order of this mail piece in this mailing file, with the two types of order previously discussed: "Item prtOrder" gives the order requested for the presorting, "seq" gives the mail piece order in the Request file. These fields should not be printed on the mail pieces ;
- The four following fields ("fieldToPrint1", "fieldToPrint2", "fieldToPrint3" and "orgInfo") give the pre-sorting information to print on the mail pieces (see the structure below). The field "orgInfo" is not always present in the feedback sent by bpost ;
- The four last fields give the conditioning information. There are present to mention when a particular mail piece is the first and/or the last one for respectively a sorting center ("icti"), a machine ("imac"), a wave ("iwav") or a distribution office ("ioff"). Their possible values are "Begin", "End" and "Begin_end" (the last is possible when the mail piece is the only one for this sorting level). These fields should not be printed on the mail pieces.

Structure

Below, an example of structure of the presorting information, which must be printed on the envelope:

A	b	-	M	1	-	W	1	/	2	0	0	0	-	R	e	g	-	0	7	5	/	1	3	9					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Legend:

1 - 8 : 8 digits for fieldToPrint1 (or for the mention "Overflow" when no office was found);

9: "/" to add to separate the fieldToPrint 1 and filedToPrint2;

10 - 21 : 12 digits for fieldToPrint2;

22 : '/' to add to separate the fieldToPrint2 and filedToPrint3;

23 - 27: up to 5 digits for the sequence of the postal destination in the round;

28 - 30 : 3 space characters to futur use.

Examples:

G	a	-	M	3	-	W	5	/	9	0	6	0	-	R	e	s	-	1	4	7	/	4	9	0					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

➔ Ga-M3-W5/9060-Res-147/490

L	a	-	M	5	-	W	1	/	0	2	9	9	-	N	o	-	R	t	e		/	9	9	9	9	9			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

➔ La-M5-W1/0299-No-Rte/99999

Printing guidelines

When printing the sequence reference, take into account that:

- the reference must be placed at the right side and above the address;
- the reference must be printed in bold and/or underlined;
- the minimal fontsize is identical to the one of the address;
- there is a blank line between the sequence reference and the address.

Example:

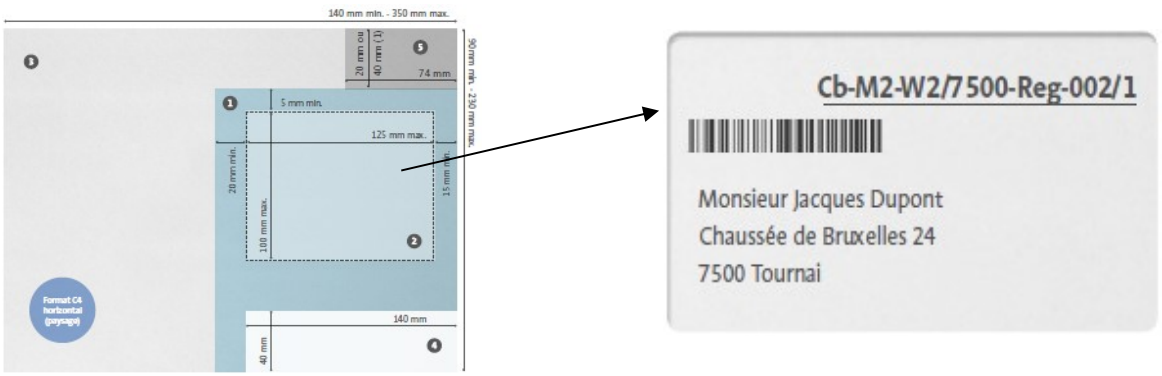


Figure 12: Example of printing of a Round & Sequence reference

For additional information on printing guidelines, the MassPost user guide can be consulted (www.bpost.be/masspost). Do not hesitate to contact bpost if there is any issues.

4. Barcode

This section gives the needed information on the barcode to print on the mail pieces for MAIL ID and Round & Sequence deposits.

In subsection 1, general information are given. In subsection 2, there are specific information about the barcode structure, and finally printing guidelines are depicted in subsection 3.

4.1. General barcode information

The barcode used by bpost for the MAIL ID system is a Code 128 barcode. Note that Code EAN 128 is not supported.

Code 128 is a very effective, high-density symbology, which permits the encoding of alphanumeric data. The symbology includes a checksum digit for verification, and the barcode may also be verified character-by-character by verifying the parity of each data byte. This symbology has been widely implemented in many applications where a relatively large amount of data must be encoded in a relatively small amount of space. Its specific structure also allows numeric data to be encoded at, effectively, double-density.

Refer to Part IV, Annexes, for more background information on constructing and representing the 128 barcode.

4.2. barcode structure

This section describes the specific content and structure of the barcode. The standard fields such as start, stop and checksum characters are not repeated.

[startcode]	JJBEA	12	12345	12345678901	[checkdigit]	[endcode]
G	A	B	C	D	F	G

Table 8: MAIL ID barcode structure

License Plate Identifier[20](A)

Indicates that the issuing agency is a Universal Postal Union (UPU) member.

This field is fixed and always contains the letter **J**

[20] Capital letters required

UPU header[21](A)

Indicates the postal organization for which this barcode is used.

This field is fixed and always contains JBEA.

Note Subset switch (optional) : Optionally a Code_C (Code 128 value 99) can be entered between the UPU Header field (A field) and the FCC field (B field). By doing so subset C will be used for all the numeric characters that follow. This will give the most compact barcode.

The MAIL ID number (B + C + D)

This number consists of three (consecutive) fields in the barcode:

- The format control code field (B)
- Customer identification (C)
- Mail piece number (D)

The MAIL ID number has to be unique over a period of 30 days at least

The Format Control Code[22] (FCC) (B)

Identifies the barcode's format: the particular configuration of fields and bars that applies to the whole.

Begin with 1 if customer generates the barcode, by 9 if it is generated by bpost. The second digit gives the number of digits of the mail piece number:

- A barcode with a 7 digits mail number receives FCC value 10 or 90.
- A barcode with a 9 digits mail number receives FCC value 11 or 91.
- A barcode with a 11 digits mail number receives FCC value 12 or 92.

Example

121234507256000001 => generated by the customer

921234507256000001 => generated by bpost

Customer identification[23] (C)

The CustomerBarcodeID identifies the party responsible for the creation of the barcodes and the mail pieces. The CustomerBarcodeID has 5 digits and will be assigned to each customer who enters the MAIL ID program

Mail piece number[24] (D)

This number is the core of the barcode. The customer is free to organize the mail number as he wishes. Within the mail number, ranges can be reserved for applications.

---

[21] Capital letters required

[22] Numeric characters required

[23] Numeric characters required

[24] Numeric characters required

Overview of current MAIL ID barcodes

Standard 1: 7-digit mail number

Field Description	Field Type	Code 128 Value	Value	N° char
Start Character		Start A = 103 Start B = 104		1
License Plate ID	alphabetic		J	1
UPU header	alphabetic		JBEA	4
Code C		99*		1
FCC	alphabetic		10	2
Barcode ID	alphabetic		12345	5
MID Number	alphabetic		1234567	7
Checksum	alphabetic		1	1
Stop Charcaters				1

Table 9: 7-Digit Mail Number

Standard 2: 9-digit mail number

Field Description	Field Type	Code 128 Value	Value	N° char
Start Character		Start A = 103 Start B = 104		1
License Plate ID	alphabetic		J	1
UPU header	alphabetic		JBEA	4
Code C		99*		1
FCC	alphabetic		11	2
Barcode ID	alphabetic		12345	5
MID Number	alphabetic		123456789	9
Checksum	alphabetic		1	1
Stop Charcaters				1

Table 10: 9-Digit Mail Number

Standard 3: 11-digit mail number

Field Description	Field Type	Code 128 Value	Value	N° char
Start Character		Start A = 103 Start B = 104		1
License Plate ID	alphabetic		J	1
UPU header	alphabetic		JBEA	4
Code C		99*		1
FCC	alphabetic		12	2
Barcode ID	alphabetic		12345	5
MID Number	alphabetic		12345678901	11
Checksum	alphabetic		1	1
Stop Charcaters				1

Table 11: 11-Digit Mail Number

* Optional

bpost business

4.3. Printin barcode

Barcode Printer

A Code 128 barcode must be printed in black on whit or on Pantone colours (see Masspost Guide) on each mail piece. The printer(s) must be able to print the barcode according to all the specifications defined within this document. Note: dot matrix printers do not provide sufficient quality to be readable by bpost's equipment.

Constraint factors

To ensure that a barcode can be read by bpost's sorting equipment, certain printing and placement constraints must be met. There is maybe some room for variance within these constraints, but it is important for software developers and customers to know the parameters tolerated.

The following constraint factors must be considered:

- Dimensions
- Skew tolerance
- Reflectance
- Quiet Zones
- Placement of barcodes
- Text Representation of the Barcode
- Measurement of barcodes in final form

Dimensions

The dimensions and spacing of individual bars within a barcode are important, as any major discrepancies can cause a barcode to be invalidated by the sorting equipment.

The minimum element size is the most important dimension.

	Minimum (mm)	Maximum (mm)
Minimum element size	0,25 mm	0,34 mm
Bar code Height	6 mm	12 mm
Standard 1 – 7 digits		
Bar code Length symbol set A or B	61 mm	83 mm
Bar code Length symbol set A or B and C	44,5 mm	60,5 mm
Standard 2 – 9 digits		
Bar code Length symbol set A or B	66,5 mm	90,4 mm
Bar code Length symbol set A or B and C	47,3 mm	64,3 mm
Standard 3 – 11 digits		
Bar code Length symbol set A or B	72 mm	97,9 mm
Bar code Length symbol set A or B and C	50 mm	68 mm

Table 12: Dimensions of barcodes

Skew tolerance

When barcodes are printed, the printer sometimes skews them. The sorting equipment tolerates a certain amount of skew. Customers must be able to recognize the limits of any skew.

Code skew

A code skew is when the entire barcode is skewed in relation to the bottom edge of a piece of mail. Code skews of less than +/- 5 degrees horizontal can still be read.

Reflectance

'Reflectance' is the degree to which light reflects from a surface. Barcode reader devices are sensitive to the reflectance of the following:

- The printed barcode;
- The space around the barcode
- The material through which the barcode is scanned

Spectral range

Barcode reader devices operate within the spectral range of 400 to 650 nanometers. Within this range, the following measurements must be met[25]:

- maximum bar reflectance (Rb) is 25%;
- minimum space reflectance (Rs) is 50%;

[25] Also when the address and barcode are behind a plastic window

- the reflectance difference (MRD) must be greater than 50%, where MRD is defined as follows:

  MRD = Rs-Rb > 50%

- the Print Contrast Signal (PCS) must be greater than 0.75 where PCS is defined as follows:

  PCS = (Rs – Rb)/Rs > 0.75

Quiet zones

'Quiet Zones' are the minimum margin spaces around a barcode that must be kept blank (free of printing or other distractions) if the barcode is to be properly scanned. Barcodes require a Quiet Zone immediately above, below, and to the right and left of the barcode.

Distractions in a Quiet Zone

Any of the following constitute 'distractions' within a Quiet Zone, and may affect a barcode's ability to be scanned:

- Any printing or other ink or marks;
- Patterns or textured paper/substrate;
- Printing showing through from another page.

Dimensions of the Quiet Zone

The following minimum dimensions must be met for each barcode:

- The smallest allowable Quiet Zone on the left and right of a barcode is **5** mm;
- The smallest allowable Quiet Zone above and below the barcode is **2** mm.

Placement of Barcodes

Certain constraints apply to the location of a barcode on an envelope. These constraints apply to any letter. The location for barcodes is above the address, and the barcode must fall in the white zone in Figure 7 on the next page, in the zone of the destination address.

Orientation

The barcode must be printed parallel to the bottom (long) edge of the letter for small format. For large format mail, the MAIL ID can be either horizontal or vertical (+/- 5%).

As the figure below shows, barcodes must be printed within the following margins:

- No less than 30 mm from the bottom edge of the piece of mail;
- No less than 15mm from either side of the piece of mail.

Other placement rule

The following placement rule also applies:

- No part of the barcode must appear in the Canceling and Metering Zone (frankeerzone).

Text representation of the barcode (Optional)

Text representation of the barcode is optional but recommended. If printed, it should appear below the barcode in 8 point type or less.

Measurement of barcodes in final form

The print quality of a barcode can only be determined in its final form as it actually appears on the letter. The correct location and reflectance can only be determined once the barcode is viewed through envelope window material or plastic wrapping, as applicable.

Figure 13: Placement of barcode

In case of a MAIL ID viewed through the envelope window, the placement of the MAIL ID shall ensure that it is completely visible through the window in any position of the content inside the envelope.

5. OptiAddress

We will briefly describe OptiAddress in section 1, followed by an overview of data exchange for this product.Introduction

OptiAddress is a tool used for address validation. Authorized mailers can submit addresses to the MAIL ID system for verification, independently of any deposit.

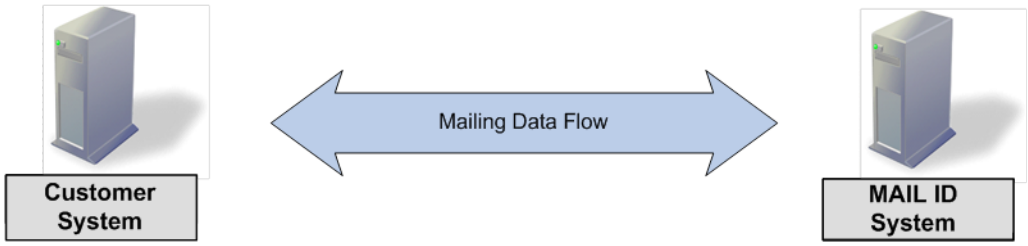


Figure 14: OptiAddress Flows Schema

1. The mailer transmits an address selection to the MAIL ID system of bpost.
2. The Mailing Check system processes the data.
3. The Mailing Check system produces detailed error feedback, i.e. not only the number of address records that are interpreted[26], but also suggestions for corrections.

5.2. Data exchange options

OptiAddress is based on the same platform as MAIL ID and uses the same electronic workflow and file structure. Hence, Mailing Check functionality uses the Mailing Files Syntax.

As described in Part I, data exchange for MAIL ID data (addresses) can only occur via structured file.

[26] Interpreted in this sense means that the given address can be matched to an existing postal address. This does not necessarily mean that the link between the addressee and the address is correct.

To use OptiAddress, the customer first needs to implement all requirements stated in Part I. Then, his ICT infrastructure must be adapted to enable the generation of structured files, whose syntax is detailed in Part III: File Syntax, Chapter Mailing Files Syntax, section Mailing Request file. To use the OptiAddress functionality, the MailingCheck action tag is needed. Refer to the MailingCheck subsection.

6. Sequence Diagrams

This part gives a more detailed overview of possible file flows between the customer and the systems of bpost.

6.1. Deposit only scenarios

Deposit (Auto Validate = N)

A simple deposit announcement, without using autovalidation.

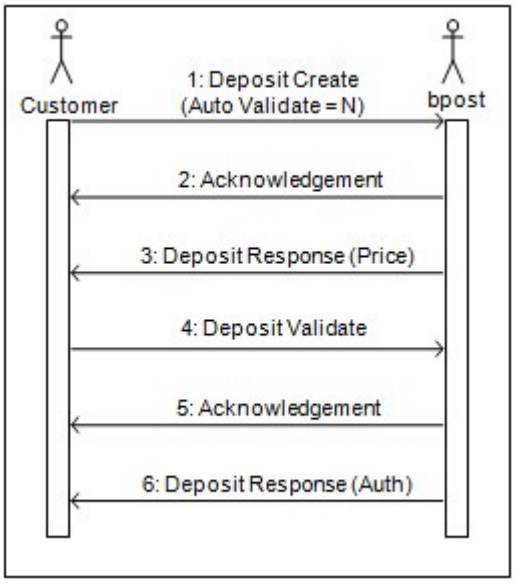


Figure 15: Deposit (Auto Validate = N)

This diagram shows an example of a typical file flow for a simple deposit. In the original deposit Request file, the autoValidate attribute is set to "N"[27]. Therefore the client still needs to send a second Deposit Request File with a depositValidate action, before he can do the physical deposit.

[27] The autovalidate attribute belongs to the DepositCreate/Deposit tag. Refer to file syntax for details.

Deposit (Auto Validate = Y)

A deposit announcement using autovalidation.

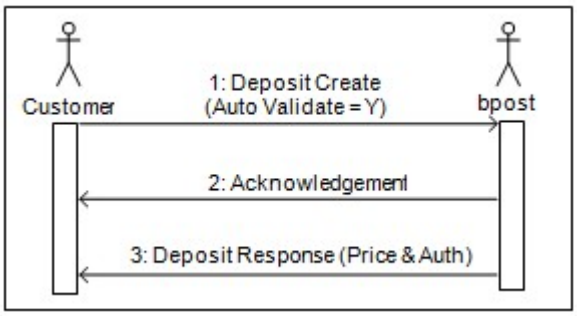


Figure 16: Deposit (Auto Validate = Y)

This diagram shows an example of a typical file flow for a simple deposit with autovalidation turned on. Hence, the client only needs to send one file.

Deposit with update

This figure illustrates a situation where the client updates a previously created deposit, e.g. because the number of mail pieces has changed.

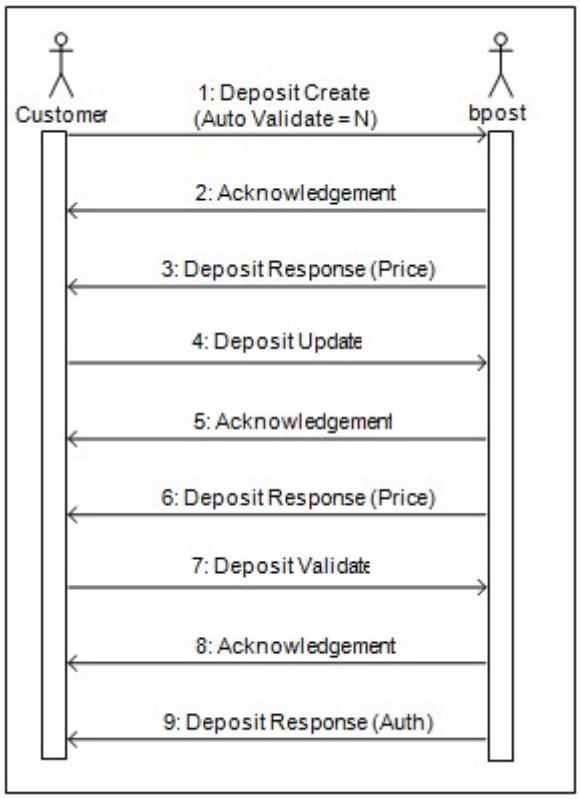


Figure 17: Deposit with Update

This graph illustrates the generic file flow Deposit Request file, followed by a deposit Acknowledgement File and finally a deposit Response File. Please note that the deposit Response File will include a new price, based upon the changes made with the depositUpdate action. After updating the deposit, the customer validates the deposit with the depositValidate action. Hence, the customer sends three Deposit Request Files; one with a depositCreate action, one with a depositUpdate action and finally one with a depositValidate action.

Deposit Delete

This figure illustrates a situation where the client deletes a previously created deposit, e.g. because the marketing department has cancelled the campaign upon which the mailing was based.

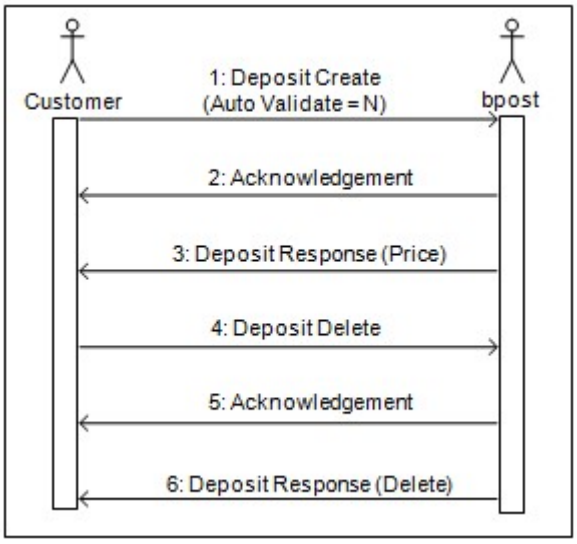


Figure 18: Deposit Delete

The deposit response contains the information to know if the deletion was successful.

In this case the customer sends 2 deposit Request Files, one with a depositCreate action and one with a depositDelete action.

6.2. Deposit Master scenarios

All deposit request actions can be realized though e-MassPost application (cfr e-MassPost Guide §5.1.2).

All deposit and mailing list requests can be uploaded via eMassPost application or via FTP.

Deposit with multiple mailing files

Consider the customer wants to send a mailing to 100.000 prospects with a new product offer, as part of a marketing campaign. Suppose he currently has a mailing list of 50.000 prospects, but has ordered new addresses from a market research bureau. He wants to use MAIL ID.

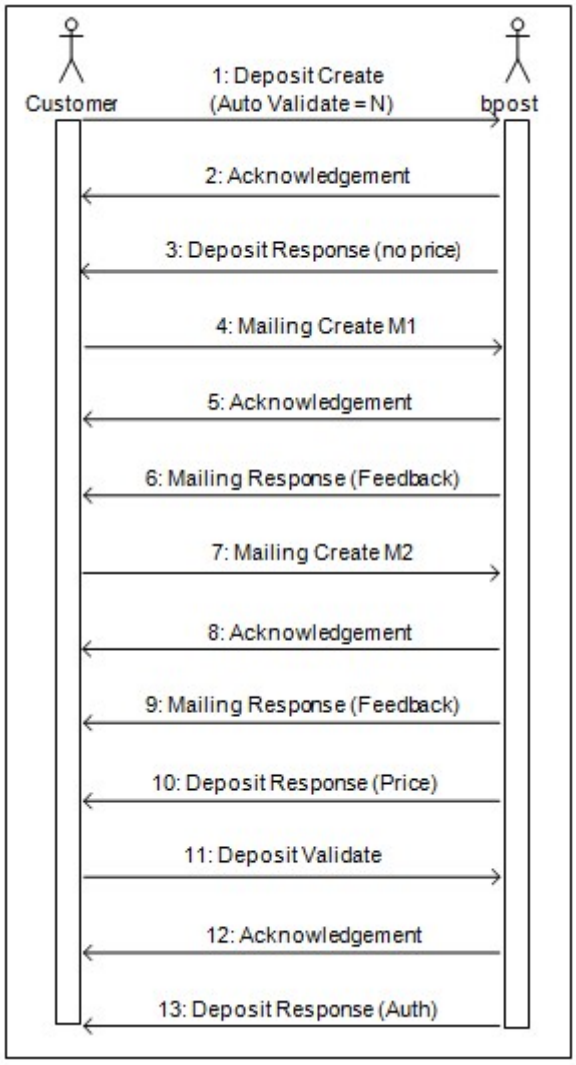


Figure 19: Deposit with multiple mailing files

As described above, in this case it makes sense to make the deposit master. Hence, the customer first sends a Deposit Request file with one depositCreate action. The customer waits until he received the Deposit Acknowledgement file and the Deposit Request file, before proceeding. The deposit Resonse file will not include a price quote, because the price quote will depend on the data quality of the addresses in the associated mailing files. Now the customer proceeds with sending the Mailing Request file with a mailingCreate action for the 50.000 addresses of prospects it already obtained. The customer receives an acknowledgement and response file. The latter contains feedback on the addresses supplied. When the customer has received the additional addresses, he creates a new Mailing Request file with a mailingCreate action and uploads it to bpost. Because the total number of mail pieces indicated in the deposit file is now equal or smaller than the total number of addresses from the mailing files, a new Deposit Response will be sent. This time the Deposit Response will include a price. The last three transfers are for validating the deposit. The customer will receive a final authorization code as feedback in the Deposit Response file.

Deposit Delete with multiple mailing files

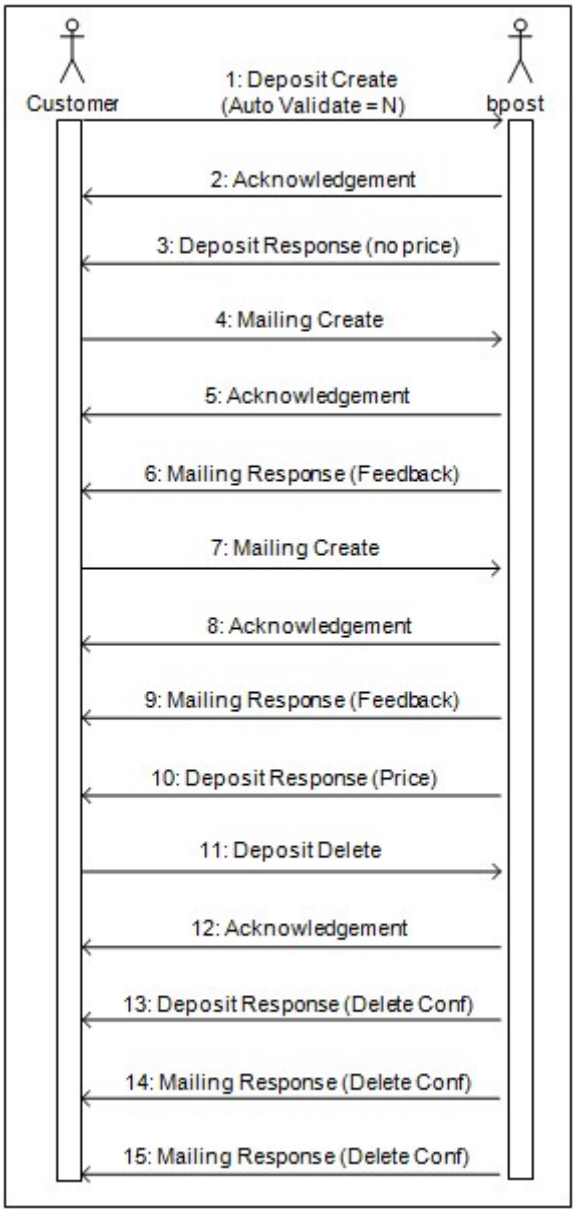


Figure 20: Deposit Delete with multiple mailing files

This example is the same as the previous one, but the customer decides in the end that the mailing should be cancelled. This is reflected in steps 11 to 15.

When the customer wants to cancel the mailing, he sends a Deposit Request File with a depositDelete action. This request will lead to the generation of a deposit acknowledgement file and a deposit response file. However, two additional files will be generated. As the deposit is master, the deletion of the master (deposit) will lead to the deletion of its children (MailingRequest 1 and MailingRequest 2). Hence, a mailing Response file will be generated for both request files, including the confirmation of the deletion.

Deposit create with mailing delete

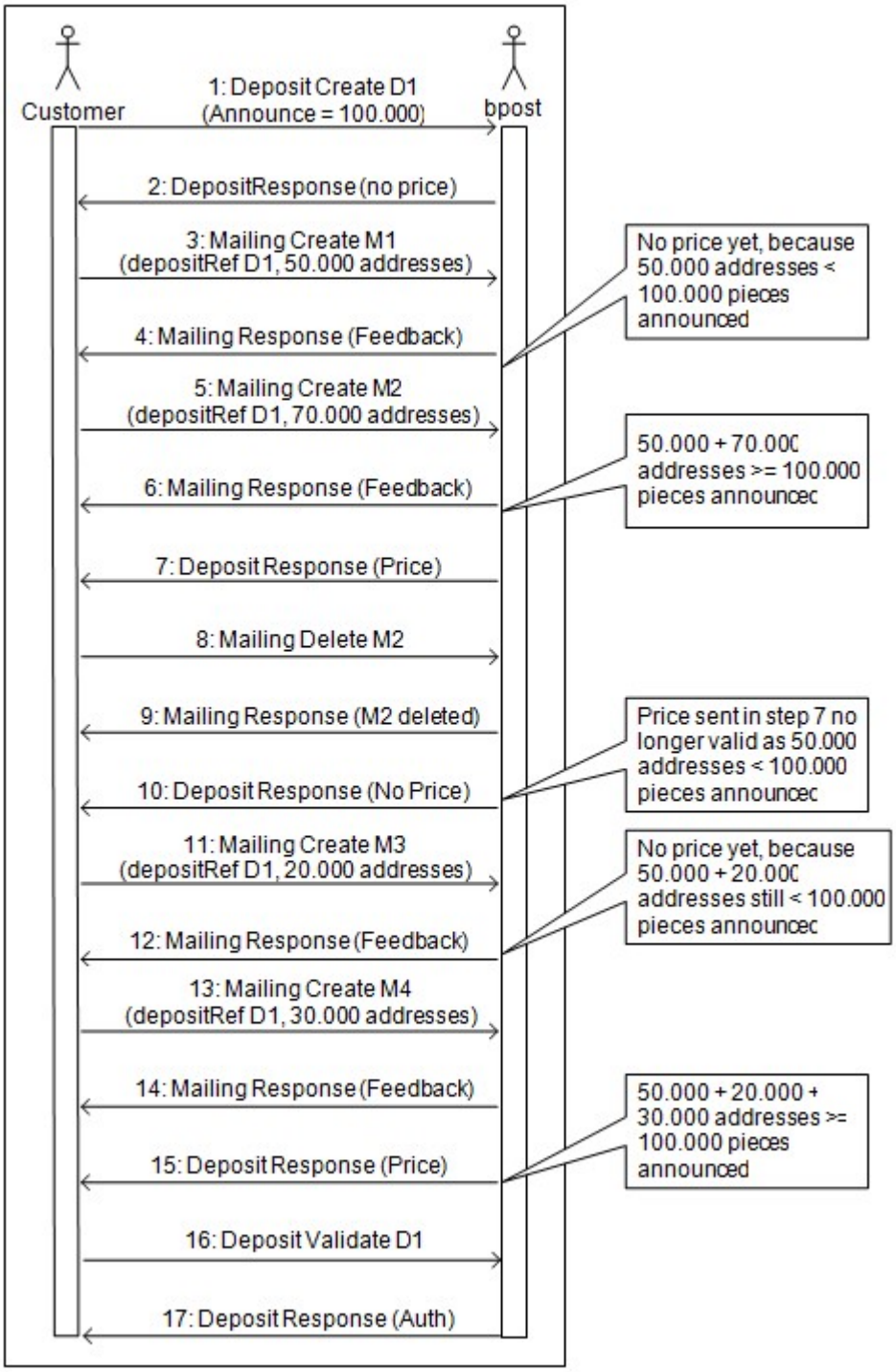


Figure 21: Deposit Response

This example is the same as the "Deposit with multiple mailing files", but this time the customer decides to change some addresses first linked to the deposit. The customer deletes then the corresponding mailing file (step 8). This will have as consequence that the previous given price is no longer valid, as there is no more enough addresses linked to the deposit. bpost will so send a

Deposit Response without price (step 10). The new price will be communicate in a new Deposit Response as soon as the customer links again enough addresses to the deposit (step 15).

6.3. Mailing file Master scenarios

Mailing file, one deposit (Auto Validate = N)

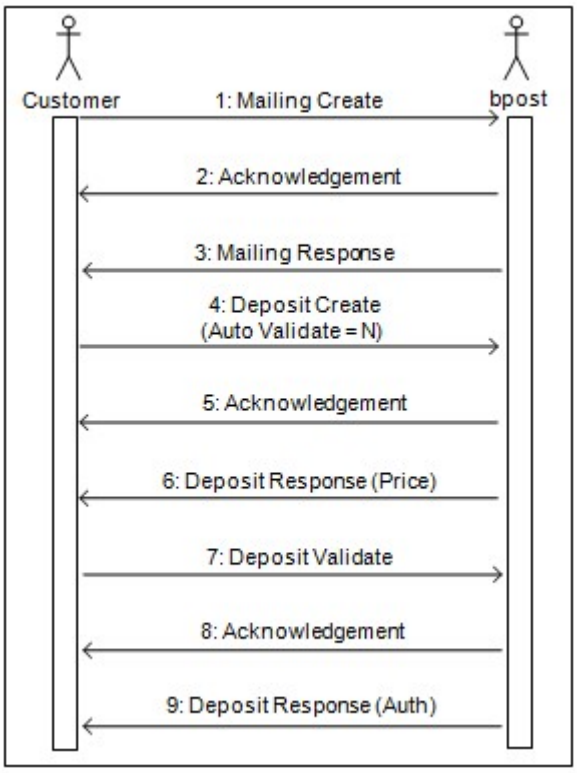


Figure 22: Mailing file, one deposit (Auto Validate = N)

In this example the customer uses the mailing file as the master and develops the mailing file before developing the deposit file. The other data flows are very straigthforward, as they just follow the generic file flow, i.e. Request, Acknowledge Response. The customer sends 3 files: the Mailing Request File with a mailingCreate action, the Deposit Request File with a depositCreate action and finally he validates the deposit in a Deposit Request File with depositValidate action. Each of these will lead to the generation of the corresponding Acknowledgement and Response files.

Mailing file, one deposit (Auto Validate = Y)

This is an example that is very similar to the one discussed before. The only difference here is that the user turned on the autoValidate function[28].

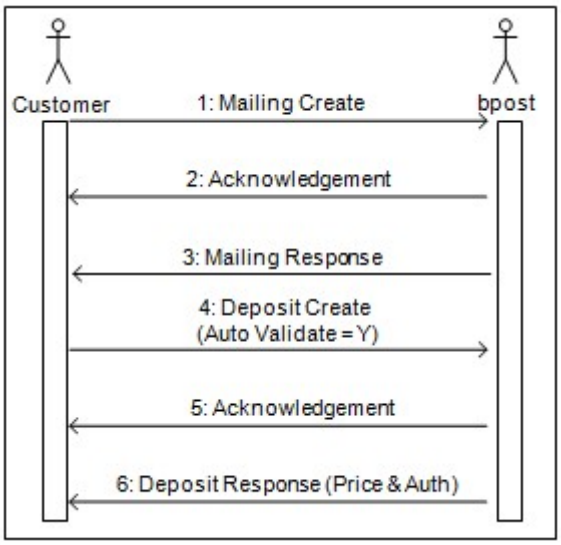


Figure 23: Mailing file, one deposit (Auto Validate = Y)

This file flow is almost identical as the one above, except that the user does not need to send another Deposit Request File with the depositValidate action. This brings the number of files transfered from nine to six.

Note : Number of mailing list items ≥ numbers of deposits create items items so that auto validate works.

Mailing file, multiple deposits (Auto Validate = N)

In this file flow, there are multiple deposit files for one mailing file. Consider a company has two product groups and an existing client base of 100.000 customers. This client base of "single product users" can be divided in customers using a product from group 1 and customer using a product from group 2. Now, assume the company wants to create a specific mailing tailored to upsell the product they are currently not using. Because there is a promo planned for Product 1 next week, the company wante the mailing for "Product 2 only users" to appear at the end of this week. The promo for Product 2 is planned for 2 weeks from today, so they want to lauch that mailing at the end of next week.

[28] Remember from the file syntax, this is an attribute from the deposit tag from the DepositCreate action tag. Refer to Part III, file syntax for more details.

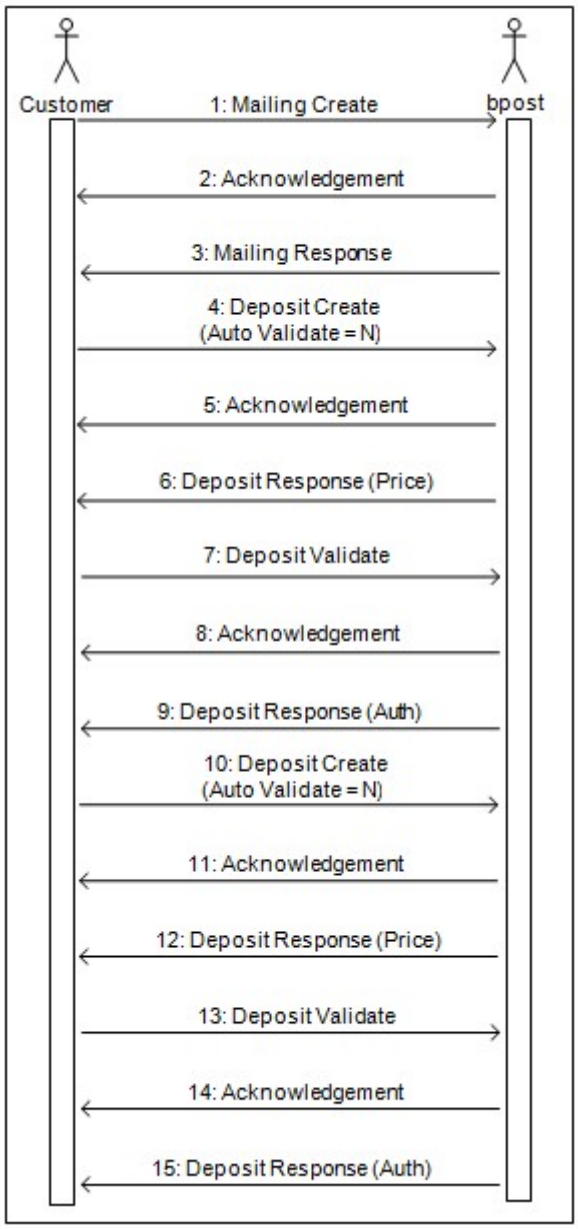


Figure 24: Mailing file, multiple deposits (Auto Validate = N)

In this case, The company would opt to create the mailing first and send the different deposit files afterwards. As described before, each time a file is sent, the system will generate the corresponding Acknowledgement and Response files. As in this case the autovalidate option is not used, each depositCreate action must be validated with a depositValidate action. Hence, five Request files are sent: one Mailing Request file with a mailingCreate action, and four Deposit Request files, two with a depositCreate action and 2 with a depositValidate action[29].

[29] Please note that it would equally be possible to include the depositValidate for the first depositCreate in the deposit Request file with the second DepositCreate, as in one Request file, all possible actions possible within the file type can be combined.

Mailing file, multiple deposits (Auto Validate = Y)

Consider the same example of above. The only difference is that the customer is sure about the deposit parameters, and therefore uses the autoValidate option.

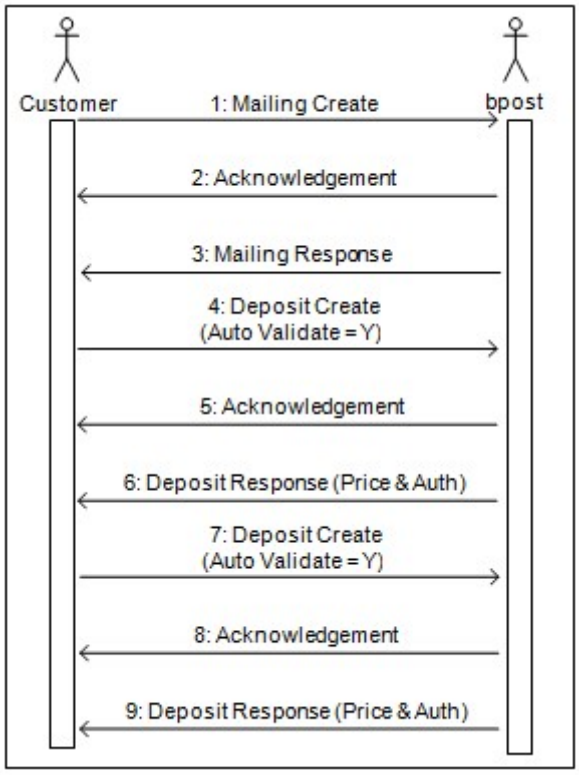


Figure 25: Mailing file, multiple deposits (Auto Validate = Y)

This file flow is almost identical as the flow of the example above. The only difference is that there are only 9 file transfers instead of 15, because the autovalidate option is used.

Mailing file Delete

Consider an example like the previous: a company has two product groups and an existing client base of 100.000 customers. This client base of "single product users" can be divided in customers using a product from group 1 and customer using a product from group 2. Now, assume the company wants to create a specific mailing tailored to upsell the product they are currently not using. Because there is a TV promo planned for Product 1 next week, the company wants the mailing for "Product 2 only users" to appear at the end of this week. The TV promo for Product 2 is planned for 2 weeks from today, so they want to lauch that mailing at the end of next week.

Now, imagine a few days before the mailing is going to be launched, it turns out there is a problem with the TV promo. Because of some problems with the planning, it turns out that the TV promo will have to be delayed with a couple of weeks. So it was decided to let the second mailing pass, and delete the first one. However, one week later, the decision to go on with the program is reconsidered, because the priorities have changed and different objectives need to be realized in short term.

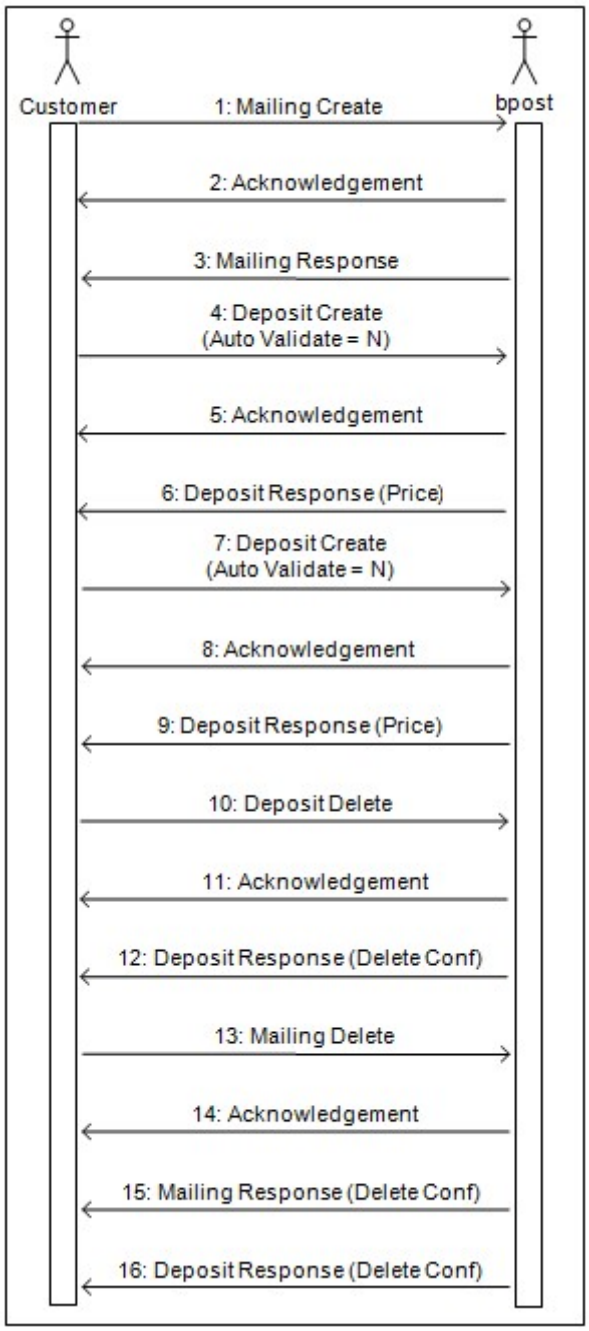


Figure 26: Mailing file Delete

In this case, the file flow would look like the one in the figure above.

It makes sense to make the Mailing Request file master in this situation, given that there is one address selection, but two different deposits. Hence, the customer first sends a Mailing Request file with a mailingCreate action. As always, he will receive a Mailing Acknowledgment file and a Mailing Response file. When he received these files, the data has been processed by bpost, so he can send the Deposit Request file with the DepositCreate action. This way he creates the Deposit for the first

mailing. In the same way a deposit is created for the second mailing[30]. Because of problems with the planning, the customer deletes this deposit.[31] The second part of the program continues as planned. After a few days, it turns out the rest of the program will be cancelled. As a consequence, a new Mailing Request file with a MailingDelete action is sent. The system processes the data and sends not only a Mailing Response file, but also a Deposit Response file, because deleting the master, will automatically implies a deletion of its children.

6.4. OptiAddress

If the customer sees that his database contains quite a lot of erroneous addresses[32], he can decide to use the OptiAddress functionality.

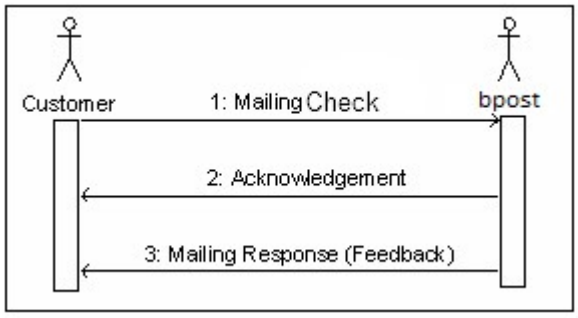


Figure 27: Mailing Check

The file flow in this case is very simple. As described in part I, all data exchanges follow the following order: Request (client), Acknowledgement, Response (bpost).

For OptiAddress, the customer only needs to send a Mailing Request File with a MailingCheck action. The system generates a Mailing Acknowledgement file, upon reception of the Request file, and a Mailing Response file, once the data has been processed. The latter will include feedback on the number of addresses that were incorrect, as well as propositions of correction (when possible) for each erroneous address.

[30] It is perfectly possible to perform both actions in one file, if all specifications of the two mailings are known at that time. Indeed, in one file syntax all actions belonging to that file syntax can be combined.

[31] Because the plan at that time is to go on with the second deposit, the master is not deleted (Mailing), but only the child (Deposit). If at that time it would have been clear the whole program would be terminated, a Mailing Request File with MailingDelete action would have been send, automatically deleting the related Deposits.

[32] Remember that, in this context, erroneous means not representing a physical address at which mail can be delivered.

Part III: File Syntax

1. General information about File Syntax

1.1. Identifiers used in files

The following identifiers may be required within the files discussed underneath. Hence a definition:

- **customerID**: a numeric identifier (8 digits max) provided by bpost identifying the certified customer. This identifier is used in the filename and in all Header sections.
- **sender**: a numeric identifier (8 digits max) provided by bpost identifying the sender of the file. It is often the same as the customerID, and this identifier is used in all the Context sections.
- **accountID**: a numeric identifier (8 digits max) provided by bpost identifying the Postal Business Contract of the customer for whom the deposit is made.
- **billTo**: a numeric identifier (8 digits max) provided by bpost identifying the party that will be invoiced.
- **depositor**: alphabetic value provided by bpost identifying the party making the physical deposit.
- **PBC**: Postal Business Contract = accountID = e-Masspost account number

There is also an identifier used in the construction of the barcodes. This is called the **customerBarcodeID**. It is a 5-digit code (left-padded with zeros if necessary) provided by bpost identifying the party responsible for the creation of the barcodes and mail pieces (this ID is so linked to the customerID). This code will appear in all MAIL ID barcodes.

Some of these identifiers could be the same. However, when divisions, subsidiaries and/or sub-contractors are involved, these set of identifiers will enable proper identification of all the parties involved.

All identifiers are regrouped in the "Customer reference Data" select that will be provided by the technical specialist of bpost.

1.2. Non-supported characters

Some characters are not supported in the XML or the TXT file syntax, while other are substituted by bpost systems or must be escaped by the customer with particular syntax. The list of non-supported, substituted or escaped characters can be found in the table underneath. The exhaustive list of characters, with also all the supported characters, is available in the annexes.

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
NUL	null	N	N	
SOH	start of heading	N	N	
STX	start of text	N	N	
ETX	end of text	N	N	
EOT	end of transmission	N	N	
ENQ	enquiry	N	N	
ACK	acknowledge	N	N	
BEL	bell	N	N	
BS	backpace	N	N	
VT	vertical tab	N	N	
FF, NP	form feed, new page	N	N	
SO	shift out	N	N	
SI	shift in	N	N	
DLE	data link escape	N	N	
DC1	device control 1	N	N	
DC2	device control 2	N	N	
DC3	device control 3	N	N	
DC4	device control 4	N	N	
NAK	negative acknowledge	N	N	
SYN	synchonous idle	N	N	
ETB	end of transmission block	N	N	
CAN	cancel	N	N	
EM	end of medium	N	N	

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
SUB	substitute	N	N	
ESC	escape	N	N	
FS	file separator	N	N	
GS	group separator	N	N	
RS	record separator	N	N	
US	unit separator	N	N	
"	double quotation mark	Substitution	To escape	TXT: Substitution: {"} -> {'} XML: Should be escaped as {"}
&	ampersand	Y	To escape	XML: Should be escaped as {&}
'	apostrophe, single quote mark	Y	To escape	XML: Should be escaped as {'}
;	semicolon	Substitution	Y	TXT: Substitution: {;} -> {,}
<	less-than sign	Y	To escape	XML: Should be escaped as {<}
>	greater-than sign	Y	To escape	XML: Should be escaped as {>}
|	vertical bar	To escape	Y	TXT: Should be escaped as \\\|

Table 13: List of non-supported characters

2. Deposit Files Syntax

Remember from Part I, that data exchange follows a generic file flow. In this section we will define the file structure for each of the three files (Request, Acknowledgement, and Response) in the case of a deposit announcement done via structured files[33].

In describing the structure of each Deposit file, we will start with a global overview of the general structure. Afterwards, the items from this general structure will be further elaborated[34].

This is followed by the description of the different elements of the file in XML format and the corresponding TXT format[35].

2.1. Deposit Request File

Global structure

The structure below is a high-level graphical representation of the XML structure of the Deposit Request file.

The root tag for the Deposit Request file is <DepositRequest>

Each tag has attributes associated to it. These are not available in the graphical representation below, but will be dealt with in the paragraphs later on. For each level one tag, a detailed description of all underlying tags will be given, including their attributes.

[33] Remember from Part II that a deposit announcement can also occurred via a Webform on the e-MassPost website. This case is not described here, as there is no structured files, and so no file syntax is needed.

[34] Some file structure descriptions may not follow this logic rigourously.

[35] Recall from Part I that the discussion starts from an XML point of view. Applying this to TXT format is relatively simple. More information on how to do this can be found later in this guide.

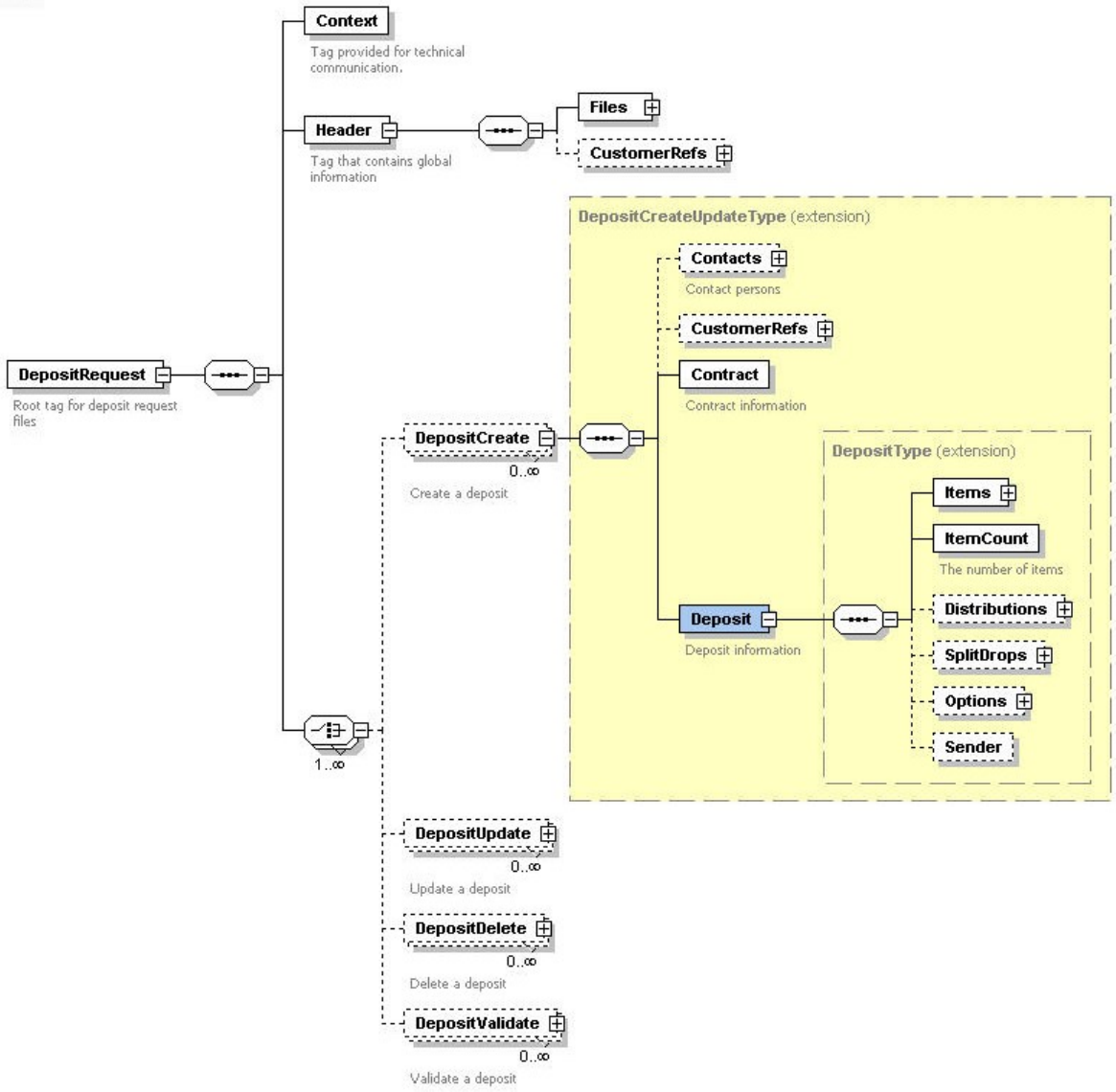


Figure 28: DepositRequest file structure

XML structure

The graphical representation of the DepositRequest file structure can be transformed to a table. In the table underneath, each column represents a level down the graphical tree from the figure above.

Note: The tags in italic are used for aggregation and have no correspondent tag in the TXT format.

Tag Level						
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**
Context						
Header						
	Files					
		RequestProps				

Tag Level						
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**
		ResponseProps				
	CustomerRefs					
		CustomerRef (#N)				
DepositCreate (#N)						
	Contacts					
		Contact (#N)				
	CustomerRefs					
		CustomerRef (#N)				
	Contract					
	Deposit					
		Items				
			Item (#N)			
				Characteristics		
					Characteristic (#N)	
				Quantities		
					Quantity (#N)	
					Quantity (g)	
				Prepayments		
					Prepayment (#N)	
		ItemCount				
		Distributions				
			Distribution (#N)			
		Options				
			Option (#N)			
				OptionQuantities		
					OptionQuantity (#N)	
		Sender				
DepositUpdate (#N)						
	… Same as DepositCreate					
DepositDelete (#N)						
	Contacts					
		Contact (#N)				
	CustomerRefs					

Tag Level						
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**
		CustomerRef (#N)				
DepositValidate (#N)						
	Contacts					
		Contact (#N)				
	CustomerRefs					
		CustomerRef (#N)				

Table 14: DepositRequest - XML structure

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the following global TXT structure for the Request File:

Context
Header
RequestProps
ResponseProps
CustomerRef
DepositCreate
Contact
CustomerRef
Contract
Deposit
Item
Characteristic
Quantity(PCE)
Quantity (g/PCE)
Prepayment
ItemCount
Option
OptionQuantity
DepositUpdate
Contact
CustomerRef
Contract
Deposit
Item
Characteristic
Quantity(PCE)
Quantity (g/PCE)
Prepayment
ItemCount
Option
OptionQuantity
DepositDelete
Contact
CustomerRef
DepositValidate
Contact
CustomerRef

Table 15: DepositRequest - TXT structure

Context tag

The Context tag is necessary for proper processing by the communication servers of bpost.

Important to note is that, if the system detects one or more errors in this tag, the entire request (i.e.: the entire file) will be rejected and no action (action tag) will be processed.

XML structure

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
Context	requestName	A constant identifying the request	Must be 'DepositRequest'	Yes	String	-
	dataset	Required by the File Handling System	Must be 'M004_MPA'	Yes	String	-
	sender	The PRS-ID of the PBC of the sender	Must match the customer identifier in the file name (see chapter "File Naming Convention")	Yes	Num	8
	receiver	Required by the File Handling System	Must be 'EMP'	Yes	String	-
	version	The file version	Must match the file version in the file name (see chapter "File Naming Convention")	Yes	String	4

Table 16: DepositRequest Context tag - XML structure

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the Context tag:

Context|*requestName*|*dataset*|*sender*|*receiver*|*version*

Table 17: DepositRequest Context tag - TXT structure

Header tag

The Header tag is used for general information.

XML structure

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
Header	customerId	The PRS-ID of the PBC of the sender	Must match the customer identifier in the file name (see chapter "File Naming Conventions")	Yes	Number	8	
	accountId	Postal Business Contract of the customer	provided by bpost	Yes	Number	8	
	Mode	A one character field	P = Production C = Certification T = Test[36]	Yes	String	1	
Files				Yes			
Files/ RequestProps	customerFileRef		Needs to match the 10 N's of the original file name	Yes	String	10 (strictly)	
Files/ ResponseProps	format	Format type for the Response file	XML or TXT. If omitted, the Response file will use the same file type as the Request file.	No	String	3	Same as Request file
	compressed	Boolean value specifying if the response should be compressed or not	Y or N. If omitted, the Response file will be compressed only if the Request file was compressed.	No	Boolean	1	Same as Request file

[36] Production: this mode can only be used after successful completion of the certification program.

Test: this mode can be used for debugging application development. Treatment is limited to 200 addresses.

Certification: this mode is to be used during the certification phase. Treatment is limited to 2000 addresses.

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
	encrypted	Boolean value specifying if the response should be encrypted or not	Possible values: N Encryption mode not yet supported	No	Boolean	1	N
	transmissionMode	Transmission mode	Possible values: HTTP(s), FTP(s) If omitted, the Response file will use the same mode as the Request file	No	String	5	Same as Request file
CustomerRefs				No			
CustomerRefs /CustomerRef (#N)	key	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	50	
	value	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	250	

Table 18: DepositRequest Header tag - XML structure

Note that the CustomerRef tag is reserved for the customer's own usage. bpost ignores the values supplied, and simply returns them in the Response file.

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the header tag:

Header|*customerId*|*accountId*|*mode*
RequestProps| *customerFileRef*
ResponseProps|*format*|*compressed*|*encrypted*|*transmissionMode*
CustomerRef|*key*|*value*

Table 19: DepositRequest Header tag - TXT structure

DepositCreate and DepositUpdate tags

Several actions are allowed in one DepositRequest file, and several instances of each action are allowed as well. A DepositCreate action is used in a DepositRequest file to create a new deposit announcement.

A DepositUpdate action is used in a DepositRequest file to update a Deposit (either in a DepositCreate action earlier in the same DepositRequest file or previously transmitted). It is not allowed to update a deposit that has been validated.

When a DepositUpdate action is received, all the current deposit data will be purged from the system and replaced by the content in the DepositUpdate action. Therefore, ALL the deposit data must be provided and not only the changes.

An example will clarify this:

Consider a DepositCreate was sent in a previous Deposit Request file for deposit D1 with options A,B,C. Now, we want to update the data, because option B changed in D. The files sent in this respect are the following:

- *<DepositCreate> deposit D1 with Options (A,B,C) → D1 created containing options (A,B,C)*
- *<DepositUpdate> deposit D1 with Options (A,C,D) → D1 updated containing options (A,C,D)… (B is purged)*

XML structure

The following table describes the structure of the DepositCreate tag.

The structure for the DepositUpdate tag is identical, except for the DepositCreate action tag which is replaced by the DepositUpdate tag.

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
DepositCreate	seq	A sequence number enabling identification of the action within the file	Needs to be unique across all actions within the file	Yes	Num	8	
	deposit Identifier	A unique reference per PBC identifying the deposit		Yes	String	20	
	deposit Identifier Type	Type of depositIdentifier	depositRef or tmpDepositNr	No[37]	String	20	deposit Ref
	mailingRef	If empty, the deposit is the master		No	String	20	Empty
DepositCreate /*Contacts*				No			
DepositCreate /Contacts /Contact (#N)	seq	A sequence number uniquely identifying the contact within the DepositCreate action	Needs to be unique within the action	Yes	Num	8	
	firstName	First name of the contact person		No	String	50	

[37] Yes if Deposit is Master

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
	lastName	Last name of the contact person		No	String	50	
	email	Email of the contact		Yes	String	100	
	lang	A 2 characters constant indicating the mother language of the contact	'fr' or 'nl'	Yes	String	2	
	phone	Phone number of the contact person		No	String	50	
	fax	Fax number of the contact person		No	String	50	
	mobile	Mobile phone number of the contact person		No	String	50	
DepositCreate /*CustomerRefs*				No			
DepositCreate /CustomerRefs /CustomerRef (#N)	key	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	50	
	value	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	250	
DepositCreate /Contract	billTo	Bill-to account of the customer or division of the customer	Provided by bpost	Yes	Num	8	
	depositor	Party making the physical deposit	Provided by bpost	No	Num	8	
	invoice Grouping	Customer owned reference used by bpost to group invoices	Depending on the customer profile in PBC	No	String	70	Empty
DepositCreate /Deposit	date	The date planned for physical delivery of the deposit at bpost	A date in the format YYYY-MM-DD	Yes	Date	10	
	modelName	The selected to be createdmodel as defined in the e-Mass Post Web interface		Yes[38]	String	70	

[38] If want avoid the models' creation, contact the Customer Operations team

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
	modelPortalUserName	The e-Masspost user name that has created this model		Yes	String	30	
	invoiceRef	The customer's invoice reference	Cannot be empty	Yes	String	30	
	meteringNumber	Metering number	This is necessary when metering type (defined in the model) is *metering* or *roll stamp*[39]	Yes	String	60	Empty
	router	Router name		No	String	200	Empty
	formByMail	Indication if the deposit declaration (PDF file) should be sent by email	Y or N	No	Boolean	1	N
	autoValidate	If *Y*, and the required number of addresses is reached, a *deposit number* will be assigned by MassPost without waiting for a Validate action. If the deposit information is not coherent, validation is not possible and the system will return an error response.	Y or N	No	Boolean	1	N
	description	Description of the deposit. The customer can add extra comments about the deposit in this field.		No	String	100	Empty
DepositCreate /Deposit /*Items*				Yes			

[39] P.B./P.P. or FAM/MAF number

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
DepositCreate /Deposit /Items /Item (#N)	seq	A sequence number uniquely identifying the item within the DepositCreate action	Needs to be unique within the DepositCreate action	Yes	Num	8	
DepositCreate /Deposit /Items /Item */Characteristics*				No			
DepositCreate /Deposit /Items /Item /Characteristics /Characteristic (#N)	key	Key of the characteristic	Only 'annexType' is allowed	Yes	String	50	
	value	Value of the characteristic	If 'annexType" is used, see values available with Download Codes in eMP	Yes	String	250	
DepositCreate /Deposit /Items /Item /Quantities				Yes			
DepositCreate /Deposit /Items /Item /Quantities /Quantity (#N)	unit	Unit in which the quantity is expressed	Only PCE is allowed	Yes	String	250	
	value	Value of the quantity		Yes	String	250	
DepositCreate /Deposit /Items /Item /Quantities /Quantity (#g)	unit	Unit in which the weight is expressed	Only g/PCE is allowed	Yes	String	250	
	value	Value of the weight or the weightband		Yes	String	250	
DepositCreate /Deposit /Items /Item */Prepayments*		This tag contains the information about the pre-payments		No			
DepositCreate /Deposit /Items /Item /Prepayments /Prepayment (#N)	key	Key of the pre-payment	Only 'meteringPrice' is allowed	Yes	String	50	
	value	Value of the pre-payment		Yes	String	250	
DepositCreate /Deposit /ItemCount	value	The number of items supplied in the action	The value must be equal to the number of Item tags	Yes	Number	8	

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
DepositCreate /Deposit /*Options*		This tag contains the information about the options		No			
DepositCreate /Deposit /Options /Option (#N)	id	Option id	Must be unique across all options within the action (available in Download Types in eMP)	Yes	String	50	
DepositCreate /Deposit /Options /Option /*OptionQuantities*				Yes			
DepositCreate /Deposit /Options /Option /OptionQuantities /OptionQuantity (#N)	unit	Unit in which the quantity is expressed	- Only 'PCE' is allowed - Must be unique within the option	Yes	String	250	
	value	Value of the quantity		Yes	String	250	

Table 20: DepositCreate/DepositUpdate tag - XML structure

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the DepositCreate tag:

DepositCreate|*seq*|*depositIdentifier*|*depositIdentifierType*|*mailingRef*
Contact|*seq*|*firstName*|*lastName*|*email*|*lang*|*phone*|*fax*|*mobile*
CustomerRef|*key*|*value*
Contract|*billTo*|*depositor\invoiceGrouping*
Deposit|*date*|*modelName*|*modelPortalUserName*|*invoiceRef*|*meteringNumber*|*router*|*formByMail*|*autoValidate*|*description*
Item|*seq*
Characteristic|*key*|*value*
Quantity|*unit*|*value (#N)*
Quantity|*unit*|*value (#g)*
Prepayment|*key*|*value*
ItemCount|*value*
Option|*id*
OptionQuantity|*unit*|*value*

Table 21: DepositCreate/DepositUpdate tag - TXT structure

The structure for the DepositUpdate tag is identical, except for the DepositCreate action tag which is replaced by the DepositUpdate tag.

DepositDelete and DepositValidate tags

A DepositDelete action is used in a DepositRequest file to delete a deposit (either in a DepositCreate action earlier in the same DepositRequest file or previously transmitted). It is not allowed to delete a deposit once it has been validated.

If Mailing files are linked to a deposit, they will also be deleted.

A DepositValidate action is used in a DepositRequest file to validate a deposit (either created in a DepositCreate action earlier in the same DepositRequest file[40] or previously transmitted). This is a necessary step in the MassPost Deposit procedure prior to physically making a deposit, unless the deposit is previously created or updated with the autoValidate option.

XML structure

The following table describes the structure of the DepositDelete tag.

The structure for the DepositValidate tag is identical, except for the DepositDelete action tag which is replaced by the DepositValidate tag.

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
DepositDelete	seq	A sequence number enabling identification of the action within the file	Needs to be unique across all actions within the file	Yes	Num	8	
	deposit Identifier	A unique customer reference identifying the deposit to delete		Yes	String	20	
	deposit Identifier Type	Type of depositIdentifier	depositRef or tmpDepositNr	No	String	20	deposit Ref
DepositDelete /*Contacts*				No			
DepositDelete /Contacts /Contact (#N)	seq	A sequence number uniquely identifying the contact within the DepositCreate action	Needs to be unique within the action	Yes	Num	8	
	firstName	First name of the contact person		No	String	50	
	lastName	Last name of the contact person		No	String	50	
	email	Email of the contact		Yes	String	100	

[40] It is possible to put a DepositValidate action for a deposit create in the same Deposit Request file only if the mailing file is the master (in this case, it is the equivalent to an autovalidate). If deposit is the master, there is not yet any mailing file related to this deposit, and it is so not possible to validate it.

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
	lang	A 2 characters constant indicating the mother language of the contact	'fr' or 'nl'	Yes	String	2	
	phone	Phone number of the contact person		No	String	50	
	fax	Fax number of the contact person		No	String	50	
	mobile	Mobile phone number of the contact person		No	String	50	
DepositDelete /*CustomerRefs*				No			
DepositDelete /Customer Refs /CustomerRef (#N)	key	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	250	
	value	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	250	

Table 22: DepositDelete/DepositValidate tag - XML structure

TXT structure

After applying the rules in Part I, section Data Exchange, subsection file formats, we obtain the following TXT format for the depositDelete tag:

DepositDelete|*seq*|*depositIdentifier*|*depositIdentifierType*
Contact|*seq*|*firstName*|*lastName*|*email*|*lang*|*phone*|*fax*|*mobile*
CustomerRef|*key*|*value*

Table 23: DepositDelete/DepositValidate tag - TXT structure

The structure for the DepositValidate tag is identical, except for the DepositDelete action tag which is replaced by the DepositValidate tag.

2.2. Deposit Acknowledgement File

Recall that the Acknowledgement file is generated by bpost and confirms that the system has received a file. It indicates the original request file name and the time when the request file was received.

The root tag of the deposit Acknowledgment file is <RequestAck>

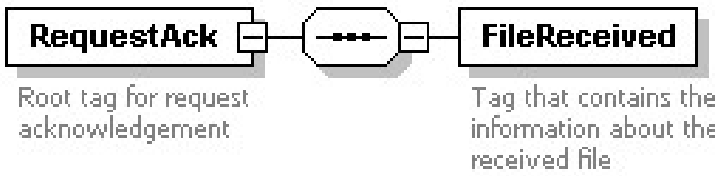


Figure 29: Deposit Acknowledgement File Structure

XML structure

The following table describes the structure of the Acknowledgement file:

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
RequestAck				Yes			
RequestAck/ FileReceived	fileName	Name of the received file	See File naming convention	Yes	String	50	
	timeStamp		Format is: YYYY-MM-DDThh:mm:ss e.g. 2001-12-17T09:30:47	Yes	Timestamp	19	

Table 24: Deposit Acknowledgement - XML Structure

Please note that the Acknowledgement file structure is a generic file structure that is identical for all Request files.

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the FileReceived tag:

FileReceived|*fileName*|*timeStamp*

Table 25: Deposit Acknowledgement - TXT Structure

2.3. Deposit Response File

Global structure

The structure found underneath is a high-level graphical representation of the XML structure of the Deposit Response File.

The root tag name for a deposit request File is < DepositResponse >.

Deposit Response files will always be generated by bpost. This section explains the structure. This will help to interpret the feedback received from the systems of bpost.

In a deposit file there is a mandatory context tag, a mandatory header tag, optional reply tags and action tags. The Replies tag appears if content errors are found in the request file or if messages need to be returned. These are errors that are not linked to a specific action, for example: errors in the request file header, invalid file name... The action tags appear for every corresponding action in the request file. For each action in the request file, the response file will include an action tag, indicating a status and the associated replies, if applicable.

Each tag has attributes associated to it. These are not available in the graphical representation below, but will be dealt with in the paragraphs below. For each level one tag, a detailed description of all underlying tags will be described, including their attributes.

Note : Response file generated by bpost (and not E-mail) is the unique relevant notification determining the correct end of the process.

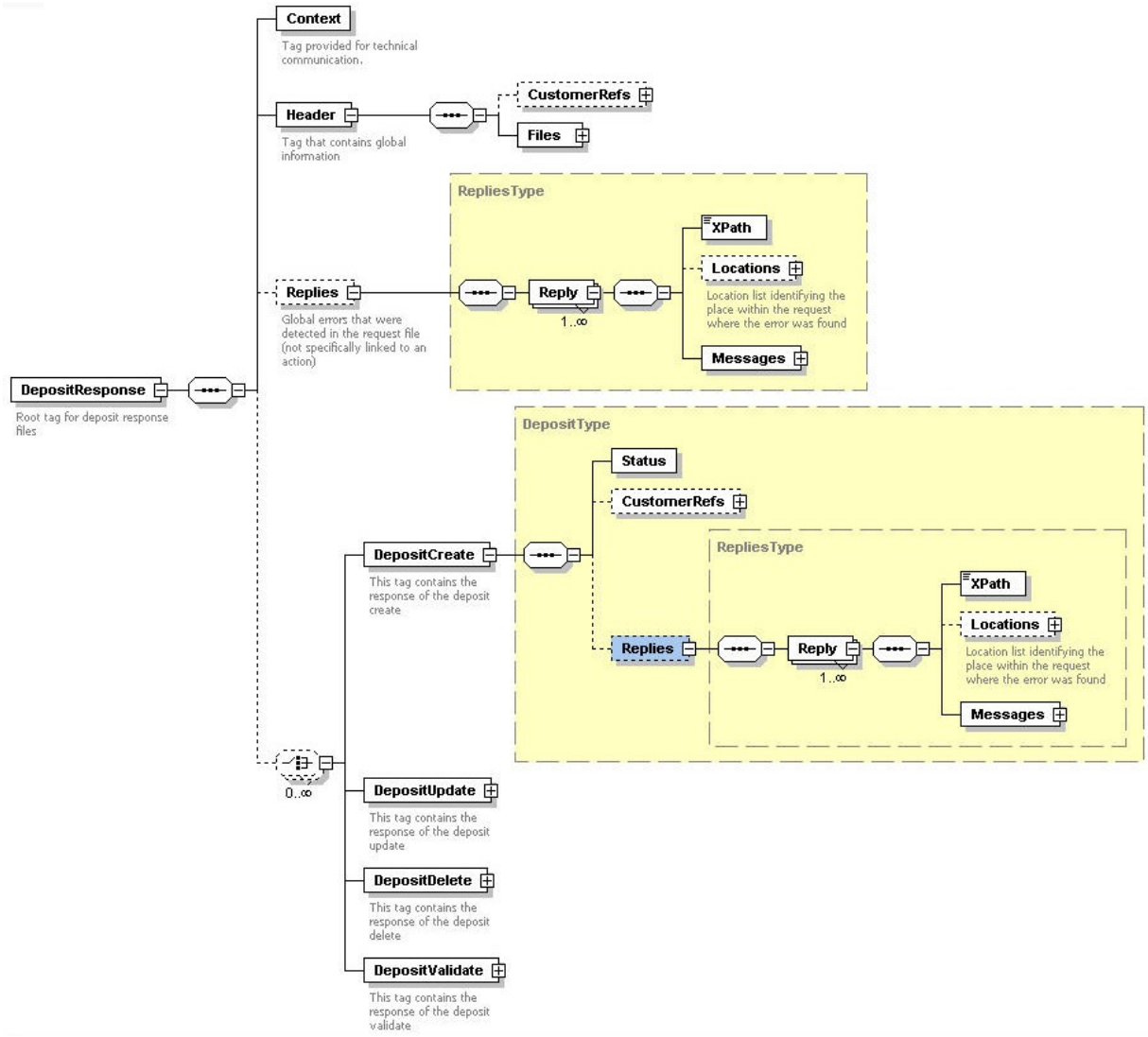


Figure 30: DepositResponse File Structure

XML structure

The graphical representation of the DepositResponse file structure can be transformed to a table. In the table below, each column represents a level down the graphical tree from the figure above.

Tag Level						
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**
Context						
Header						
	CustomerRefs					
		CustomerRef (#N)				
	Files					
		RequestProps				
Replies						
	Reply(#N)					
		XPath				
		Locations				
			Location (#N)			
		Messages				
			Message (#N)			
				Description		
				MessageContents		
					MessageContent(#N)	
DepositCreate (#N)						
	Status					
	CustomerRefs					
		CustomerRef (#N)				
	Replies					
		Reply(#N)				
			XPath			
			Locations			
				Location(#N)		
			Messages			
				Message(#N)		
					Description	
					MessageContents	
						MessageContent(#N)

Tag Level						
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**
DepositValidate (#N)	… Same as DepositCreate					
DepositUpdate (#N)	… Same as DepositCreate					
DepositDelete (#N)	… Same as DepositCreate					

Table 26: DepositResponse - XML Structure

Important to note is that the tags in italic are used for aggregation and have no correspondent form in the TXT request format. The structure is identical for all action tags (DepositCreate, DepositValidate, DepositUpdate and DepositDelete).

TXT structure

The TXT format for the Deposit Response is the following:

Context
Header
CustomerRef
RequestProps
Reply
Location
Message
Description
MessageContent
DepositCreate
Status
CustomerRef
Reply
Location
Message
Description
MessageContent
DepositValidate
Status
CustomerRef
Reply
Location
Message
Description
MessageContent
DepositUpdate
Status
CustomerRef
Reply
Location
Message
Description
MessageContent
DepositDelete
Status

CustomerRef
Reply
Location
Message
Description
MessageContent

Table 27: DepositResponse - TXT Structure

Context tag

The Context tag is necessary for proper processing by the communication servers of bpost.

It is important to note that, if the system detects one or more errors in this tag, the entire request (the entire file) will be rejected and no action (action tag) will be processed.

XML structure

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
Context	requestName	A constant identifying the request.	Must be 'DepositResponse'	Yes	String	-
	dataset	Required by the File Handling System	Must be 'M004_MPA'	Yes	String	-
	sender	Required by the File Handling System	Must be 'EMP'	Yes	String	-
	receiver	The PRS-ID of the PBC of the sender	Must match the customer identifier in the file name (see chapter "File Naming Convention")	Yes	Num	8
	version	The file version (= 0100)	Must match the file version in the file name (see chapter "File Naming Convention").	Yes	String	4

Table 28: DepositResponse Context Tag - XML Structure

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the follwing TXT format for the Context tag:

Context|*requestName*|*dataset*|*sender*|*receiver*|*version*

Table 29: DepositResponse Context Tag - TXT Structure

Header tag

The Header tag is used for general information.

XML structure

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
Header	customerId	The PRS-ID of the PBC of the sender	Must match the customer identifier in the file name (see File Naming Convention section)	Yes	Number	8	
CustomerRefs				No			
CustomerRefs /CustomerRef (#N)	key	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	250	
	value	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	250	
Files				Yes			
Files /RequestProps	fileName	The file name of the request file	Will match the actual full name of the request file	Yes	String	100	
	customer FileRef		Needs to match the 10 N's of the original file name	Yes	String	10	

Table 30: DepositResponse Header Tag - XML Structure

Note that the CustomerRef tag is reserved for the customer's own usage. bpost ignores the values supplied, and simply returns them in the Response file.

TXT structure

After applying the rules from Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the Header tag:

Header|*customerId*
CustomerRef|*key*|*value*
RequestProps|*fileName*|*customerFileRef*

Table 31: DepositResponse Header Tag - TXT Structure

Action tags

The structure of the Action tag in the response is the same for all the actions (DepositCreate, DepositUpdate, DepositDelete and DepositValidate,). For each of these actions, the Response file describes:

- A unique identifier and deposit reference;
- A status indicating whether the action was successful or not;
- The customer references for the action as far as they were present in the DepositRequest file;
- If applicable, the replies associated with the action. The structure of the replies is further detailed in the paragraph below.

XML structure

The following table describes the structure of the DepositCreate tag. The structure for the other response action tags is identical, except for the DepositCreate action tag which is replaced by the appropriate action tag (DepositUpdate, DepositValidate and Depositdelete).

Field Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
DepositCreate	seq	The sequence number of the DepositCreate action in the request file		Yes	Num	8	
	depositRef	The deposit reference that was supplied in the request file		Yes	String	20	
	depositIdentifierType	Type of depositIdentifier	depositRef or tmpDepositNr				depositRef
DepositCreate/Status	code	Status code (see status codes table in annexes, §1.1)		Yes	String	10	
DepositCreate/CustomerRefs				No			
DepositCreate/CustomerRefs/CustomerRef (#N)	key	Value copied from the request file		Yes	String	50	
	value	Value copied from the request file		Yes	String	250	
DepositCreate/Replies				No			
	See the Replies tag description below						

Table 32: DepositResponse Action Tag - XML Structure

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the action tags:

DepositCreate|*seq*|*depositRef*
Status|*code*
CustomerRef|*key*|*value*
Reply -> See replies tag description below

Table 33: DepositResponse Action Tag - TXT Structure

Replies tag

Replies tags are used everywhere in the Response file where errors or other messages are described.

The Response file contains a number of replies. Each reply is related to a specific location in the Request file. A reply may contain one or more messages. All messages within a reply are related to the same location defined for the reply.

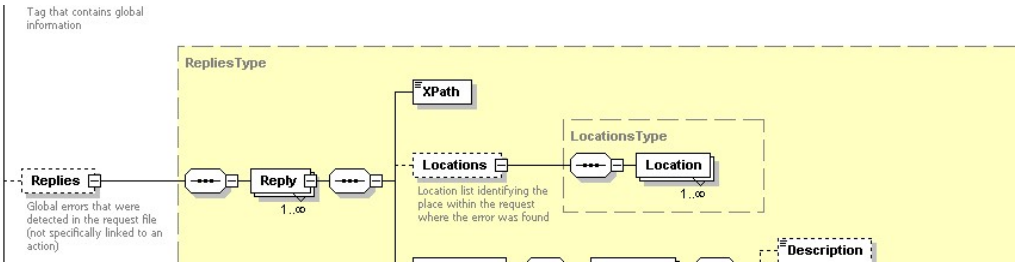


Figure 31: Replies Tag Structure

XML structure

The following table describes the structure of this tag.

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
Replies				No		
Replies /Reply (#N)	seq	The sequence number of the reply within the Replies tag		Yes	Num	8
Replies /Reply */XPath*		The XPath expression identifying the exact location where the reply is related to.		Yes	String	50
Replies /Reply */Locations*				No		
Replies /Reply /Locations /Location	tagName	The name of the tag		Yes	String	50
	attributeName	The name of the tag attribute that uniquely identifies the element		No	String	50
	attributeValue	The value of the attribute (that is defined in attributeName) to look for		No	String	250

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
Replies /Reply /*Messages*				Yes		
Replies /Reply /Messages /Message	code	Message code (see message code table in the annexes, §1.1)		Yes	String	10
	severity	Message severity: "FATAL", "ERROR", "WARN", "INFO"		Yes	String	10
Replies /Reply /Messages /Message /Description		Message description supplying extra information		No	String	50
Replies /Reply /Messages /Message /*MessageContents*		Tag containing extra information about the message		No		
Replies /Reply /Messages /Message /MessageContents /MessageContent	key	Key of the extra information	Possible keys depend on the action (See MessageContent element)	Yes	String	50
	value	Value of the extra information		Yes	String	250

Table 34: Replies Tag - XML Structure

For each reply, the XPath element[41] contains the XPath expression identifying the exact path (starting from the document root) to the location where the reply applies to.

Since XPath is a specific XML construct, this information cannot be included in the TXT format. Therefore, an alternative structure is provided in the Locations tag. Within this tag, a sequence of tags within the XML tree is defined. To arrive at the location where the error was encountered requires navigating down the tree following this sequence of tags.

When, at a certain level, multiple child tags with the same name exist, the attributeName value defines which attribute forms the unique key for this field. The attributeValue value then defines which key attribute value to look for.

Considering the following Request in XML:

```
<A lang="en">
	<B id="1">
		<C key="1">
			<D>some text</D>
		</C>
		<C key="2">
			<D>some text</D>
		</C>
		<C key="3">
			<D>some text</D>
		</C>
	</B>
	<B id="2">
		<C key="1">
```

41 XPath is a language for addressing parts of an XML document. For more information about XPath, please refer to the XPath specification at http://www.w3.org/TR/xpath.

```
                <D>some text</D>
            </C>
            <C key=”2”>
                <D>Some erroneous text</D>
            </C>
            <C key=”3”>
                <D>some text</D>
            </C>
        </B>
</A>
```

To identify an error in the bold **<D>Some erroneous text</D>** element, the <Replies> tag would look like this:

```
<Replies>
    <Reply seq=”1”>
        <XPath>/A[@lang=”en”]/B[@id="2"]/C[@key="2"]/D</XPath>
        <Locations>
            <Location tagName="A" attributeName="lang" attributeValue="en"/>
            <Location tagName="B" attributeName="id" attributeValue="2"/>
            <Location tagName="C" attributeName="key" attributeValue="2"/>
            <Location tagName="D"/>
        </Locations>
        <Messages>
          <Message code=”1003” severity=”FATAL”>
              <Description>Erroneous text found</Description>
          </Message>
        </Messages>
    </Reply>
</Replies>
```

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the replies tag:

Reply|*seq*
Location|*tagName*|*attributeValue*
Message|*code*|*severity*
Description|*description*
MessageContent|*key*|*value*

Table 35: Replies Tag - TXT Structure

Transforming the XML example above translates into TXT like this:

```
A|en
B|1
C|1
D|Some text
C|2
D|Some text
C|3
D|Some text
B|2
C|1
D|Some text
C|2
```

D|Some erroneous text
C|3
D|Some text

Reply|1
Location|A|en
Location|B|2
Location|C|2
Location|D|
Message|1003|FATAL
Description|Erroneous text found

The information can be parsed sequentially, without any knowledge of a tree structure:

1. Look for the first occurrence of 'A|en'
2. From that point on, look for the first occurrence of 'B|2'
3. From that point on, look for the first occurrence of 'C|2'
4. From that point on, look for the first occurrence of 'D'

=> this is the message location. The message describes a FATAL error with code 1003.

MessageContent element

The MessageContent element contains extra parameters for the message. The type of information supplied in this element varies depending on the action.

The following table describes the MessageContent keys that may appear for each action:

Action	MessageContent key	Description
DepositCreate		
	tmpDepositNumber	The temporary number of the created deposit. This MessageContent only appears when the action was successful.
	price	The price calculated by MassPost based on all submitted parameters
DepositValidate		
	depositNumber	The final deposit number. This MessageContent only appears when the action was successful.
	price	The price calculated by MassPost based on all submitted parameters
DepositUpdate		
	price	The price calculated by MassPost based on all submitted parameters
DepositDelete		No MessageContent

Table 36: MessageContent Keys

3. Mailing Files Syntax

Remember from Part I that data exchange follows a generic file flow. In this section we will define the file structure for each of the three files (Request, Acknowledgement, and Response) in the case of a Mailing File Syntax.

In describing the structure of each Mailing file, we will start with a global overview of the structure. Afterwards, the items from this general structure will be further elaborated on.[42]

This is followed by the description of the different elements of the file in XML format and the corresponding TXT format[43].

3.1. Mailing Request file

We will discuss a high level overview of the Mailing Request file, followed by a more detailed description of the level 1 tags and their attributes.

Global structure

The structure found underneath is a high-level graphical representation of the XML structure of the Mailing Request file.

The root tag name for a mailing Request file is <MailingRequest>.

The possible action tags for the MailingRequest file are:

- MailingCreate
- MailingDelete
- MailingCheck

Attention: due to the OptiAddress service restrictions, having one or more MailingCheck actions with other actions is not supported (e.g. having one MailingCreate, two MailingChecks and one MailingDelete). In that case, all MailingCheck actions will not be processed returning a warning for each MailingCheck. All other actions will be processed.

Each tag has attributes associated to it. These are not available in the graphical representation below, but will be dealt with in the paragraphs below. For each level one tag, a detailed description of all underlying tags will be described, including their attributes.

---

[42] Some file structure descriptions may not follow this logic rigorously.

[43] Recall from Part I that the discussion starts from an XML point of view. Applying this to TXT format is relatively simple.

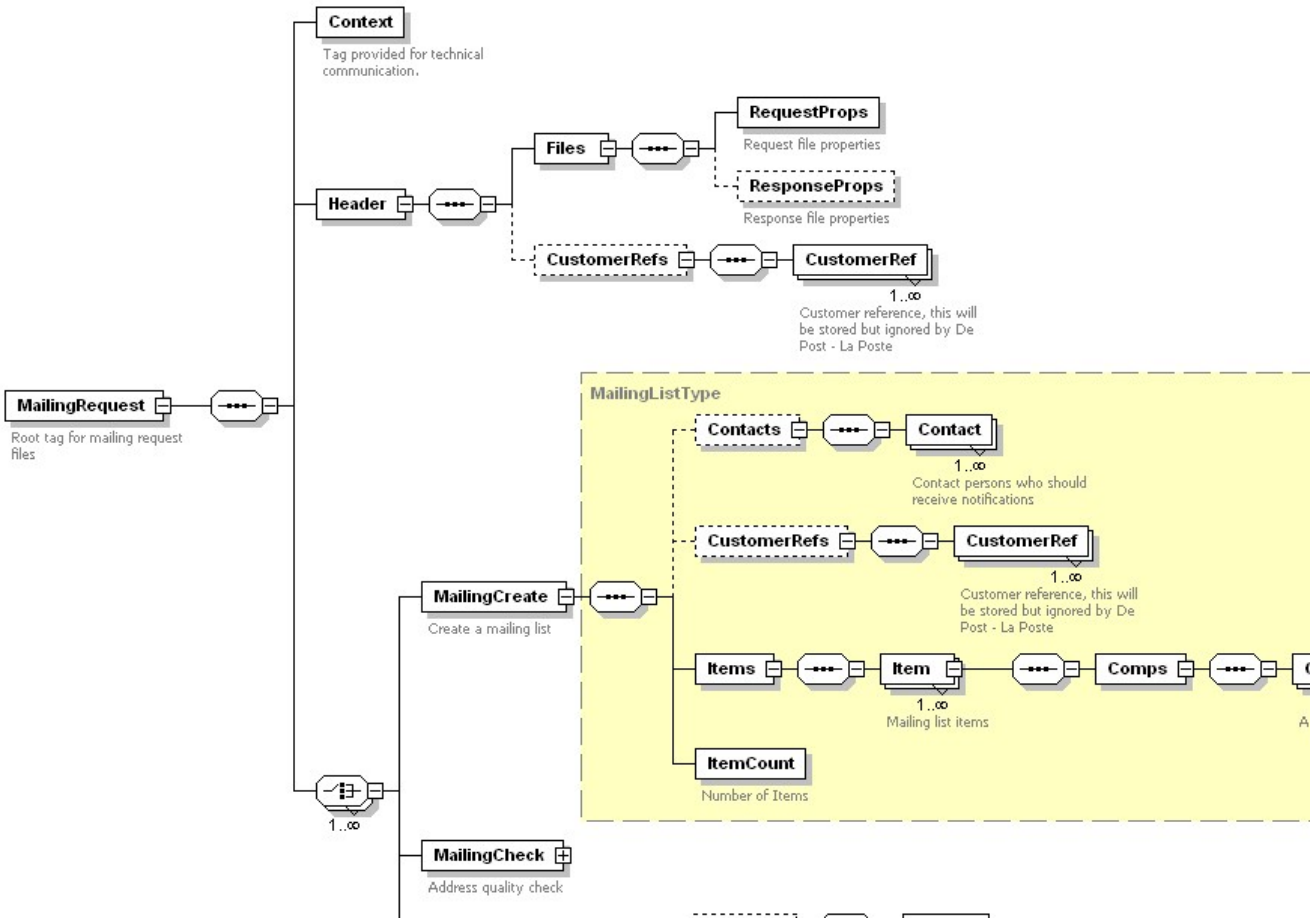


Figure 32: MailingRequest File Structure

XML structure

The graphical representation of the Mailing Request file structure can be transformed to a table. In the table underneath, each column represents a level down the graphical tree from the figure above.

Tag Level				
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**
Context				
Header				
	Files			
		RequestProps		
		ResponseProps		
	CustomerRefs			
		CustomerRef(#N)		
MailingCreate(#N)				
	FileInfo			

Tag Level				
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**
	Format			
	PresortingCodeVersion			
	Contacts			
		Contact(#N)		
	CustomerRefs			
		CustomerRef(#N)		
	Items			
		Item(#N)		
			Comps	
				Comp(#N)
	ItemCount			
MailingCheck(#N)				
	… Same as MailingCreate			
MailingDelete(#N)				
	Contacts			
		Contact(#N)		
	CustomerRefs			
		CustomerRef(#N)		

Table 37: MailingRequest - XML Structure

Important to note is that the tags in italic are used for aggregation.

TXT structure

As visible in the txt structure below, the tags in italic have no correspondent tag in the TXT request file format.

Context
Header
RequestProps
ResponseProps
CustomerRef
MailingCreate
FileInfo
Format
PresortingCodeFile
Contact
CustomerRef
Item
Comp
ItemCount
MailingCheck
Contact
CustomerRef
Item
Comp
ItemCount
MailingDelete

Table 38: MailingRequest - TXT Structure

XLS Structure

As said in Part I, subchapter 4.2 "Supported File Formats", the structure for XLS files is simplified. Especially, the data given in the Context tag, the Header tag and the actions tags (MailingCreate, MailingDelete) are given via webform on the e-MassPost website. The only information given via the XLS file are those contains in the tag "Items" (tag of level 2 below the "MailingCreate" tag).

Context tag

The Context tag is necessary for proper processing by the communication servers of bpost. It is important to note that the entire request will be rejected and no action will be processed if the system detects one or more error in this tag (code 998 of the status codes, see Annexes, chapter 1, subchapter "Status codes").

XML structure

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
Context	request Name	A constant identifying the request.	Must be 'MailingRequest'	Yes	String	-
	Dataset	Required by the File Handling System	Must be 'M037_MID'	Yes	String	-
	Sender	The PRS-ID of the PBC of the sender	Must match the customer identifier in the file name (see chapter "File Naming Convention")	Yes	String	8
	Receiver	Required by the File Handling System	Must be 'MID'	Yes	String	-
	Version	The file version	Must match the file version in the file name (see chapter "File Naming Convention").	Yes	String	4

Table 39: MailingRequest Context Tag - XML Structure

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the Context tag:

Context|*requestName*|*dataset*|*sender*|*receiver*|*version*

Table 40: MailingRequest Context Tag - TXT Structure

Header tag

The Header tag is used for general information.

XML structure

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
Header	customerId	The PRS-ID of the PBC of the sender[44]	Must match the customer identifier in the file name (see File Naming Convention)	Yes	Number	8	
	accountId	Postal Business Contract of the customer	Provided by bpost	Yes	Num	8	
	mode	A one character field	P = Production C = Certification T = Test	Yes	String	1	
Files				Yes			
Files /RequestProps	customerFileRef		Needs to match the 10 N's of the original file name	Yes	String	10	
Files /ResponseProps	format	Format type for the Response File	XML or TXT. If omitted, the Response File will use the same file type as the Request File	No	String	3	same as request file
	compressed	Boolean value specifying if the response should be compressed or not	Y or N. If omitted, the Response File will be compressed only if the Request File was compressed.	No	Boolean	1	same as request file
	Encrypted	Boolean value specifying if the response should be encrypted or not	Possible values: N Encryption mode not yet supported	No	Boolean	1	N

[44] See rules in case of use of multiple barcode customer ID in Annex with a technical solutions specialist

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
	transmission Mode	Transmission mode	Possible values: HTTPS, FTP, FTPS, FTP If omitted, the Response File will use the same mode as the Request File	No	String	5	Same as request file
CustomerRefs				No			
CustomerRefs /CustomerRef (#N)	key	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	50	
	value	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	250	

Table 41: MailingRequest Header Tag - XML Structure

Note that the CustomerRef tag is reserved for the customer's own usage. bpost ignores the values supplied, and simply returns them in the Response file.

TXT structure

After applying the rules from Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the DepositCreate tag:

Header|*customerId*|*accountId*|*mode*
RequestProps| *customerFileRef*
ResponseProps|*format*|*compressed*|*encrypted*|*transmissionMode*
CustomerRef|*key*|*value*

Table 42: MailingRequest Header Tag - TXT Structure

MailingCreate tag

A MailingCreate action is used in a Mailing Request file to create a new mailing list. The structure discussed here is for the latest version of the MAIL ID files, the version 2.00 (for older versions, see previous versions of this document).

XML structure

The following table describes the structure of the MailingCreate tag.

Field Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
MailingCreate	seq	A sequence number enabling identification of the action within the file	Needs to be unique across all actions within the file	Yes	Num	8	
	mailing Ref	A unique customer reference identifying the mailing list		Yes	String	20	
	deposit Identifier	If empty, the mailing list is master		No[45]	String	20	
	deposit Identifier Type	Type of depositIdentifier	depositRef or tmpDepositNr	No[46]	String	20	
	genMID	MAIL ID Number flag. N = No The customer supplies MAIL ID numbers 7 = bpost will generate a 7-digit MAIL ID number 9 = bpost will generate a 9-digit MAIL ID number 11 = bpost will generate an 11-digit MAIL ID number	N 7 9 11	No	String	2	N
	genPSC	Pre sorting code flag. Y = Yes bpost should generate pre-sorting codes. N = No. Only for non sorted items	Y or N	No	String	1	N
	expectedDeliveryDate	The date on which the drop is expected	Format: YYYY-MM-DD Example: 2020-10-24	Yes	Date	10	
MailingCreate/ FileInfo	type	Treatment type of the letters (Mail ID, Round & Sequence or both)	MID2 RS3 MID2,RS3	Yes	String		
MailingCreate/ Format		Information about the type of letters to be dropped	Small Large	Yes	String		Small

[45] This tag is mandatory if mailinglist is slave. This value must be the temporary deposit number or the deposit identifier

[46] This tag is mandatory if mailinglist is slave. This value must be the depositRef if deposit identifier is used or the tmpDepositNr if the temporary deposit number is used.

Field Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
	responseSortingMode	For Round&Sequence, requested order of the information in the Response file. May be in the customer order (order of the requested file) or in the print order (order in which letters must be placed in the bundles).	CU (for customer order) PO (for print order)	No	String		PO
MailingCreate/PresortingCodeVersion	version	The version number of the presorting code	Simple interger value	No	Num	8	Default version
MailingCreate /Contacts			Necessary to receive email answer	No			
MailingCreate /Contacts /Contact (#N)	seq	A sequence number uniquely identifying the contact within the MailingCreate action	Needs to be unique within the action	Yes	Num	8	
	firstName	First name of the contact person		No	String	50	
	lastName	Last name of the contact person		No	String	50	
	email	Email of the contact person		Yes	String	100	
	lang	A 2 characters constant indicating the mother language of the contact	'fr' or 'nl'	Yes	String	2	
	phone	Phone number of the contact person		No	String	50	
	fax	Fax number of the contact person		No	String	50	
	mobile	Mobile phone number of the contact person		No	String	50	
MailingCreate /CustomerRefs				No			
MailingCreate /CustomerRefs /CustomerRef (#N)	key	Ignored by bpost		Yes	String	50	
	value	Ignored by bpost		Yes	String	250	
MailingCreate /Items				Yes			
MailingCreate /Items /Item (#N)	seq	A sequence number uniquely identifying the item within the MailingCreate action	Needs to be unique within the MailingCreate action	Yes	Num	8	

Field Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
	lang	Language in which the address is expressed	'fr', 'nl', or 'de'	No	String	2	
	midNum	The MAIL ID number (see section "MAIL ID barcode structure" of the subchapter 2.3 "Barcode" of part II)		Yes, if genMID in Mailing Create tag = N	String	18	
	psCode	The pre-sorting code		No	String	20	
	priority	The priority for the item	'P' for Prior or 'NP" for Non-Prior	Yes	String	2	
MailingCreate /Items /Item /Comps				Yes			
MailingCreate /Items /Item /Comps /Comp (#N)	code	Address component code	- See address components table (Table 46: Address Components) - Needs to be unique within the Item	Yes	Num	2	
	value	Value of the address component		Yes	String	70	
MailingCreate /ItemCount	value	The number of items supplied in the action	The value must be equal to the number of Item tags	Yes	Number	8	

Table 43: MailingCreate Tag - XML Structure

TXT structure

After applying the rules from Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the MailingCreate tag:

MailingCreate|*seq*| *mailingRef*|*depositIdentifier*|*depositIdentifierType*|*genMID*|*genPSC*|*expectedDeliveryDate*
FileInfo|*type*
Format|*requestedFormat*|*responseSortingMode*
PresortingCodeFile|*PresortingCodeFile*
Contact|*seq*|*firstName*|*lastName*|*email*|*lang*|*phone*|*fax*|*mobile*
CustomerRef|*key*|*value*
Item|*seq*|*lang*|*midNum*|*psCode*|*priority*
Comp|*code*|*value*
ItemCount|*value*

Table 44: MailingCreate Tag - TXT Structure

XLS Structure

As said above, only the information given in the tag "Items" of the XML structure are contained by the XLS(X) or CSV file, so the tags of level 3 "Item" and "Comps" with their attributes (sequence, MAIL ID barcode, address components, …).

In a XLS(X) file, these information are given in these different columns :

Column	Attribute
A	seq
B to T	comp (structured, see below)
U to X	comp (unstructured, see below)
Y	midNum
Z	psCode
AA	lang
AB	priority
AC to AM	Do not complete : fields reserved for the bpost response

Table 45: MailingCreate Tag – XLS Structure

The structure of the CSV file is similar to the one of the XLS file, the only difference is that the different data are not separated by multiple columns, but with a separator. The only accepted separator between the fields in CSV file is the pipe '|'.

Address components

The following table lists all the possible address component codes that can be used in the address item components (attribute code of element Items/Item/Comps/Comp).

The customer cannot send empty address component field (enum for codes 70-79). If there is nothing to put in one of the component field, this field must not be mention in the Request file.

Component number	Code Description	Max Field Length
1	Greeting	10
2	First Name	42
3	Middle Name	20
4	Last Name	42
5	Suffix	10
6	Company Name	42
7	Department	42
8	Building	42
9	Address Line 1	42
12	House Number	12
13	Box Number	8
14	P.O. Box Number	42

Component number	Code Description	Max Field Length
15	Postal Code	12
16	City	30
17	ISO Country Code	2
18	Country Name	42
19	State	42
70-79	Reserved for customer use, verified but not used by bpost	70
90	Unstructured Name (01-05)	50
91	Unstructured Company/Department/Building (06-08)	50
92	Unstructured Street/House/Box (09-13, or 14)	50
93	Unstructured Post Code City (15-16)	50

Table 46: Address Components

The address is subdivided into different groups:

1. Individual recipient group, fields 1 to 5 or unstructured field 90
2. Organisation and geolocation group, fields 6 to 8 or unstructured field 91
3. Street, house number and box number group, fields 9 to 14 or unstructured field 92
4. Postcode and locality group, fields 15 and 16 or unstructured field 93
5. Country group, fields 17 to 19

A Mail ID address can be deposited with bpost in 2 ways: in a structured (every field of the file contains only one detail) or unstructured way (more details of the same group in one field). If the database allows it, for optimal recognition it is best to send the information in a structured way. If not, the addresses can be sent in an unstructured format. However, this might have a negative influence on recognition.

Groups 3 and 4 are very important because they are the basis of the home address (in Belgium). Group 5, including the complete name of the destination country, is indispensable for international mail items. Under certain conditions, groups 1 and 2 can help bpost to clarify certain ambiguities in the recognition of addresses.

It is possible to use the structured way of recording for a certain group of fields and an unstructured way for another group. It is however not possible to use both a structured and unstructured way of recording within the same group.

For example:

- Permitted:

  92= Rue Courtejoie 17 bte 1

  15= 5590

  16= Ciney

- Not permitted:

  9= Rue Courtejoie

  92= 17 bte 1

  15= 5590

  16= Ciney

For a good interpretation of the addresses, it is extremely important that the correct fields are used. For instance, if a postcode is entered into the house number field, the system will not recognise the address. And when a box number is entered into the house number field, the address will not be read correctly and the mail item may be delayed.

More technical details are given in Part IV: Annexes, Chapter 4 : Addressing rules for MAIL ID.

MailingCheck tag

A MailingCheck action is used in a MailingRequest file to use the OptiAddress functionality, so in order to interpret all the addresses in the mailing list and returns a compliancy rate[47] and a detailed error feedback. A MailingCheck cannot be linked to a deposit.

In a Request file, the structure for the MailingCheck tag is close to the MailingCreate tag, with some differences, for example for the depositRef field, which has no meaning for this action and is therefore ignored by the system, or for the expectedDeliveryDate field, which is not allowed.

As already said, a MailingCheck action cannot be combined with other actions in the same Request file.

XML structure

The following table describes the structure of the MailingCheck tag.

Field Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
MailingCheck	seq	A sequence number enabling identification of the action within the file	Needs to be unique across all actions within the file	Yes	Num	8	
	mailing Ref	A unique customer reference identifying the mailing list		Yes	String	20	
	deposit Identifier	No meaning for MailingCheck tag : Allowed but ignored		No	String	20	
	Deposit Identifier Type	No meaning for MailingCheck tag : Allowed but ignored		No	String	20	
	genMID	No interest for MailingCheck tag, but allowed	N 7 9 11	No	String	2	N

[47] Compliancy rate is based on the number of addresses that bpost was able to match with a postal address.

Field Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
	genPSC	Pre sorting code flag. Y = Yes bpost should generate pre-sorting codes N = No The customer supplies pre-sorting code	Y or N	No	String	1	N
	copy Request Item	If 'Yes', the system rewrite all the addresses in the Response file	Y or N	No	String	1	N
	suggestions Count	The maximal number of suggestions that the system will return for one address		No	Num	4	5
	suggestions MinScore	The minimal Levenshtein score that a suggestion must have to be returned in the Response file	Must be a number from 1 to 100	No	Num	3	60
MailingCheck /Contacts			Necessary to receive email answer	No			
MailingCheck /Contacts /Contact (#N)	seq	A sequence number uniquely identifying the contact within the MailingCreate action	Needs to be unique within the action	Yes	Num	8	
	firstName	First name of the contact person		No	String	50	
	lastName	Last name of the contact person		No	String	50	
	email	Email of the contact person		Yes	String	100	
	lang	A 2 characters constant indicating the mother language of the contact	'fr' or 'nl'	Yes	String	2	
	phone	Phone number of the contact person		No	String	50	
	fax	Fax number of the contact person		No	String	50	
	mobile	Mobile phone number of the contact person		No	String	50	
MailingCheck /CustomerRefs				No			
MailingCheck /CustomerRefs /CustomerRef (#N)	key	Ignored by bpost		Yes	String	50	
	value	Ignored by bpost		Yes	String	250	

Field Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
MailingCheck /Items				Yes			
MailingCheck /Items /Item (#N)	seq	A sequence number uniquely identifying the item within the MailingCreate action	Needs to be unique within the MailingCreate action	Yes	Num	8	
	lang	Language in which the address is expressed	'fr', 'nl', or 'de'	Yes	String	2	
	midNum	The MAIL ID number (see 4.2)		No	String	18	
	psCode	The pre-sorting code		No	String	20	
	priority	The priority for the item	'P' for Prior or 'NP" for Non-Prior	Yes	String	2	
MailingCheck /Items /Item /Comps				Yes			
MailingCheck /Items /Item /Comps /Comp (#N)	code	Address component code	- See address components table (Table 46: Address Components) - Needs to be unique within the Item	Yes	Num	2	
	value	Value of the address component		Yes	String	70	
MailingCheck /ItemCount	value	The number of items supplied in the action	The value must be equal to the number of Item tags	Yes	Number	8	

Table 47: MailingCheck Tag - XML Structure

XML structure

After applying the rules from Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the MailingCreate tag:

MailingCheck|*seq*|*mailingRef*|*depositIdentifier*|*depositIdentifierType*|*genMID*|*genPSC*|*copyRequestItem*|*suggestionsCount*|*suggestionsMinScore*
Contact|*seq*|*firstName*|*lastName*|*email*|*lang*|*phone*|*fax*|*mobile*
CustomerRef|*key*|*value*
Item|*seq*|*lang*|*midNum*|*psCode*|*priority*
Comp|*code*|*value*
ItemCount|*value*

Table 48: MailingCheck Tag - TXT Structure

MailingDelete tag

A MailingDelete action is used in a Mailing Request file to delete existing mailing list(s). If the deleted mailing list is master, all deposits linked to this mailing list are also deleted.

XML structure

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
MailingDelete	seq	A sequence number enabling identification of the request within the file	Needs to be unique across all actions within the file	Yes	Num	8	
	mailingRef	A unique customer reference identifying the mailing list to delete		Yes	String	20	
MailingDelete */Contacts*				No			
MailingDelete /Contacts /Contact (#N)	seq	A sequence number uniquely identifying the contact within the MailingCreate action	Needs to be unique within the action	Yes	Num	8	
	firstName	First name of the contact person		No	String	50	
	lastName	Last name of the contact person		No	String	50	
	email	Email of the contact person		Yes	String	100	
	lang	A 2 characters constant indicating the mother language of the contact	'fr' or 'nl'	Yes	String	2	
	phone	Phone number of the contact person		No	String	50	
	fax	Fax number of the contact person		No	String	50	
	mobile	Mobile phone number of the contact person		No	String	50	
MailingDelete */CustomerRefs*				No			
MailingDelete /CustomerRefs /CustomerRef *(#N)*	key	Ignored by bpost		Yes	String	50	
	value	Ignored by bpost		Yes	String	250	

Table 49: MailingDelete Tag - XML Structure

TXT structure

After applying the rules from Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the MailingDelete tag:

MailingDelete|*seq*|*mailingRef*
Contact|*seq*|*firstName*|*lastName*|*email*|*lang*|*phone*|*fax*|*mobile*
CustomerRef|*key*|*value*

Table 50: MailingDelete Tag - TXT Structure

3.2. Mailing Acknowledgement file

The Acknowledgement file confirms that the system has received a file. Recall that this file is generated by bpost. It indicates the original Request file name and the time when the Request file was received.

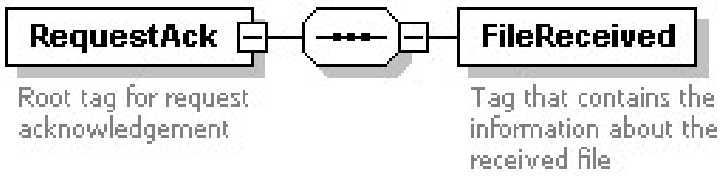


Figure 33: Mailing Acknowledgement File Structure

XML structure

The following table describes the structure of the Acknowledgement file:

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
RequestAck				Yes			
RequestAck /FileReceived	filename	Name of the received file	See file naming convention, (see part on File Naming Convention)	Yes	String	50	
	timeStamp		Format is: YYYY-MM-DDThh:mm:ss e.g. 2001-12-17T09:30:47	Yes	Timestamp	19	

Table 51: Mailing Ackowledgement - XML Structure

Remark:

The Acknowledgement file structure is a generic file structure that is identical for all Request files.

TXT structure

After applying the rules from Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the DepositCreate tag:

FileReceived|*fileName*|*timeStamp*

Table 52: Mailing Ackowledgement - TXT Structure

XLS Structure

With the Address File Tool, there is no Acknowledgement file. It is the presence of the Request file in the files list on the AFT part of the e-MassPost website that acknowledges the good reception of the Request file.

3.3. Mailing Response file

Global structure

Below is a high-level graphical representation of the XML structure of the Mailing Response File.

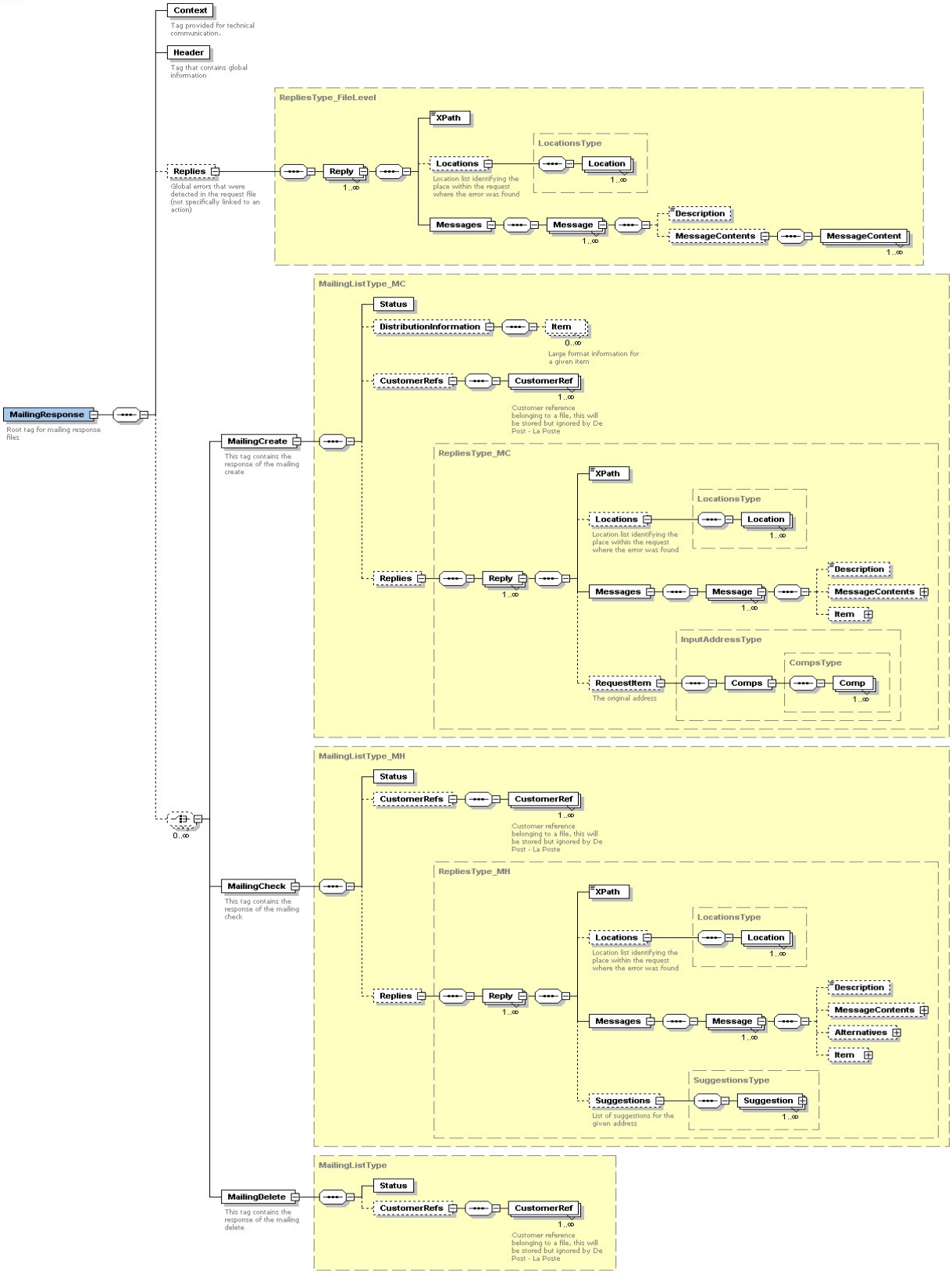


Figure 34: MailingResponse File Structure

The root tag name for a Mailing Response File is <MailingResponse>.

As said in Part I, the Replies tag appears if content errors are found in the Request file or if messages need to be returned. These are errors that are not linked to a specific action, for example: errors in the Request file header, invalid file name... The action tags appear for every corresponding action in the Request file. For each action in the Request file, the Response file will include an action tag, indicating a status and the associated replies, if applicable.

Each tag has attributes associated to it. These are not available in the graphical representation above, but will be dealt with in the paragraphs below. For each level one tag, a detailed description of all underlying tags will be described, including their attributes.

XML structure

The graphical representation of the Mailing Response file structure can be transformed to a table. In the table underneath, each column represents a level down the graphical tree from the figure above.

Tag Level								
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**	**Level 8**	**Level 9**
Context								
Header								
	CustomerRefs							
		Customer Ref (#N)						
	Customer							
	Files							
		Request Props						
Replies								
	Reply(#N)							
		Xpath						
		Locations						
			Location (#N)					
		Messages						
			Message (#N)					
				Description				
				Message *Contents*				
					Message Content (#N)			
MailingCreate (#N)								
	Status							
	DistributionInfo rmation							

Tag Level								
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**	**Level 8**	**Level 9**
		Item (#N)						
	CustomerRefs							
		Customer Ref (#N)						
	Replies							
		Reply (#N)						
			XPath					
			Locations					
				Location (#N)				
			Messages					
				Message (#N)				
					Description			
					Message Contents			
						Message Content (#N)		
MailingCheck (#N)								
	Status							
	CustomerRefs							
		Customer Ref (#N)						
	Replies							
		Reply (#N)						
			XPath					
			Locations					
				Location (#N)				
			Messages					
				Message (#N)				
					Description			
					Message Contents			

Tag Level								
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**	**Level 8**	**Level 9**
						Message Content (#N)		
					Alternatives			
						Alternative (#N)		
							Comps	
								Comp (#N)
					Item			
						Comps		
							Comp (#N)	
			Suggestions					
				Suggestion (#N)				
					Comps			
						Comp (#N)		
MailingDelete (#N)								
	Status							
	CustomerRefs							
		Customer Ref (#N)						

Table 53: MailingResponse - XML Structure

TXT structure

It is important to note that the tags in italic are used for aggregation and that the structure is identical for all action tags (MailingCreate, MailingCheck and MailingDelete)

As visible in the TXT structure below, the tags in italic have no correspondent tag in the TXT request file format.

Context
Header
CustomerRef
RequestProps
Reply
Location
Message

Description
MessageContent
MailingCreate
Status
CustomerRef
Item
Reply
Location
Message
Description
MessageContent
MailingCheck
Status
CustomerRef
Reply
Location
Message
Description
MessageContent
Comp
Suggestion
Comp
MailingDelete
Status
CustomerRef

Table 54: MailingResponse - TXT Structure

XLS Structure

As said before, the structure for XLS(X) or CSV files is simplified. Especially, the data given in the Context tag, the Header tag and the actions tags (MailingCreate, MailingDelete) are not given in the file. The only information available in the file are those contained in the tag "DistributionInformation" and in the Attribute "Code" of the tag "Messages" (tag of level 3 under the tag "Replies") ("DistributionInformation" and "Replies" are tags of level 2 below the "MailingCreate" tag).

Context tag

XML structure

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
Context	requestName	A constant identifying the request.	Must be 'MailingResponse'	Yes	String	-
	dataset	Required by the File Handling System	Must be 'M037_MID'	Yes	String	-
	sender	Required by the File Handling System	Must be 'MID'	Yes	String	-
	receiver	The PRS-ID of the PBC of the sender	Must match the customer identifier in the file name (see part on File Naming Convention)	Yes	Num	8

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
	version	The file version	Must match the file version in the file name (see 4.2 "File Naming Convention").	Yes	String	4

Table 55: MailingResponse Context Tag - XML Structure

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the Context tag:

Context|*requestName*|*dataset*|*sender*|*receiver*|*version*

Table 56: MailingResponse Context Tag - TXT Structure

Header tag

The Header tag is used for general information.

XML structure

TagName	Attributes	Description	Rule	Manda-tory	Field Type	Max Field Length
Header	customerId	The PRD-ID of the PBC of the sender	Must match the customer identifier in the file name (see part File Naming Convention)	Yes	Number	8
CustomerRefs				No		
CustomerRefs /CustomerRef (#N)	key	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	50
	value	Field reserved for the customer's own usage. Ignored by bpost.		Yes	String	250
Files				Yes		
Files/ RequestProps	fileName	The file name of the request file	Will match the actual full name of the request file	Yes	String	100
	customerFileRef		Needs to match the 10 N's of the original file name	Yes	String	10

Table 57: MailingResponse Header Tag - XML Structure

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the following TXT format for the Header tag:

Header|*customerId*
CustomerRef|*key*|*value*
RequestProps|*filename*|*customerFileRef*

Table 58: MailingResponse Header Tag - TXT Structure

Action tags

The structure of the Action tag in the response is the same for all the actions (MailingCreate, MailingCheck, and MailingDelete).

For each of these actions, the Response file describes:

- A unique identifier and deposit reference;
- A status indicating whether the action was successful or not;
- The customer references for the action as far as they were present in the Deposit Request file;
- If applicable, the replies associated with the action.

XML structure

The following table describes the structure of the MailingCreate tag.

TagName	Attributes	Description	Rule	Manda-tory	Field Type	Max Field Length
MailingCreate	seq	The sequence number of the MailingCreate action in the request file		Yes	Num	8
	mailingRef	The mailing list reference that was supplied in the request file		Yes	String	20
MailingCreate /Status	code	Status code (see status codes table in the annexes, §1.1)		Yes	String	10
MailingCreate /DistributionInformation				No [48]		

[48] Depends on the data present on the FileInfo tag of the request.

TagName	Attributes	Description	Rule	Manda-tory	Field Type	Max Field Length
MailingCreate/DistributionInformation/DistributionInformation (#N)	prtOrder	Order in which the letters should be placed in the bundle to comply with the Round&Sequence requirements		Yes	Num	
	seq	The sequence number of the requested item corresponding to route and sequence information		No	Num	8
	fieldToPrint1	The sorting Plan that is applicable to the item – To print on the mailing		No		
	fieldToPrint2	The ZIP code of the distributing office and the name of the route applicable to the item – To print on the mailing and		No		
	fieldToPrint3	The sequence on the route that is applicable to the item – To print on the mailing		No		
	orgInfo	Complementary information for the distribution office on the treatment to give to the bundles – To print on the mailing (if present)		No	String	3
	icti	Indication that the address is the first, last or only letter for a sorting center	Begin : first letter End : last letter Begin_end : only one letter	No	String	
	izon	Indication that the address is the first, last or only letter for a distribution zone	Begin : first letter End : last letter Begin_end : only one letter	No	String	
	imac	Indication that the address is the first, last or only letter for a machine	Begin : first letter End : last letter Begin_end : only one letter	No	String	
	iwav	Indication that the address is the first, last or only letter for a wave	Begin : first letter End : last letter Begin_end : only one letter	No	String	
	ioff	Indication that the address is the first, last or only letter for an office	Begin : first letter End : last letter Begin_end : only one letter	No	String	
MailingCreate/CustomerRefs				No		
MailingCreate/CustomerRefs/CustomerRef (#N)	key	Value copied from the request file		Yes	String	50
	value	Value copied from the request file		Yes	String	250
MailingCreate/Replies				No		
Mailing check						
		See the Replies tag description below				

Table 59: MailingResponse Action Tag - XML Structure

TXT structure

After applying the rules in Part I, section Data Exchange, subsection File Formats, we obtain the follwing TXT format for the MailingCreate tag:

MailingCreate|*seq*|*mailingRef*
Status|*code*
Item|*fieldToPrint1*|*fieldToPrint2*|*fieldToPrint3*|*icti*|*izon*|*imac*|*iwav*|*ioff*|*orgInfo*|*prtOrder*| *(repeat n times)*
CustomerRef|*key*|*value*
Reply -> See replies tag description below

Table 60: MailingResponse Action Tag - TXT Structure for a RS3

XLS Structure

As said above, only the information given in the tag "DistributionInformation" and in the Attribute "Code" of the tag "Messages" (tag of level 3 under the tag "Replies") of the XML structure are contained by the XLS(X) or CSV file.

In a XLS(X) file, these information are given in these different columns :

Column	Attributes
A	seq
B to X, AA to AB	Information given in the Request file
Y	midNum
Z	psCode
AC	fieldToPrint1
AD	fieldToPrint2
AE	fieldToPrint3
AF	code (from tag "Replies")
AG	orgInfo
AH	icti
AI	izon
AJ	imac
AK	iwav
AL	ioff
AM	prtOrder

Table 61: MailingResponse Action Tag – XLS Structure

The structure of the CSV file is similar to the one of the XLS file, the only difference is that the different data are not separated in different columns, but with the pipe separator '|'.

Replies tag for Mailing Create

Replies tags are used everywhere in the Response file where errors or other messages are described.

The Response file contains a number of replies. Each reply is related to a specific location in the Request file. A reply may contain one or more messages. All messages within a reply are related to the same location defined for the reply.

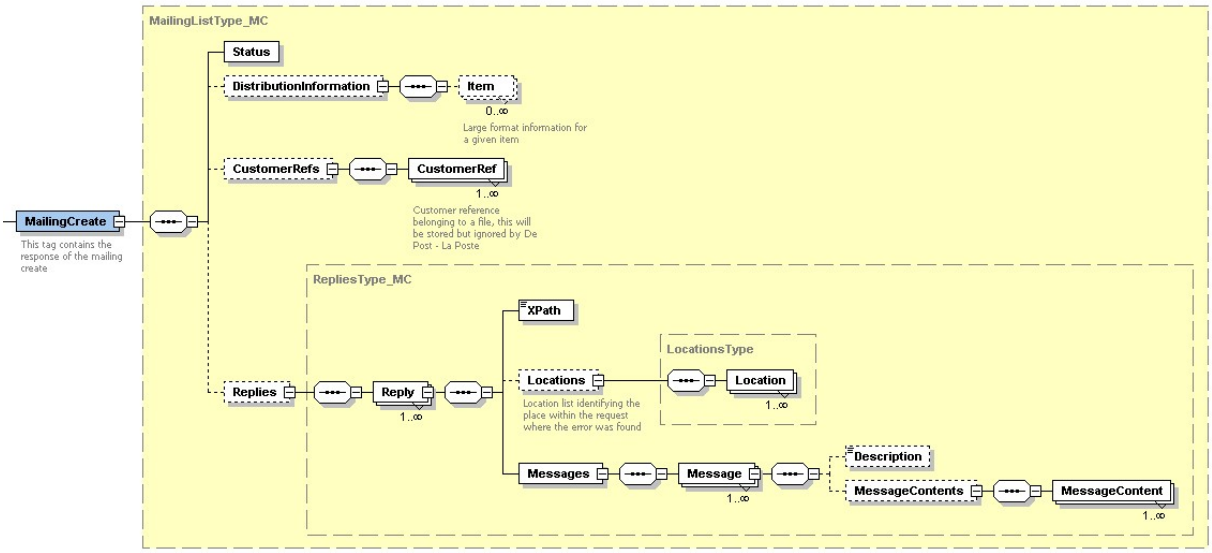


Figure 35: Replies Tag Structure for MailingCreate

XML structure

The following table describes the structure of this tag.

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
Replies				No		
Replies/Reply(#N)	seq	The sequence number of the reply within the Replies tag		Yes	Num	8
Replies/Reply/*XPath*		The XPath expression identifying the exact location where the reply is related to.		Yes	String	50
Replies/Reply/*Locations*				No		
Replies/Reply/Locations /Location	tagName	The name of the tag		Yes	String	50
	attributeName	The name of the tag attribute that uniquely identifies the element		No	String	250

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
	attributeValue	The value of the attribute (that is defined in attributeName) to look for		No	String	250
Replies/Reply/*Messages*				Yes		
Replies/Reply/Messages/Message	code	Message code (see message code table in the annexes)		Yes	String	10
	severity	Message severity: "FATAL", "ERROR", "WARN", "INFO"		Yes	String	10
Replies/Reply/Messages/Message/Description		Message description supplying extra information		No	String	250
Replies/Reply/Messages/Message/*MessageContents*		Tag containing extra information about the message		No		
Replies/Reply/Messages/Message/MessageContents/MessageContent	key	Key of the extra information	Possible keys depend on the action	Yes	String	50
	value	Value of the extra information		Yes	String	250

Table 62: MailingResponse Replies Tag - XML Structure

TXT structure

After applying the rules in Part I, File Formats, we obtain the following TXT format for the Reply tag:

Reply|*seq*
Location|*tagName*|*attributeValue*
Message|*code*|*severity*
Description|*description*
MessageContent|*key*|*value*

Table 63: MailingResponse Replies Tag - TXT Structure

MessageContent element

The MessageContent element contains extra parameters for the message. The type of information supplied in this element varies depending on the action. The following table describes the MessageContent keys that may appear for the MailingCreate action:

Action	MessageContent key	Description
MailingCreate		
	complianceRate	Percentage of valid addresses in submitted Mailing file

	compCode	The code of the incorrect address component If a mailing list item contains an incorrect address component value, this MessageContent contains the code of the incorrect component
	midNum	The MAIL ID number that was generated by bpost This MessageContent only appears if the customer indicated in the request file that bpost should generate MAIL ID numbers
	psCode	The pre-sorting code that was generated by bpost This MessageContent only appears if the customer indicated in the request file that bpost should generate pre-sorting codes

Table 64: MessageContent Keys

Example 1 (for OptiAddress file):

An item contained an incorrect address component. The incorrect component was the component with code "9". The corrected value for this component is "Suikerkaai".

```
<Message code="MID-4000" severity="WARN">
<MessageContents>
      <MessageContent key="compCode" value="9"/>
      <MessageContent key="compCorrection" value="Suikerkaai"/>
   </MessageContents>
</Message>
```

Example 2 (for MAIL ID file):

bpost generated a MAIL ID number and/or a pre-sorting code.

The MAIL ID number is "11123458025112" and the pre-sorting code is "Ga-W2-L2".

```
<Message code="MID-4030" severity="INFO">
   <MessageContents>
      <MessageContent key="midNum" value="11123458025112>
      <MessageContent key="psCode" value="Ga-W2-L2"/>
   </MessageContents>
</Message>
```

Example 3 (for Round & Sequence file):

The content of the distribution info in response to Mailing Create using large format (round and sequence info).

```
<DistributionInformation>
       <Item prtOrder="1" seq="1" fieldToPrint1="Ba-M1-W3" fieldToPrint2="1040-Reg-102"
fieldToPrint3="131" icti="Begin" izon="Begin_End imac="Begin_End" iwav="Begin_End"
ioff="Begin_End"/>
       <Item prtOrder="2" seq="2" fieldToPrint1="Bb-M2-W5" fieldToPrint2="1050-Reg-078"
fieldToPrint3="11" izon="begin" imac="Begin_End" iwav="Begin_End" ioff="Begin_End"/>
       <Item prtOrder="3" seq="3" fieldToPrint1="Bb-M9-W7" fieldToPrint2="1340-Rbu-212"
fieldToPrint3="31" icti="End" izon="End" imac="Begin_End" iwav="Begin_End" ioff="Begin_End"/>
      </DistributionInformation>
```

Replies tag for mailing check

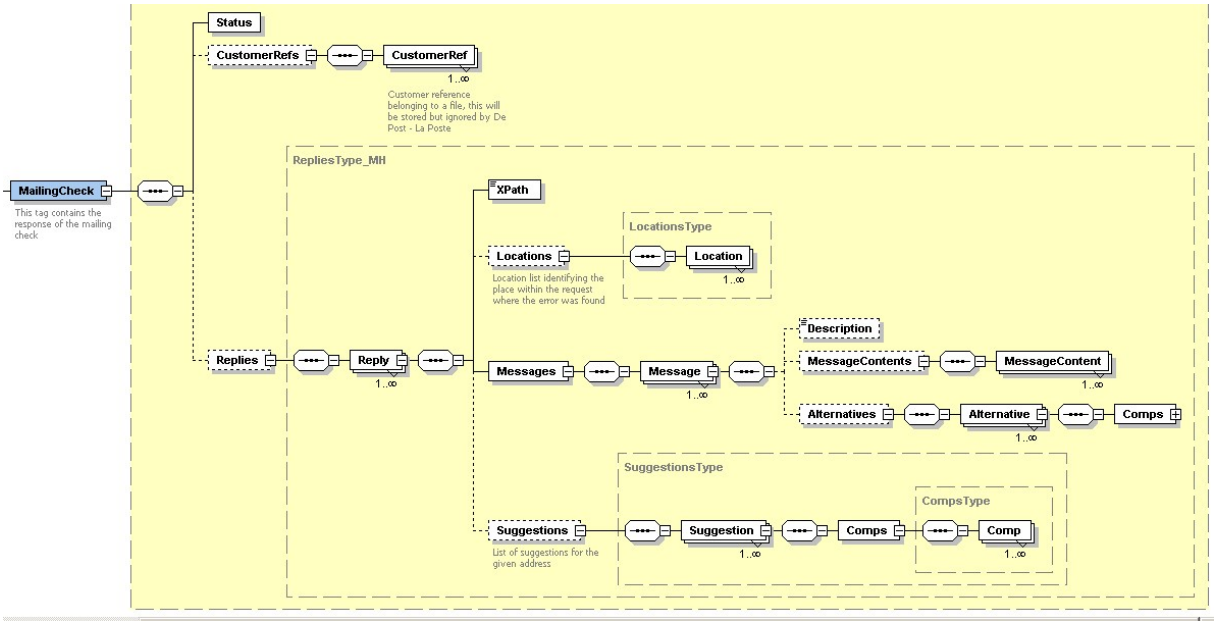


Figure 36: Replies Tag Structure for MailingCheck

The previous picture represents the MailingCheck response tag containing "Suggestions".

XML structure

The following table describes the structure of this tag.

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
Replies				No		
Replies/Reply(#N)	seq	The sequence number of the reply within the Replies tag		Yes	Num	8
Replies/Reply/*XPath*		The XPath expression identifying the exact location where the reply is related to.		Yes	String	50
Replies/Reply/*Locations*				No		
Replies/Reply/Locations /Location	tagName	The name of the tag		Yes	String	50
	attributeName	The name of the tag attribute that uniquely identifies the element		No	String	250

TagName	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length
	attributeValue	The value of the attribute (that is defined in attributeName) to look for		No	String	250
Replies/Reply/*Messages*				Yes		
Replies/Reply/Messages /Message	code	Message code (see message code table)		Yes	String	10
	severity	Message severity: "FATAL", "ERROR", "WARN", "INFO"		Yes	String	10
Replies/Reply/Messages /Message/Description		Message description supplying extra information		No	String	250
Replies/Reply/Messages /Message/*MessageContents*		Tag containing extra information about the message		No		
Replies/Reply/Messages /Message/MessageContents /MessageContent	key	Key of the extra information	Possible keys depend on the action	Yes	String	50
	value	Value of the extra information		Yes	String	250
Replies/Reply/Messages /Message/*Alternatives*				No		
Replies/Reply/Messages /Message/Alternatives /Alternative(#N)	seq	The sequence of the alternative		Yes	Num	8
Replies/Reply/Messages /Message/Alternatives /Alternative/*Comps*				Yes		
Replies/Reply/Messages /Message/Alternatives /Alternative/Comps/Comp(#N)	code	Address component code		Yes	Num	2
	value	Address component value		Yes	String	70
Replies/Reply/*Suggestions*						
Replies/Reply/Suggestions /Suggestion (#N)	seq	Suggestion sequence		Yes	Num	8
	score	Suggestion score		Yes	Num	3
Replies/Reply/Messages /Message/Suggestions /Suggestion/*Comps*				Yes		
Replies/Reply/Messages /Message/Suggestions /Suggestion/Comps/Comp(#N)	code	Address component code		Yes	Num	2
	value	Address component value		Yes	String	70

Table 65: MailingResponse Replies Tag - XML Structure

TXT structure

After applying the rules in Part I, File Formats, we obtain the following TXT format for the Reply tag:

Reply|*seq*
Location|*tagName*|*attributeValue*
Message|*code*|*severity*
Description|*description*
MessageContent|*key*|*value*
Alternative|*seq*
Comp|*code*|*value*
Suggestion|*seq*|*score*
Comp|*code*|*value*

Table 66: MailingResponse Replies Tag - TXT Structure

MessageContent element

The MessageContent element contains extra parameters for the message. The type of information supplied in this element varies depending on the action. The following table describes the MessageContent keys that may appear for the MailingCheck action:

Action	MessageContent key	Description
MailingCheck		
	compCode	The code of the incorrect address component If a mailing list item contains an incorrect address component value, this MessageContent contains the code of the incorrect component
	compCorrection	The corrected value for the incorrect address component If a mailing list item contains an incorrect address component value that could be corrected, this MessageContent contains the corrected value.
	psCode	The pre-sorting code that was generated by bpost This MessageContent only appears if the customer indicated in the request file that bpost should generate pre-sorting codes.

Table 67: MessageContent Keys

Replies tag for mailing delete

Replies tags are used everywhere in the Response file where errors or other messages are described.

The Response file contains a number of replies. Each reply is related to a specific location in the Request file. A reply may contain one or more messages. All messages within a reply are related to the same location defined for the reply.

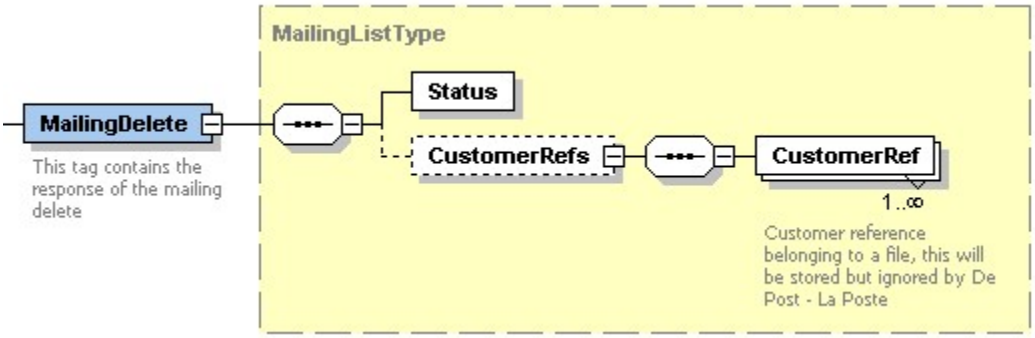


Figure 37: Replies Tag Structure for MailingDelete

XML structure

There is no message content for the Delete tag.

TXT structure

After applying the rules in Part I, file formats, we obtain the follwing TXT format for the Reply tag:

Reply|*seq*
Status

Table 68: MailingDelete Replies Tag - TXT Structure

MessageContent element

The MessageContent element contains extra parameters for the message. The type of information supplied in this element varies depending on the action. The following table describes the MessageContent keys that may appear for the MailingDelete action:

Action		Description
MailingDelete		
	Status	The status of the delete action. A value '100' indicates the delete has been processed.

Table 69: MailingDelete

4. The presorting code file syntax

bpost publishes the presorting code files on the FTP publisher account in two formats: XML and TXT (a XLS version is available on the website of bpost). Here are the specifications for each format.

The XML presorting code file fomat

The XML file has the following global structure:

Tag Level					
Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**
Context					
Header					
PreSoringCodes					
	PreSortingCode				

Table 70: Presorting codes - XML Structure

The XSD representation of the presorting codes file is the following:

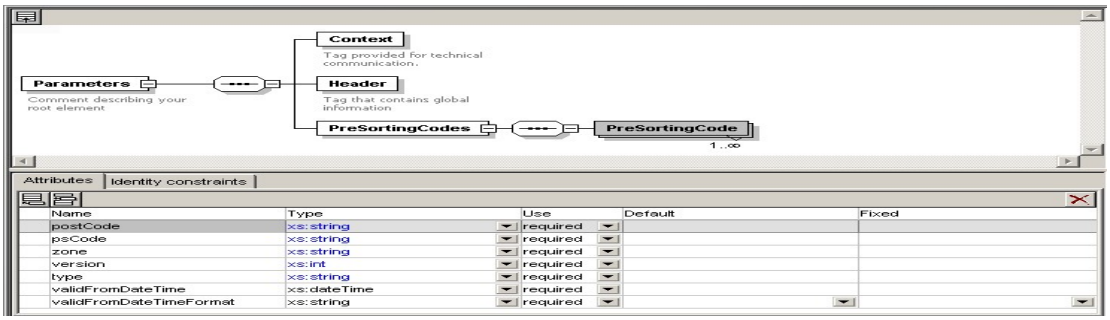


Figure 38: Presorting codes - XSD Representation

As for common responses, the file contains a context and a header but with fixed values:

Context tag:

<Context requestName="PreSortingCodesUpdate" dataset="M037_MID" sender="MID" receiver="MidPublisher" version="0100"/>

Header tag:

<Header customerId="00000000"/>

PreSortingCode tag

Here is the description table for the XML structure:

Tag Name	Attributes	Description	Rule	Mandatory	Field Type	Max Field Length	Default value
PreSortingCodes							
PreSortingCodes /PreSortingCode (#N)	postcode	The post code		Yes	Num	4	
	psCode	The presorting code		Yes	String	20	
	Zone	The zone		Yes	String	10	
	Type	The presorting code type	NF (small format) GF (large format)	Yes	String	20	
	validFromDateTime	The date from which the pscode is valid	Format is given by the attribute validFromDateTimeFormat	Yes	Date	20	
	validFromDateTimeFormat	The format used by the validFromDateTime attribute		Yes	String	20	

Table 71: PreSortingCode Tag - XML Structure

Here is a full xml small format exemple[49]:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Parameters>
<Context requestName="PreSortingCodesUpdate" dataset="M037_MID" sender="MID"
         receiver="MidPublisher" version="0100"/>
   <Header customerId="00000000"/>
   <PreSortingCodes>
      <PreSortingCode postCode="1000" psCode="B-W2-E7" zone="2" type="NF_Presort"
                      validFromDateTime="2009/09/30 06:00:00"
                      validFromDateTimeFormat="yyyy/MM/dd HH:mm:ss" version="122"/>
      <PreSortingCode postCode="1005" psCode="B-W2-E7" zone="2" type="NF_Presort"
                      validFromDateTime="2009/09/30 06:00:00"
                      validFromDateTimeFormat="yyyy/MM/dd HH:mm:ss" version="122"/>
      <PreSortingCode postCode="1006" psCode="B-W2-E7" zone="2" type="NF_Presort"
                      validFromDateTime="2009/09/30 06:00:00"
                      validFromDateTimeFormat="yyyy/MM/dd HH:mm:ss" version="122"/>
   </PreSortingCodes>
</Parameters>
```

[49] That exemple only contains few presorting code. The real file will contains more records.

The TXT presorting code file fomat

The TXT file has the following global structure:

```
--------------------------------------------------------------------------------
Context
Header
PreSortingCode
--------------------------------------------------------------------------------
```

Table 72: PreSortingCodes - TXT Structure

Here is a large format exemple with the same data as the one presented for the xml format[50]:

```
Context|PreSortingCodesUpdate|M037_MID|MID|MidPublisher|0100
Header|00000000
PreSortingCode|0099|B-M2-W2|GF|2016/10/07 15:14:10|yyyy/MM/dd HH:mm:ss|1420|B
PreSortingCode|0612|B-M2-W2|GF|2016/10/07 15:14:10|yyyy/MM/dd HH:mm:ss|1420|B
PreSortingCode|1000|B-M2-W2|GF|2016/10/07 15:14:10|yyyy/MM/dd HH:mm:ss|1420|B
```

Here is a small format exemple with the same data as the one presented for the xml format[51]:

```
Context|PreSortingCodesUpdate|M037_MID|MID|MidPublisher|0100
Header|00000000
PreSortingCode|0099|B-W4-S6|4|1440|NF|2016/10/21 05:00:00|yyyy/MM/dd HH:mm:ss
PreSortingCode|0612|B-W4-S6|4|1440|NF|2016/10/21 05:00:00|yyyy/MM/dd HH:mm:ss
PreSortingCode|1000|B-W4-S6|4|1440|NF|2016/10/21 05:00:00|yyyy/MM/dd HH:mm:ss
```

[50] Each PreSortingCode record spanson only one line. The wrapping here is only needed for presentation

[51] Each PreSortingCode record spanson only one line. The wrapping here is only needed for presentation

Part IV: Annexes

1. Errors

In this chapter we will discuss the different errors that can arrise. First status codes are explained, afterwards Message codes are described.

1.1. Status codes

The Response file describes for each action in the Request file the status of this action. The following table gives an overview of the possible status codes:

Status Code	Description
100	The action was successful
998	The action failed because the system detected at least one fatal error in the **Header** or **Context** tag. In this case, all the actions within the request file will have status 998.
999	The action failed because the system encountered at least one fatal error while processing the action.

Table 73: Status Codes

Note:

This table of status codes may evolve over time. At all times, an up-to-date status code table can be downloaded on the e-MassPost website http://www.bpost.be/emasspost, in the tab "Informations" of the menu "Files".

1.2. Message codes

The system reports errors and other information (about a Request file) within the Replies tag in the response file. As described in Part III File Syntax, the Response Files contain one or more messages.

Each message has a severity and a message code. Message severity is common accross all Response file fyntaxes and will be dealt with in the first paragraph. Message codes are specific for each Response file syntax and will be dealt with in the second paragraph.

Message severity

The message severity indicates the kind of message.

FATAL: fatal error. The system encountered an error that caused it to stop processing the action.

ERROR: non-fatal error. The system encountered a non-fatal error. The action could still be processed.

WARN: warning.

INFO: the system reports information.

Note that message and status codes are related to each other. Indeed, if an action tag within the response file contains no FATAL messages, the action could be processed successfully. Consequently, it will have status code 100. On the other hand, whenever an action triggers at least one message with severity FATAL, the status code will be 998 or 999[52].

Message code

Each message code has an associated content.The following tables (one per file syntax) give an overview of the possible message codes and their severity.

Deposit Message Codes

CODE	SEVERITY	DESCRIPTION
MPW-0000	INFO	Success
MPW-0001	FATAL	Password successfully changed
MPW-0002	FATAL	user object is expired
MPW-0003	FATAL	duplicate object
MPW-0004	FATAL	object not found
MPW-0005	FATAL	Unable to delete object. Object is referenced.
MPW-0006	FATAL	Invalid request. The id format is incorrect.
MPW-0007	FATAL	Invalid state of struts synchronization token.
MPW-1000	FATAL	User not found
MPW-1001	FATAL	geen native user voor deze routeur
MPW-1002	FATAL	Unable to build menu for the current user
MPW-1003	FATAL	No web roles found
MPW-1004	FATAL	help not found
MPW-1005	FATAL	User s have to be of the same type (normal/intermediate)

[52] Refer to status code (supra) for more information as to which Fatal status code will be used for which case.

bpost business

CODE	SEVERITY	DESCRIPTION
MPW-1006	FATAL	Native user may not be changed
MPW-1008	FATAL	only intermediate users may perform this action
MPW-1009	FATAL	Deposit not found
MPW-1010	FATAL	Product group already exists
MPW-1011	FATAL	Package with the current name already exists
MPW-1013	FATAL	native intermediate admin of the routeur not found
MPW-1014	FATAL	only administrators may perform this action
MPW-1015	FATAL	only normal users may perform this action
MPW-1016	FATAL	Annex not found
MPW-1017	FATAL	pbc_num not found
MPW-1018	FATAL	invoice grouping already exists
MPW-1019	FATAL	User already exists
MPW-1020	FATAL	Models not found for current user
MPW-1021	FATAL	no webcounter destination center could be fond
MPW-1022	FATAL	the given model was not found
MPW-1023	FATAL	model allready exists
MPW-1024	FATAL	module allready exists
MPW-1025	FATAL	item allready exists for this module
MPW-1026	FATAL	dyna table is empty
MPW-1028	FATAL	metering prijzen mogen niet leeg gelaten worden
MPW-1029	FATAL	delivery date can not be empty
MPW-1030	FATAL	no modifyable deposits found
MPW-1031	FATAL	modelname cannot be empty
MPW-1032	FATAL	PBC with the given name already exists
MPW-1033	FATAL	Given PBC number not found
MPW-1034	FATAL	intermediate native user may only be filled out for routeurs
MPW-1035	FATAL	Wrong data table passed
MPW-1036	FATAL	Only native users may be created using this function
MPW-1037	FATAL	No clients found
MPW-1038	FATAL	No Products found
MPW-1039	FATAL	No Postal Business constracts found
MPW-1040	FATAL	administrtors not filled in
MPW-1041	FATAL	Administartor and intermediate administrator have to be different persons
MPW-1042	FATAL	No routeurs found
MPW-1043	FATAL	deposit date is a non working day
MPW-1044	FATAL	Convention nr niet gevonden

CODE	SEVERITY	DESCRIPTION
MPW-1045	FATAL	Invoice niet gevonden
MPW-1046	FATAL	PRS niet gevonden
MPW-1047	FATAL	Convention reeds toegevoegd
MPW-1048	FATAL	Invoice reeds toegevoegd
MPW-1049	FATAL	Niet in staat de barcode samen te stellen
MPW-1050	FATAL	No invoice Clients found
MPW-1051	FATAL	Model Not found for the user
MPW-1052	FATAL	No PBC users found
MPW-1053	FATAL	A Postal Business Contract has to be chosen
MPW-1054	FATAL	Report <p1> not handled by the program logic
MPW-1055	FATAL	user may not view autorisation report or report does not exist
MPW-1056	FATAL	user may not view summary report or report does not exist
MPW-1059	FATAL	No deposit places found
MPW-1060	FATAL	No conventions found
MPW-1061	FATAL	The user has not the necessary rights.
MPW-1062	FATAL	Invoice client of pbc already exists
MPW-1080	FATAL	Translations saved successfull
MPW-1081	FATAL	No changes to save
MPW-1090	FATAL	Client with given PRS number not found
MPW-1091	FATAL	Error using the PRS service
MPW-1096	FATAL	Only one weight range allowed for deposit with same format and weight range
MPW-1097	FATAL	Dummy conventions may not be used
MPW-1098	FATAL	No product-rights defined
MPW-1099	FATAL	Maximum number of pieces is 9.999.999
MPW-1100	FATAL	Field is mandatory
MPW-1101	FATAL	Field has minimum length
MPW-1102	FATAL	Passwords to not match
MPW-1103	FATAL	Please specify a date format (<p1>) for field <p2>
MPW-1104	FATAL	Reference is required field
MPW-1105	FATAL	Machine nr is required field
MPW-1106	FATAL	Fields <p1> must be different for the 2 users
MPW-1107	FATAL	Account may not correct format
MPW-1108	FATAL	max value for weights : 99999999999
MPW-1109	FATAL	incorrect deciaml value for weights
MPW-1110	FATAL	no decimal numbers allowed for quantity
MPW-1111	FATAL	Setup Error, no webcounters defined

bpost business

CODE	SEVERITY	DESCRIPTION
MPW-1112	FATAL	This date does not exist
MPW-1113	FATAL	Incorrect timeformat for <p1> [hh:mm]
MPW-1114	FATAL	This time does not exist: <p1>
MPW-1117	FATAL	No single (unique) annex combination could be found.
MPW-1120	FATAL	concurrent access validation
MPW-1177	FATAL	incorrect deciaml value for metering prices
MPW-1178	FATAL	max value for metering prices : 999999999
MPW-1189	FATAL	concurrent access modify
MPW-1190	FATAL	model name all ready exist
MPW-1191	FATAL	deposit date expired
MPW-1313	FATAL	user has no access to the function
MPW-1330	FATAL	product van composition heeft prijs lager dan minimum prijs , <p1>=gewicht, <p2>=minimum prijs
MPW-2000	FATAL	Dup value Deposit centre
MPW-2001	FATAL	Dup value Nature types
MPW-2002	FATAL	Dup value Metering types
MPW-2003	FATAL	Dup value sort types
MPW-2004	FATAL	Dup value payment types
MPW-2005	FATAL	Dup value convention types
MPW-2006	FATAL	Dup value flexibilities
MPW-2007	FATAL	Dup value destination types
MPW-2008	FATAL	Dup value product types
MPW-2009	FATAL	MD Char dup value
MPW-2010	FATAL	MD Product dup value
MPW-2011	FATAL	Dup value MD periodicities
MPW-2012	FATAL	Dup value sales
MPW-2013	FATAL	Passed annex not found in the database
MPW-2014	FATAL	Dup value destination centre
MPW-2015	FATAL	Dup value quality types
MPW-2016	FATAL	Dup value adresses
MPW-2017	FATAL	Could not convert timestamp
MPW-2018	FATAL	Could not convert number
MPW-2019	FATAL	Passed Convention no found in the database
MPW-2020	FATAL	Convention with the same number already exists
MPW-2021	FATAL	No linked annex found
MPW-2022	FATAL	Annex with the same name already exists

CODE	SEVERITY	DESCRIPTION
MPW-2023	FATAL	Passed weight range is not defined in the database
MPW-2024	FATAL	Option price already exists
MPW-2052	FATAL	Option already exists
MPW-2120	FATAL	Option already exists for this date
MPW-2219	FATAL	Both parameters are filled out when only one was expected
MPW-2222	FATAL	locked pbc
MPW-2223	FATAL	start date before end date
MPW-2285	FATAL	No downloadable summary reports
MPW-2286	FATAL	No downloadable codes
MPW-2296	FATAL	Dup value VAT Code
MPW-2300	FATAL	Start date is after end date
MPW-2301	FATAL	Convention already exists
MPW-2302	FATAL	Not all fields were filled
MPW-2303	FATAL	Incorrect dateformat
MPW-2304	FATAL	A string was entered where a number is required
MPW-2305	FATAL	The pricetable is empty
MPW-2306	FATAL	Erp Code already exists
MPW-2307	FATAL	Annex already exists
MPW-2308	FATAL	No initial pricetable stored yet to copy
MPW-2309	FATAL	invalid BTW Rate
MPW-2310	FATAL	No clients added to the convention
MPW-2311	FATAL	Copyset table not saved yet
MPW-2312	FATAL	This client was already added
MPW-2313	FATAL	At least one criteria must be given
MPW-2314	FATAL	Weight ranges must differ with exactly one unit
MPW-2315	FATAL	Error while using ARS Webservice
MPW-2316	FATAL	No contracting client added to the convention
MPW-2317	FATAL	The date in <p1> is before today
MPW-2318	FATAL	At least one pricetable must be entered or the annex must be linked to another annex
MPW-2319	FATAL	Weight ranges are max 8 digit-long and have no comma in them
MPW-2320	FATAL	if payment method is domicilation post, a domiciliation number must be entered
MPW-2321	FATAL	only one decimal point is allowed in number <p1>
MPW-2322	FATAL	the ars data of this contract was not found on ars
MPW-2323	FATAL	if you use domiciliation post as payment method, each invoice clients must have a domiciliation number
MPW-2324	FATAL	The value in <p1> must be a positive number

CODE	SEVERITY	DESCRIPTION
MPW-2325	FATAL	Both Just-In-Time & Day Plus cannot be specified
MPW-2326	FATAL	The value in <p1> must be a positive number with <p2> decimals
MPW-2327	FATAL	The value in <p1> must be a number greater or equal to -1
MPW-2328	FATAL	An overlap is found in the weightranges
MPW-2329	FATAL	The minimum weight must be lower than the maximum weight
MPW-2330	FATAL	At least one weightrange is required
MPW-2331	FATAL	Maximum weight cannot be zero
MPW-2332	FATAL	Convention Nr must be a number
MPW-2340	FATAL	Value is too large.
MPW-2350	FATAL	The date cannot be before <p1>
MPW-2360	FATAL	The date should be in the future
MPW-2361	FATAL	The date should be after the begin date of the convention
MPW-2399	FATAL	At least one product needs to be added to the annex
MPW-2526	FATAL	Date already exists in othert pricetable
MPW-2527	FATAL	Maxweight < MinWeight
MPW-2563	FATAL	announced quantity cannot be zero
MPW-2589	FATAL	Party must exist in prs
MPW-2610	FATAL	Model not complete
MPW-2663	FATAL	only integral values allowed as option quantity
MPW-2664	FATAL	no blanc option quantity allowed
MPW-2665	FATAL	no 0 as option quantity allowed
MPW-2666	FATAL	Option quantity cannot be above the maximum number for the option
MPW-2667	FATAL	The option quantity is not modifiable
MPW-2678	FATAL	ARS billingcondition-paymenttype service contract does not conform delayed payment flag of part <p1>
MPW-2688	FATAL	Upload succes
MPW-2713	FATAL	warning schema has changed
MPW-2714	FATAL	model not found
MPW-2715	FATAL	metering price not needed
MPW-2716	FATAL	metering price must be at least equal to price for linked convention
MPW-2717	FATAL	metering machine needed
MPW-2718	FATAL	metering price needed
MPW-2719	FATAL	metering machine not needed
MPW-2720	FATAL	user not autorised for prs_id
MPW-2721	FATAL	double weight ranges detected
MPW-2796	FATAL	streetname, POBOX or otheraddress must be filled in

CODE	SEVERITY	DESCRIPTION
MPW-2797	FATAL	not a valid legal site
MPW-2800	FATAL	Deposit center can't process this volume
MPW-2888	FATAL	combination does not exist in annex
MPW-2896	FATAL	account not valid
MPW-2897	FATAL	account size <> 12
MPW-2899	FATAL	Customer Barcode ID already exists
MPW-2913	FATAL	Name is mandatory
MPW-2933	FATAL	Not a valid barcode
MPW-2934	FATAL	start date and end date can differ max 1 month
MPW-2936	FATAL	Vatcode, customer nr or customer name must be filled in
MPW-2946	FATAL	Minstens 1 delimiter value invullen
MPW-2992	FATAL	metering or product constraint
MPW-2993	FATAL	no prizes specified
MPW-2994	FATAL	valid date befor today
MPW-2995	FATAL	VAT already exists for this date
MPW-2996	FATAL	A new PRS with number <p1> is created
MPW-2997	FATAL	List of parties: <p1>
MPW-2998	FATAL	Referred PRS <p1> is not legal site!
MPW-2999	FATAL	invalid Belgian postcode
MPW-3000	FATAL	generic errors message for Double formats
MPW-3001	FATAL	this client was already added as a registred depositor
MPW-3002	FATAL	No products found for this product group
MPW-3013	FATAL	credit nota created !
MPW-3501	FATAL	overlapping ranges
MPW-3502	FATAL	active discount
MPW-3503	FATAL	illegal session state
MPW-3504	FATAL	illegal request
MPW-3505	FATAL	existing validity
MPW-3506	FATAL	existing price
MPW-3507	FATAL	incorrect decimal format
MPW-3508	FATAL	incorrect range
MPW-3509	FATAL	Product group mismatch
MPW-3510	FATAL	Not a valid percentage.
MPW-3511	FATAL	empty annex name
MPW-3512	FATAL	Discount already exists
MPW-3520	FATAL	Message shown when 'new composition' is invoked for a date that already exists.

CODE	SEVERITY	DESCRIPTION
MPW-3600	FATAL	Invalid weightbandheader
MPW-3601	FATAL	Invalid columnheader
MPW-3602	FATAL	Invalid weightbandvalue
MPW-3603	FATAL	Invalid column name
MPW-3604	FATAL	Invalid price value
MPW-3605	FATAL	No file uploaded
MPW-3606	FATAL	Invalid account id provided
MPW-3650	FATAL	Unexpected error during pricetable upload
MPW-5000	FATAL	No product was found for the requested unit weight.
MPW-5001	FATAL	No product was found for the requested option.
MPW-5002	FATAL	A mandatory option was not specified by the customer.
MPW-5003	FATAL	The customer wants to update a deposit with a depositRef / tmpDepositNr that does not exist.
MPW-5004	FATAL	The customer wants to update a deposit with a different model name. The model cannot be updated.
MPW-5005	FATAL	We received a request message that could not be read or parsed.
MPW-5006	FATAL	Request file not found.
MPW-5007	FATAL	A file with the same filename already exists.
MPW-5008	FATAL	Received a request message for a file that does not have status OK (100).
MPW-5009	FATAL	The file name is not compliant with the file naming conventions.
MPW-5011	FATAL	The version in the file name does not match the version in the file context.
MPW-5012	FATAL	The customer id in the file name does not match the customer id in the file header.
MPW-5013	FATAL	The customer reference in the file name does not match the customer reference in the file header.
MPW-5014	FATAL	Unable to create/update a slave deposit for a master mailing list that does not exist.
MPW-5015	FATAL	The customer id in the file name does not match the sender in the file header.
MPW-5016	FATAL	Unable to create a deposit. The user does not have authorisation to create a deposit.
MPW-5017	FATAL	Not allowed to link new mailing lists to a deposit that is already validated.
MPW-5018	FATAL	Not allowed to link new mailing list to a deposit that does not exist.
MPW-5019	FATAL	Incorrect mailinglist reference, no match found.
MPW-5020	FATAL	Action not allowed, mailing list already attached.
MPW-5021	FATAL	Action not allowed, mailing list is linked to a deposit that is already validated.
MPW-5022	FATAL	Not allowed to delete this mailing list, because it is linked to a validated deposit.
MPW-5023	FATAL	Action not allowed because of Slave-Master relationship constraint.
MPW-5024	FATAL	Could not create the mailing list, because the given mailing reference already exists for this customer.
MPW-5025	FATAL	Could not create the mailing list, because the given invoice client (bill-to) is not allowed for the user.

CODE	SEVERITY	DESCRIPTION
MPW-5026	FATAL	Could not create the mailing list, because the used model was created by a user (modelPoralUserName) who belongs to a different customer than the user that is creating the deposit.
MPW-5027	FATAL	Could not update the mailing list, because the given invoice client (bill-to) is not allowed for the user.
MPW-5028	FATAL	Unable to update a deposit. The user does not have authorisation to update a deposit.
MPW-5029	FATAL	Unable to validate a deposit. The user does not have authorisation to validate a deposit.
MPW-5030	FATAL	Unable to delete a deposit. The user does not have authorisation to delete a deposit.
MPW-5031	FATAL	Could not create the deposit, because the given deposit reference already exists for this customer.
MPW-5032	FATAL	Could not create the mailing list, because the given model user does not exist.
MPW-5033	FATAL	Could not validate the deposit, because the given deposit doesn't exists.
MPW-5034	FATAL	Could not validate the deposit, because there were not enough addresses attached.
MPW-5035	FATAL	Model is incomplete; missing product
MPW-5036	FATAL	Model is incomplete; missing deposit place
MPW-5037	FATAL	Model is incomplete; missing nature type
MPW-5038	FATAL	Model is incomplete; missing destination type
MPW-5039	FATAL	Portal could not authenticate the given user id and account id
MPW-5040	FATAL	Model is incomplete; missing normalisation
MPW-5041	FATAL	Model is incomplete; missing mechanisation
MPW-5042	FATAL	Model is incomplete; missing sorting type
MPW-5043	FATAL	Model is incomplete; missing day plus
MPW-5044	FATAL	Model is incomplete; missing deposit until
MPW-5045	FATAL	Model is incomplete; missing metering type
MPW-5047	FATAL	Unable to create a mailing list. The user does not have authorisation to create a mailing list.
MPW-5048	FATAL	Unable to check a mailing list. The user does not have authorisation to check a mailing list.
MPW-5049	FATAL	Unable to delete a mailing list. The user does not have authorisation to delete a mailing list.
MPW-5050	FATAL	Unable to delete the deposit. The deposit was already validated.
MPW-5051	FATAL	Unable to delete the master mailing list. Unable to delete the slave deposits.
MPW-5052	FATAL	Could not check the mailing list, because the given mailing reference already exists for this customer.
MPW-5053	FATAL	Could not delete the deposit, because the given deposit doesn't exists.
MPW-5054	FATAL	Metering price is missing but required
MPW-5055	FATAL	Metering price is given but not allowed
MPW-5056	FATAL	Unable to update the deposit. The deposit was already validated.
MPW-5057	FATAL	Unable to validate the deposit. The deposit was already validated.

CODE	SEVERITY	DESCRIPTION
MPW-5058	FATAL	Unable to validate the deposit. A mailing list was already purged
MPW-5059	FATAL	Could not update the deposit, because the given deposit reference already exists for this customer.
MPW-5060	FATAL	Annex type is required since current annex has multiple annex types
MPW-5061	FATAL	Annex type is not allowed since current annex has only one annex type.
MPW-5062	FATAL	The given depositor is not valid for this annex.
MPW-5063	FATAL	Unable to validate the deposit. A mailing list was not yet fully processed.
MPW-5064	FATAL	Unable to create the deposit. The master mailing list does not contain enough addresses.
MPW-5065	FATAL	Unable to update the deposit. A master deposit cannot be updated to be a slave deposit.
MPW-5066	FATAL	Unable to update the deposit. A slave deposit cannot be updated to be a master deposit.
MPW-5067	FATAL	Unable to update the deposit. The master mailing list does not contain enough addresses.
MPW-5068	FATAL	Could not check the mailing list, no certification information was found for this customer.
MPW-5069	FATAL	Could not create the mailing list, no barcode or certification information was found for this customer.
MPW-5070	FATAL	Unable to process request in production mode. The customer is not certified.
MPW-5071	FATAL	Unable to update deposit. The deposit was created in a different mode.
MPW-5072	FATAL	Unable to create mailing list in production mode. The customer is not certified.
MPW-5073	FATAL	Unable to create slave mailing list. The deposit was created in a different mode.
MPW-5074	FATAL	Unable to check mailing list in production mode. The customer is not certified.
MPW-5075	FATAL	Unable to attach mailing list. The mailing list was created in a different mode.
MPW-5076	FATAL	Unable to delete mailing list. The mailing list was created in a different mode.
MPW-5077	FATAL	Unable to create mailing list. Customer does not have an MID customer id.
MPW-5078	FATAL	Unable process request file. Syntax error.
MPW-5079	FATAL	Unable to validate deposit. The deposit was created in a different mode.
MPW-5080	FATAL	Unable to delete deposit. The deposit was created in a different mode.
MPW-5081	FATAL	Unable to process request file. Could not decompress request file.
MPW-5082	FATAL	Unable to create/update the slave deposit. Mailing list was created in a different mode.
MPW-5083	FATAL	Unable to update execution mode of the master deposit. Deposit is still linked to slave mailing lists.
MPW-5084	FATAL	Unable to update the master deposit so that it becomes a slave deposit. Deposit is still linked to slave mailing lists.
MPW-5085	FATAL	Unable to update the master deposit so that it no longer is a MID deposit. Deposit is still linked to slave mailing lists.
MPW-5086	WARN	Invalid contact email address.
MPW-5087	FATAL	The account id does not match the account id in the file header.
MPW-5088	FATAL	Unable to upload file. Max file size exceeded.
MPW-5089	FATAL	File has allready been deleted
MPW-5090	FATAL	File cannot be opened

CODE	SEVERITY	DESCRIPTION
MPW-5091	FATAL	Please enter a filename in the field
MPW-5092	FATAL	Unable to upload file. Could not determine portal user.
MPW-5093	FATAL	Unable to upload file. File not Found
MPW-5094	FATAL	Maximum number of items exceeded for a Mail ID deposit in Certification mode
MPW-5095	FATAL	Not allowed to link new mailing list to a deposit that belongs to another PBC.
MPW-5097	FATAL	The delivery date is not compliant with the distribution period in model
MPW-5100	FATAL	It's not allowed to use different expected deposit places for this deposit group
MPW-5102	FATAL	There is no right to create a deposit group for this convention
MPW-5103	FATAL	There is no right to create a deposit group for different places within this convention
MPW-5104	FATAL	The deposit has at least one characteristic in the model that differs from the first deposit of the deposit group
MPW-5105	FATAL	The customer wants to update/delete/validate a deposit group with a depositGroupRef / tmpDepositGroupNr that does not exist.
MPW-5106	FATAL	Unable to update/delete/validate the deposit group. The deposit group was already validated.
MPW-5108	FATAL	Unable to update/delete/validate the deposit group. The deposit group was created in a different mode.
MPW-5109	FATAL	A deposit belonging to a deposit group cannot be updated/deleted/validated
MPW-5110	FATAL	Unable to update/delete/validate a deposit group belonging to another account
MPW-5111	FATAL	As a router, it's impossible to update a deposit group belonging to another administrator
MPW-5112	FATAL	Unable to update/delete/validate a deposit belonging to another account
MPW-5113	FATAL	As a router, it's impossible to update/delete/validate a deposit belonging to another administrator
MPW-5114	FATAL	A deposit not associated with a file cannot be linked to a mailing list.
MPW-5115	FATAL	A Mail-ID deposit should be associated with a Mail-ID file
MPW-5116	FATAL	Model is incomplete; missing fileType
MPW-5117	WARN	File type "Data Quality" was chosen initially, but isn't applicable for the current annex. The file type is reset to "No File"
MPW-5118	FATAL	Could not create the parent holding because it already exist
MPW-5128	FATAL	Not allowed to delete this mailing list, because it is linked to a booking drop.
MPW-5129	FATAL	An Intelligent Bundling deposit should be associated with a mailing list that has Intelligent Bundling format.
MPW-5130	FATAL	A split drop is only possible for a minimum of <p1> items.
MPW-5131	FATAL	The total Volume over the days (<p1>) is different from the announced volume (<p2>).
MPW-5132	FATAL	The total volume of the split drops is different from the announced volume.
MPW-5133	FATAL	A split drop deposit must be done over consecutive working days
MPW-5134	FATAL	No split drop possible for the selected Mechanisation.
MPW-5135	FATAL	Volume per day should at least be <p1>

CODE	SEVERITY	DESCRIPTION
MPW-5136	FATAL	The volume for a drop day can not be empty
MPW-5137	FATAL	Negative volume for a drop day is not allowed.
MPW-5138	FATAL	Unable to delete a mailing plan. The user does not have authorization to delete a mailing plan.
MPW-5139	FATAL	Unable to replace a mailing plan. The user does not have authorization to replace a mailing plan.
MPW-5140	FATAL	Unable to create a mailing plan. The user does not have authorization to create a mailing plan.
MPW-5141	FATAL	Could not create the mailing plan, because the given mailing reference already exists for this customer.
MPW-5142	FATAL	Not allowed to delete a mailing plan because the mailing plan reference does not exists.
MPW-5143	INFO	It will be required to pass at the counter where the specimen check will happen
MPW-5999	FATAL	Unable to resolve the cause of the exception. Unexpected error.
MPW-6000	WARN	Invalid erp-code
MPW-6001	FATAL	Name is already used
MPW-6002	FATAL	count to large
MPW-6003	FATAL	Cannot insert null value
MPW-7000	FATAL	Missing value in search criteria for Quality Observation Search: Deposit Nr
MPW-7001	FATAL	Missing value in search criteria for Quality Observation Search: Customer Nr
MPW-7002	FATAL	Missing value in search criteria for Quality Observation Search: PB Nr
MPW-7003	FATAL	Missing value in search criteria for Quality Observation Search: FRM Nr
MPW-7004	FATAL	Data could not be updated/created, due to data integrity constraints
MPW-7005	FATAL	No client could be found for this deposit
MPW-7006	FATAL	No client could be found for this customer nr
MPW-7007	FATAL	Field freetext and reason for changing must be field in
MPW-7010	FATAL	Quality Criteria is mandatory
MPW-7011	FATAL	Observation Location is mandatory
MPW-7039	FATAL	The quality criteria description is not of format "000 Description"
MPW-7040	FATAL	quality criteria not found
MPW-7041	FATAL	duplicate quality criteria
MPW-7042	FATAL	franking machine number already bound to an other customer
MPW-7043	FATAL	postage paid number already bound to an other customer
MPW-7044	FATAL	Wrong value for remark and/or quantity
MPW-7045	FATAL	Location must be checked
MPW-7046	FATAL	City must be selected
MPW-7047	FATAL	No city are corresponding to this city and zipcode
MPW-7048	FATAL	Null value not allowed

CODE	SEVERITY	DESCRIPTION
MPW-7049	FATAL	Weight Range is not valid
MPW-7050	FATAL	First row must begin with 0
MPW-7051	FATAL	PRC Threshold range is not valid
MPW-7052	FATAL	No valid contracts found for the current bill-to
MPW-7053	FATAL	Unable to create slave mailing list. Referring to a parcel deposit isn't allowed
MPW-7054	FATAL	Unable to create slave mailing list. Referring to a deposit associated with an annex having multiple products isn't allowed
MPW-7055	FATAL	Unable to create slave mailing list. Referring to a deposit having a pricing type different than unit weight pricing isn't allowed
MPW-7056	FATAL	An identical deposit can contain only one composition
MPW-7057	FATAL	All compositions of an identical deposit should contain the same unit weight
MPW-7058	FATAL	Each product should have at most one composition
MPW-7059	FATAL	All unit weights should be unique
MPW-7060	FATAL	For each product, the unit weights should be unique
MPW-7061	FATAL	The product group doesn't allow multi annex lines, the remaining annexes must have the same pricing type
MPW-7065	ERROR	A category deposit should contain several compositions
MPW-8000	ERROR	Wrong value for field
MPW-9000	FATAL	Total weight must be lower than 10 000 000 000 g (in the past it is <1> should be less than <2>)
MPW-9001	ERROR	<p1> cannot be zero
MPW-9002	ERROR	<p1> should be a real number
MPW-9003	ERROR	<p1> must be between <p2> and <p3>
MPW-9004	ERROR	No account found for the given id
MPW-9005	ERROR	<p1> is mandotory
MPW-9006	ERROR	Account id <p1> was not found
MPW-9007	ERROR	Account id <p1> is already in this group
MPW-9008	ERROR	Account id <p1> is already in the group <p2>
MPW-9009	ERROR	<p1> is not a number
MPW-9010	ERROR	At least one field should be filled in
MPW-9011	ERROR	Description must be filled in
MPW-9012	ERROR	Unable to find the given invoice grouping
MPW-9013	ERROR	<p1> already exists
MPW-9014	ERROR	Unable to create deposit with this bill to and this periodical agreement
MPW-9015	ERROR	Unable to process a deposit with a cancelled agreement
MPW-9016	ERROR	A penalty is already applied on this deposit
MPW-9017	ERROR	The deposit has no periodical or the periodical is cancelled

CODE	SEVERITY	DESCRIPTION
MPW-9020	ERROR	Pricetable already exists
MPW-9021	ERROR	Annex already linked to other pricetable
MPW-9026	INFO	Upload Successful
MPW-9027	ERROR	Incorrect filetype
MPW-9028	ERROR	Incorrect file: Number of columns is incorrect for line <p1>
MPW-9029	ERROR	The address file cannot contain blank rows
MPW-9030	ERROR	At least 1 address should be specified
MPW-9031	WARNING	<p1> avcs cards have been printed
MPW-9032	ERROR	card <p1> with barcode <p2> cannot be printed
MPW-9033	WARNING	Cannot print avcs card because deposit does not exist or is blocked
MPW-9034	ERROR	Cell <p1> at row <p2> contains a non string or numeric value
MPW-9035	ERROR	Invalid destination center number provided
MPW-9036	ERROR	The field <p1> must contain <p2> characters.
MPW-9037	ERROR	Unable to delete the field: No rights for this file.
MPW-9038	INFO	Delete successful
MPW-9039	ERROR	The mailing reference is already used
MPW-9040	ERROR	The field <p1> can not be larger than <p2> characters
MPW-9041	ERROR	Response generation successful
MPW-9042	ERROR	The presorting type does not match the presorting type of the annex it is linked to.
MPW-9043	ERROR	The field <p1> should be a valid number
MPW-9044	ERROR	Invalid column header: Expected <p1>, but was <p2>
MPW-9045	ERROR	The field <p1> is a required field, but was not filled in at row <p2>
MPW-9046	ERROR	The field <p1> should be a number, but was not at row <p2>
MPW-9047	ERROR	The value for <p1> at row <p2> is larger than <p3> characters
MPW-9048	ERROR	A pipe is not allowed in a cell (Row <p1>, Cell <p2>)
MPW-9049	ERROR	The address on row <p1> should contain more than the seq and priority
MPW-9051	ERROR	The total amount fee must be equal to total quantity subscription fee
MPW-9053	ERROR	The field <p1> at row <p2> should contain one of the following values: <p3>
MPW-9054	ERROR	The field <p1> at row <p2> should be between <p3> and <p4> characters
MPW-9055	ERROR	The field <p1> at row <p2> is a response field, and should be empty
MPW-9056	WARNING	No subscription validity was found for <p1>. Please make sure to add it later.
MPW-9057	ERROR	Subscription validity date is incorrect.
MPW-9058	WARNING	There was at least 1 valid annex that didn’t contain a pricegrid for <p1>
MPW-9059	WARNING	No annexes found for this convention. Do not forget to make one for <p1>
MPW-9060	ERROR	Can not upload an empty excel file
MPW-9061	ERROR	The sequence should be unique, but <p1> was used more than once in this file

CODE	SEVERITY	DESCRIPTION
MPW-9062	ERROR	Can not have this Sorting type for this File type
MPW-9063	ERROR	Can not have this File type for this Mechanisation
MPW-9064	ERROR	You need to upload a file
MPW-9065	ERROR	Validity date cannot be before Convention start date
MPW-9066	ERROR	The expected deposit date must not be before <p1>.
MPW-9067	ERROR	If the unit weight is larger then <p1> then every sub drop can only have a maximum of <p2> items
MPW-9068	ERROR	A dimension (HxWxD) must contain at least both height and width
MPW-9069	ERROR	No duplicate <p1> allowed
MPW-9070	ERROR	<p1> should be <p2> or more.
MPW-9071	ERROR	The <p1> mailing list is selected more than once.
MPW-9072	ERROR	The selected mailing lists for drop day x do not contain the required number of addresses
MPW-9073	ERROR	The selected mailing lists for drop day <p1> contain too many addresses
MPW-9074	ERROR	You can not add more then <p1> consecutive days.
MPW-9075	ERROR	A drop cannot have more than 1 composition if the booking format is Identical
MPW-9076	ERROR	You can only get an Authorisation report for a booking that is announced and validated
MPW-9077	ERROR	You cannot choose a sender without a fee.
MPW-9078	ERROR	Split drop is not allowed for less than <1>
MPW-9079	ERROR	The sender is mandatory.
MPW-9080	WARNING	Message shown when 'new composition' is invoked for a date & level that already exists.
MPW-9081	WARNING	Level number
MPW-9082	ERROR	<p1> is not a correct level. Should be a number.
MPW-9083	ERROR	Level <p1> is missing for column <p2>
MPW-9084	ERROR	Level(s) <p1> are not configured in the application. Level(s) <p2> are missing in the uploaded file. Affects price table <p3>
MPW-9085	ERROR	No levels defined for price table <p1>
MPW-9086	ERROR	The price table <p1> must be unique for non intermediary subscription.
MPW-9087	ERROR	Annex doesn't have all prices level
MPW-9088	ERROR	Cannot change a normal contract to an Intermediary Subscription contract
MPW-9089	ERROR	Level <p1> is defined multiple times for column <p2>.
MPW-9090	ERROR	Minimum volume for per drop is <p1>.
MPW-9091	ERROR	A requalified version of a deposit can only be canceled.
MPW-9092	ERROR	Contact info is missing a mandatory field
MPW-9093	ERROR	Multiple IS conventions (<p1>) found for invoice client <p2>
MPW-9094	ERROR	No booking found
MPW-9095	ERROR	The deposit is linked to a booking <p1>
MPW-9096	ERROR	E-return validity date is incorrect.

CODE	SEVERITY	DESCRIPTION
MPW-9097	ERROR	The total amount fee must be equal to total quantity e-return fee
MPW-9098	ERROR	Convention with e-Return fee can only have a price model Progressive or Instant.
MPW-9099	ERROR	A subscription fee should be first created.
MPW-9100	ERROR	E-Return validity date should not be before subscription validity date.
MPW-9101	ERROR	A deposit DM Connect without file will lack discounts.
MPW-9102	ERROR	A deposit Admin without file will lack discounts.
MPW-9103	ERROR	The use of Data Quality file is not allowed for this deposit with product Admin.
MPW-9104	ERROR	Midnumber is missing but is mandatory
MPW-9105	ERROR	A yearly plan already exists for this year.
MPW-9106	ERROR	The volume deposit and the drop range volume totals are different.
MPW-9107	ERROR	The comments must be provided in order to change the status.
MPW-9108	ERROR	The ARR (<p1>) is too low to be able to validate the deposit.
MPW-9109	ERROR	It is too late to define the deposit as early drop for today.
MPW_9110	WARNING	"R&S multiple/R&S unique" will be replaced by "R&S multiple with MID/R&S unique with MID"
MPW-9111	ERROR	Deposit with temp deposit number <p1> has no pallet box data
MPW-9112	ERROR	Mailinglist <p1> has no pallet box data
MPW-9113	ERROR	EVD lost due to mailing address deviation is more than expected
MPW-9114	ERROR	EVD lost due to deposit not validated on time
MPW-9115	ERROR	Distribution moment not selected
MPW-9116	ERROR	Week Certain Distribution Moment should not be mixed with BOW and EOW
MPW-9221	ERROR	MPW-9221: Convention is not available for the selected drop date.
MPW-9800	ERROR	New booking rules : The booking discount question must be answered.
MPW-9801	ERROR	New booking rules : The deposit cannot be postponed more than <1> days comparing to the last booked date.
MPW-9802	ERROR	New booking rules : The maximum days a deposit can be postponed comparing to its last booked date must be configured.
MPW-9803	ERROR	New booking rules : The booking discount option must be answered.
MPW-9901	ERROR	Specimen Validation : Email notification can not be empty.
MPW-9902	ERROR	Specimen Validation : Title can not be empty.
MPW-9903	ERROR	Specimen Validation : File name can not be empty.
MPW-9904	ERROR	Specimen Validation : File can not be empty.
MPW-9905	ERROR	Specimen Validation : File type is not supported (pdf only).
MPW-9906	ERROR	Specimen Validation : File size is exceeded.
MPW-9911	ERROR	Specimen Validation : Start date is incorrect.
MPW-9912	ERROR	Specimen Validation : End date is incorrect.
MPW-9913	ERROR	Specimen Validation : End date cannot be before the start date.

CODE	SEVERITY	DESCRIPTION
MPW-9914	ERROR	Specimen Validation : The id is incorrect.
MPW-9915	ERROR	Your file size is too big.
MPW-9920	INFO	Specimen Validation : The specimen validation does not exist.
MPW-9921	INFO	Specimen Validation : No specimen validation attached.
MPW-9922	INFO	Specimen Validation : The specimen validation request has been deactivated
MPW-9990	ERROR	Specimen Validation : Unknown portal user.

Table 74: Deposit Message Codes

Mailing Message Codes

Code	Severity	Description
MID-1000	FATAL	The user is not authenticated (wrong user id and/or account id)
MID-1010	FATAL	The user is not authorized to upload files for this customer
MID-1020	FATAL	Not allowed to send data with Production mode. The customer is not certified
MID-1030	FATAL	The user is not authorized to perform this action.
MID-2000	FATAL	Request file version is not supported.
MID-2010	FATAL	Request filename does not conform to the file naming convention.
MID-2020	FATAL	Unable to decompress the file.
MID-2021	FATAL	Compressed file contains multiple files
MID-2026	FATAL	The compressed file name does not match the compressed archive's name
MID-2027	FATAL	The compressed archive is empty
MID-2030	FATAL	Unable to decrypt the file.
MID-2040	FATAL	Syntax error in the request file.
MID-2041	FATAL	Invalid mailing action sequence number.
MID-2042	FATAL	Duplicate item sequence number.
MID-2050	FATAL	A file with the same file name already exists.
MID-2055	FATAL	File contains invalid content
MID-2060	FATAL	File not received from File Handling System.
MID-2070	WARN	The number of items does not match the number of items specified in ItemCount
MID-2072	FATAL	The customer id in the file name does not match the customer id in the file header
MID-2073	FATAL	The customer file reference in the file name does not match the customer file reference in the file header

Code	Severity	Description
MID-2074	FATAL	The version in the file name does not match the version in the file context
MID-2075	FATAL	The actual sender does not match the sender in the file context
MID-2076	FATAL	The account id in the file is not the same as the account id with which you logged in.
MID-2080	FATAL	Invalid file type for the transformation, only TXT is supported
MID-2081	FATAL	Incorrect content format for the transformation
MID-3000	WARN	Invalid contact email address.
MID-3008	FATAL	The mailinglist contains no MID Numbers
MID-3009	FATAL	It is not allowed to reuse midnumbers generated by bpost which are not yet invalidated
MID-3010	FATAL	Invalid or duplicate MID number.
MID-3011	FATAL	Range of generared MID number is exhausted.
MID-3012	FATAL	Inconistent mid number usage: either all addresses should have (generated) MID numbers or no addresses should have MID numbers.
MID-3012a	FATAL	Inconsistent mid number usage: either all addresses should have (generated) MID numbers or no addresses should have MID numbers.
MID-3012b	WARN	Generation of midnumber has been asked and midnumber are provided
MID-3012c	FATAL	Address must contain a midNumer
MID-3012d	FATAL	Address may not contain a mid number!
MID-3013	FATAL	Invalid MID customer id in MID number.
MID-3014	FATAL	MID number already exists.
MID-3014a	FATAL	MID number Invalidated due to Fatal Error
MID-3015	FATAL	Invalid MID number format.
MID-3016	FATAL	Incorrect expected delivery date.
MID-3020	WARN	Invalid pre sorting code.
MID-3021	WARN	The specified presorting code file does not exist
MID-3022	WARN	Incorrect presorting code version.
MID-3023	WARN	Presorting code version does not exists.
MID-3024	FATAL	Presorting code type not compliant with the mailinglist format
MID-3030	FATAL	Not allowed to link new mailing lists to a deposit that is already validated.
MID-3031	FATAL	Not allowed to link a new mailing list to a deposit that does not exist.
MID-3032	FATAL	Not allowed to create a new mailing list because the mailing list reference is already in use.
MID-3033	WARN	Action ignored because mailing list reference already exists.
MID-3034	FATAL	Incorrect mailing list, no match found.

Code	Severity	Description
MID-3035	FATAL	Action not allowed, mailing list already attached.
MID-3036	FATAL	Action not allowed, mailing list is linked to a deposit that is already validated.
MID-3037	FATAL	Not allowed to link a new mailing list to a non MID deposit.
MID-3038	FATAL	Mailing lists may only be linked to a MID or Data Quality deposit.
MID-3039	FATAL	MID deposits can only be linked to MID mailing lists.
MID-3040	FATAL	Not allowed to delete this mailing list, because it is linked to a validated deposit.
MID-3041	FATAL	Unable to delete the mailing list because the slave deposits could not be deleted.
MID-3042	ERROR	Not allowed to delete this mailing list, because it is linked to a booking drop.
MID-3043	FATAL	An Intelligent Bundling deposit should be associated with a mailing list that has Intelligent Bundling format.
MID-3044	FATAL	RS with MID deposits can only be linked to MID,RS mailing lists
MID-3045	FATAL	a sort plan deposit can be only link with a MID mailinglist and not with a RS or MID,RS mailinglist
MID-3050	FATAL	Action not allowed because of Master - Slave relationship constraint
MID-3060	FATAL	Could not delete the mailing list, because the given mailingRef doesn't exist for this account.
MID-3060a	ERROR	Not allowed to delete this mailing list, because it is defined in the same request file.
MID-3061	FATAL	Could not reuse the mailing list, the given mailingRef does not exist for this account.
MID-3062	FATAL	Could not reuse the source mailing list, because the source mailing list was created manually.
MID-3070	FATAL	Could not create the mailing list, because the given mailingRef already exists for this account.
MID-3071	FATAL	Could not check the mailing list, because the given mailingRef already exists for this account.
MID-3072	FATAL	Could not check the mailing list, no certification information was found for this customer.
MID-3073	FATAL	Could not create the mailing list, no barcode or certification information was found for this customer.
MID-3074	FATAL	Unable to create slave mailing list. The deposit was created in a different mode.
MID-3075	FATAL	Unable to attach mailing list. The mailing list was created in a different mode.
MID-3076	FATAL	Unable to delete mailing list. The mailing list was created in a different mode.
MID-3077	FATAL	Unable to create mailing list. Customer does not have an MID customer id.
MID-3078	FATAL	Could not create the mailing list, because a mailing list with the same mailingRef was created manually.
MID-3079	FATAL	The mailing list has been deleted during processing.
MID-3080	WARN	Not allowed to use other action with MailingCheck. No MailingCheck actions processed!
MID-3081	FATAL	Incorrect FileInfo value

Code	Severity	Description
MID-4000	WARN	Incorrect address component value.
MID-4001	WARN	Empty address component found, this component will be ignored
MID-4010	ERROR	Incorrect address, no match found.
MID-4011	ERROR	Incorrect address, no match found but distribution office recognized.
MID-4020	ERROR	Incorrect address, multiple matches found.
MID-4030	INFO	The system generated an MID number and/or a pre-sorting code.
MID-4040	INFO	The system calculated a compliance rate.
MID-4050	WARN	A round was found for the address but no PDP.
MID-4060	WARN	Building found but no perfect match
MID-4061	INFO	PDP-ID
MID-4062	INFO	PDP-ID Suffix.
MID-4070	ERROR	A round was found for the address but no PDP.
MID-4080	ERROR	Street found but no round and no pdp.
MID-4090	INFO	Street found but no round and no pdp.
MID-4100	WARN	This address is not mid + compliant
MID-4200	FATAL	DataQuality is no longer supported
MID-4210	FATAL	RS is no longer supported
MID-4300	WARN	Not enough Pdp match
MID-5101	ERROR	reference list is empty
MID-5102	ERROR	target list is empty
MID-5103	ERROR	reference list contains duplicated values
MID-5104	ERROR	target list contains duplicated values
MID-5105	ERROR	Some mailing list not found in database
MID-5106	ERROR	reference or target list has no addresses
MID-5107	ERROR	some mailing list are not bound to mailing create or purged
MID-5120	ERROR	dynamic conditions check failed for urgent non schedulable Request
MID-5121	ERROR	found aitRequest without response
MID-5122	ERROR	at least one mailing list is refreshing
MID-5123	ERROR	at least one mailing list is in progress
MID-5124	ERROR	at least one mailing list is in error
MID-5125	ERROR	maximum volume difference exceeded

Code	Severity	Description
MID-5126	ERROR	At least one mailing list is not a number
MID-5127	FATAL	CML_InvalidCustomerID
MID-5128	FATAL	CML_SomeMailingListNotOwnedByCustomer
MID-5129	FATAL	CML_ReferenceListMappingNotFound
MID-5130	FATAL	CML_TargetListMappingNotFound
MID-5131	FATAL	CML_MaximumVolumeForInteractiveRequestExceeded
MID-5132	WARN	CML_ErrorWhilePerformingSearch
MID-7001	WARN	An address component contains an error.
MID-7002	WARN	An address component is missing.
MID-7003	ERROR	An address component contains an error.
MID-7004	ERROR	An address component is missing.
MID-7005	WARN	Missing box number
MID-7999	WARN	Generic error on address component.
MID-8001	ERROR	ITLS Error - E_buildMailinglistInfo
MID-8002	ERROR	ITLS Error - E_transformBarcodeInfoAsMailinglistInfo
MID-8003	ERROR	ITLS Error - E_enrichWithDepositInfo
MID-8010	ERROR	ITLS Error - E_unableToFindDeposit
MID-8100	ERROR	ITLS Error - E_NoDepositLinkedToMessage
MID-8101	ERROR	ITLS Error - E_ErrorWhilePreparingItlsCTISender
MID-8200	WARN	Technical problem (Portal)
MID-8210	WARN	Technical problem (Masspost)
MID-8220	WARN	Technical problem (Unknow)
MID-9999	FATAL	An unexpected error occurred.

Table 75: Mailing Message Codes

Note:

This table of message codes may evolve over time. At all times, an up-to-date message code table can be downloaded on the e-MassPost website, in the tab "Informations" of the menu "Files".

2. Barcode Background

2.1. General[53]

Code 128 is a very effective, high-density symbology, which permits the encoding of alphanumeric data. The symbology includes a checksum digit for verification, and the barcode may also be verified character-by-character by verifying the parity of each data byte. This symbology has been widely implemented in many applications where a relatively large amount of data must be encoded in a relatively small amount of space. Its specific structure also allows numeric data to be encoded at, effectively, double-density.

An example of alphanumeric encoding in a single Code 128 barcode is:

Figure 39: Code 128 Example

Computing the Checksum Digit

Before a Code 128 symbol may be encoded, the software must compute the correct checksum digit, which will be included in the barcode. The checksum digit is based on a modulo 103 calculation based on the weighted sum of the values of each of the digits in the message that is being encoded, including the start character.

The steps for calculating the check digit are as follows:

- Take the value of the start character (103, 104, or 105 for start codes A, B, C respectively) and make that the starting value of the running checksum.
- Starting with the first data character following the start character, take the value of the character (between 0 and 102, inclusive) multiply it by its character position (1) and add that to the running checksum.
- Take each additional character in the message, take its value, and multiply it by its character position, and add the total to the running checksum.
- Divide the resulting running checksum by 103. The remainder becomes the checksum digit, which is added to the end of the message.
- The stop character is appended after the checksum digit.

[53] See Barcodeisland.com No EAN barcode is allowed.

- This is easier to understand with an example. Let's calculate the checksum digit for the sample barcode above, "**HI345678**". The checksum digit is included in all Code 128 barcodes, but it isn't printed as part of the text below the barcode symbol (as is the case with UPC and EAN symbols).

Barcode	START-A	H	I	CODE-C	34	56	78
Character Value	103	40	41	99	34	56	78
Character Position	-	1	2	3	4	5	6
Calculation	103	40 * 1	41 * 2	99 * 3	34 * 4	56 * 5	78 * 6
Weighted Sum	103	40	82	297	136	280	468

Table 76: Checksum Calculation

Summing up the running checksum for each digit: 103 + 40 + 82 + 297 + 136 + 280 + 468 = 1406. This value divided by 103 is 1406 / 103 = 13 with a remainder of 67. Thus the checksum digit is the character, which has a value of 67.

NOTE: Note that the checksum starts with the first Start Character, with a weight of 1, and that the first data character also has a weight of 1.

ATTENTION: Note that the checksum calculation for numeric characters in character set C is done in pairs.

2.2. Encoding the Symbol

Once the checksum digit has been calculated, the entire message which must be encoded in the bars and spaces is known. Continuing with our example, encoding from zero, the Code 128 barcode used in our example above: **HI345678** with a checksum digit of 67.

Discussion of the encoding of the barcode by considering that the number "1" represents a "dark" or "bar" section of the barcode whereas a "0" represents a "light" or "space" section of the barcode. Thus the numbers 1101 represents a doublewide bar (11), followed by a singlewide space (0), followed by a singlewide bar (1). This would be printed in the barcode as:

Figure 40: Symbol Encoding

Code 128 Encoding Table

This table indicates how to encode each digit of a Code 128 barcode. Note that it is easiest to think of each character as a value between 0 and 105, inclusive, rather than thinking of them as

characters. The character that a value represents depends on what mode (or character set), so rather than thinking of a character as "A" or "B", etc. it is more appropriate to think of it as 33, 34, etc.

VALUE	WHICH REPRESENTS IN CHARACTER SET			ENCODING	VALUE	WHICH REPRESENTS IN CHARACTER SET			ENCODING
	A	B	C			A	B	C	
00	SP	SP	00	11011001100	53	U	U	53	11011101110
01	!	!	01	11001101100	54	V	V	54	11101011000
02	"	"	02	11001100110	55	W	W	55	11101000110
03	#	#	03	10010011000	56	X	X	56	11100010110
04	$	$	04	10010001100	57	Y	Y	57	11101101000
05	%	%	05	10001001100	58	Z	Z	58	11101100010
06	&	&	06	10011001000	59	[	[	59	11100011010
07	'	'	07	10011000100	60	\	\	60	11101111010
08	(	(	08	10001100100	61	]	]	61	11001000010
09	)	)	09	11001001000	62	^	^	62	11110001010
10	*	*	10	11001000100	63	_	_	63	10100110000
11	+	+	11	11000100100	64	NUL	`	64	10100001100
12	,	,	12	10110011100	65	SOH	a	65	10010110000
13	-	-	13	10011011100	66	STX	b	66	10010000110
14	.	.	14	10011001110	67	ETX	c	67	10000101100
15	/	/	15	10111001100	68	EOT	d	68	10000100110
16	0	0	16	10011101100	69	ENQ	e	69	10110010000
17	1	1	17	10011100110	70	ACK	f	70	10110000100
18	2	2	18	11001110010	71	BEL	g	71	10011010000
19	3	3	19	11001011100	72	BS	h	72	10011000010
20	4	4	20	11001001110	73	HT	I	73	10000110100
21	5	5	21	11011100100	74	LF	j	74	10000110010
22	6	6	22	11001110100	75	VT	k	75	11000010010
23	7	7	23	11101101110	76	FF	l	76	11001010000
24	8	8	24	11101001100	77	CR	m	77	11110111010
25	9	9	25	11100101100	78	SO	n	78	11000010100
26	:	:	26	11100100110	79	SI	o	79	10001111010

bpost business

VALUE	WHICH REPRESENTS IN CHARACTER SET			ENCODING	VALUE	WHICH REPRESENTS IN CHARACTER SET			ENCODING
	A	B	C			A	B	C	
27	;	;	27	11101100100	80	DLE	p	80	10100111100
28	<	<	28	11100110100	81	DC1	q	81	10010111100
29	=	=	29	11100110010	82	DC2	r	82	10010011110
30	>	>	30	11011011000	83	DC3	s	83	10111100100
31	?	?	31	11011000110	84	DC4	t	84	10011110100
32	@	@	32	11000110110	85	NAK	u	85	10011110010
33	A	A	33	10100011000	86	SYN	v	86	11110100100
34	B	B	34	10001011000	87	ETB	w	87	11110010100
35	C	C	35	10001000110	88	CAN	x	88	11110010010
36	D	D	36	10110001000	89	EM	y	89	11011011110
37	E	E	37	10001101000	90	SUB	z	90	11011110110
38	F	F	38	10001100010	91	ESC	{	91	11110110110
39	G	G	39	11010001000	92	FS	|	92	10101111000
40	H	H	40	11000101000	93	GS	}	93	10100011110
41	I	I	41	11000100010	94	RS	~	94	10001011110
42	J	J	42	10110111000	95	US	DEL	95	10111101000
43	K	K	43	10110001110	96	FNC3	FNC3	96	10111100010
44	L	L	44	10001101110	97	FNC2	FNC2	97	11110101000
45	M	M	45	10111011000	98	SHIFT	SHIFT	98	11110100010
46	N	N	46	10111000110	99	Code C	Code C	99	10111011110
47	O	O	47	10001110110	100	Code B	FNC4	Code B	10111101110
48	P	P	48	11101110110	101	FNC4	Code A	Code A	11101011110
49	Q	Q	49	11010001110	102	FNC1	FNC1	FNC1	11110101110
50	R	R	50	11000101110	103	START A	START A	START A	11010000100
51	S	S	51	11011101000	104	START B	START B	START B	11010010000
52	T	T	52	11011100010	105	START C	START C	START C	11010011100
						STOP	STOP	STOP	11000111010

Table 77: Code 128 Encoding

Code 128 Encoding Example

The above example, **HI345678**, can now be coded in Code 128. As calculated in the Checksum Digit Calculation section, the checksum digit is **67**. The checksum digit is coded at the end of the message. Each digit is encoded using the encoding table above.

1. The START-A character: **11010000100**
2. The digit "H" encoded as: **11000101000**
3. The digit "I" encoded as: **11000100010**
4. The "CODE-C" character: **10111011110**
5. The digits "34" encoded as: **10001011000**
6. The digits "56" encoded as: **11100010110**
7. The digits "78" encoded as: **11000010100**
8. The checksum digit of 67 encoded as: **10000101100**
9. The STOP character: **11000111010**
10. The termination bar: **11**

This is shown in the following graphical representation where the barcode has been sectioned-off into areas that reflect each of the 10 components just mentioned.

Figure 41: Code 128 Encoding Example

A Code 128 barcode consists of a leading quiet zone, one of three start codes, the data itself, a check character, a stop character, and a trailing quiet zone.

The Code 128 specification defines three "character sets" or "character modes." The start code that is used determines which character set will be used. The character set may also be changed in the middle of the barcode. For example, in the barcode above the barcode starts in "Character set A" to encode the text "HI", and then switches to "Character set C" to more efficiently encode the numbers that follow.

To encode a value as a Code 128 barcode, the checksum digit must first be calculated (see procedure above) and the entire barcode, including check digit, may then be encoded as a sequence of bars and spaces.

A Code 128 barcode has the following physical structure:

Start code, which is the code 103, 104, or 105 from the encoding table below (either 11010000100 (Start-A), 11010010000 (Start-B), or 11010011100 (Start-C).

Each of the data bytes of the message, encoded with the encoding table below.

The checksum byte, calculated as described above and encoded using the table below.

Stop character of 11000111010.

Termination bar of 11.

3. List of supported and non-supported characters

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
NUL	null	N	N	
SOH	start of heading	N	N	
STX	start of text	N	N	
ETX	end of text	N	N	
EOT	end of transmission	N	N	
ENQ	enquiry	N	N	
ACK	acknowledge	N	N	
BEL	bell	N	N	
BS	backpace	N	N	
HT	horizontal tab	Y	Y	
LF, NL	line feed, new line	Y	Y	
VT	vertical tab	N	N	
FF, NP	form feed, new page	N	N	
CR	carriage return	Y	Y	
SO	shift out	N	N	
SI	shift in	N	N	
DLE	data link escape	N	N	
DC1	device control 1	N	N	
DC2	device control 2	N	N	
DC3	device control 3	N	N	

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
DC4	device control 4	N	N	
NAK	negative acknowledge	N	N	
SYN	synchonous idle	N	N	
ETB	end of transmission block	N	N	
CAN	cancel	N	N	
EM	end of medium	N	N	
SUB	substitute	N	N	
ESC	escape	N	N	
FS	file separator	N	N	
GS	group separator	N	N	
RS	record separator	N	N	
US	unit separator	N	N	
	space	Y	Y	
!	exclamation mark	Y	Y	
"	double quotation mark	Substitution	To escape	TXT: Jcommerce substitution: {"} -> {'}. XML:Should be escaped as {"}
#	number sign, pound	Y	Y	
$	dollar sign	Y	Y	
%	percent sign	Y	Y	
&	ampersand	Y	To escape	XML: Should be escaped as {&}
'	apostrophe, single quote mark	Y	To escape	XML: Should be escaped as {'}
(	left parenthesis	Y	Y	
)	right parenthesis	Y	Y	
*	asterisk	Y	Y	

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
+	plus sign	Y	Y	
,	comma	Y	Y	
-	minus sign, hyphen	Y	Y	
.	period, decimal point, full stop	Y	Y	
/	slash, virgule, solidus	Y	Y	
0	digit 0	Y	Y	
1	digit 1	Y	Y	
2	digit 2	Y	Y	
3	digit 3	Y	Y	
4	digit 4	Y	Y	
5	digit 5	Y	Y	
6	digit 6	Y	Y	
7	digit 7	Y	Y	
8	digit 8	Y	Y	
9	digit 9	Y	Y	
:	colon	Y	Y	
;	semicolon	Substitution	Y	TXT: Jcommerce substitution: {;} -> {,}
<	less-than sign	Y	To escape	XML: Should be escaped as {<}
=	equal sign	Y	Y	
>	greater-than sign	Y	To escape	XML: Should be escaped as {>}
?	question mark	Y	Y	
@	commercial at sign	Y	Y	
A	capital A	Y	Y	

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
B	capital B	Y	Y	
C	capital C	Y	Y	
D	capital D	Y	Y	
E	capital E	Y	Y	
F	capital F	Y	Y	
G	capital G	Y	Y	
H	capital H	Y	Y	
I	capital I	Y	Y	
J	capital J	Y	Y	
K	capital K	Y	Y	
L	capital L	Y	Y	
M	capital M	Y	Y	
N	capital N	Y	Y	
O	capital O	Y	Y	
P	capital P	Y	Y	
Q	capital Q	Y	Y	
R	capital R	Y	Y	
S	capital S	Y	Y	
T	capital T	Y	Y	
U	capital U	Y	Y	
V	capital V	Y	Y	
W	capital W	Y	Y	
X	capital X	Y	Y	
Y	capital Y	Y	Y	

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
Z	capital Z	Y	Y	
[	left square bracket	Y	Y	
****	backslash, reverse solidus	Y	Y	
]	right square bracket	Y	Y	
^	spacing circumflex accent	Y	Y	
_	spacing underscore, low line, horizontal bar	Y	Y	
`	spacing grave accent, back apostrophe	Y	Y	
a	small a	Y	Y	
b	small b	Y	Y	
c	small c	Y	Y	
d	small d	Y	Y	
e	small e	Y	Y	
f	small f	Y	Y	
g	small g	Y	Y	
h	small h	Y	Y	
i	small i	Y	Y	
j	small j	Y	Y	
k	small k	Y	Y	
l	small l	Y	Y	
m	small m	Y	Y	
n	small n	Y	Y	
o	small o	Y	Y	
p	small p	Y	Y	

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
q	small q	Y	Y	
r	small r	Y	Y	
s	small s	Y	Y	
t	small t	Y	Y	
u	small u	Y	Y	
v	small v	Y	Y	
w	small w	Y	Y	
x	small x	Y	Y	
y	small y	Y	Y	
z	small z	Y	Y	
{	left brace, left curly bracket	Y	Y	
|	vertical bar	To escape	Y	Should be escaped in TXT by using "|"
}	right brace, right curly bracket	Y	Y	
~	tilde accent	Y	Y	
DEL	delete	Y	Y	
€		Y	Y	
‚		Y	Y	
ƒ		Y	Y	
„		Y	Y	
…		Y	Y	
†		Y	Y	
‡		Y	Y	
ˆ		Y	Y	

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
‰		Y	Y	
Š		Y	Y	
‹		Y	Y	
Œ		Y	Y	
Ž		Y	Y	
‘		Y	Y	
’		Y	Y	
“		Y	Y	
”		Y	Y	
•		Y	Y	
-		Y	Y	
–		Y	Y	
~		Y	Y	
™		Y	Y	
š		Y	Y	
›		Y	Y	
œ		Y	Y	
ž		Y	Y	
Ÿ		Y	Y	
	non-breaking space	Y	Y	
¡	inverted exclamation mark	Y	Y	
¢	cent sign	Y	Y	
£	pound sterling sign	Y	Y	
¤	general currency sign	Y	Y	

Character	Description	**Supported in TXT ?** (Y/N/ Substitution/ To escape)	**Supported in XML ?** (Y/N/ Substitution/ To escape)	Remark
¥	yen sign	Y	Y	
¦	broken vertical bar	Y	Y	
§	section sign	Y	Y	
¨	spacing dieresis or umlaut	Y	Y	
©	copyright sign	Y	Y	
ª	feminine ordinal sign	Y	Y	
«	left double angle quote or guillemet	Y	Y	
¬	logical not sign	Y	Y	
	soft hyphen	Y	Y	
®	registered trademark sign	Y	Y	
¯	spacing macron long accent	Y	Y	
°	degree sign	Y	Y	
±	plus-or-minus sign	Y	Y	
²	superscript 2	Y	Y	
³	superscript 3	Y	Y	
´	spacing accute accent	Y	Y	
µ	micro sign, mu	Y	Y	
¶	paragraph sign, pilcrow sign	Y	Y	
·	middle dot, centered dot	Y	Y	
¸	spacing cedilla	Y	Y	
¹	superscript 1	Y	Y	
º	masculine ordinal indicator	Y	Y	
»	right double angle quote or guillemet	Y	Y	

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
¼	fraction 1/4	Y	Y	
½	fraction 1/2	Y	Y	
¾	fraction 3/4	Y	Y	
¿	inverted question mark	Y	Y	
À	capital A grave	Y	Y	
Á	capital A acute	Y	Y	
Â	capital A circumflex	Y	Y	
Ã	capital A tilde	Y	Y	
Ä	capital A dieresis or umlaut	Y	Y	
Å	capital A ring	Y	Y	
Æ	capital AE ligature	Y	Y	
Ç	capital C cedilla	Y	Y	
È	capital E grave	Y	Y	
É	capital E acute	Y	Y	
Ê	capital E circumflex	Y	Y	
Ë	capital E dieresis or umlaut	Y	Y	
Ì	capital I grave	Y	Y	
Í	capital I acute	Y	Y	
Î	capital I circumflex	Y	Y	
Ï	capital I dieresis or umlaut	Y	Y	
Ð	capital ETH	Y	Y	
Ñ	capital N tilde	Y	Y	
Ò	capital O grave	Y	Y	
Ó	capital O acute	Y	Y	

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
Ô	capital O circumflex	Y	Y	
Õ	capital O tilde	Y	Y	
Ö	capital O dieresis or umlaut	Y	Y	
×	multiplication sign	Y	Y	
Ø	capital O slash	Y	Y	
Ù	capital U grave	Y	Y	
Ú	capital U acute	Y	Y	
Û	capital U circumflex	Y	Y	
Ü	capital U dieresis or umlaut	Y	Y	
Ý	capital Y acute	Y	Y	
Þ	capital THORN	Y	Y	
ß	small sharp s, sz ligature	Y	Y	
à	small a grave	Y	Y	
á	small a acute	Y	Y	
â	small a circumflex	Y	Y	
ã	small a tilde	Y	Y	
ä	small a dieresis or umlaut	Y	Y	
å	small a ring	Y	Y	
æ	small ae ligature	Y	Y	
ç	small c cedilla	Y	Y	
è	small e grave	Y	Y	
é	small e acute	Y	Y	
ê	small e circumflex	Y	Y	
ë	small e dieresis or umlaut	Y	Y	

Character	Description	Supported in TXT ? (Y/N/ Substitution/ To escape)	Supported in XML ? (Y/N/ Substitution/ To escape)	Remark
ì	small i grave	Y	Y	
í	small i acute	Y	Y	
î	small i circumflex	Y	Y	
ï	small i dieresis or umlaut	Y	Y	
ð	small eth	Y	Y	
ñ	small n tilde	Y	Y	
ò	small o grave	Y	Y	
ó	small o acute	Y	Y	
ô	small o circumflex	Y	Y	
õ	small o tilde	Y	Y	
ö	small o dieresis or umlaut	Y	Y	
÷	division sign	Y	Y	
ø	small o slash	Y	Y	
ù	small u grave	Y	Y	
ú	small u acute	Y	Y	
û	small u circumflex	Y	Y	
ü	small u dieresis or umlaut	Y	Y	
ý	small y acute	Y	Y	
þ	small thorn	Y	Y	
ÿ	small y dieresis or umlaut	Y	Y	

Table 78: List of supported and non-supported characters

4. Addressing rules for Mail ID

Introduction

A Mail ID address can be deposited with bpost in 2 ways: in a structured (every field of the file contains only one detail) or unstructured way (more details of the same group in one field). Your database permitting, for optimal recognition it is best to send us the information in a structured way. If not, you can send us the addresses in an unstructured format. You must be aware that this might have a negative influence on recognition.

Component number	Code Description	Max Field Length
1	Greeting	10
2	First Name	42
3	Middle Name	20
4	Last Name	42
5	Suffix	10
6	Company Name	42
7	Department	42
8	Building	42
9	Address Line 1	42
12	House Number	12
13	Box Number	8
14	P.O. Box Number	42
15	Postal Code	12
16	City	30
17	ISO Country Code	2
18	Country Name	42
19	State	42
70-79	Reserved for customer use, verified but not used by bpost	70
90	Unstructured Name (01-05)	50
91	Unstructured Company/Department/Building (06-08)	50
92	Unstructured Street/House/Box (09-13, or 14)	50
93	Unstructured Post Code City (15-16)	50

Table 79: Address Components

The address is subdivided into different groups:

1. Individual recipient group, fields 1 to 5 or unstructured field 90
2. Organisation and geolocation group, fields 6 to 8 or unstructured field 91
3. Street, house number and box number group, fields 9 to 14 or unstructured field 92
4. Postcode and locality group, fields 15 and 16 or unstructured field 93
5. Country group, fields 17 to 19

Groups 3 and 4 are very important because they are the basis of the home address (in Belgium). Group 5, including the complete name of the destination country, is indispensable for international mail items. Under certain conditions, groups 1 and 2 can help bpost to clarify certain ambiguities in the recognition of addresses.

It is possible to use the structured way of recording for a certain group of fields and an unstructured way for another group. It is however not possible to use both a structured and unstructured way of recording within the same group.

For example:

- Permitted:

  92= Rue Courtejoie 17 bte 1

  15= 5590

  16= Ciney

- Not permitted:

  9= Rue Courtejoie

  92= 17 bte 1

  15= 5590

  16= Ciney

For a good interpretation of the addresses, it is extremely important that the correct fields are used. For instance, if a postcode is entered into the house number field, the system will not recognise the address. And when a box number is entered into the house number field, the address will not be read correctly and the mail item may be delayed. Addresses can be verified on the website via the link http://bpost.be/validationadresse.

Content of the address components groups

Individual recipient group, fields 1 to 5 or unstructured field 90

a. Structured

i. Field # 1 Greeting: the full or abbreviated title: Mr, Mrs, Ms, etc.

ii. Field # 2 First Name: the first name should not be abbreviated to avoid confusion

iii. Field # 3 Middle Name: this field is hardly ever used in Belgium

iv. Field # 4 Last Name: please fill in the complete last name

b. Unstructured: the information is best presented in the following order: 'Greeting + first name + last name', separated by a space and in compliance with the guidelines for the individual fields.

Organisation and geolocation group, fields 6 to 8 or unstructured field 91

a. Structured

i. Field # 6 Company Name: state the organisation or the company

ii. Field # 7 Department: state the department, if applicable

iii. Field # 8 Building (including floor, stairs, flat numbers): not to be confused with 'box number information' from group 31. Any details regarding the physical address must be entered in field 8,'Building'.

b. Unstructured: the information is best presented in the following order: 'company + department + building' separated by a space and in compliance with the guidelines for the individual fields.

Street, house number and box number group, fields 9 to 14 or unstructured field 92

a. Structured

i. Field # 9 Street 1: enter the type and name of the street here

- Use the right street type such as street, lane, boulevard, etc.
- The street type can only be abbreviated if absolutely necessary to keep the address on one line in the address box. The abbreviation should be done in compliance with the table below.

STREET NAMES IN FRENCH		STREET NAMES IN DUTCH		STREET NAMES IN GERMAN	
Avenue	Av.	straat	str.	Strasse	Str.
Boulevard	Bd	laan	ln	Allee	All.
Centre	Ctre	plein	pl.	Platz	Pl.
Place	Pl.	steenweg	stwg	Gewerbegebiet	GG.
Route	Rte	industriezone	I.Z.	Residenz	Res.
Square	Sq.	gebouw	geb.		
Zone Industrielle	Z.I.	square	sq.		
Chaussée	Chée				
Impasse	Imp.				

Table 80: Street types abbreviations list

- The street name must follow the street type immediately.
- The street name must only be abbreviated if absolutely necessary. In this case, it must be made sure that there can be no confusion with similar street names in the same municipality.
- Do not state the street name in several languages ('Rue de la Paix / Vredesstraat', 'Rue du Merlostraat').
- Avoid the use of punctuation.
- Avoid the use of special characters (such as "/", "#", "&", "§", "n°" or even brackets or quotation marks.
- Dates and cardinal numbers must be written as Arabic numerals (e.g. Rue du 11 novembre, rue des 4 saisons, etc).
- Exception: names of kings or popes. These names usually consist of a first name and an ordinal number. This number is written in Roman numerals. (e.g.: Rue du Roi Albert II, Rue du Pape Benoit XVI, etc).

ii. Field # 12 House Number: state the house number or the building number here. The number of the floor or corridor must not be mentioned in this field. If necessary, use field 8 for this.

- Compound numbers: use '-' to separate the numbers. Do not use a space or a '/'.
  For example: Avenue Louise 43-45
- Numeric extension of building numbers: not to be confused with the box number of a building (you can write this in the dedicated field or preceded by 'box').
  For example: Dieweg 61/2 (no spaces)
- Alphabetic extension of building numbers: not to be confused with box numbers.
  For example: Allee de la Meuse 1A

iii. Field # 13 Box Number: state the box number here. Indications such as 'bus', 'bte', 'box' or 'boite', 'b', 'Bt', '#', '-', '/'… are not permitted here because this information is inherent to this field. The indication of the floor or corridor number must not be mentioned in this field. If necessary, use field 8 for this.

iv. Field # 14 Post Office Box Number: write the number of the post box or poste restante here. The mention 'Post box' must not precede the number because this information is inherent to the field.

b. Unstructured: the information must be stated in the order 'street + number', separated by a space and if necessary with the mention 'box' + box number. All of this needs to be in compliance with the guidelines for the individual fields. In Dutch, the type of street follows its name, in French, the type of street precedes its name.

- The house number follows the street name immediately.
- Do not use punctuation (full stop, comma or other) to separate the different elements from one another.
- The mention of the floor or corridor number is not permitted in this field. If necessary, this can be stated in field 91.
- If the mail item is addressed to a building with box numbers, the box number is preceded by 'bus', 'bte', 'box' or 'boite'.
- Mentions such as 'b', 'Bt', '#', '-', '/'… are not permitted.
- If a PO Box number is mentioned in an unstructured field, the number must be stated AND must be preceded by the words 'Post Office box'.

  For example: Post Office box 12

Postcode and place group, fields 15 and 16 or unstructured field 93

a. Structured

i. Field # 15 Postal Code: insert a valid Belgian postcode with 4 numbers (for a list of all postcodes, go to www.bpost.be, click on 'Particulieren' > 'Klantendienst' > 'Postcodes') or a foreign postcode.

- Never put 'B' or 'BE' in front of a postcode.
- Never use the ISO code (F-, FR…) for international addresses

ii. Field # 16 City: state the name of the municipality without mention of anything else (such as hamlets or boroughs).

b. Unstructured: the information must be stated in the order 'postcode + municipality', separated by a space and in compliance with the guidelines for the individual fields.

Country group, fields 17 to 19

Mail ID is currently not used for international mail items. However, it is not possible to use these fields. The name of the country must not be stated for mail items with a Belgian destination. For mail items with a foreign destination, however, the complete name of the country must be stated. The ISO country code is not enough.

a. Structured

i. Field # 17 ISO Country Code: optional field. If you use this field, the code must be valid. (http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2_country_code#Officially_assigned_code_elements)

ii. Field # 18 Country Name: the complete country name must be stated in one of the 3 national languages or in English.

iii. Field # 19 State: for some countries, it can be useful to state in a state or province.

b. Unstructured: it is essential that you state the name of the country in compliance with the guidelines for the individual fields.

5. Comprehensive Examples

This chapter presents some file examples for each sort of product developed in this document (deposit files, MAIL ID files, Round & Sequence files and OptiAddress files). For each of them, there are examples of a Request, an Acknowledgement and a Response files, in TXT and in XML, with before a possible file name for this file.

5.1. Deposit files

Deposit Request

XML format

- File name : EMP_0100_12345678_EXMPL-ERR1_121106164945_0RQ.XML
- Content :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DepositRequest>
   <Context requestName="DepositRequest" dataset="M004_MPA" sender="12345678"
receiver="EMP" version="0100"/>
   <Header customerId="12345678" accountId="456987" mode="P">
      <Files>
        <RequestProps customerFileRef="exmpl-err1"/>
        <ResponseProps format="XML" compressed="Y" encrypted="N" transmissionMode="HTTP"/>
      </Files>
      <CustomerRefs>
        <CustomerRef key="custRef1" value="this is a customer private value"/>
        <CustomerRef key="custRef2" value="this is a customer private value"/>
      </CustomerRefs>
   </Header>
   <DepositCreate seq="1" depositIdentifier="CME79" depositIdentifierType="depositRef"
mailingRef="mailingRef631">
      <Contacts>
        <Contact seq="1" firstName="Lucien" lastName="Dupont" email="lucien.dupont@machin.be"
lang="fr" phone="+32 4 897.45.48" fax="+32 4 897.45.49"
mobile="+32 487 12.34.56"/>
        <Contact seq="2" email="dominique.provoost@machin.be" lang="fr"/>
      </Contacts>
      <Contract billTo="65432112" depositor="65432112" invoiceGrouping="XYZ123"/>
      <Deposit date="2005-02-18" modelName="modFN024" modelPortalUserName="john"
invoiceRef="my reference" meteringNumber="B-008" router="myRouter" formByMail="Y"
autoValidate="N" description="the description of the deposit">
        <Items>
               <Item seq="1">
                      <Characteristics>
                             <Characteristic key="SpecialCharacteristic" value="SPEC"/>
                      </Characteristics>
                      <Quantities>
                             <Quantity unit="PCE" value="12000"/>
                      </Quantities>
                      <Prepayments>
                             <Prepayment key="MeteringPrice" value="4800"/>
                      </Prepayments>
               </Item>
```

```
                <Item seq="2">
                        <Quantities>
                                <Quantity unit="PCE" value="4000"/>
                        </Quantities>
                </Item>
        </Items>
        <ItemCount value="2"/>
        <Options>
                <Option id="651">
                        <OptionQuantities>
                                <OptionQuantity unit="PCE" value="16000"/>
                        </OptionQuantities>
                </Option>
                <Option id="3587"/>
        </Options>
      </Deposit>
   </DepositCreate>
   <DepositUpdate seq="2" depositIdentifier="CME62" depositIdentifierType="depositRef">
      <Contacts>
        <Contact seq="1" email="dominique.provoost@machin.be" lang="fr"/>
      </Contacts>
      <Contract billTo="654321" depositor="65432112" invoiceGrouping="XYZ123"/>
      <Deposit date="2005-02-21" modelName="modFN023" modelPortalUserName="john"
invoiceRef="my reference2" formByMail="Y" autoValidate="Y" description="the description of the
deposit">
        <Items>
                <Item seq="1">
                        <Quantities>
                                <Quantity unit="PCE" value="11000"/>
                        </Quantities>
                </Item>
        </Items>
        <ItemCount value="1"/>
      </Deposit>
   </DepositUpdate>
   <DepositDelete seq="3" depositIdentifier="CME55" depositIdentifierType="depositRef"/>
   <DepositValidate seq="4" depositIdentifier="123157"
depositIdentifierType="tmpDepositNumber"/>
   <DepositValidate seq="5" depositIdentifier="ABC007"/>
   <DepositCreate seq="6" depositIdentifier="CME80" depositIdentifierType="depositRef">
      <Contacts>
        <Contact seq="1" email="dominique.provoost@machin.be" lang="fr"/>
      </Contacts>
      <Contract billTo="654321" invoiceGrouping="XYZ123"/>
      <Deposit date="2005-03-01" modelName="modFN023" modelPortalUserName="john"
invoiceRef="my reference 3" formByMail="Y" autoValidate="Y" description="the description of
the deposit">
        <Items>
                <Item seq="1">
                        <Quantities>
                                <Quantity unit="PCE" value="53000"/>
                        </Quantities>
                </Item>
        </Items>
        <ItemCount value="1"/>
      </Deposit>
   </DepositCreate>
   <DepositDelete seq="7" depositIdentifier="CME55" depositIdentifierType="depositRef"/>
</DepositRequest>
```

TXT format

- File name : EMP_0100_12345678_EXMPL-ERR1_121106164945_0RQ.TXT
- Content :

```
        Context|DepositRequest|M004_MPA|12345678|EMP|0100
Header|12345678|456987|P
```

```
RequestProps|exmpl-err1
ResponseProps|TXT|Y|N|HTTP
CustomerRef|custRef1|this is a customer private value
CustomerRef|custRef2|this is a customer private value
DepositCreate|1|CME79|depositRef|mailingRef631
Contact|1|Lucien|Dupont|lucien.dupont@machin.be|fr|+32 4 897.45.48|+32 4 897.45.49|+32 487
12.34.56
Contact|2|||dominique.provoost@machin.be|fr|||
Contract|65432112|65432112|XYZ123
Deposit|2005-02-18|modFN024|john|my reference|B-008|myRouter|Y|N|the description of the
deposit
Item|1
Characteristic|SpecialCharacteristic|SPEC
Quantity|PCE|12000
Prepayment|MeteringPrice|4800
Item|2
Quantity|PCE|4000
ItemCount|2
Option|651
OptionQuantity|PCE|16000
Option|3587
DepositUpdate|2|CME62|depositRef||
Contact|1|||dominique.provoost@machin.be|fr|||
Contract|654321|654321|XYZ123
Deposit|2005-02-21|modFN023|john|my reference2|||Y|Y|the description of the deposit
Item|1
Quantity|PCE|11000
ItemCount|1
DepositDelete|3|CME55|depositRef||
DepositValidate|4|123157|tmpDepositNumber|
DepositValidate|5|ABC007|depositRef||
DepositCreate|6|CME80|depositRef|
Contact|1|||dominique.provoost@machin.be|fr|||
Contract|654321|654321|XYZ123
Deposit|2005-03-01|modFN023|john|my reference 3|||Y|Y|the description of the deposit
Item|1
Quantity|PCE|53000
ItemCount|1
DepositDelete|7|456987|CME55|depositRef||56565
```

Deposit Acknowledgement

XML format

- File name : EMP_0100_12345678_EXMPL-ERR1_121106165045_1AK.XML
- Content :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<RequestAck>
   <FileReceived fileName="EMP_0100_12345678_EXMPL-GOOD_050224134046_0RQ.XML"
 timeStamp="2005-12-17T09:30:47"/>
</RequestAck>
```

TXT format

- File name : EMP_0100_12345678_EXMPL-ERR1_121106165045_1AK.TXT
- Content :

```
FileReceived|EMP_0100_12345678_EXMPL-GOOD_050224134046_0RQ.TXT|2001-12-17T09:30:47
```

bpost business

Deposit response

Imagine a customer sends the following request (containing errors)

- File name : EMP_0100_12345678_EXMPL-ERR1_121106164945_0RQ.XML
- Content :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DepositRequest>
   <Context requestName="DepositRequest" dataset="M004_MPA" sender="12345678"
receiver="EMP" version=”0100”/>
   <Header customerId="12345678” accountId="456987" mode="P">
      <Files>
        <RequestProps customerFileRef="exmpl-err1"/>
        <ResponseProps format="XML" compressed="Y" encrypted="N" transmissionMode="HTTP"/>
      </Files>
      <CustomerRefs>
        <CustomerRef key="custRef1" value="this is a customer private value"/>
        <CustomerRef key="custRef2" value="this is a customer private value"/>
      </CustomerRefs>
   </Header>
   <DepositCreate seq="1" depositIdentifier="CME79" depositIdentifierType="depositRef"
mailingRef="mailingRef631">
      <Contacts>
        <Contact seq="1" firstName="Lucien" lastName="Dupont" email="lucien.dupont@machin.be"
lang="fr" phone="+32 4 897.45.48" fax="+32 4 897.45.49"
mobile="+32 487 12.34.56"/>
        <Contact seq="2" email="dominique.provoost@@machin.be" lang="fr"/>
      </Contacts>
      <Contract billTo="65432112" depositor="65432112" invoiceGrouping="XYZ123"/>
      <Deposit date="2005-02-18" ="modFN024" modelPortalUserName="john" invoiceRef="my
reference" meteringNumber="B-008" router="myRouter" formByMail="Y" autoValidate="N"
description="the description of the deposit">
        <Items>
               <Item seq="1">
                      <Characteristics>
                             <Characteristic key="SpecialCharacteristic" value="SPEC"/>
                      </Characteristics>
                      <Quantities>
                             <Quantity unit="PCE" value="12000"/>
                      </Quantities>
                      <Prepayments>
                             <Prepayment key="MeteringPrice" value="4800"/>
                      </Prepayments>
               </Item>
               <Item seq="2">
                      <Quantities>
                             <Quantity unit="PCE" value="4000"/>
                      </Quantities>
               </Item>
        </Items>
        <ItemCount value="2"/>
        <Options>
               <Option id="651XXX">
                      <OptionQuantities>
                             <OptionQuantity unit="PCE" value="16000"/>
                      </OptionQuantities>
               </Option>
               <Option id="3587"/>
        </Options>
      </Deposit>
   </DepositCreate>
   <DepositUpdate seq="2" depositIdentifier="CME62" depositIdentifierType="depositRef">
      <Contacts>
        <Contact seq="1" email="dominique.provoost@machin.be" lang="fr"/>
      </Contacts>
      <Contract billTo="654321" depositor="65432112" invoiceGrouping="XYZ123"/>
      <Deposit date="2005-02-21" modelName="modFN023" modelPortalUserName="john"
invoiceRef="my reference2" formByMail="Y" autoValidate="Y" description="the description of the
deposit">
```

```
        <Items>
                <Item seq="1">
                        <Quantities>
                                <Quantity unit="PCE" value="11000"/>
                        </Quantities>
                </Item>
        </Items>
        <ItemCount value="1"/>
      </Deposit>
   </DepositUpdate>
   <DepositDelete seq="3" depositIdentifier="CME55" depositIdentifierType="depositRef"/>
   <DepositValidate seq="4" depositIdentifier="123157"
depositIdentifierType="tmpDepositNumber"/>
   <DepositValidate seq="5" depositIdentifier="ABC007" depositIdentifierType="depositRef"/>
   <DepositCreate seq="6" depositIdentifier="CME80" depositIdentifierType="depositRef">
      <Contacts>
        <Contact seq="1" email="dominique.provoost@machin.be" lang="fr"/>
      </Contacts>
      <Contract billTo="654321" depositor="65432112" invoiceGrouping="XYZ123"/>
      <Deposit date="2005-03-01" modelName="modFN023" modelPortalUserName="john"
invoiceRef="my reference 3" formByMail="Y" autoValidate="Y" description="the description of
the deposit">
        <Items>
                <Item seq="1">
                        <Quantities>
                                <Quantity unit="PCE" value="53000"/>
                        </Quantities>
                </Item>
        </Items>
        <ItemCount value="1"/>
      </Deposit>
   </DepositCreate>
   <DepositDelete seq="7" depositIdentifier="CME55" depositIdentifierType="depositRef"/>
</DepositRequest>
```

XML format

The Response file for this request would be:

- File name : EMP_0100_12345678_EXMPL-ERR1_121106165955_2RS.XML
- Content :

```
<?xml version="1.0" encoding="UTF-8"?>
<DepositResponse>
   <Context requestName="DepositResponse" dataset="M004_MPA" sender="EMP"
receiver="12345678" version="0100"/>
   <Header customerId="12345678">
      <CustomerRefs>
        <CustomerRef key="custRef1" value="this is a customer private value"/>
        <CustomerRef key="custRef2" value="this is a customer private value"/>
      </CustomerRefs>
      <Files>
        <RequestProps fileName="EMP_0100_12345678_EXMPL-ERR1_050224134253_2RS.XML"
customerFileRef="exmpl-err1"/>
      </Files>
   </Header>
   <DepositCreate seq="1" depositRef="CME79">
      <Status code="999"/>
      <CustomerRefs>
        <CustomerRef key="custRef1_1" value="my cust ref 1.1"/>
        <CustomerRef key="custRef1_2" value="my cust ref 1.2"/>
      </CustomerRefs>
      <Replies>
        <Reply seq="1">
<XPath>
/DepositRequest/DepositCreate[@seq="1"]/Contacts/Contact[@seq="2"]
</XPath>
                <Locations>
                        <Location tagName="Contact" attributeName="seq" attributeValue="2"/>
```

```
                </Locations>
                <Messages>
                        <Message code="123" severity="WARN">
                                <Description>E-mail address 'dominique@@machine.be' is
invalid</Description>
                        </Message>
                </Messages>
        </Reply>
        <Reply seq=”2”>
<XPath>
/DepositRequest/DepositCreate[@seq="1"]/Deposit/Options/[@id="651XXX"]
</XPath>
                <Locations>
                        <Location tagName="Option" attributeName="id" attributeValue="651XXX"/>
                </Locations>
                <Messages>
                        <Message code="789" severity="FATAL">
                                <Description>Option id '651XXX' is unknown</Description>
                        </Message>
                </Messages>
        </Reply>
      </Replies>
   </DepositCreate>
   <DepositUpdate seq="2" depositRef="CME62">
      <Status code="100"/>
   </DepositUpdate>
   <DepositDelete seq="3" depositRef="CME55">
      <Status code="100"/>
   </DepositDelete>
   <DepositValidate seq="4" depositRef="ABC006XXX">
      <Status code="999"/>
      <Replies>
        <Reply seq=”1”>
<XPath>
/DepositRequest/DepositValidate[@seq="4"]
</XPath>
                <Messages>
                        <Message code="168" severity="FATAL">
                                <Description>Deposit reference 'ABC006XXX' is
invalid</Description>
                        </Message>
                </Messages>
        </Reply>
      </Replies>
   </DepositValidate>
   <DepositValidate seq="5" depositRef="ABC007">
      <Status code="100"/>
      <Replies>
        <Reply seq=”1”>
<XPath>
/DepositRequest/DepositValidate[@seq="5"]
                </XPath>
                <Messages>
                        <Message code="0" severity="INFO">
                                <MessageContents>
                                        <MessageContent key="depositNumber" value="25487"/>
                                </MessageContents>
                        </Message>
                </Messages>
        </Reply>
      </Replies>
   </DepositValidate>
   <DepositCreate seq="6" depositRef="CME80">
      <Status code="100"/>
      <Replies>
        <Reply seq=”1”>
<XPath>
/DepositRequest/DepositCreate[@seq="6"]
</XPath>
                <Messages>
                        <Message code="0" severity="INFO">
                                <MessageContents>
```

```
                                    <MessageContent key="tmpDepositNumber" value="123457"/>
                                </MessageContents>
                            </Message>
                        </Messages>
                    </Reply>
                </Replies>
            </DepositCreate>
            <DepositDelete seq="7" depositRef="CME55">
                <Status code="100"/>
            </DepositDelete>
        </DepositResponse>
```

Note:

This is a sample deposit Response file. The used message codes and status codes are arbitrarily chosen.

This Response file contains the following information:

- DepositCreate with seq=1: the action was not successful (status 999)
- Warning: the second Contact email address is invalid (indeed, contains 2 "@" characters)
- Fatal: the deposit has an option with an unknown id '651XXX'
- DepositUpdate with seq=2: the action was successful (status 100)
- DepositDelete with seq=3: the action was successful (status 100)
- DepositValidate with seq=4: the action was not successful (status 999)
- Fatal: invalid deposit reference
- DepositValidate with seq=5: the action was successful (status 100)
- The final deposit number = 25487.
- DepositCreate with seq=6: the action was successful (status 100)
- The temporary deposit number = 123457
- DepositDelete with seq=7: the action was not successful (status 999)
- Fatal: both tmpDepositNumber and depositNumber cannot filled in

TXT Format

The txt format of this Response file would be:

- File name : EMP_0100_12345678_EXMPL-ERR1_121106165955_2RS.TXT
- Content :

```
Context|DepositResponse|M004_MPA|EMP|12345678|0100
Header|12345678
CustomerRef|custRef1|this is a customer private value
CustomerRef|custRef2|this is a customer private value
RequestProps|EMP_0100_12345678_EXMPL-ERR1_050224134253_2RS.TXT|exmpl-err1
DepositCreate|1|CME79
Status|999
CustomerRef|custRef1_1|my cust ref 1.1
CustomerRef|custRef1_2|my cust ref 1.2
Reply|1
Location|Contact|2
Message|123|WARN
Description|E-mail address 'dominique@@machine.be' is invalid
Reply|2
Location|Option|651XXX
Message|789|FATAL
Description|Option id '651XXX' is unknown
```

```
DepositUpdate|2|CME62
Status|100
DepositDelete|3|CME55
Status|100
DepositValidate|4|ABC006XXX
Status|999
Reply|1
Message|168|FATAL
Description|Deposit reference 'ABC006XXX' is invalid
DepositValidate|5|ABC007
Status|100
Reply|1
Message|0|INFO
MessageContent|depositNumber|25487
DepositCreate|6|CME80
Status|100
Reply|1
Message|0|INFO
MessageContent|tmpDepositNumber|123457
DepositDelete|7|CME55
Status|999
Reply|1
Message|587|FATAL
Description|Giving both tmpDepositNumber and depositNumber is prohibited
```

5.2. Mailing list files

Mailing Request

XML format

- File name : MID_0200_12345678_EXMPL-GOOD_121106164945_0RQ.XML
- Content :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MailingRequest>
   <Context requestName="MailingRequest" dataset="M037_MID" sender="12345678" receiver=”MID"
version=”0200”/>
   <Header customerId="12345678” accountId="81279" mode="P">
      <Files>
        <RequestProps customerFileRef=" EXMPL-GOOD"/>
        <ResponseProps format="XML" compressed="Y" encrypted="N" transmissionMode="HTTP"/>
      </Files>
      <CustomerRefs>
        <CustomerRef key="CK" value="this is a customer private value"/>
        <CustomerRef key="PV" value="this is another customer private value"/>
      </CustomerRefs>
   </Header>
   <MailingCreate seq="1" mailingRef="mailref001" depositIdentifier="abcd678"
depositIdentifierType="depositRef" genMID="N" genPSC="N" expectedDeliveryDate="2020-10-27">
      <FileInfo type="MID2"/>
      <Format responseSortingMode = "CU">Small</Format>
      <PresortingCodeFile/>
      <Contacts>
<Contact seq="1" firstName="lucien" lastName="dupont" email="lucien.dupont@machin.be"
lang="fr" phone="(032)13547686" fax="(032)765323" mobile="(0478)675.164"/>
        <Contact seq="2" email="dominique.provoost@machin.be" lang="fr"/>
      </Contacts>
      <CustomerRefs>
        <CustomerRef key="AB" value="this is a customer private value"/>
```

```
      <CustomerRef key="AC" value="this is another customer private value"/>
    </CustomerRefs>
    <Items>
      <Item seq="1" lang="fr" midNum="10123453332521" priority="P">
            <Comps>
                  <Comp code="2" value="Luc"/>
                  <Comp code="4" value="Dupont"/>
                  <Comp code="9" value="Rue de l'eglise"/>
                  <Comp code="12" value="123"/>
                  <Comp code="15" value="1000"/>
                  <Comp code="16" value="Bruxelles"/>
            </Comps>
      </Item>
      <Item seq="2" lang="nl" midNum="10123453332522" priority="P">
            <Comps>
                  <Comp code="2" value="Maria"/>
                  <Comp code="4" value="Peeters"/>
                  <Comp code="9" value="Koningsstraat"/>
                  <Comp code="12" value="10"/>
                  <Comp code="13" value="2"/>
                  <Comp code="15" value="1000"/>
                  <Comp code="16" value="Brussel"/>
            </Comps>
      </Item>
      <Item seq="3" lang="nl" midNum="10123453332523" priority="P">
            <Comps>
                  <Comp code="2" value="James"/>
                  <Comp code="4" value="Smith"/>
                  <Comp code="9" value="Belliardstraat"/>
                  <Comp code="12" value="105"/>
                  <Comp code="15" value="1000"/>
                  <Comp code="16" value="Brussel"/>
            </Comps>
      </Item>
    </Items>
    <ItemCount value="3"/>
  </MailingCreate>

  <MailingDelete seq="2" mailingRef="Atp789"/>

</MailingRequest>
```

TXT format

- File name : MID_0200_12345678_EXMPL-GOOD_121106164945_0RQ.TXT
- Content :

```
Context|MailingRequest|M037_MID|12345678|MID|0200
Header|12345678|81279|P
RequestProps|EXMPL-GOOD
ResponseProps|TXT|Y|N|HTTP
CustomerRef|CK|this is a customer private value
CustomerRef|PV|this is another customer private value
MailingCreate|1|mailref001|abcd678|depositRef|N|N|2013-10-27
FileInfo|MID2
Format|Small|
Contact|1|lucien|dupont|lucien.dupont@machin.be|fr|(032)13547686|(032)765323|(0478)675.164
Contact|2|||dominique.provoost@machin.be|fr|||
CustomerRef|AB|this is a customer private value
CustomerRef|AC|this is another customer private value
Item|1|fr|10123453332521||P
Comp|2|Luc
Comp|4|Dupont
Comp|9|Rue de l'eglise
Comp|13|123
Comp|15|1000
Comp|16|Bruxelles
Item|2|nl|10123453332522||P
```

```
Comp|2|Maria
Comp|4|Peeters
Comp|9|Koningsstraat
Comp|13|10
Comp|15|1000
Comp|16|Brussel
Item|3|nl|10123453332523||P
Comp|2|James
Comp|4|Smith
Comp|9|Belliardstraat
Comp|13|105
Comp|15|1000
Comp|16|Brussel
ItemCount|3
MailingDelete|2|Atp789
```

XLS Format

In the XLS(X) and CSV file formats, the file is the same for all the mailing files (e.g. for MAIL ID deposit and Round and sequence deposit).

- File name : EXMPL-GOOD.XLS
- Content :

	A	C	E	J	M	N	P	Q	Y	AB
1	SEQ	FIRST_NAME	LAST_NAME	ADDRESS_LINE_1	HOUSE_NUMBER	BOX_NUMBER	POSTAL_CODE	CITY	MIDNUMBER	PRIORITY
2	1	Luc	Dupont	Rue de l'eglise	123		1000	Brussel	10123453332521	P
3	2	Maria	Peeters	Koningsstraat	10	2	1000	Brussel	10123453332522	P
4	3	James	Smith	Belliardstraat	105		1000	Brussel	10123453332523	P

Figure 42: Mailing Request file – XLS Example

Mailing Acknowledgement

XML format

- File name : MID_0200_12345678_EXMPL-GOOD_121106165045_1AK.XML
- Content :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<RequestAck>
   <FileReceived fileName="MID_0200_12345678_EXMPL-GOOD_121017134046_0RQ.XML"
 timeStamp="2012-10-17T09:30:47"/>
</RequestAck>
```

TXT format

- File name : MID_0200_12345678_EXMPL-GOOD_121106165045_1AK.TXT
- Content :

```
FileReceived|MID_0200_12345678_EXMPL-GOOD_121017134046_0RQ.TXT|2012-10-17T09:30:47
```

Mailing Response

Imagine a customer sends the following request (containing errors):

bpost business

- File name : MID_0200_12345678_EXMPL-ERR1_121106164945_0RQ.XML
- Content :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MailingRequest>
   <Context requestName="MailingRequest" dataset="M037_MID" sender="12345678"
receiver="MID" version="0200"/>
   <Header customerId="12345678" accountId="81279" mode="P">
      <Files>
        <RequestProps customerFileRef="EXMPL-ERR1"/>
        <ResponseProps format="XML" compressed="Y" encrypted="N" transmissionMode="HTTP"/>
      </Files>
      <CustomerRefs>
        <CustomerRef key="CK" value="this is a customer private value"/>
        <CustomerRef key="PV" value="this is another customer private value"/>
      </CustomerRefs>
   </Header>
   <MailingCreate seq="1" mailingRef="mr1" depositIdentifier="abcd678"
depositIdentifierType="depositRef" genMID="N" genPSC="N" expectedDeliveryDate="2020-10-27">
      <FileInfo type="MID2"/>
      <Format>Small</Format>
      <PresortingCodeFile/>
<Contacts>
        <Contact seq="1" firstName="lucien" lastName="dupont" email="lucien.dupont@machin.be"
lang="fr" phone="(032)13547686" fax="(032)765323" mobile="(0478)675.164"/>
        <Contact seq="2" email="dominique.provoost@@machin.be" lang="fr"/>
      </Contacts>
<CustomerRefs>
        <CustomerRef key="AB" value="this is a customer private value"/>
        <CustomerRef key="AC" value="this is another customer private value"/>
      </CustomerRefs>
      <Items>
        <Item seq="1" lang="fr" midNum="10123453332531" priority="P">
               <Comps>
                      <Comp code="9" value="Rue de l'eglise"/>
                      <Comp code="13" value="10"/>
                      <Comp code="15" value="1000"/>
                      <Comp code="16" value="Bruxelles"/>
               </Comps>
        </Item>
        <Item seq="2" lang="nl" midNum="10123453332531" priority="P">
               <Comps>
                      <Comp code="9" value="Koningsstraat"/>
                      <Comp code="13" value="10"/>
                      <Comp code="15" value="1000"/>
                      <Comp code="16" value="Brussel"/>
               </Comps>
        </Item>
        <Item seq="3" lang="nl" midNum="10123453332532" priority="P">
               <Comps>
                      <Comp code="9" value="Belliardstraat"/>
                      <Comp code="13" value="105"/>
                      <Comp code="15" value="1000"/>
                      <Comp code="16" value="Brussel"/>
               </Comps>
        </Item>
      </Items>
      <ItemCount value="3"/>
   </MailingCreate>
   <MailingCreate seq="2" mailingRef="mr2" depositIdentifier="abcd678"
depositIdentifierType="depositRef" genMID="N" genPSC="N" expectedDeliveryDate="2020-10-29">
      <FileInfo type="MID2"/>
      <Format>Small</Format>
      <PresortingCodeFile/>
      <Contacts>
        <Contact seq="1" firstName="lucien" lastName="dupont" email="lucien.dupont@machin.be"
lang="fr" phone="(032)13547686" fax="(032)765323"/>
      </Contacts>
      <Items>
        <Item seq="1" lang="nl" midNum="10123453332533" psCode="G-W1-L2" priority="P">
               <Comps>
```

```
                        <Comp code="9" value="Suikerstraat"/>
                        <Comp code="13" value="28"/>
                        <Comp code="15" value="9050"/>
                        <Comp code="16" value="Zelzate"/>
                </Comps>
        </Item>
        <Item seq="2" lang="nl" midNum="10123453332534" psCode="B-W1-L8" priority=”P”>
                <Comps>
                        <Comp code="9" value="Watstraat"/>
                        <Comp code="13" value="10"/>
                        <Comp code="15" value="1030"/>
                        <Comp code="16" value="Brussel"/>
                </Comps>
        </Item>
        <Item seq="3" lang="nl" midNum="10123453332535" psCode="B-W1-L8" priority=”P”>
                <Comps>
                        <Comp code="9" value="Belliardstraat"/>
                        <Comp code="13" value="15"/>
                        <Comp code="15" value="1000"/>
                        <Comp code="16" value="Brussel"/>
                </Comps>
        </Item>
      </Items>
      <ItemCount value="3"/>
   </MailingCreate>
   <MailingCreate seq="3" mailingRef="mr3" depositIdentifier="abcd678"
depositIdentifierType="depositRef" genMID="N" genPSC="N" expectedDeliveryDate="2020-10-29">
      <FileInfo type="MID2"/>
      <Format>Small</Format>
      <PresortingCodeFile/>
      <Contacts>
        <Contact seq="1" firstName="lucien" lastName="dupont" email="lucien.dupont@machin.be"
lang="fr" phone="(032)13547686" fax="(032)765323"/>
      </Contacts>
      <Items>
        <Item seq="1" lang="nl" psCode="G-W1-L1" priority="P">
                <Comps>
                        <Comp code="9" value="Veldstraat"/>
                        <Comp code="13" value="3"/>
                        <Comp code="15" value="9000"/>
                        <Comp code="16" value="Gent"/>
                </Comps>
        </Item>
        <Item seq="2" lang="nl" psCode="G-W1-L1" priority=”P”>
                <Comps>
                        <Comp code="9" value="Lange Violettestraat"/>
                        <Comp code="13" value="8"/>
                        <Comp code="15" value="9000"/>
                        <Comp code="16" value="Gent"/>
                </Comps>
        </Item>
        <Item seq="3" lang="nl" psCode="G-W1-L1" priority=”P”>
                <Comps>
                        <Comp code="9" value="Kouter"/>
                        <Comp code="13" value="10"/>
                        <Comp code="15" value="9000"/>
                        <Comp code="16" value="Gent"/>
                </Comps>
        </Item>
      </Items>
      <ItemCount value="3"/>
   </MailingCreate>
   <MailingDelete seq="4" mailingRef="Atp789">
<Contacts>
<Contact seq="1" firstName="lucien" lastName="dupont" email="lucien.dupont@machin.be"
lang="fr" phone="(032)13547686" fax="(032)765323" mobile="(0478)675.164"/>
        <Contact seq="2" email="dominique.provoost@machin.be" lang="fr"/>
      </Contacts>
<CustomerRefs>
        <CustomerRef key="D1" value="this is a customer private value"/>
        <CustomerRef key="D2" value="this is another customer private value"/>
      </CustomerRefs>
```

```
   </MailingDelete>
</MailingRequest>
```

XML format

The Response file for this request would be:

- File name : MID_0200_12345678_EXMPL-ERR1_121106165955_2RS.XML
- Content :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MailingResponse>
<Context requestName="MailingResponse" dataset="M037_MID" sender="MID" receiver="12345678"
version="0200"/>
   <Header customerId="12345678”>
      <CustomerRefs>
        <CustomerRef key="CK" value="this is a customer private value"/>
        <CustomerRef key="PV" value="this is another customer private value"/>
      </CustomerRefs>
      <Files>
        <RequestProps fileName="MID_0200_12345678_EXMPL-ERR1_121106165955_2RS.XML"
customerFileRef="EXMPL-ERR1"/>
      </Files>
   </Header>
   <MailingCreate seq="1" mailingRef="mr1">
      <Status code="999"/>
      <CustomerRefs>
        <CustomerRef key="AB" value="this is a customer private value"/>
        <CustomerRef key="AC" value="this is another customer private value"/>
      </CustomerRefs>
      <Replies>
        <Reply seq="1">

   <XPath>/MailingRequest/MailingCreate[@seq="1"]/Contacts/Contact[@seq="2"]</XPath>
               <Locations>
                      <Location tagName="Contact" attributeName="seq" attributeValue="2"/>
               </Locations>
               <Messages>
                      <Message code="MID-3000" severity="WARN"/>
               </Messages>
        </Reply>
        <Reply seq="2">
               <XPath>/MailingRequest/MailingCreate[@seq="1"]/Items/Item[@seq="2"]</XPath>
               <Locations>
                      <Location tagName="Item" attributeName="seq" attributeValue="2"/>
               </Locations>
               <Messages>
                      <Message code="MID-3010" severity="FATAL"/>
               </Messages>
        </Reply>
      </Replies>
   </MailingCreate>
   <MailingCreate seq="2" mailingRef="mr2">
      <Status code="100"/>
      <Replies>
        <Reply seq=”1”>
<XPath>
/MailingRequest/MailingCreate[@seq="2"]
               </XPath>
               <Messages>
                      <Message code="MID-4040" severity="INFO">
                             <MessageContents>
                                    <MessageContent key="complianceRate" value="33.3%"/>
                             </MessageContents>
                      </Message>
               </Messages>
        </Reply>
        <Reply seq="2">
               <XPath>/MailingRequest/MailingCreate[@seq="2"]/Items/Item[@seq="1"]</XPath>
```

```
                <Locations>
                        <Location tagName="Item" attributeName="seq" attributeValue="1"/>
                </Locations>
                <Messages>
                        <Message code="MID-4000" severity="WARN">
                                <MessageContents>
                                        <MessageContent key="compCode" value="9"/>
                                </MessageContents>
                        </Message>
                        <Message code="MID-4000" severity="WARN">
                                <MessageContents>
                                        <MessageContent key="compCode" value="15"/>
                                </MessageContents>
                        </Message>
                </Messages>
        </Reply>
        <Reply seq="3">
                <XPath>/MailingRequest/MailingCreate[@seq="2"]/Items/Item[@seq="2"]</XPath>
                <Locations>
                        <Location tagName="Item" attributeName="seq" attributeValue="2"/>
                </Locations>
                <Messages>
                        <Message code="MID-4020" severity="ERROR"/>
                </Messages>
        </Reply>
        <Reply seq="3">
                <XPath>/MailingRequest/MailingCreate[@seq="2"]/Items/Item[@seq="3"]</XPath>
                <Locations>
                        <Location tagName="Item" attributeName="seq" attributeValue="3"/>
                </Locations>
                <Messages>
                        <Message code="MID-3020" severity="WARN"/>
                </Messages>
        </Reply>
      </Replies>
   </MailingCreate>
   <MailingCreate seq="3" mailingRef="mr3">
      <Status code="100"/>
      <Replies>
        <Reply seq=”1”>
<XPath>
/MailingRequest/MailingCreate[@seq="3"]
                </XPath>
                <Messages>
                        <Message code="MID-4040" severity="INFO">
                                <MessageContents>
                                        <MessageContent key="complianceRate" value="100.0%"/>
                                </MessageContents>
                        </Message>
                </Messages>
        </Reply>
        <Reply seq="2">
                <XPath>/MailingRequest/MailingCreate[@seq="3"]/Items/Item[@seq="1"]</XPath>
                <Locations>
                        <Location tagName="Item" attributeName="seq" attributeValue="1"/>
                </Locations>
                <Messages>
                        <Message code="MID-4030" severity="INFO">
                                <MessageContents>
                                        <MessageContent key="midNum" value="11123458025111"/>
                                </MessageContents>
                        </Message>
                </Messages>
        </Reply>
        <Reply seq="3">
<XPath>/MailingRequest/MailingCreate[@seq="3"]/Items/Item[@seq="2"]</XPath>
                <Locations>
                        <Location tagName="Item" attributeName="seq" attributeValue="2"/>
                </Locations>
                <Messages>
                        <Message code="MID-4030" severity="INFO">
                                <MessageContents>
```

```
                                        <MessageContent key="midNum" value="11123458025112"/>
                                </MessageContents>
                        </Message>
                        <Message code="MID-4000" severity="WARN">
                                <MessageContents>
                                        <MessageContent key="compCode" value="16"/>
                                </MessageContents>
                        </Message>
                </Messages>
        </Reply>
        <Reply seq="4">
                <XPath>/MailingRequest/MailingCreate[@seq="3"]/Items/Item[@seq="3"]</XPath>
                <Locations>
                        <Location tagName="Item" attributeName="seq" attributeValue="3"/>
                </Locations>
                <Messages>
                        <Message code="MID-4030" severity="INFO">
                                <MessageContents>
                                        <MessageContent key="midNum" value="11123458025113"/>
                                </MessageContents>
                        </Message>
                </Messages>
        </Reply>
      </Replies>
   </MailingCreate>

   <MailingDelete seq="4" mailingRef="Atp789">
      <Status code="100"/>
      <CustomerRefs>
        <CustomerRef key="D1" value="this is a customer private value"/>
        <CustomerRef key="D2" value="this is another customer private value"/>
      </CustomerRefs>
   </MailingDelete>

</MailingResponse>
```

This Response file contains the following information:

- MailingCreate with seq=1: the action was not successful (status 999)
- Warning: the second Contact email address is invalid (indeed, contains 2 "@" characters)
- Fatal: the second item has a MAIL ID number that is not unique (indeed, same MAIL ID number as the first item)
- MailingCreate with seq=2: the action was successful (status 100)
- Warning: the first item is incorrect, but could be matched -> It had 2 incorrect components: component 9 and 15.
- Error: the second item is incorrect; multiple matches were found
- Warning: the third item has an incorrect pre-sorting code
- MailingCreate with seq=3: the action was successful (status 100)
- bpost generated MAIL ID number "11123458025111" for item 1.
- bpost generated MAIL ID number "11123458025112" for item 2.
- Warning: the second item is incorrect, but could be matched. -> It had 1 incorrect component: component 16.
- bpost generated MAIL ID number "11123458025113" for item 3.
- MailingCheck with seq=4: the action was successful (status 100)
- Warning: the first item is incorrect, but could be matched.
- Component with code "9" was corrected with value "Suikerkaai".
- Component with code "15" was corrected with value "9060".

- Error: The second item is incorrect; multiple matches are found and returned in the Alternatives tag.
- MailingDelete with seq=5: action was successful (status 100)

TXT Format

The TXT format of this Response file would be:

- File name : MID_0200_12345678_EXMPL-ERR1_121106165955_2RS.TXT
- Content :

```
Context|MailingResponse|M037_MID|MID|12345678|0200
Header|12345678
CustomerRef|CK|this is a customer private value
CustomerRef|PV|this is another customer private value
RequestProps|MID_0200_12345678_EXMPL-ERR1_121106165955_2RS.XML|exmpl-err1
MailingCreate|1|mr1
Status|999
CustomerRef|AB|this is a customer private value
CustomerRef|AC|this is another customer private value
Reply|1
Location|Contact|2
Message|MID-3000|WARN
Reply|2
Location|Item|2
Message|MID-3010|FATAL
MailingCreate|2|mr2
Status|100
Reply|1
Message|MID-4040|INFO
MessageContent|complianceRate|33.3%
Reply|2
Location|Item|1
Message|MID-4000|WARN
MessageContent|compCode|9
Message|MID-4000|WARN
MessageContent|compCode|15
Reply|3
Location|Item|2
Message|MID-4020|ERROR
Reply|4
Location|Item|3
Message|MID-3020|WARN
MailingCreate|3|mr3
Status|100
Reply|1
Message|MID-4040|INFO
MessageContent|complianceRate|100.0%
Reply|2
Location|Item|1
Message|MID-4030|INFO
MessageContent|midNum|11123458025111
Reply|3
Location|Item|2
Message|MID-4030|INFO
MessageContent|midNum|11123458025112
Message|MID-4000|WARN
MessageContent|compCode|16
Reply|4
Location|Item|3
Message|MID-4030|INFO
MessageContent|midNum|11123458025113
MailingCheck|4|mr4
Status|100
CustomerRef|CC|this is a customer private value
CustomerRef|DD|this is another customer private value
Reply|1
```

```
Message|MID-4040|INFO
MessageContent|complianceRate|50.0%
Reply|2
Location|Item|1
Message|MID-4000|WARN
MessageContent|compCode|9
MessageContent|compCorrection|Suikerkaai
Message|MID-4000|WARN
MessageContent|compCode|15
MessageContent|compCorrection|9060
Reply|3
Location|Item|2
Message|MID-4020|ERROR
Alternative|1
Comp|9|James Wattstraat
Comp|13|10
Comp|15|1030
Comp|16|Brussel
Alternative|2
Comp|9|Wetstraat
Comp|13|10
Comp|15|1030
Comp|16|Brussel
MailingDelete|4|Atp789
Status|100
CustomerRef|D1|this is a customer private value
CustomerRef|D2|this is another customer private value
```

XLS Format

- File name : 130910134314_EXMPL-GOOD.XLS
- Content :

	Y	Z	AA	AB	AC	AD	AE	AF
1	MIDNUMBER	PRESORTING_CODE	LANGUAGE	PRIORITY	DISTRIBUTIONOFFICE	ROUTENAME	ROUTESEQ	FEEDBACK
	90034709524400	B-W1-L8		NP				MID-4030
	90034709524401	B-W1-L8		NP				MID-4030
	90034709524402	B-W1-L8		NP				MID-4030

Figure 43: MAIL ID Response file – XLS Example

5.3. Round and sequence

Mailing Request

XML Format

- File name : MID_0200_12345678_RSEXAMPLE1_121024185043_0RQ.XML
- Content :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MailingRequest>
  <Context requestName="MailingRequest" dataset="M037_MID" sender="12345678"
receiver="MID" version="0200"/>
  <Header customerId="12345678" accountId="32352" mode="P">
    <Files>
            <RequestProps customerFileRef=" RSEXAMPLE1"/>
```

```
          <ResponseProps format="XML" compressed="N" encrypted="N"
transmissionMode="HTTP"/>
    </Files>
  </Header>
  <MailingCreate seq="1" mailingRef=" RSEXAMPLE1" genMID="7" genPSC="Y"
expectedDeliveryDate="2016-10-13">
    <FileInfo type="RS3"/>
    <Format responseSortingMode = "CU">Large</Format>
    <Contacts>
          <Contact seq="1" firstName="lucien" lastName="dupont" email="
lucien.dupont@machin.be " lang="fr"/>
    </Contacts>
    <Items>
          <Item seq="1" priority="NP">
                <Comps>
                      <Comp code="2" value="Luc"/>
                      <Comp code="4" value="Dupont"/>
                      <Comp code="9" value="geheimzinnigevrouwstraat"/>
                      <Comp code="12" value="21"/>
                      <Comp code="15" value="3840"/>
                      <Comp code="16" value="Borgloon"/>
                </Comps>
          </Item>
          <Item seq="2" priority="NP">
                <Comps>
                      <Comp code="2" value="Maria"/>
                      <Comp code="4" value="Peeters"/>
                      <Comp code="9" value="rue du piroy"/>
                      <Comp code="12" value="4"/>
                      <Comp code="15" value="1370"/>
                      <Comp code="16" value="Jodoigne"/>
                </Comps>
          </Item>
          <Item seq="3" priority="NP">
                <Comps>
                      <Comp code="2" value="James"/>
                      <Comp code="4" value="Smith"/>
                      <Comp code="9" value="rue du moulin"/>
                      <Comp code="12" value="325"/>
                      <Comp code="15" value="4020"/>
                      <Comp code="16" value="Liege"/>
                </Comps>
          </Item>
    </Items>
    <ItemCount value="3"/>
  </MailingCreate>
</MailingRequest>
```

TXT Format

- File name : MID_0200_12345678_RSEXAMPLE1_121024185043_0RQ.TXT
- Content :

```
Context|MailingRequest|M037_MID|12345678|MID|0200
Header|12345678|32352|P
RequestProps|RSEXAMPLE1
ResponseProps|TXT|N|N|
MailingCreate|1|RSEXAMPLE1|||7|N|2016-10-20
FileInfo|RS3
Format|Large|CU
Contact|1|lucien|dupont|lucien.dupont@machin.be|fr|||
Item|1|fr|||P
Comp|2|Annie
Comp|4|HEURSEL
Comp|92|Vieille Rue 1
```

bpost business

```
Comp|15|1450
Item|2|fr|||P
Comp|2|Annie
Comp|4|HEURSEL
Comp|92|Rue Vieille Cure 1
Comp|15|1476
ItemCount|2
```

Mailing Acknowledgement

XML format

- File name : MID_0200_12345678_RSEXAMPLE1_121024185143_1AK.XML
- Content :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<RequestAck>
<FileReceived fileName=" MID_0200_12345678_RSEXAMPLE1_121017134046_0RQ.XML"
timeStamp="2016-10-10T10:58:30"/>
</RequestAck>
```

TXT format

- File name : MID_0200_12345678_RSEXAMPLE1_121017134046_1AK.TXT
- Content :

```
FileReceived|MID_0200_12345678_ RSEXAMPLE1_121024185043_0RQ.TXT|2016-10-10T11:08:37
```

Mailing Response

XML format

- File name : MID_0200_12345678_RSEXAMPLE1_121024190043_2RS.XML
- Content :

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<MailingResponse>
	<Context requestName="MailingResponse" dataset="M037_MID" sender="MID"
receiver="12345678" version="0200"/>
	<Header customerId="12345678">
		<Files>
		<RequestProps fileName="MID_0200_12345678_RSEXEMPLE1_120521190043_0RQ.XML"
customerFileRef="RSEXEMPLE1"/>
		</Files>
	</Header>
	<MailingCreate seq="1" mailingRef="RSEXEMPLE1">
		<Status code="100"/>
		<DistributionInformation>
			<Item prtOrder="1" seq="1" fieldToPrint1="Aa-M2-W6" fieldToPrint2="3840-
No-Rte" fieldToPrint3="99999" icti="Begin_End" izon=”Begin_End” imac="Begin_End"
iwav="Begin_End" ioff="Begin_End"/>
```

```
                        <Item prtOrder="2" seq="2" fieldToPrint1="Bb-M9-W7" fieldToPrint2="1370-
Reg-306" fieldToPrint3="663" icti="Begin_End" izon="Begin_End" imac="Begin_End"
iwav="Begin_End" ioff="Begin_End"/>
                        <Item prtOrder="3" seq="3" fieldToPrint1="La-M3-W5" fieldToPrint2="4020-
Res-173" fieldToPrint3="549" icti="Begin_End" izon="Begin_End" imac="Begin_End"
iwav="Begin_End" ioff="Begin_End"/>
                </DistributionInformation>
                <Replies>
                        <Reply seq="1">
                                <XPath>/MailingRequest/MailingCreate[@seq=&amp;quot;1&amp;quot;]</XPath>
                                <Messages>
                                        <Message code="MID-4040" severity="INFO">
                                                <MessageContents>
                                                        <MessageContent
key="compliancyRateAtBuildingLevel" value="66.67%"/>
                                                        <MessageContent
key="presortingCodeCompliancyRate" value="100.00%"/>
                                                        <MessageContent
key="addressesWithRecipientCompliancyRate" value="100.00 % "/>
                                                </MessageContents>
                                        </Message>
                                        <Message code="MID-4300" severity="WARN">
                                                <MessageContents>
                                                        <MessageContent
key="compliancyRateAtBuildingLevelUnderLimit" value=" Your building compliancy rate of 66.67 %
is lower than 70.00 %"/>
                                                </MessageContents>
                                        </Message>
                                </Messages>
                        </Reply>
                        <Reply seq="2">

        <XPath>/MailingRequest/MailingCreate[@seq=&amp;quot;1&amp;quot;]/Items/Item[@seq=&amp;quot;1&amp;quot;]
</XPath>
                                <Locations>
                                        <Location tagName="Item" attributeName="seq"
attributeValue="1"/>
                                </Locations>
                                <Messages>
                                        <Message code="MID-4010" severity="ERROR"/>
                                        <Message code="MID-4030" severity="INFO">
                                                <MessageContents>
                                                        <MessageContent key="midNum"
value="90034704113200"/>
                                                        <MessageContent key="psCode" value="A-M2-
W6"/>
                                                </MessageContents>
                                        </Message>
                                </Messages>
                        </Reply>
                        <Reply seq="3">

        <XPath>/MailingRequest/MailingCreate[@seq=&amp;quot;1&amp;quot;]/Items/Item[@seq=&amp;quot;2&amp;quot;]
</XPath>
                                <Locations>
                                        <Location tagName="Item" attributeName="seq"
attributeValue="2"/>
                                </Locations>
                                <Messages>
                                        <Message code="MID-4030" severity="INFO">
                                                <MessageContents>
                                                        <MessageContent key="midNum"
value="90034704113201"/>
                                                        <MessageContent key="psCode" value="B-M9-
W7"/>
                                                </MessageContents>
                                        </Message>
                                </Messages>
                        </Reply>
                        <Reply seq="4">
```

```
        <XPath>/MailingRequest/MailingCreate[@seq=&quot;1&quot;]/Items/Item[@seq=&quot;3&quot;]
</XPath>
                                <Locations>
                                        <Location tagName="Item" attributeName="seq"
attributeValue="3"/>
                                </Locations>
                                <Messages>
                                        <Message code="MID-4030" severity="INFO">
                                                <MessageContents>
                                                        <MessageContent key="midNum"
value="90034704113202"/>
                                                        <MessageContent key="psCode" value="L-M3-
W5"/>
                                                </MessageContents>
                                        </Message>
                                </Messages>
                        </Reply>
                </Replies>
        </MailingCreate>
</MailingResponse>
```

TXT format

- File name : MID_0200_12345678_RSEXAMPLE1_121024190043_2RS.TXT
- Content :

```
Context|MailingResponse|M037_MID|MID|12345678|0200
Header|12345678
RequestProps|MID_0200_12345678_RSEXAMPLE1_121024190043_0RQ.TXT| RSEXAMPLE1
MailingCreate|1| RSEXAMPLE1
Status|100
Item|1|Ba-M9-W7|1450-Reg-405|42||Begin|Begin|Begin|Begin|Begin_End|1
Item|2|Ba-M9-W7|1470-Reg-611|39||End|End|End|End|Begin_End|2
Reply|1
Message|MID-4040|INFO
MessageContent|compliancyRateAtBuildingLevel|100.00%
MessageContent|presortingCodeCompliancyRate|0.00%
MessageContent|addressesWithRecipientCompliancyRate|100.00 %
Reply|2
Message|MID-2070|WARN
Description|Number of items (2) does not match expected number of items specified in
ItemCount (1)
Reply|3
Location|Item|1
Message|MID-3020|WARN
MessageContent|psCode|B-M9-W7
Message|MID-4060|WARN
Message|MID-4030|INFO
MessageContent|midNum|90034704113400
Reply|4
Location|Item|2
Message|MID-3020|WARN
MessageContent|psCode|B-M9-W7
Message|MID-4060|WARN
Message|MID-4030|INFO
MessageContent|midNum|90034704113401
```

XLS Format

- File name : 130910134314_EXMPL-GOOD.XLS
- Content :

Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM
PRESORTING	LANGUAGE	PRIORITY	FIELDTOPRIN	FIELDTOPRINT2	FIELDTOPRIN	FEEDBACK	ORGINFO	ICTI	IZON	IMAC	IWAV	IOFF	PRINTORDER
B-W1-L4		NP	Overflow	299-No-Rte	99999	MID-4010 / MID-4030		Begin_End		Begin_End	Begin_End	Begin_End	1
B-M1-W5		NP	B-M1-W5	1050-Res-053	540	MID-3020 / MID-4060 / MID-4030		Begin		Begin_End	Begin_End	Begin_End	496
A-M5-W1		NP	A-M5-W1	2000-Reg-015	988	MID-3020 / MID-4060 / MID-4030							105
B-M5-W4		NP	B-M5-W4	2610-Reg-612	99999	MID-3020 / MID-4010 / MID-4030						Begin_End	499
A-M5-W1		NP	A-M5-W1	2000-Reg-008	47	MID-3020 / MID-4060 / MID-4030							52

Figure 44: Round and Sequence Response file – XLS Example

5.4. OptiAddress

Mailing Request

XML format

- File name : MID_0200_12345678_OPTIADDRES_120823132442_0RQ.XML
- Content :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MailingRequest>
   <Context requestName="MailingRequest" dataset="M037_MID" sender="6805" receiver="MID"
version="0200"/>
   <Header customerId="6805" accountId="32352" mode="T">
      <Files>
        <RequestProps customerFileRef="OPTIADDRES"/>
        <ResponseProps format="XML" compressed="N" encrypted="N" transmissionMode="FTPS"/>
      </Files>
      <CustomerRefs>
        <CustomerRef key="HeaderCRefKey01" value="HeaderCRefVal01"/>
        <CustomerRef key="HeaderCRefKey02" value="HeaderCRefVal02"/>
      </CustomerRefs>
   </Header>
   <MailingCheck seq="1" mailingRef="OPTIADDRES" genMID="N" genPSC="Y" copyRequestItem="N"
suggestionsCount="10" suggestionsMinScore="50">
      <Contacts>
        <Contact seq="1" email="lucien.dupont@machin.be" lang="fr"/>
      </Contacts>
      <Items>
        <Item seq="1" priority="NP">
               <Comps>
                      <Comp code="9" value="Avenue de Boetendael"/>
                      <Comp code="12" value="20"/>
                      <Comp code="15" value="1180"/>
                      <Comp code="16" value="Uccle"/>
               </Comps>
        </Item>
        <Item seq="2" priority="NP">
               <Comps>
                      <Comp code="9" value="Avenue de Boetendael"/>
                      <Comp code="12" value="20"/>
                      <Comp code="15" value="1180"/>
                      <Comp code="16" value="Ucle"/>
               </Comps>
        </Item>
        <Item seq="3" priority="NP">
               <Comps>
                      <Comp code="9" value="Avenue de Botendael"/>
                      <Comp code="12" value="22A"/>
                      <Comp code="15" value="1190"/>
                      <Comp code="16" value="Ucle"/>
```

```
            </Comps>
        </Item>
        <Item seq="4" priority="NP">
            <Comps>
                <Comp code="14" value="1159"/>
                <Comp code="15" value="1000"/>
                <Comp code="16" value="Bruxelles"/>
            </Comps>
        </Item>
        <Item seq="5" priority="NP">
            <Comps>
                <Comp code="9" value="Bloemendaelelaan"/>
                <Comp code="12" value="6"/>
                <Comp code="15" value="9990"/>
                <Comp code="16" value="Maldegem"/>
            </Comps>
        </Item>
        <Item seq="6" priority="NP">
            <Comps>
                <Comp code="9" value="Abdij"/>
                <Comp code="12" value="14"/>
                <Comp code="15" value="3000"/>
                <Comp code="16" value="Leuven"/>
            </Comps>
        </Item>
        <Item seq="7" priority="NP">
            <Comps>
                <Comp code="92" value="Avenue de Boetendael 20"/>
                <Comp code="93" value="1180 Uccle"/>
            </Comps>
        </Item>
      </Items>
      <ItemCount value="7"/>
   </MailingCheck>
</MailingRequest>
```

TXT format

- File name : MID_0200_12345678_OPTIADDRES_120823132442_0RQ.TXT
- Content :

```
Context|MailingRequest|M037_MID|6805|MID|0200
Header|6805|32352|P
RequestProps|OPTIADDRES
ResponseProps|TXT|N|N|FTPS
MailingCheck|1|OPTIADDRES|||N|Y|N|10|50
Contact|1|||lucien.dupont@machin.be|fr|||
Item|1||||NP
Comp|9|Avenue de Boetendael
Comp|12|20
Comp|15|1180
Comp|16|Uccle
Item|2||||NP
Comp|9|Avenue de Boetendael
Comp|12|20
Comp|15|1180
Comp|16|Ucle
Item|3||||NP
Comp|9|Avenue de Botendael
Comp|12|20
Comp|15|1190
Comp|16|Ucle
Item|4||||NP
Comp|14|1159
Comp|15|1000
Comp|16|Bruxelles
Item|5||||NP
Comp|9|Bloemendaelelaan
```

```
Comp|12|6
Comp|15|9990
Comp|16|Maldegem
Item|6|||NP
Comp|9|Abdij
Comp|12|14
Comp|15|3000
Comp|16|Leuven
Item|7|||NP
Comp|92|Avenue de Boetendael 20
Comp|93|1180 Uccle
ItemCount|7
```

Mailing Acknowledgement

XML format

- File name : MID_0200_12345678_OPTIADDRES_120823132542_1AK.XML
- Content :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<RequestAck>
   <FileReceived fileName="MID_0200_6805_OPTIADDRES_120823132542_0RQ.XML" timeStamp="2012-08-
23T14:12:30"/>
</RequestAck>
```

TXT format

- File name : MID_0200_12345678_OPTIADDRES_120823132542_1AK.TXT
- Content :

```
FileReceived|MID_0200_6805_OPTIADDRES_120823132542_0RQ.TXT|2012-08-23T14:09:02
```

Mailing Response

XML format

- File name : MID_0200_12345678_OPTIADDRES_120823133042_2RS.XML
- Content :

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<MailingResponse>
   <Context requestName="MailingResponse" dataset="M037_MID" sender="MID" receiver="6805"
version="0200"/>
   <Header customerId="6805">
      <CustomerRefs>
        <CustomerRef key="HeaderCRefKey01" value="HeaderCRefVal01"/>
        <CustomerRef key="HeaderCRefKey02" value="HeaderCRefVal02"/>
      </CustomerRefs>
      <Files>
        <RequestProps fileName="MID_0200_6805_OPTIADDRES_120823132442_0RQ.XML"
customerFileRef="OPTIADDRES"/>
      </Files>
   </Header>
   <MailingCheck seq="1" mailingRef="OPTIADDRE5">
```

```
      <Status code="100"/>
      <Replies>
        <Reply seq="1">
               <XPath>/MailingRequest/MailingCheck[@seq=&quot;1&quot;]</XPath>
               <Messages>
                      <Message code="MID-4040" severity="INFO">
                             <MessageContents>
                                    <MessageContent key="compliancyRateAtRoundLevel"
value="85.71%"/>
                                    <MessageContent key="compliancyRateAtPdpLevel"
value="71.43%"/>
                                    <MessageContent key="presortingCodeCompliancyRate"
value="0.00%"/>
                             </MessageContents>
                      </Message>
               </Messages>
        </Reply>
        <Reply seq="2">
<XPath>/MailingRequest/MailingCheck[@seq=&quot;1&quot;]/Items/Item[@seq=&quot;2&quot;]</XPath>
               <Locations>
                      <Location tagName="Item" attributeName="seq" attributeValue="2"/>
               </Locations>
               <Messages>
                      <Message code="MID-4060" severity="INFO"/>
                      <Message code="7001" severity="WARN">
                             <MessageContents>
                                    <MessageContent key="compCode" value="16"/>
                                    <MessageContent key="compCorrection" value="UCCLE"/>
                             </MessageContents>
                      </Message>
               </Messages>
        </Reply>
        <Reply seq="3">
<XPath>/MailingRequest/MailingCheck[@seq=&quot;1&quot;]/Items/Item[@seq=&quot;3&quot;]</XPath>
               <Locations>
                      <Location tagName="Item" attributeName="seq" attributeValue="3"/>
               </Locations>
               <Messages>
                      <Message code="MID-4060" severity="INFO"/>
                      <Message code="7001" severity="WARN">
                             <MessageContents>
                                    <MessageContent key="compCode" value="9"/>
                                    <MessageContent key="compCorrection" value="AVENUE DE
BOETENDAEL"/>
                             </MessageContents>
                      </Message>
                      <Message code="7001" severity="WARN">
                             <MessageContents>
                                    <MessageContent key="compCode" value="12"/>
                                    <MessageContent key="compCorrection" value="22-22A"/>
                             </MessageContents>
                      </Message>
                      <Message code="7001" severity="WARN">
                             <MessageContents>
                                    <MessageContent key="compCode" value="15"/>
                                    <MessageContent key="compCorrection" value="1180"/>
                             </MessageContents>
                      </Message>
                      <Message code="7001" severity="WARN">
                             <MessageContents>
                                    <MessageContent key="compCode" value="16"/>
                                    <MessageContent key="compCorrection" value="UCCLE"/>
                             </MessageContents>
                      </Message>
               </Messages>
        </Reply>
        <Reply seq="4">
<XPath>/MailingRequest/MailingCheck[@seq=&quot;1&quot;]/Items/Item[@seq=&quot;4&quot;]</XPath>
               <Locations>
                      <Location tagName="Item" attributeName="seq" attributeValue="4"/>
               </Locations>
               <Messages>
```

```
                        <Message code="MID-4060" severity="INFO"/>
                        <Message code="7001" severity="WARN">
                                <MessageContents>
                                        <MessageContent key="compCode" value="16"/>
                                        <MessageContent key="compCorrection" value="BRUXELLES DE
BROUCKERE"/>
                                </MessageContents>
                        </Message>
                </Messages>
        </Reply>
        <Reply seq="5">
<XPath>/MailingRequest/MailingCheck[@seq=&quot;1&quot;]/Items/Item[@seq=&quot;5&quot;]</XPath>
                <Locations>
                        <Location tagName="Item" attributeName="seq" attributeValue="5"/>
                </Locations>
                <Messages>
                        <Message code="MID-4070" severity="ERROR"/>
                        <Message code="MID-7003" severity="ERROR">
                                <MessageContents>
                                        <MessageContent key="compCode" value="9"/>
                                </MessageContents>
                        </Message>
                        <Message code="MID-4020" severity="ERROR">
                                <Alternatives>
                                        <Alternative seq="1">
                                                <Comps>
                                                        <Comp code="9" value="BLOEMENDALELAAN"/>
                                                        <Comp code="12" value="6"/>
                                                        <Comp code="15" value="9990"/>
                                                        <Comp code="16" value="MALDEGEM"/>
                                                </Comps>
                                        </Alternative>
                                        <Alternative seq="2">
                                                <Comps>
                                                        <Comp code="9" value="BLOEMESTRAAT"/>
                                                        <Comp code="12" value="6"/>
                                                        <Comp code="15" value="9990"/>
                                                        <Comp code="16" value="MALDEGEM"/>
                                                </Comps>
                                        </Alternative>
                                </Alternatives>
                        </Message>
                </Messages>
        </Reply>
        <Reply seq="6">
<XPath>/MailingRequest/MailingCheck[@seq=&quot;1&quot;]/Items/Item[@seq=&quot;6&quot;]</XPath>
                <Locations>
                        <Location tagName="Item" attributeName="seq" attributeValue="6"/>
                </Locations>
                <Messages>
                        <Message code="MID-4011" severity="ERROR"/>
                </Messages>
                <Suggestions>
                        <Suggestion seq="1" score="72">
                                <Comps>
                                        <Comp code="9" value="ABDIJSTRAAT"/>
                                </Comps>
                        </Suggestion>
                        <Suggestion seq="2" score="64">
                                <Comps>
                                        <Comp code="9" value="ABDIJ VAN PARK"/>
                                </Comps>
                        </Suggestion>
                        <Suggestion seq="3" score="63">
                                <Comps>
                                        <Comp code="9" value="ABDIJ VLIERBEEK"/>
                                </Comps>
                        </Suggestion>
                        <Suggestion seq="4" score="62">
                                <Comps>
                                        <Comp code="9" value="SINT-GEERTRUIABDIJ"/>
                                </Comps>
```

```
                    </Suggestion>
                </Suggestions>
            </Reply>
        </Replies>
    </MailingCheck>
</MailingResponse>
```

TXT format

- File name : MID_0200_12345678_OPTIADDRES_120823133042_2RS.TXT
- Content :

```
Context|MailingResponse|M037_MID|MID|6805|0200
Header|6805
RequestProps|MID_0200_6805_OPTIADDRES_120823132542_0RQ.TXT|OPTIADDRES
MailingCheck|1|OPTIADDRES
Status|100
Reply|1
Message|MID-4040|INFO
MessageContent|compliancyRateAtRoundLevel|85.71%
MessageContent|compliancyRateAtPdpLevel|71.43%
MessageContent|presortingCodeCompliancyRate|0.00%
Reply|2
Location|Item|2
Message|MID-4060|INFO
Message|7001|WARN
MessageContent|compCode|16
MessageContent|compCorrection|UCCLE
Reply|3
Location|Item|3
Message|MID-4060|INFO
Message|7001|WARN
MessageContent|compCode|9
MessageContent|compCorrection|AVENUE DE BOETENDAEL
Message|7001|WARN
MessageContent|compCode|15
MessageContent|compCorrection|1180
Message|7001|WARN
MessageContent|compCode|16
MessageContent|compCorrection|UCCLE
Reply|4
Location|Item|4
Message|MID-4060|INFO
Message|7001|WARN
MessageContent|compCode|16
MessageContent|compCorrection|BRUXELLES DE BROUCKERE
Reply|5
Location|Item|5
Message|MID-4070|ERROR
Message|MID-7003|ERROR
MessageContent|compCode|9
Message|MID-4020|ERROR
Alternative|1
Comp|9|BLOEMENDALELAAN
Comp|12|6
Comp|15|9990
Comp|16|MALDEGEM
Alternative|2
Comp|9|BLOEMESTRAAT
Comp|12|6
Comp|15|9990
Comp|16|MALDEGEM
Reply|6
Location|Item|6
Message|MID-4011|ERROR
Suggestion|1|72
Comp|9|ABDIJSTRAAT
Suggestion||64
```

bpost business

```
Comp|9|ABDIJ VAN PARK
Suggestion|3|63
Comp|9|ABDIJ VLIERBEEK
Suggestion|4|62
Comp|9|SINT-GEERTRUIABDIJ
```

Index of Tables

Index of Figures