PD CEN/TS 14014:2015

BSI Standards Publication

Postal services — Hybrid mail — XML definition of encapsulation of letters for automated postal handling

...making excellence a habit.™

National foreword

This Published Document is the UK implementation of CEN/TS 14014:2015. It supersedes DD CEN/TS 14014:2006 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee SVS/4, Postal services.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

ISBN 978 0 580 87750 6
ICS 03.240; 35.240.99

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 May 2015.

Amendments/corrigenda issued since publication

Date	Text affected

TECHNICAL SPECIFICATION

SPÉCIFICATION TECHNIQUE

TECHNISCHE SPEZIFIKATION

CEN/TS 14014

May 2015

ICS 35.240.99; 03.240

Supersedes CEN/TS 14014:2006

English Version

Postal services - Hybrid mail - XML definition of encapsulation of letters for automated postal handling

Services postaux - Courrier hybride - Définition XML de l'encapsulation des lettres pour un traitement postal automatisé

This Technical Specification (CEN/TS) was approved by CEN on 14 March 2015 for provisional application.

The period of validity of this CEN/TS is limited initially to three years. After two years the members of CEN will be requested to submit their comments, particularly on the question whether the CEN/TS can be converted into a European Standard.

CEN members are required to announce the existence of this CEN/TS in the same way as for an EN and to make the CEN/TS available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force (in parallel to the CEN/TS) until the final decision about the possible conversion of the CEN/TS into an EN is reached.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Contents

Page

Foreword

This document (CEN/TS 14014:2015) has been prepared by Technical Committee CEN/TC 331 “Postal services”, the secretariat of which is held by NEN.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

This document supersedes CEN/TS 14014:2006. An explanation of the differences between this Technical Specification and CEN/TS 14014:2006 is given in Annex C.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to announce this Technical Specification: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

Introduction

Hybrid Mail is the technology whereby input in one communication medium is converted for delivery on another communication medium according to the sender's instructions and/or the recipient's capabilities. The typical application of Hybrid Mail is to provide a Hybrid Mail operator with printing data as well as processing and delivery instructions, and request the operator to secure the print, enveloping and delivery of the physical letters. Hybrid Mail operators may also exchange data.

The transfer of data to a Hybrid Mail operator or between Hybrid Mail operators requires that the printing data be linked to a number of data items related to the management, production, finishing, etc. of the data to be printed. Such data items secure that all relevant information is accompanying the printing data. Also it will enable the Hybrid Mail operator to automate his processes with customers and other Hybrid Mail operators.

There is a need for a standardised yet flexible way to present the data to the Hybrid Mail operator or to exchange data between Hybrid Mail operators. This will enable customers and Hybrid Mail operators to have a seamless exchange of information. It will allow makers of applications for document creation (letters, marketing mailing, etc.) and output management from other applications (accounting systems, production management, etc.), to add here to the same data presentment and to offer the seamless data interchange.

1 Scope

The purpose of this Technical Specification is to define the syntax rules for a data stream for the submission of printing data to a Hybrid Mail operator or between Hybrid Mail operators. The Technical Specification defines an XML Schema Definition (XSD) describing the data stream.

The description is based upon the XML (eXtended Mark-up Language) definition of rules and semantics for defining an XSD. The purpose of this is to offer a generalised syntax description that can provide seamless integration with a number of existing applications generating data that is liable to be forwarded to or from a Hybrid Mail operator.

The use of an XSD will ensure that the documents confirm to the standard defined and that the output has the correct syntax. Software manufacturers can use an XSD to program applications that will produce “correct” outputs.

This Technical Specification defines the syntax for creating a data stream that will eventually be converted into a deliverable. The overall object (a batch) can be divided into one or more objects that again can be divided into objects. The hierarchy includes bundles that contain a common part and letters. Each object has a number of characteristics attached to it.

This diagram shows the structure of a HML (Hybrid Mail Language) document: each letter is self-contained (contains all the necessary information to be delivered on a certain destination).

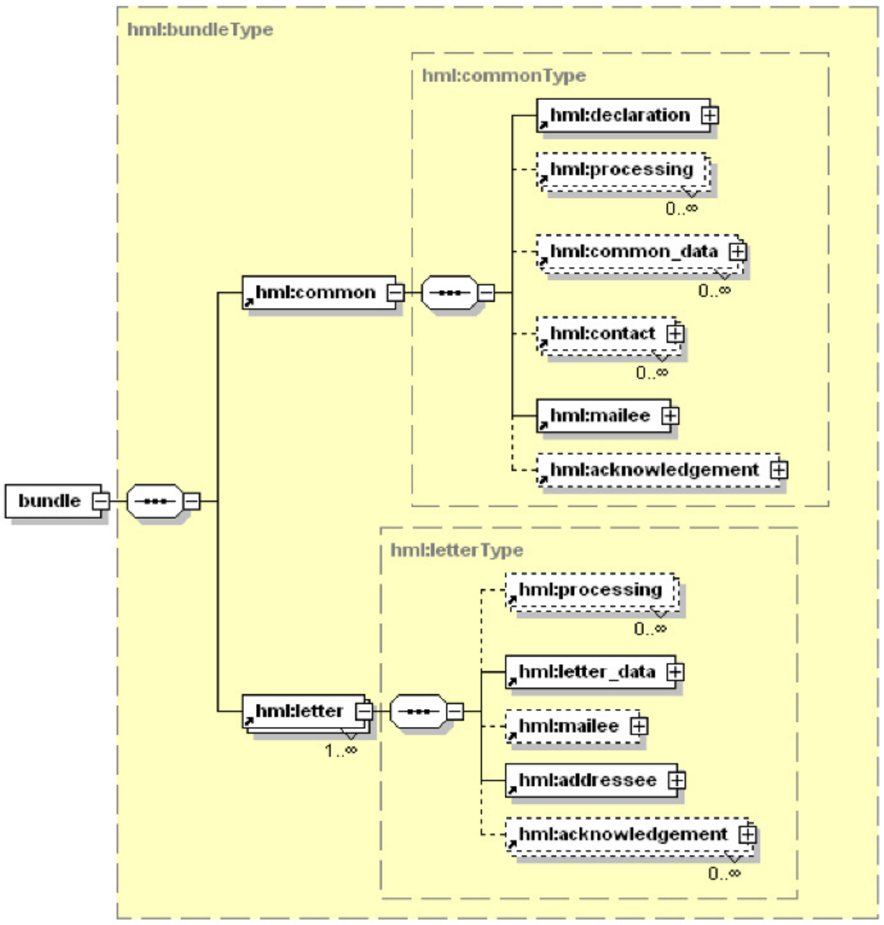


Figure 1 — Structure of a HML (Hybrid Mail Language)

Each letter can have one contact. Each contact can have multiple alternatives for delivery.

This Technical Specification does not define the specific services offered by local operators (Hybrid Mail operators).

This Technical Specification does not define the communication method used. It does only define the format of Hybrid Mail as such.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
mailbag
data structure that contains bundles as well as administrative and other data common to all bundles

Note 1 to entry: One HML document will contain one mailbag. A mailbag may contain one or several bundles.

3.2
bundle
data structure that contains letters that are processed as a group as well as administrative and other data common to these letters. A bundle is equivalent with a batch. Usually a sender is sending a mailbag with only one batch

Note 1 to entry: A bundle may contain one or more letters.

3.3
letter
data structure that contains the data to be rendered as one integral piece of information which is to be delivered to one recipient in physical or electronic format

3.4
contact
data structure that contains delivery information for letters

Note 1 to entry: The contact may be relevant to only one letter or may be shared between several letters.

3.5
target language
language to be defined in this document and to be later used for writing documents, and the result of a possible translation of existing data structure(s). In the present document the target language is HML

Note 1 to entry: Clause 4 gives further description of the syntax of the target language.

4 Symbols and abbreviated terms

For the purposes of this Technical Specification, the following symbols and abbreviated terms apply.

AFP	Advance Function Presentation – PDL defined by IBM
HML	Hybrid Mail Language
IEC	International Electrotechnical Commission http://www.iec.ch
ISO	International Organization for Standardization
PCL	Print Control Language – PDL defined by HP
PDF	Portable Document Format – PDL defined by Adobe
PDL	Print Description Language
PI	Processing Instruction – part of the XML standard
SGML	Standard Generalized Mark-up Language
UCS	Universal Coded Character set
URL	Universal Resource Locator
W3C	World Wide Web Consortium – see http://www.w3.org/
XML	eXtensible Mark-up Language
XSD	XML Schema Definition
XSL	eXtensible Stylesheet Language

5 Meta-syntax

This clause introduces a syntactic notation, later used in this Technical Specification. The notation is adopted to define the syntactic rules of the target language: in this sense, the notation is a meta-syntax for the syntax of the target language.

HML is based on XML version 1.1 as described in [XML-2006]. This is a subset of Standard Generalized Mark up Language (SGML) as defined in ISO 8879.

For the sake of generality, in the following the term target language will be used for specifying the language to be defined and to be later used for writing documents.

In this Technical Specification the target language is HML.

Syntactic rules of the target language are defined by means of syntactic clauses, classified as either element declarations, attribute list declarations or comments. In the following, the first two of these syntactic clauses will be described in detail. Here, only their abstract characteristics are introduced.

Element declarations define elements, which are logical parts of the documents.

Elements may contain other elements, to be considered as sub-parts of the first ones. To describe this relationship among elements, element declarations can define elements in terms of other elements.

The whole document itself is considered as an element, and is described by an element declaration: this element is the unique root, which all the other elements start from.

On the other side, elements not further subdivided in parts are simply streams of characters allowed in the documents. They are defined as well by element declarations.

In this way, element declarations express the inner structure of documents: this structure can be easily reconstructed by going through the chain of element declarations, starting from the one that declares the root element.

An element declaration is defined by the following syntactical construction:

```
<xs:element name="element_name" >
   <xs:choice>
         <xs:element ref="contentspec"/>
      </xs:choice>
   </xs:element>
```

where *element_name* is the name of the element defined and *contentspec* is the list of elements which constitute the set of elements which defines the named element.

Attribute list declarations define characteristics of elements.

In describing the notation, some rules will be followed.

In the syntactic clauses:

— syntactic items independent of the particular target language (i.e. keywords, symbols and so on) are written in regular font (that is, without using bold or italic forms);

— identifiers used as placeholders for other things to be later made actual in the syntax of the target language (as for example syntactic items dependent of the particular target language) are written in italic font.

In the examples:

— syntactic items independent of the particular target language (i.e. keywords, symbols and so on) are written in regular font (that is, without using bold or italic forms);

— syntactic items dependent of the particular target language (as for example constant names of the target language) are written in italic font.

An attribute declaration is defined by the following syntactical construction:

```
   <xs:attribute name="element_name" type="attribute_type"
default="default_decl"/>
```

where the *element_name* identifies which element the attributes belong to, *attribute_name* is the name of the attribute, *attribute_type* is the type of the attribute and *default_decl* informs whether the attribute has a default value that is used if the attribute is not present.

For a more detailed description of the syntax of XML please see [XML-2006]

6 Definition

6.1 General

In this clause, syntactic rules of HML are given in XML. They completely define the concrete syntax of HML. The structure of the HML XSD is illustrated in Figure 2:

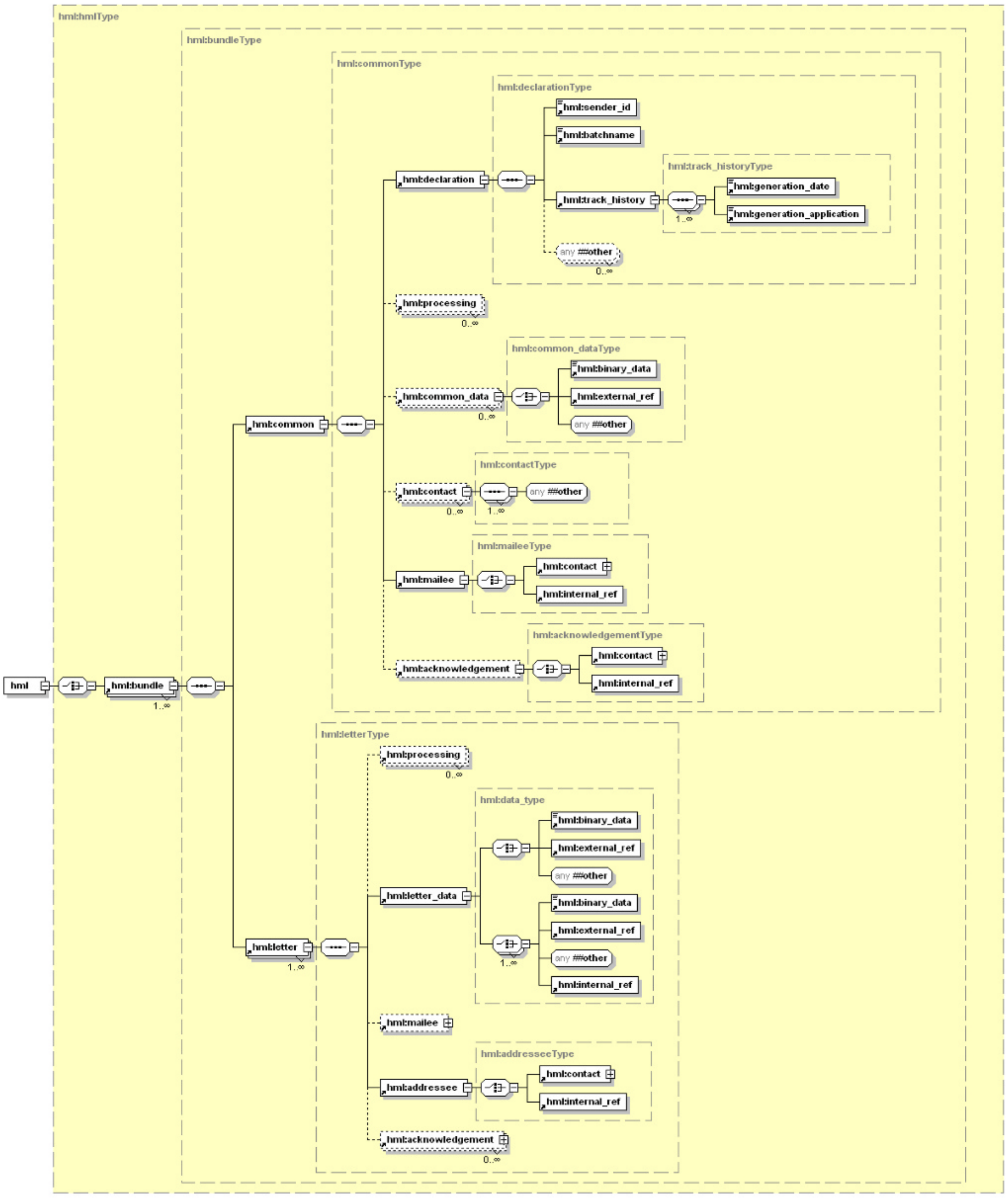


Figure 2 — Illustration of the structure of the HML XSD

6.2 General rule for HML documents

6.2.1 General

HML is based on eXtensible Markup Language (XML) version 1.1 as described in [XML-2006]. This is a subset of Standard Generalized Markup Language (SGML) as defined in ISO 8879.

As HML is specified as an XSD, the general rules for XML compliant XSD's apply to HML. However this chapter specifies how a compliant HML document shall be processed when using the following components from the XML standard, which are not unique to HML:

— Comments

— Processing Instructions

— Name Spaces

6.2.2 Comments

```
Comment ::= '<!--' ((Char - '-') | ('-' (Char - '-')))* '-->'
```

A conforming HML system shall make the comments in a conforming HML document available to external processors on request.

6.2.3 Processing Instructions

```
PI ::= '<?' PITarget (S (Char* - (Char* '?>' Char*)))? '?>'
PITarget ::= Name - (('X' | 'x') ('M' | 'm') ('L' | 'l'))
```

A conforming HML system shall ensure that the processing instructions are presented to the relevant processor if this is available. This can either be pre- or postprocessor. Otherwise the processing instructions shall be ignored.

6.2.4 Name space

The use of name spaces in HML shall comply with the W3C recommendations – see [XML-names-2004]. The processing of names from other name spaces than the HML name space shall be accepted and either be handled by the HML system or alternatively by a pre- or postprocessor.

6.2.5 Hybrid Mail Language (HML) Extensions:

In the future new equipment shall be controllable in Hybrid Mail operations. To include new features to make use of this equipment and to fulfil the Hybrid Mail Language it is allowed to include new elements.

The way of doing this is make use of the defined way as mentioned in Chapter 2.8 of the [XML-2004]: the document type declaration can point to an external subset containing mark-up declarations.

6.3 General elements

6.3.1 General

The encoding for any text strings in this Technical Specification shall be the ISO/IEC 10646. This will have to be specified in the XML header section of the HML document.

6.3.2 acknowledgement

This is an optional tag to provide a contact point for a recipient of bundle acknowledgement information. The preference attribute is a list of references defined in the contact to specify the preference of the media on which the delivery should be made. If there is none specified then the delivery order is the order of the defined entries in the contact element.

6.3.3 addressee

The addressee section contains the possible contact points of the recipient. The preference attribute is a list of references defined in the contact to specify the preference of the media on which the delivery should be made. If there is none specified then the delivery order is the order of the defined entries in the contact element.

6.3.4 batchname

Is an identifier for the bundle.

6.3.5 binary_data

Holds data of any sort, base64 encoded. See [Base64] for details. This can be used for data that is not XML structured.

6.3.6 bundle

Represents a collection of letters and their common properties. An HML document may contain a number of bundles. A bundle contains a number of letters.

6.3.7 common

Contains the data and information that is common to all the letters of a bundle. Similar to a global declaration for the bundle. Contains descriptions of the bundle as well as common data.

6.3.8 common_data

Declaration of data common to a set of letters. The common_data is identified with a unique identifier that may be referred to with an internal_ref tag somewhere else in the bundle.

6.3.9 contact

Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery.

6.3.10 declaration

Contains meta-information for the encapsulating bundle.

6.3.11 external_ref

External URL reference to a data segment. The attribute type has to specify what kind of data is pointing to.

6.3.12 generation_application

The application that processed this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking.

6.3.13 generation_date

The date and time of the processing of this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking.

6.3.14 hml

The root level tag defines the document.

6.3.15 internal_ref

Internal reference to a unique key id declared in the common data section.

6.3.16 letter

The smallest message unit. A message is constituted by an instance of a letter. A bundle contains a list of letters.

6.3.17 letter_data

Contains the data of the letter, either indirectly as internal or external references, or as actual data. Letter data is an open slot for any external data structure and does not attempt to describe the data itself.

6.3.18 mailee

The mailee section contains the possible contact points of the sender.

6.3.19 processing

Defines a property to follow the data contained in the current encapsulating document fragment and fragments inside (and after the processing tag), except if the type is redefined in a document sub fragment. This is similar to an environment variable for the data. Processing holds and transports the properties of the data contained in the document fragment. These options may or may not be relevant for pre-processing, processing and post-processing, depending on the HML application.

6.3.20 sender_id

The id used by the HML processor to identify the originator of the file.

6.3.21 track_history

Contains one record of a processing event. A list of track_history records the processing of the bundle.

Annex A
(informative)

List of elements

A.1 Elements

Element	Type	Reference
acknowledgement	Element	See 6.3.2 and B.1.1.
addressee	Element	See 6.3.3 and B.1.2.
batchname	Element	See 6.3.4 and B.1.3.
binary_data	Element	See 6.3.5 and B.1.4.
bundle	Element	See 0, 6.3.6 and B.1.5.
common	Element	See 6.3.7 and B.1.6.
common_data	Element	See 6.3.8 and B.1.7.
contact	Element	See 6.3.9 and B.1.8.
declaration	Element	See 6.3.10 and B.1.9.
external_ref	Element	See 6.3.11 and B.1.10.
generation_application	Element	See 6.3.12 and B.1.11.
generation_date	Element	See 6.3.13 and B.1.12.
hml	Element	See 6.3.14 and B.1.13.
internal_ref	Element	See 6.3.15 and B.1.14.
letter	Element	See 0, 6.3.16 and B.1.15.
letter_data	Element	See 6.3.17 and B.1.16.
mailee	Element	See 6.3.18 and B.1.17.
processing	Element	See 6.3.19 and B.1.18.
sender_id	Element	See 6.3.20 and B.1.19.
track_history	Element	See 6.3.21 and B.1.20.

A.2 Complex types

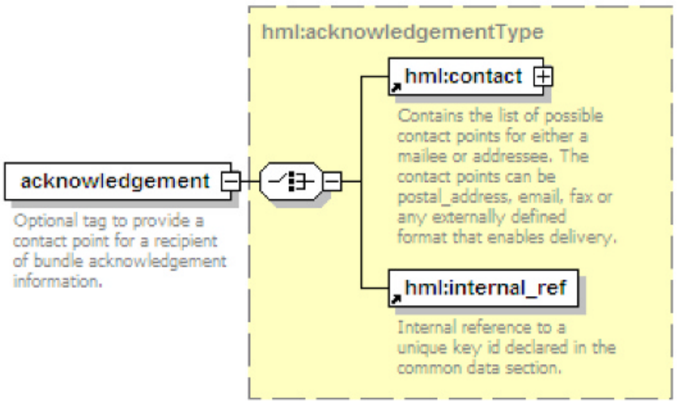
Element	Type	Reference
acknowledgementType	Complex type	See B.2.1
addresseeType	Complex type	See B.2.2
bundleType	Complex type	See B.2.4
commonType	Complex type	See B.2.6
contactType	Complex type	See B.2.7
data_type	Complex type	See B.2.8
declarationType	Complex type	See B.2.9
external_refType	Complex type	See B.2.10
hmlType	Complex type	See B.2.11
Internal_refType	Complex type	See B.2.12
letterType	Complex type	See B.2.13
maileeType	Complex type	See B.2.14
restricted_dataType	Complex type	See B.2.16
track_historyType	Complex type	See B.2.17

Annex B
(informative)

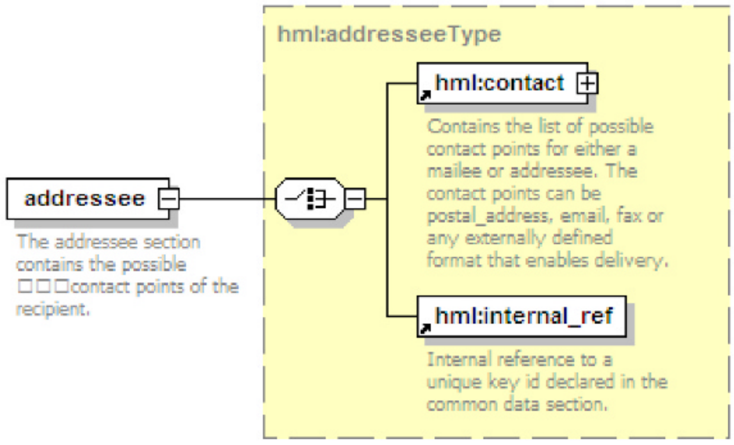
HML description

B.1 Elements

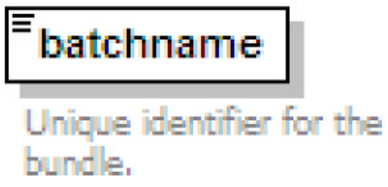
B.1.1 acknowledgement

diagram	 hml:acknowledgementType hml:contact Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery. acknowledgement Optional tag to provide a contact point for a recipient of bundle acknowledgement information. hml:internal_ref Internal reference to a unique key id declared in the common data section.
type	**hml:acknowledgementType**
children	**hml:contact hml:internal_ref**
used by	complexTypes **commonType letterType**
attributes	Name Type Use Annotation preference xs:IDREFS optional Preference attribute is specifying what is the order of preferred delivery channel. The IDREFS has to be the one defined in the contact. If none specified then the order in which the channels had been defined will be used.
annotation	documentation Optional tag to provide a contact point for a recipient of bundle acknowledgement information.
source	<xs:element name="acknowledgement" type="hml:acknowledgementType" block="#all"> <xs:annotation> <xs:documentation>Optional tag to provide a contact point for a recipient of bundle acknowledgement information.</xs:documentation> </xs:annotation> </xs:element>-

B.1.2 addressee

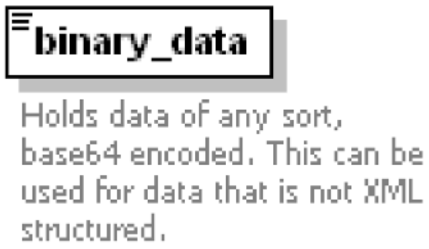
diagram	 hml:addresseeType hml:contact Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery. addressee The addressee section contains the possible □□□contact points of the recipient. hml:internal_ref Internal reference to a unique key id declared in the common data section.
type	**hml:addresseeType**
children	**hml:contact hml:internal_ref**
used by	complexType **letterType**
attributes	Name Type Use Annotation preference xs:IDREFS optional preference attribute is specifying what is the order of preferred delivery channel. The IDREFS has to be the one defined in the contact. If none specified then the order in which the channels had been defined will be used.
annotation	documentation The addressee section contains the possible contact points of the recipient.
source	<xs:element name="addressee" type="hml:addresseeType" block="#all"> <xs:annotation> <xs:documentation>The addressee section contains the possible contact points of the recipient.</xs:documentation> </xs:annotation> </xs:element>

B.1.3 batchname

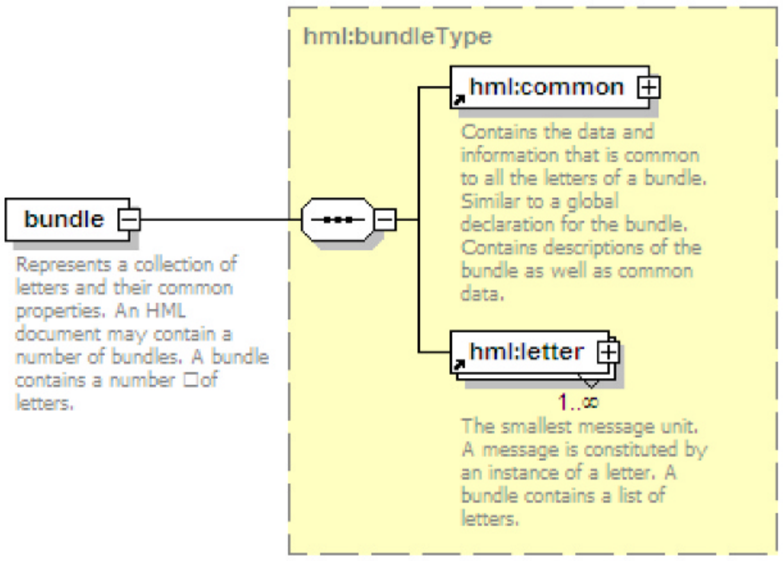
diagram	
type	**xs:anyURI**
used by	complexType **declarationType**

annotation	documentation Unique identifier for the bundle.
source	<xs:element name="batchname" type="xs:anyURI" block="#all"> <xs:annotation> <xs:documentation>Unique identifier for the bundle. </xs:documentation> </xs:annotation> </xs:element>

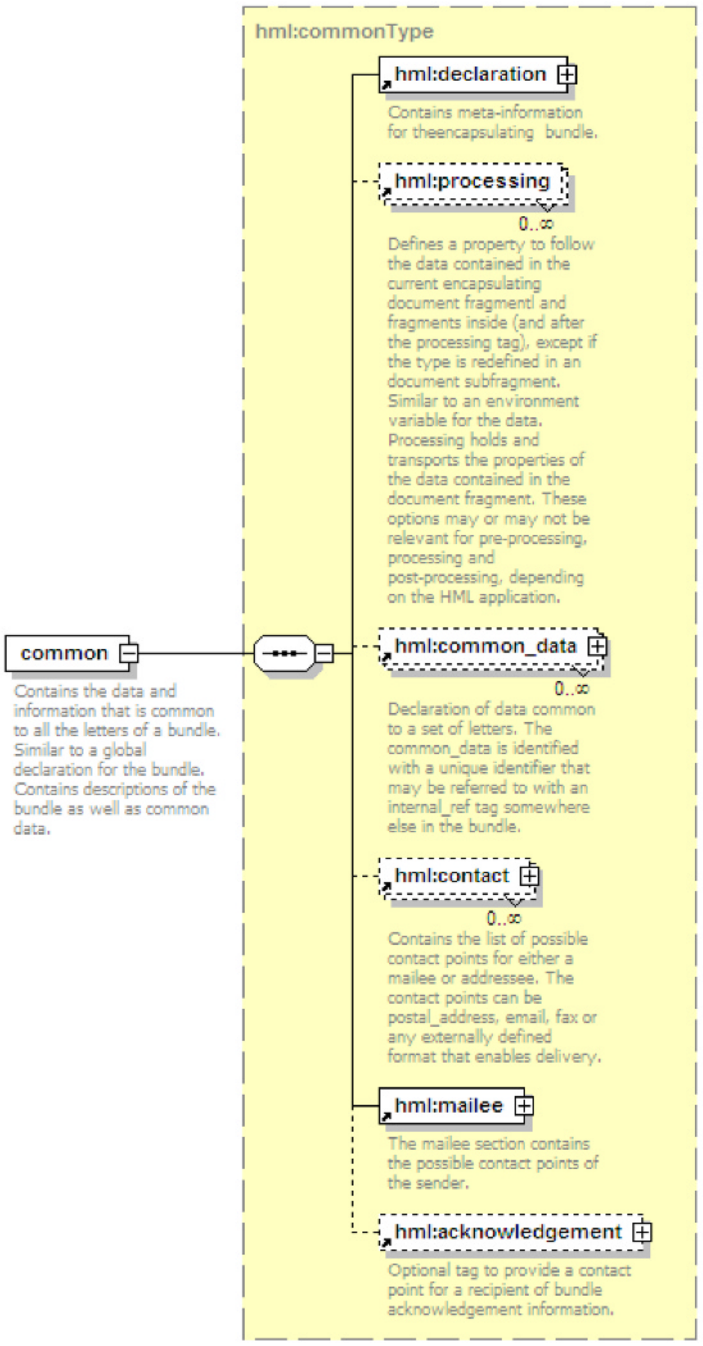
B.1.4 binary_data

diagram	 binary_data Holds data of any sort, base64 encoded. This can be used for data that is not XML structured.
type	**hml:binary_dataType**
used by	complexTypes **data_type restricted_dataType**
attributes	Name Type Use Default Fixed Annotation type xs:string required
annotation	documentation Holds data of any sort, base64 encoded. This can be used for data that is not XML structured.
source	<xs:element name="binary_data" type="hml:binary_dataType"> <xs:annotation> <xs:documentation>Holds data of any sort, base64 encoded. This can be used for data that is not XML structured.</xs:documentation> </xs:annotation> </xs:element>

B.1.5 bundle

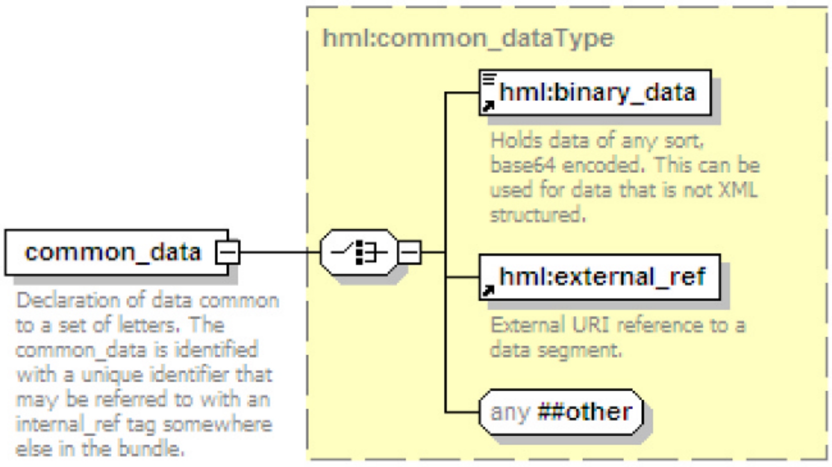
diagram	 hml:bundleType hml:common Contains the data and information that is common to all the letters of a bundle. Similar to a global declaration for the bundle. Contains descriptions of the bundle as well as common data. bundle Represents a collection of letters and their common properties. An HML document may contain a number of bundles. A bundle contains a number of letters. hml:letter 1..∞ The smallest message unit. A message is constituted by an instance of a letter. A bundle contains a list of letters.
type	**hml:bundleType**
children	**hml:common hml:letter**
used by	complexType **hmlType**
attributes	Name Type Use Default Fixed Annotation bundleid xs:ID optional
annotation	documentation Represents a collection of letters and their common properties. An HML document may contain a number of bundles. A bundle contain at least one letter.
source	<xs:element name="bundle" type="hml:bundleType" block="#all"> <xs:annotation> <xs:documentation>Represents a collection of letters and their common properties. An HML document may contain a number of bundles. A bundle contains a number of letters.</xs:documentation> </xs:annotation> </xs:element>

B.1.6 common

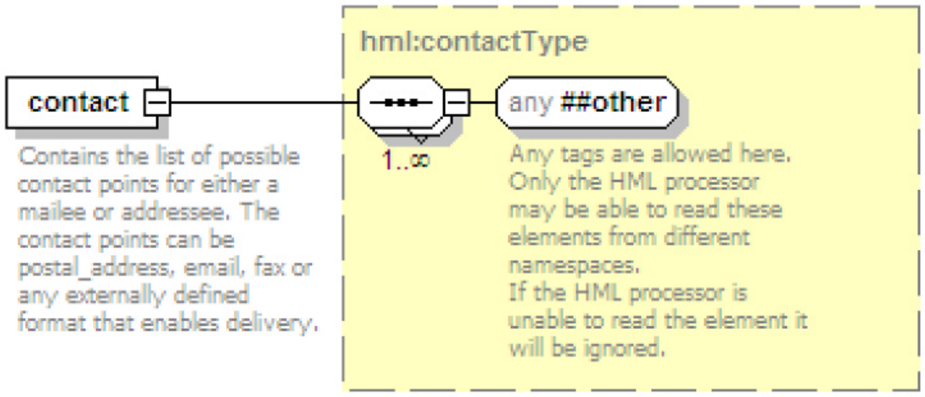
diagram	 hml:commonType hml:declaration Contains meta-information for theencapsulating bundle. hml:processing 0..∞ Defines a property to follow the data contained in the current encapsulating document fragmentl and fragments inside (and after the processing tag), except if the type is redefined in an document subfragment. Similar to an environment variable for the data. Processing holds and transports the properties of the data contained in the document fragment. These options may or may not be relevant for pre-processing, processing and post-processing, depending on the HML application. common Contains the data and information that is common to all the letters of a bundle. Similar to a global declaration for the bundle. Contains descriptions of the bundle as well as common data. hml:common_data 0..∞ Declaration of data common to a set of letters. The common_data is identified with a unique identifier that may be referred to with an internal_ref tag somewhere else in the bundle. hml:contact 0..∞ Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery. hml:mailee The mailee section contains the possible contact points of the sender. hml:acknowledgement Optional tag to provide a contact point for a recipient of bundle acknowledgement information.
type	**hml:commonType**
children	**hml:declaration hml:processing hml:common_data hml:contact hml:mailee hml:acknowledgement**
used by	complexType **bundleType**
annotation	documentation Contains the data and information that is common to all the letters of a bundle. Similar to a global declaration for the bundle. Contains descriptions of the bundle as well as common data.

source	<xs:element name="common" type="hml:commonType" block="#all"> <xs:annotation> <xs:documentation>Contains the data and information that is common to all the letters of a bundle. Similar to a global declaration for the bundle. Contains descriptions of the bundle as well as common data.</xs:documentation> </xs:annotation> </xs:element>

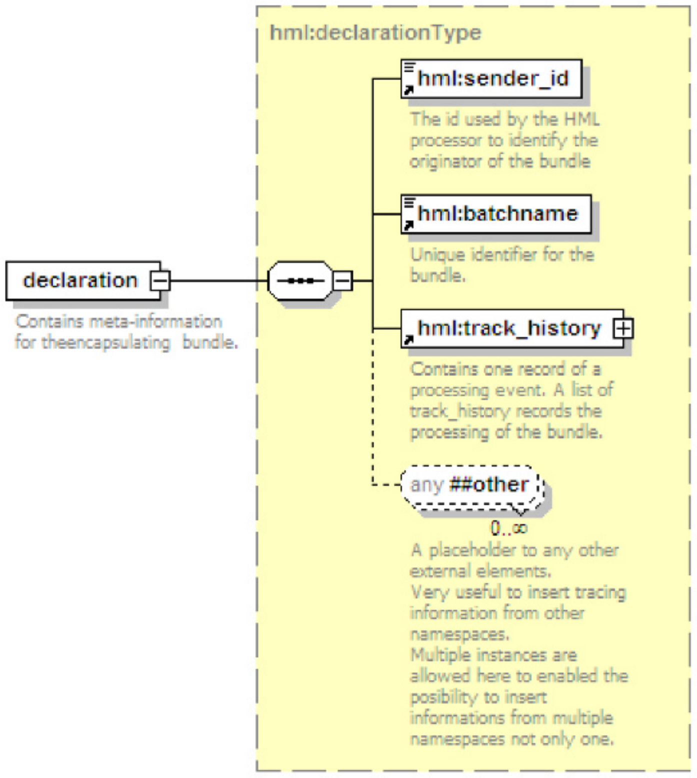
B.1.7 common_data

diagram	 hml:common_dataType hml:binary_data Holds data of any sort, base64 encoded. This can be used for data that is not XML structured. common_data Declaration of data common to a set of letters. The common_data is identified with a unique identifier that may be referred to with an internal_ref tag somewhere else in the bundle. hml:external_ref External URI reference to a data segment. any ##other
type	**hml:common_dataType**
children	**hml:binary_data hml:external_ref**
used by	complexType **commonType**
attributes	Name Type Use Default Fixed Annotation id xs:ID required
annotation	documentation Declaration of data common to a set of letters. The common_data is identified with a unique identifier that may be referred to with an internal_ref tag somewhere else in the bundle.
source	<xs:element name="common_data" type="hml:common_dataType" block="#all"> <xs:annotation> <xs:documentation>Declaration of data common to a set of letters. The common_data is identified with a unique identifier that may be referred to with an internal_ref tag somewhere else in the bundle.</xs:documentation> </xs:annotation> </xs:element>

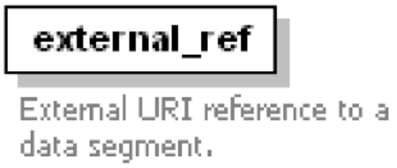
B.1.8 contact

diagram	 hml:contactType contact Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery. 1..∞ any ##other Any tags are allowed here. Only the HML processor may be able to read these elements from different namespaces. If the HML processor is unable to read the element it will be ignored.
type	**hml:contactType**
used by	complexTypes **acknowledgementType addresseeType commonType maileeType**
attributes	Name Type Use Default Fixed Annotation id xs:ID
annotation	documentation Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery.
source	<xs:element name="contact" type="hml:contactType" block="#all"> <xs:annotation> <xs:documentation>Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery.</xs:documentation> </xs:annotation> </xs:element>

B.1.9 declaration

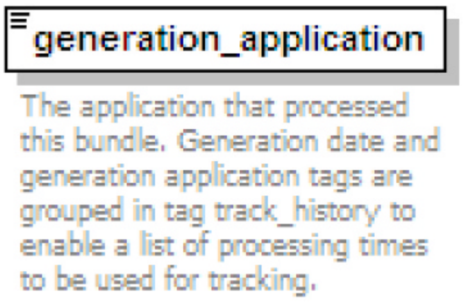
diagram	 hml:declarationType hml:sender_id The id used by the HML processor to identify the originator of the bundle hml:batchname Unique identifier for the bundle. declaration Contains meta-information for theencapsulating bundle. hml:track_history Contains one record of a processing event. A list of track_history records the processing of the bundle. any ##other 0..∞ A placeholder to any other external elements. Very useful to insert tracing information from other namespaces. Multiple instances are allowed here to enabled the posibility to insert informations from multiple namespaces not only one.
type	**hml:declarationType**
children	**hml:sender_id hml:batchname hml:track_history**
used by	complexType **commonType**
annotation	documentation Contains meta-information for theencapsulating bundle.
source	<xs:element name="declaration" type="hml:declarationType" block="#all"> <xs:annotation> <xs:documentation>Contains meta-information for theencapsulating bundle.</xs:documentation> </xs:annotation> </xs:element>

B.1.10 external_ref

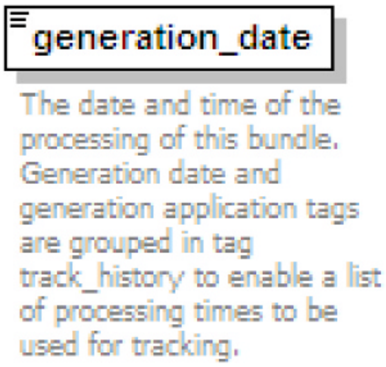
diagram	
type	**hml:external_refType**
used by	complexTypes **data_type restricted_dataType**

attributes	Name Type Use Default Fixed Annotation id xs:anyURI required type xs:string required
annotation	documentation External URI reference to a data segment.
source	<xs:element name="external_ref" type="hml:external_refType"> <xs:annotation> <xs:documentation>External URI reference to a data segment.</xs:documentation> </xs:annotation> </xs:element>

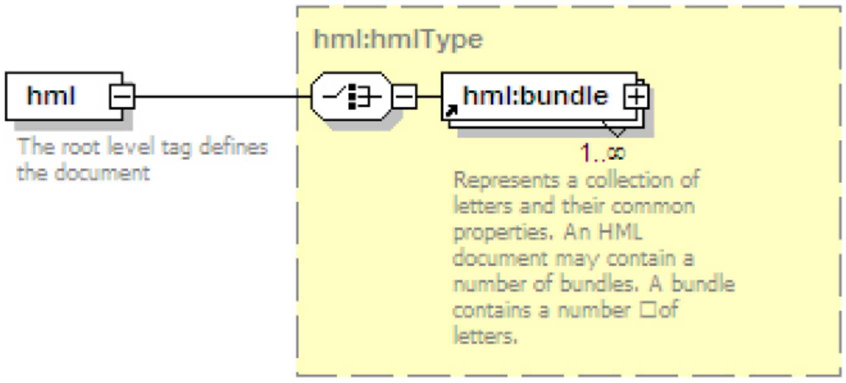
B.1.11 generation_application

diagram	 generation_application The application that processed this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking.
type	**xs:string**
used by	complexType **track_historyType**
annotation	documentation The application that processed this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking.
source	<xs:element name="generation_application" type="xs:string" block="#all"> <xs:annotation> <xs:documentation>The application that processed this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking.</xs:documentation> </xs:annotation> </xs:element>

B.1.12 generation_date

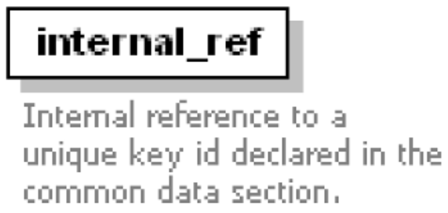
diagram	 generation_date The date and time of the processing of this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking.
type	**xs:dateTime**
used by	complexType **track_historyType**
annotation	documentation The date and time of the processing of this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking.
source	<xs:element name="generation_date" type="xs:dateTime" block="#all"> <xs:annotation> <xs:documentation>The date and time of the processing of this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking.</xs:documentation> </xs:annotation> </xs:element>

B.1.13 hml

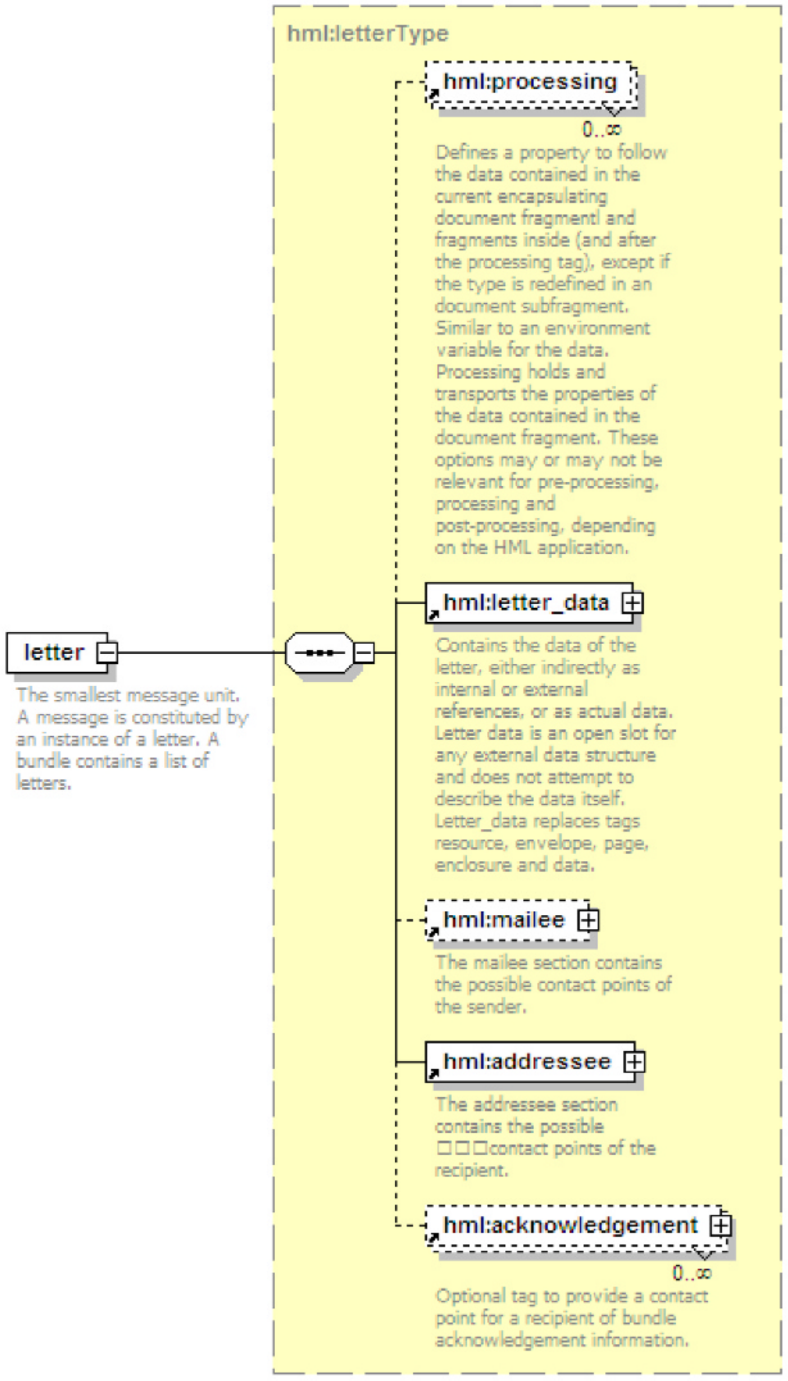
diagram	 hml:hmlType hml The root level tag defines the document hml:bundle 1..∞ Represents a collection of letters and their common properties. An HML document may contain a number of bundles. A bundle contains a number of letters.
type	**hml:hmlType**
children	**hml:bundle**
attributes	Name Type Use Default Fixed Annotation version xs:string required id xs:ID optional

annotation	documentation The root level tag defines the document
source	<xs:element name="hml" type="hml:hmlType" block="#all"> <xs:annotation> <xs:documentation>The root level tag defines the document</xs:documentation> </xs:annotation> </xs:element>

B.1.14 internal_ref

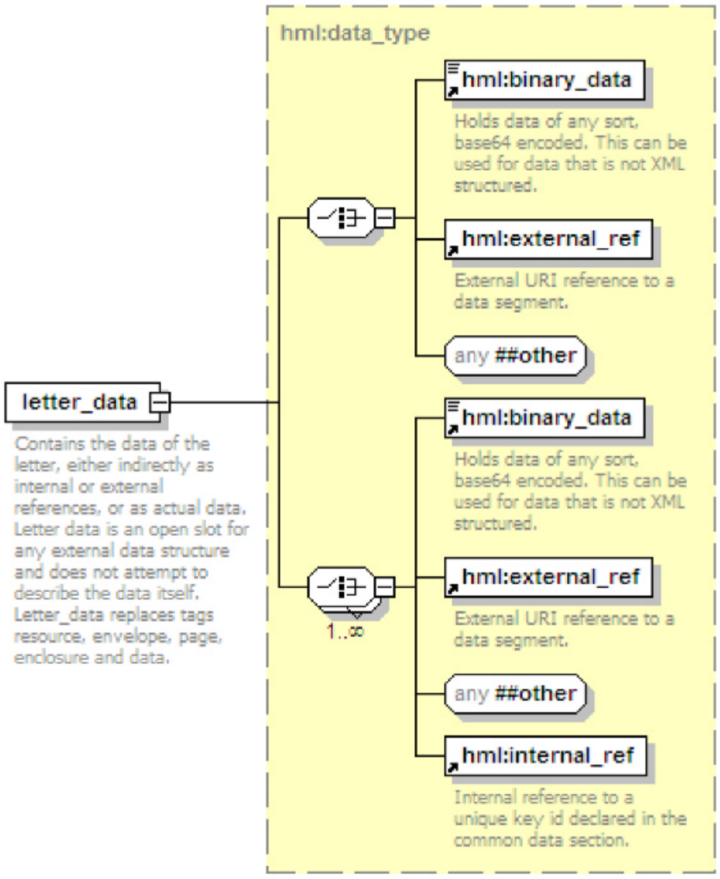
diagram	 internal_ref Internal reference to a unique key id declared in the common data section.
type	**hml:internal_refType**
used by	complexTypes **acknowledgementType addresseeType data_type maileeType**
attributes	Name Type Use Default Fixed Annotation key_id xs:IDREF required
annotation	documentation Internal reference to a unique key id declared in the common data section.
source	<xs:element name="internal_ref" type="hml:internal_refType" block="#all"> <xs:annotation> <xs:documentation>Internal reference to a unique key id declared in the common data section.</xs:documentation> </xs:annotation> </xs:element>

B.1.15 letter

diagram	 hml:letterType hml:processing 0..∞ Defines a property to follow the data contained in the current encapsulating document fragmentl and fragments inside (and after the processing tag), except if the type is redefined in an document subfragment. Similar to an environment variable for the data. Processing holds and transports the properties of the data contained in the document fragment. These options may or may not be relevant for pre-processing, processing and post-processing, depending on the HML application. hml:letter_data Contains the data of the letter, either indirectly as internal or external references, or as actual data. Letter data is an open slot for any external data structure and does not attempt to describe the data itself. Letter_data replaces tags resource, envelope, page, enclosure and data. letter The smallest message unit. A message is constituted by an instance of a letter. A bundle contains a list of letters. hml:mailee The mailee section contains the possible contact points of the sender. hml:addressee The addressee section contains the possible □□□contact points of the recipient. hml:acknowledgement 0..∞ Optional tag to provide a contact point for a recipient of bundle acknowledgement information.
type	**hml:letterType**
children	**hml:processing hml:letter_data hml:mailee hml:addressee hml:acknowledgement**
used by	complexType **bundleType**
attributes	Name Type Use Default Fixed Annotation letterclass xs:string required letterid xs:ID required

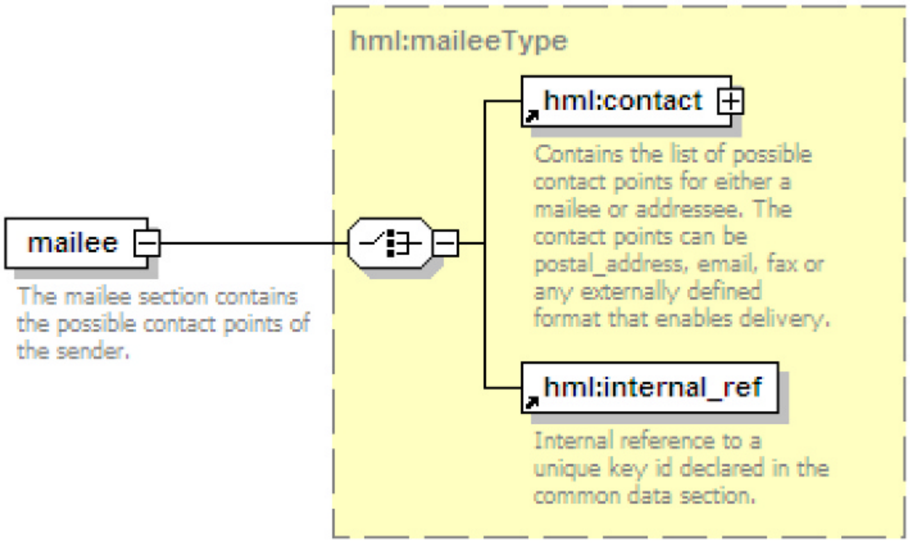
annotation	documentation The smallest message unit. A message is constituted by an instance of a letter. A bundle contains a list of letters.
source	<xs:element name="letter" type="hml:letterType" block="#all"> <xs:annotation> <xs:documentation>The smallest message unit. A message is constituted by an instance of a letter. A bundle contains a list of letters.</xs:documentation> </xs:annotation> </xs:element>

B.1.16 letter_data

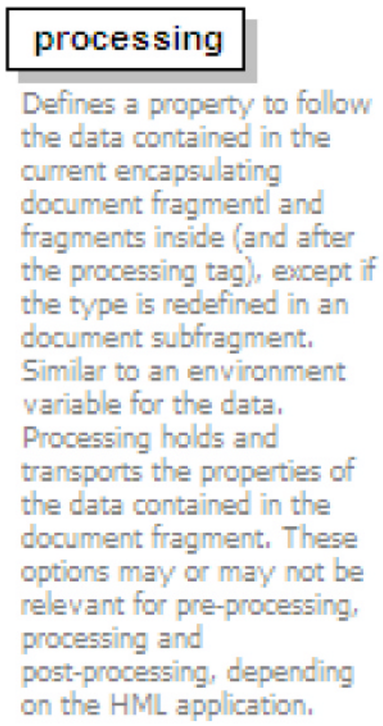
diagram	 hml:data_type hml:binary_data Holds data of any sort, base64 encoded. This can be used for data that is not XML structured. hml:external_ref External URI reference to a data segment. any ##other letter_data Contains the data of the letter, either indirectly as internal or external references, or as actual data. Letter data is an open slot for any external data structure and does not attempt to describe the data itself. Letter_data replaces tags resource, envelope, page, enclosure and data. 1..∞ hml:binary_data Holds data of any sort, base64 encoded. This can be used for data that is not XML structured. hml:external_ref External URI reference to a data segment. any ##other hml:internal_ref Internal reference to a unique key id declared in the common data section.
type	**hml:data_type**
children	**hml:binary_data hml:external_ref hml:internal_ref**
used by	complexType **letterType**
annotation	documentation Contains the data of the letter, either indirectly as internal or external references, or as actual data. Letter data is an open slot for any external data structure and does not attempt to describe the data itself. Letter_data replaces tags resource, envelope, page, enclosure and data.
source	<xs:element name="letter_data" type="hml:data_type" block="#all">

	<xs:annotation> <xs:documentation>Contains the data of the letter, either indirectly as internal or external references, or as actual data. Letter data is an open slot for any external data structure and does not attempt to describe the data itself. Letter_data replaces tags resource, envelope, page, enclosure and data.</xs:documentation> </xs:annotation> </xs:element>

B.1.17 mailee

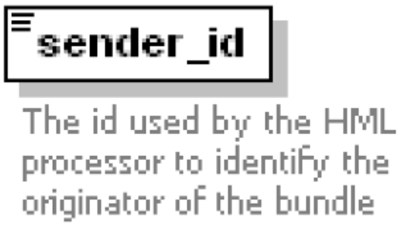
diagram	 hml:maileeType mailee The mailee section contains the possible contact points of the sender. hml:contact Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery. hml:internal_ref Internal reference to a unique key id declared in the common data section.
type	**hml:maileeType**
children	**hml:contact hml:internal_ref**
used by	complexTypes **commonType letterType**
attributes	Name Type Use Annotation preference xs:IDREFS optional preference attribute is specifying what is the order of preferred delivery channel. The IDREFS has to be the one defined in the contact. If none specified then the order in which the channels had been defined will be used.
annotation	documentation The mailee section contains the possible contact points of the sender.
source	<xs:element name="mailee" type="hml:maileeType" block="#all"> <xs:annotation> <xs:documentation>The mailee section contains the possible contact points of the sender.</xs:documentation> </xs:annotation> </xs:element>

B.1.18 processing

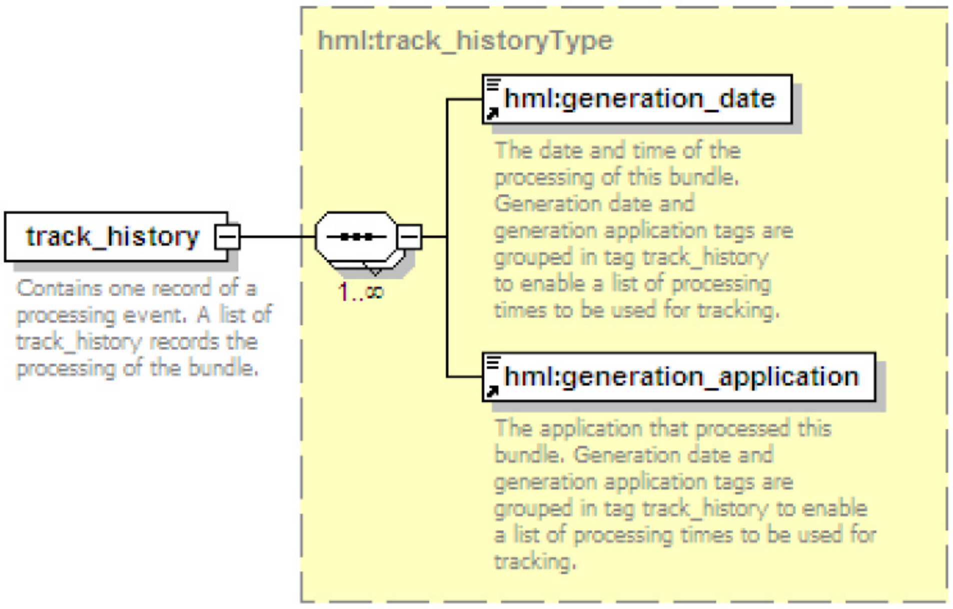
diagram	 processing Defines a property to follow the data contained in the current encapsulating document fragmentl and fragments inside (and after the processing tag), except if the type is redefined in an document subfragment. Similar to an environment variable for the data. Processing holds and transports the properties of the data contained in the document fragment. These options may or may not be relevant for pre-processing, processing and post-processing, depending on the HML application.
type	**hml:processingType**
used by	complexTypes **commonType letterType**
attributes	<table><tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr><tr><td>value</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr><tr><td>metric</td><td>xs:string</td><td>optional</td><td></td><td></td><td></td></tr><tr><td>type</td><td>xs:string</td><td>required</td><td></td><td></td><td></td></tr></table>
annotation	documentation Defines a property to follow the data contained in the current encapsulating document fragmentl and fragments inside (and after the processing tag), except if the type is redefined in an document subfragment. Similar to an environment variable for the data. Processing holds and transports the properties of the data contained in the document fragment. These options may or may not be relevant for pre-processing, processing and post-processing, depending on the HML application.
source	<xs:element name="processing" type="hml:processingType" block="#all"> <xs:annotation> <xs:documentation>Defines a property to follow the data contained in the current encapsulating document fragmentl and fragments inside (and after the processing tag), except if the type is redefined in an document subfragment. Similar to an environment variable for the data. Processing holds and transports the properties of the data contained in the document fragment. These options may or may not be relevant for pre-processing, processing and post-processing, depending on the HML application. </xs:documentation> </xs:annotation>

	</xs:element>

B.1.19 sender_id

diagram	 sender_id The id used by the HML processor to identify the originator of the bundle
type	**xs:string**
used by	complexType **declarationType**
annotation	documentation The id used by the HML processor to identify the originator of the bundle
source	<xs:element name="sender_id" type="xs:string" block="#all" final="#all"> <xs:annotation> <xs:documentation>The id used by the HML processor to identify the originator of the bundle</xs:documentation> </xs:annotation> </xs:element>

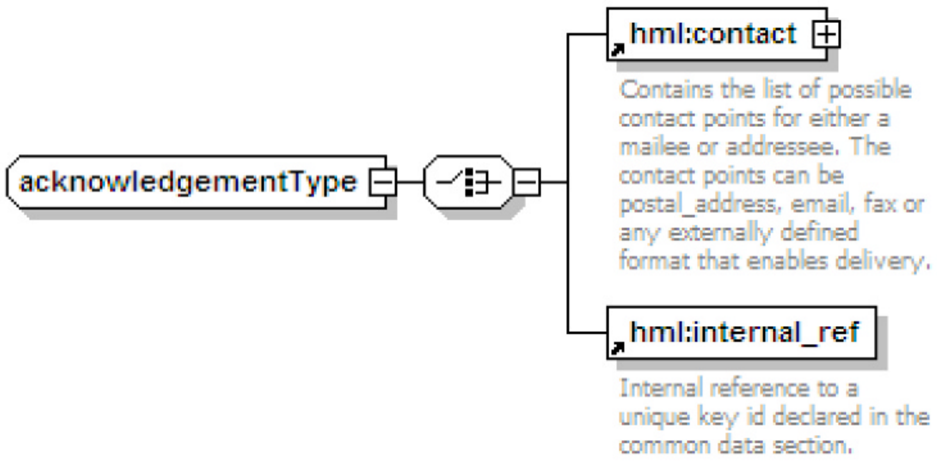
B.1.20 track_history

diagram	 hml:track_historyType hml:generation_date The date and time of the processing of this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking. track_history Contains one record of a processing event. A list of track_history records the processing of the bundle. 1..∞ hml:generation_application The application that processed this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking.
type	**hml:track_historyType**
children	**hml:generation_date hml:generation_application**
used by	complexType **declarationType**
annotation	documentation Contains one record of a processing event. A list of track_history records the

	processing of the bundle.
source	<xs:element name="track_history" type="hml:track_historyType" block="#all"> <xs:annotation> <xs:documentation>Contains one record of a processing event. A list of track_history records the processing of the bundle.</xs:documentation> </xs:annotation> </xs:element>

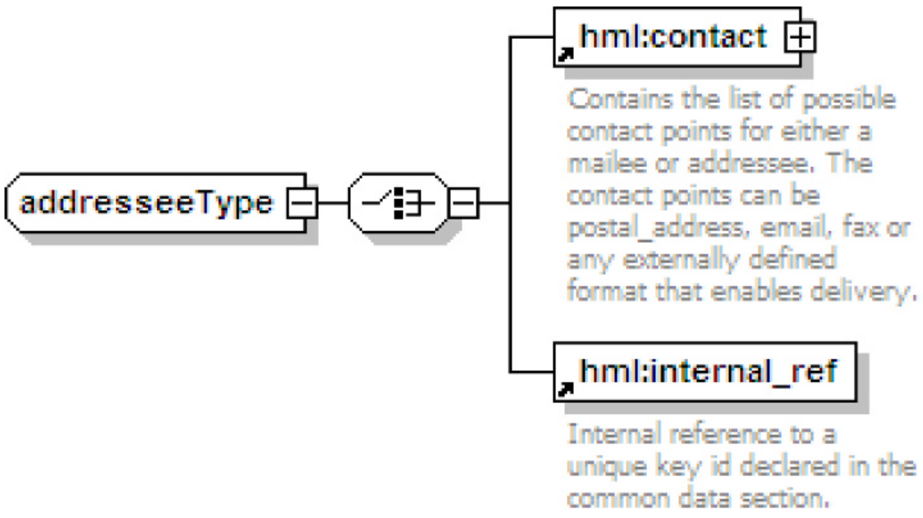
B.2 Complex types

B.2.1 acknowledgementType

diagram	 acknowledgementType hml:contact Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery. hml:internal_ref Internal reference to a unique key id declared in the common data section.
children	**hml:contact hml:internal_ref**
used by	element **acknowledgement**
attributes	Name Type Use Annotation preference xs:IDREFS optional preference attribute is specifying what is the order of preferred delivery channel. The IDREFS has to be the one defined in the contact. If none specified then the order in which the channels had been defined will be used.
source	<xs:complexType name="acknowledgementType" final="#all"> <xs:choice> <xs:element ref="hml:contact"/> <xs:element ref="hml:internal_ref"/> </xs:choice> <xs:attribute name="preference" type="xs:IDREFS" use="optional"> <xs:annotation> <xs:documentation>preference attribute is specifying what is the order of preferred delivery channel.

	The IDREFS has to be the one defined in the contact. If none specified then the order in which the channels had been defined will be used.</xs:documentation> </xs:annotation> </xs:attribute> </xs:complexType>

B.2.2 addresseeType

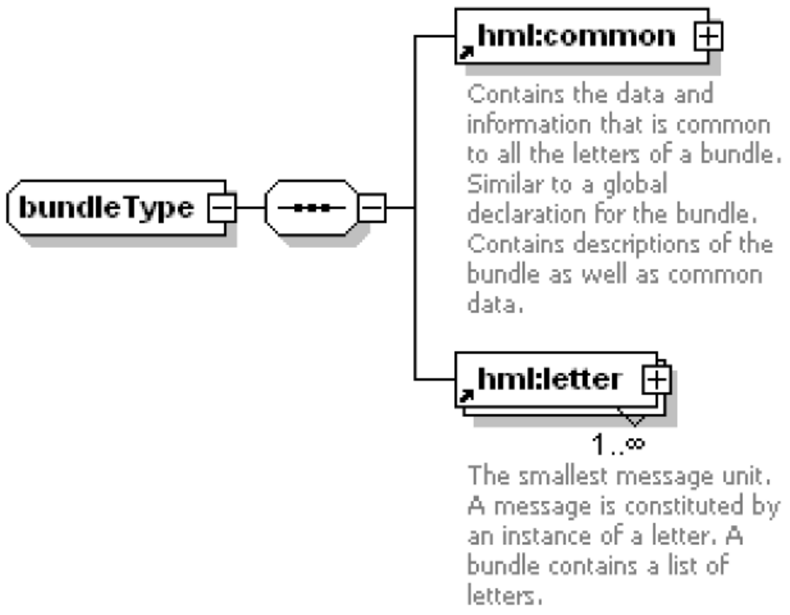
diagram	 hml:contact Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery. addresseeType hml:internal_ref Internal reference to a unique key id declared in the common data section.
children	**hml:contact hml:internal_ref**
used by	element **addressee**
attributes	Name Type Use Annotation preference xs:IDREFS optional preference attribute is specifying what is the order of preferred delivery channel. The IDREFS has to be the one defined in the contact. If none specified then the order in which the channels had been defined will be used.
source	<xs:complexType name="addresseeType" final="#all"> <xs:choice> <xs:element ref="hml:contact"/> <xs:element ref="hml:internal_ref"/> </xs:choice> <xs:attribute name="preference" type="xs:IDREFS" use="optional"> <xs:annotation> <xs:documentation>preference attribute is specifying what is the order of preferred delivery channel. The IDREFS has to be the one defined in the contact. If none specified then the order in which the channels had been defined will be used.</xs:documentation>

	</xs:annotation> </xs:attribute> </xs:complexType>

B.2.3 binary_dataType

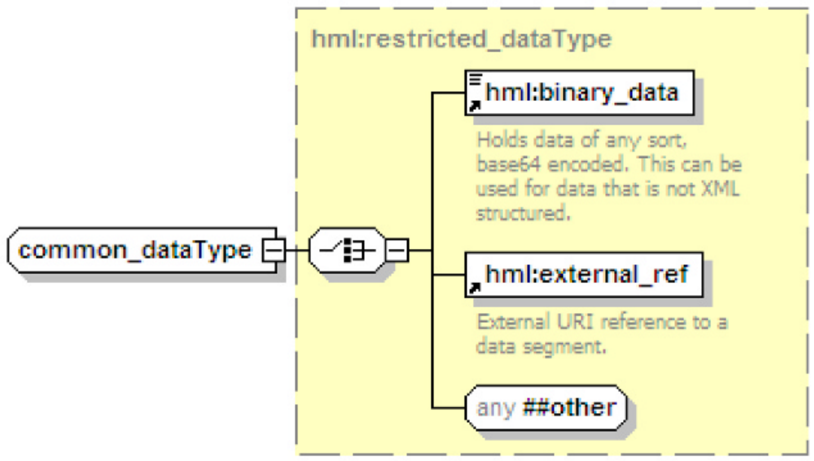
diagram	
type	extension of **xs:base64Binary**
used by	element **binary_data**
attributes	Name Type Use Default Fixed Annotation type xs:string required
source	<xs:complexType name="binary_dataType"> <xs:simpleContent> <xs:extension base="xs:base64Binary"> <xs:attribute name="type" type="xs:string" use="required"/> </xs:extension> </xs:simpleContent> </xs:complexType>

B.2.4 bundleType

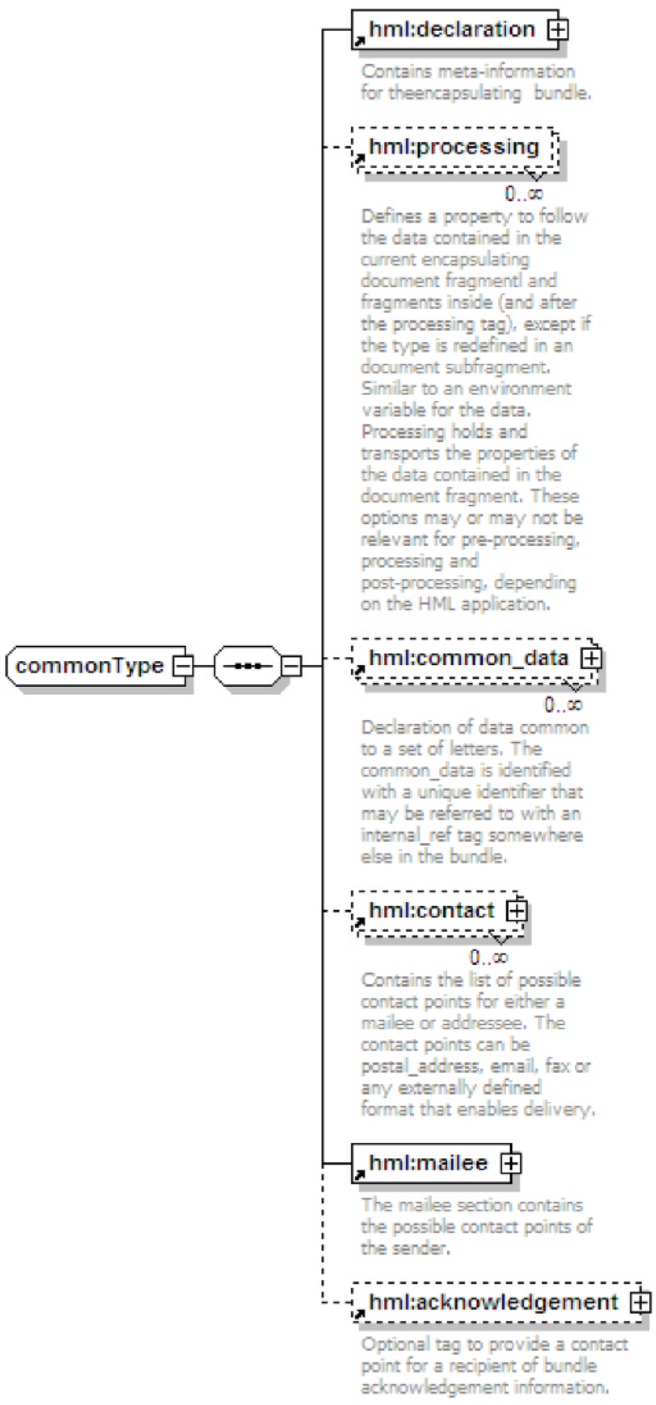
diagram	 bundleType hml:common Contains the data and information that is common to all the letters of a bundle. Similar to a global declaration for the bundle. Contains descriptions of the bundle as well as common data. hml:letter 1..∞ The smallest message unit. A message is constituted by an instance of a letter. A bundle contains a list of letters.
children	**hml:common hml:letter**
used by	element **bundle**

attributes	Name Type Use Default Fixed Annotation bundleid xs:ID optional
source	<xs:complexType name="bundleType"> <xs:sequence> <xs:element ref="hml:common"/> <xs:element ref="hml:letter" maxOccurs="unbounded"/> </xs:sequence> <xs:attribute name="bundleid" type="xs:ID" use="optional"/> </xs:complexType>

B.2.5 common_dataType

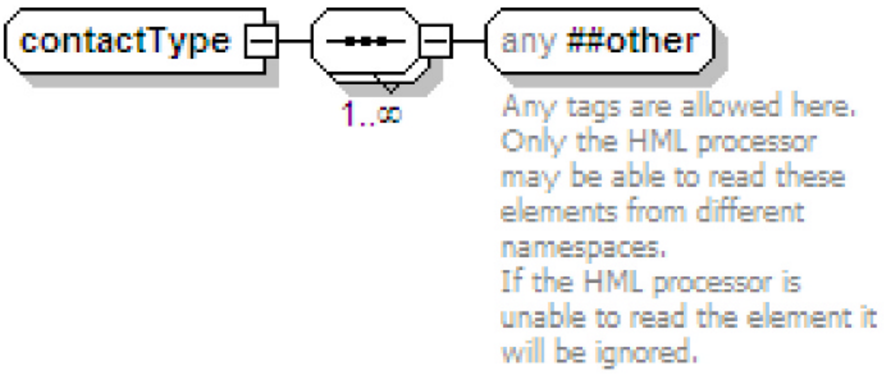
diagram	 hml:restricted_dataType common_dataType hml:binary_data Holds data of any sort, base64 encoded. This can be used for data that is not XML structured. hml:external_ref External URI reference to a data segment. any ##other
type	extension of **hml:restricted_dataType**
children	**hml:binary_data hml:external_ref**
used by	element **common_data**
attributes	Name Type Use Default Fixed Annotation id xs:ID required
source	<xs:complexType name="common_dataType"> <xs:complexContent> <xs:extension base="hml:restricted_dataType"> <xs:attribute name="id" type="xs:ID" use="required"/> </xs:extension> </xs:complexContent> </xs:complexType>

B.2.6 commonType

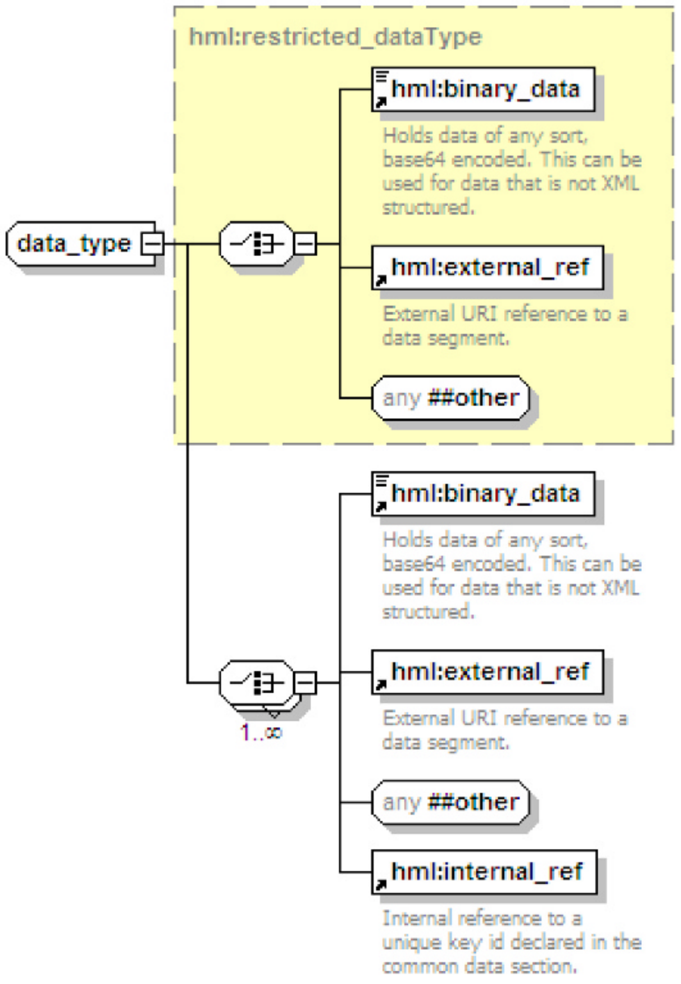
diagram	 hml:declaration Contains meta-information for theencapsulating bundle. hml:processing 0..∞ Defines a property to follow the data contained in the current encapsulating document fragmentl and fragments inside (and after the processing tag), except if the type is redefined in an document subfragment. Similar to an environment variable for the data. Processing holds and transports the properties of the data contained in the document fragment. These options may or may not be relevant for pre-processing, processing and post-processing, depending on the HML application. commonType hml:common_data 0..∞ Declaration of data common to a set of letters. The common_data is identified with a unique identifier that may be referred to with an internal_ref tag somewhere else in the bundle. hml:contact 0..∞ Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery. hml:mailee The mailee section contains the possible contact points of the sender. hml:acknowledgement Optional tag to provide a contact point for a recipient of bundle acknowledgement information.
children	**hml:declaration hml:processing hml:common_data hml:contact hml:mailee hml:acknowledgement**
used by	element **common**
source	<xs:complexType name="commonType" final="#all"> <xs:sequence> <xs:element ref="hml:declaration"/>

	<xs:element ref="hml:processing" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="hml:common_data" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="hml:contact" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="hml:mailee"/> <xs:element ref="hml:acknowledgement" minOccurs="0"/> </xs:sequence> </xs:complexType>

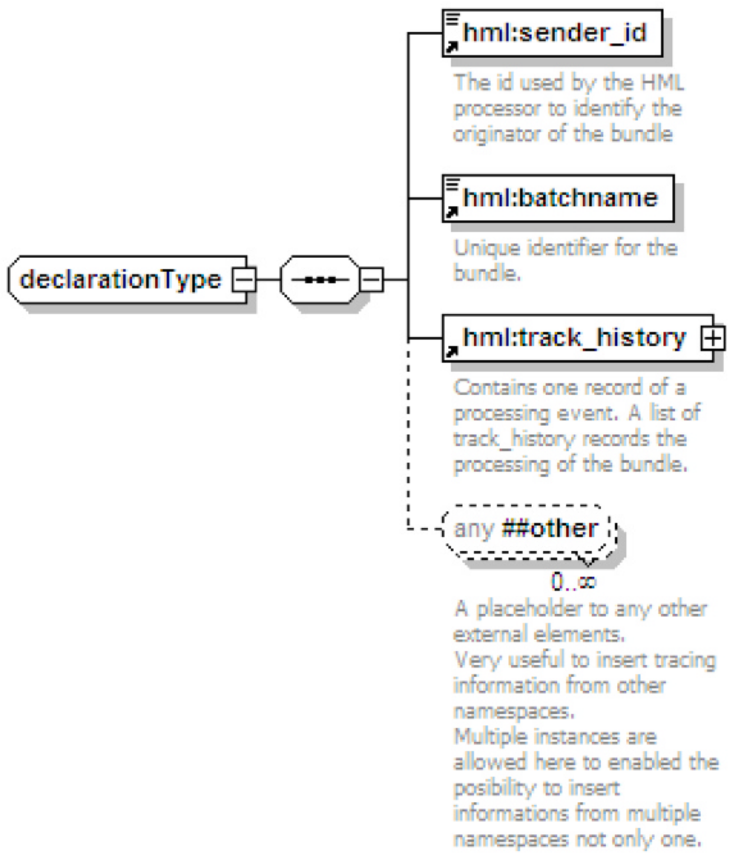
B.2.7 contactType

diagram	 contactType 1..∞ any ##other Any tags are allowed here. Only the HML processor may be able to read these elements from different namespaces. If the HML processor is unable to read the element it will be ignored.
used by	element **contact**
attributes	Name Type Use Default Fixed Annotation id xs:ID
source	<xs:complexType name="contactType" final="restriction"> <xs:sequence maxOccurs="unbounded"> <xs:any namespace="##other" processContents="strict"> <xs:annotation> <xs:documentation>Any tags are allowed here. Only the HML processor may be able to read these elements from different namespaces. If the HML processor is unable to read the element it will be ignored.</xs:documentation> </xs:annotation> </xs:any> </xs:sequence> <xs:attribute name="id" type="xs:ID"/> </xs:complexType>

B.2.8 data_type

diagram	 hml:restricted_dataType data_type hml:binary_data Holds data of any sort, base64 encoded. This can be used for data that is not XML structured. hml:external_ref External URI reference to a data segment. any ##other hml:binary_data Holds data of any sort, base64 encoded. This can be used for data that is not XML structured. hml:external_ref External URI reference to a data segment. any ##other hml:internal_ref Internal reference to a unique key id declared in the common data section. 1..∞
type	extension of **hml:restricted_dataType**
children	**hml:binary_data hml:external_ref hml:internal_ref**
used by	element **letter_data**
source	<xs:complexType name="data_type"> <xs:complexContent> <xs:extension base="hml:restricted_dataType"> <xs:choice maxOccurs="unbounded"> <xs:element ref="hml:binary_data"/> <xs:element ref="hml:external_ref"/> <xs:any namespace="##other" processContents="strict"/> <xs:element ref="hml:internal_ref"/> </xs:choice> </xs:extension> </xs:complexContent> </xs:complexType>

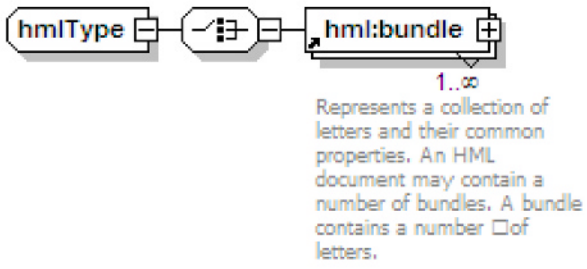
B.2.9 declarationType

diagram	 declarationType hml:sender_id The id used by the HML processor to identify the originator of the bundle hml:batchname Unique identifier for the bundle. hml:track_history Contains one record of a processing event. A list of track_history records the processing of the bundle. any ##other 0..∞ A placeholder to any other external elements. Very useful to insert tracing information from other namespaces. Multiple instances are allowed here to enabled the posibility to insert informations from multiple namespaces not only one.
children	**hml:sender_id hml:batchname hml:track_history**
used by	element **declaration**
source	<xs:complexType name="declarationType" final="#all"> <xs:sequence> <xs:element ref="hml:sender_id"/> <xs:element ref="hml:batchname"/> <xs:element ref="hml:track_history"/> <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>A placeholder to any other external elements. Very useful to insert tracing information from other namespaces. Multiple instances are allowed here to enabled the posibility to insert informations from multiple namespaces not only one.</xs:documentation> </xs:annotation> </xs:any> </xs:sequence> </xs:complexType>

B.2.10 external_refType

diagram	
used by	element **external_ref**
attributes	Name Type Use Default Fixed Annotation id xs:anyURI required type xs:string required
source	<xs:complexType name="external_refType"> <xs:attribute name="id" type="xs:anyURI" use="required"/> <xs:attribute name="type" type="xs:string" use="required"/> </xs:complexType>

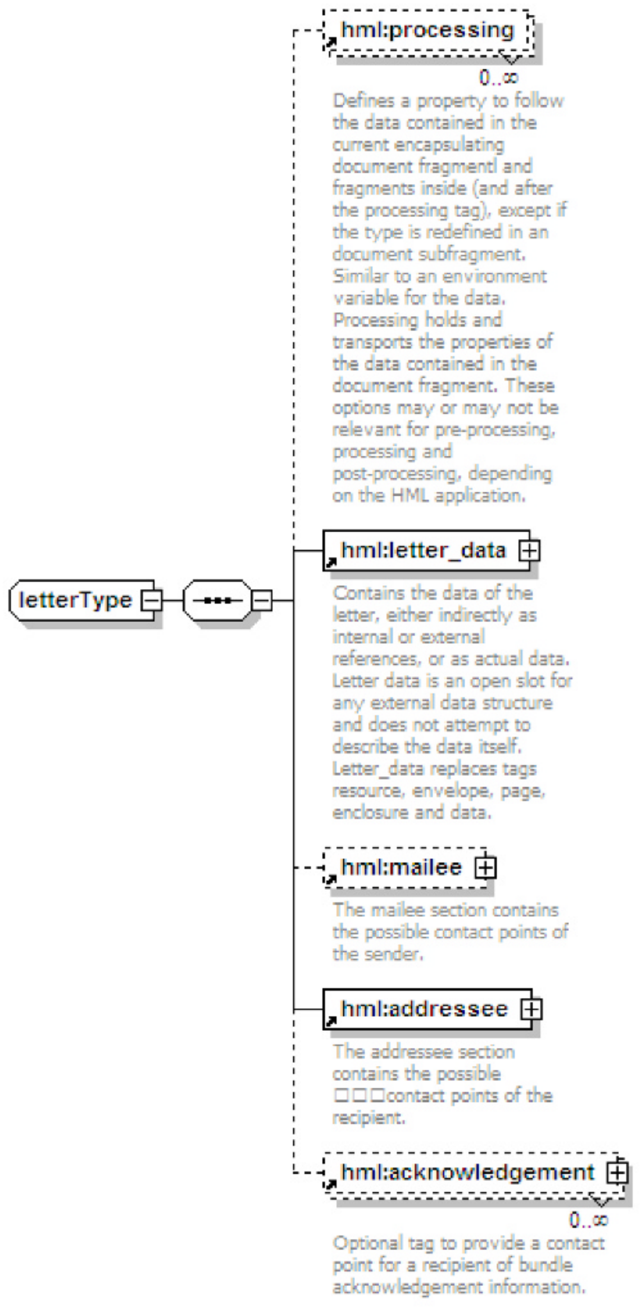
B.2.11 hmlType

diagram	 hmlType hml:bundle 1..∞ Represents a collection of letters and their common properties. An HML document may contain a number of bundles. A bundle contains a number of letters.
children	**hml:bundle**
used by	element **hml**
attributes	Name Type Use Default Fixed Annotation version xs:string required id xs:ID optional
source	<xs:complexType name="hmlType" final="#all"> <xs:choice> <xs:element ref="hml:bundle" maxOccurs="unbounded"/> </xs:choice> <xs:attribute name="version" type="xs:string" use="required"/> <xs:attribute name="id" type="xs:ID" use="optional"/> </xs:complexType>

B.2.12 internal_refType

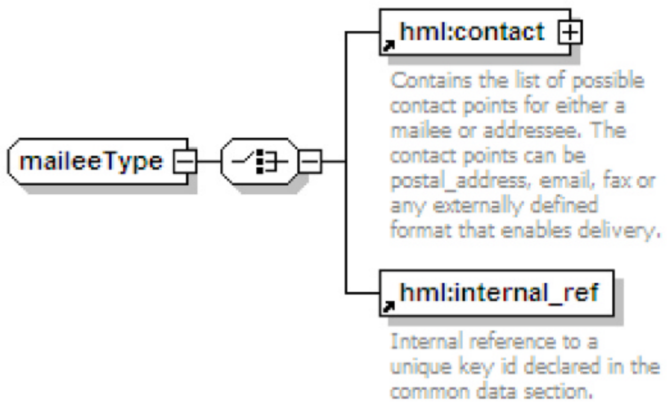
diagram	
used by	element **internal_ref**
attributes	Name Type Use Default Fixed Annotation key_id xs:IDREF required
source	<xs:complexType name="internal_refType"> <xs:attribute name="key_id" type="xs:IDREF" use="required"/> </xs:complexType>

B.2.13 letterType

diagram	 hml:processing 0..∞ Defines a property to follow the data contained in the current encapsulating document fragmentl and fragments inside (and after the processing tag), except if the type is redefined in an document subfragment. Similar to an environment variable for the data. Processing holds and transports the properties of the data contained in the document fragment. These options may or may not be relevant for pre-processing, processing and post-processing, depending on the HML application. letterType hml:letter_data Contains the data of the letter, either indirectly as internal or external references, or as actual data. Letter data is an open slot for any external data structure and does not attempt to describe the data itself. Letter_data replaces tags resource, envelope, page, enclosure and data. hml:mailee The mailee section contains the possible contact points of the sender. hml:addressee The addressee section contains the possible □□□contact points of the recipient. hml:acknowledgement 0..∞ Optional tag to provide a contact point for a recipient of bundle acknowledgement information.
children	**hml:processing hml:letter_data hml:mailee hml:addressee hml:acknowledgement**
used by	element **letter**
attributes	Name Type Use Default Fixed Annotation letterclass xs:string required letterid xs:ID required
source	<xs:complexType name="letterType" final="#all"> <xs:sequence>

	<xs:element ref="hml:processing" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="hml:letter_data"/> <xs:element ref="hml:mailee" minOccurs="0"/> <xs:element ref="hml:addressee"/> <xs:element ref="hml:acknowledgement" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> <xs:attribute name="letterclass" type="xs:string" use="required"/> <xs:attribute name="letterid" type="xs:ID" use="required"/> </xs:complexType>

B.2.14 maileeType

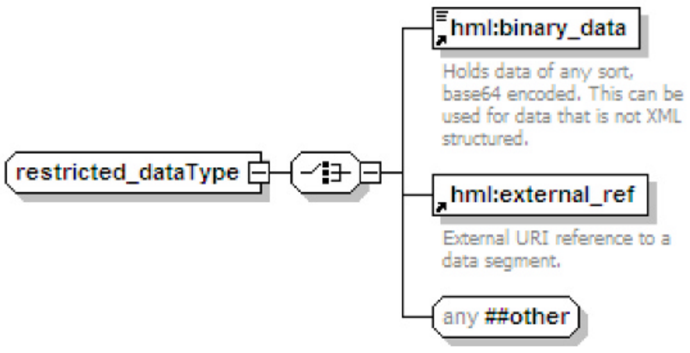
diagram	 hml:contact Contains the list of possible contact points for either a mailee or addressee. The contact points can be postal_address, email, fax or any externally defined format that enables delivery. maileeType hml:internal_ref Internal reference to a unique key id declared in the common data section.
children	**hml:contact hml:internal_ref**
used by	element **mailee**
attributes	Name Type Use Annotation preference xs:IDREFS optional preference attribute is specifying what is the order of preferred delivery channel. The IDREFS has to be the one defined in the contact. If none specified then the order in which the channels had been defined will be used.
source	<xs:complexType name="maileeType" final="#all"> <xs:choice> <xs:element ref="hml:contact"/> <xs:element ref="hml:internal_ref"/> </xs:choice> <xs:attribute name="preference" type="xs:IDREFS" use="optional"> <xs:annotation> <xs:documentation>preference attribute is specifying what is the order of preferred delivery channel. The IDREFS has to be the one defined in the contact. If none specified then the order in which the channels had been defined will be

	used.</xs:documentation> </xs:annotation> </xs:attribute> </xs:complexType>

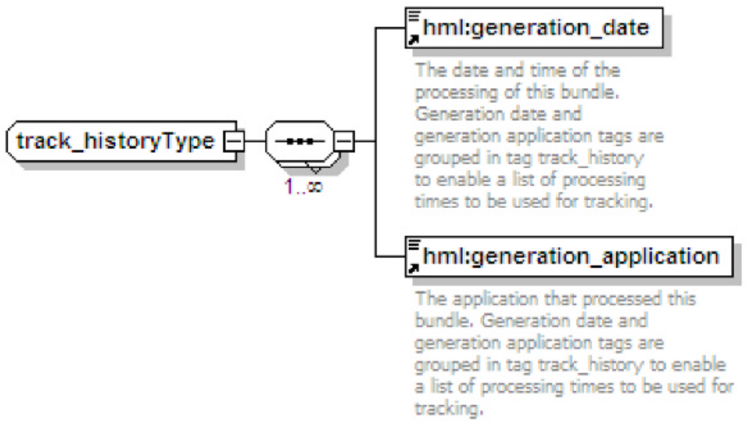
B.2.15 processingType

diagram	
used by	element **processing**
attributes	Name Type Use Default Fixed Annotation value xs:string optional metric xs:string optional type xs:string required
source	<xs:complexType name="processingType"> <xs:attribute name="value" type="xs:string" use="optional"/> <xs:attribute name="metric" type="xs:string" use="optional"/> <xs:attribute name="type" type="xs:string" use="required"/> </xs:complexType>

B.2.16 restricted_dataType

diagram	 restricted_dataType hml:binary_data Holds data of any sort, base64 encoded. This can be used for data that is not XML structured. hml:external_ref External URI reference to a data segment. any ##other
children	**hml:binary_data hml:external_ref**
used by	complexTypes **common_dataType data_type**
source	<xs:complexType name="restricted_dataType"> <xs:choice> <xs:element ref="hml:binary_data"/> <xs:element ref="hml:external_ref"/> <xs:any namespace="##other"/> </xs:choice> </xs:complexType>

B.2.17 track_historyType

diagram	 track_historyType 1..∞ hml:generation_date The date and time of the processing of this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking. hml:generation_application The application that processed this bundle. Generation date and generation application tags are grouped in tag track_history to enable a list of processing times to be used for tracking.
children	**hml:generation_date hml:generation_application**
used by	element **track_history**
source	<xs:complexType name="track_historyType" final="#all"> <xs:sequence maxOccurs="unbounded"> <xs:element ref="hml:generation_date"/> <xs:element ref="hml:generation_application"/> </xs:sequence> </xs:complexType>

Annex C
(informative)

Differences between this document and CEN/TS 14014:2006

The main divergences from the HML specification issued by CEN/TC 331 are:

1 Schema definitions have been updated from XML 1.0 to XML 1.1

2 References to XML 1.0 have been replaced by references to XML 1.1, notably to [XML-2006] in the Bibliography.

HML XSD (Annex D) has been updated

Annex D
(informative)

HML definition, copy of the HML XSD

```
<?xml version="1.1" encoding="UTF-8"?>

<!-- edited by Constantin Florescu (ErgoIDP AS) Bernard Rouille (DTC) and Jacob
Johnsen (Ipostes.com)-->

<xs:schema targetNamespace="http://www.ipostes.com/xml/"
xmlns:hml="http://www.ipostes.com/xml/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.1">

   <xs:element name="acknowledgement" type="hml:acknowledgementType"
block="#all">

      <xs:annotation>

         <xs:documentation>Optional tag to provide a contact point for a
recipient of bundle acknowledgement information.</xs:documentation>

      </xs:annotation>

   </xs:element>

   <xs:element name="addressee" type="hml:addresseeType" block="#all">

      <xs:annotation>

         <xs:documentation>The addressee section contains the possible

         contact points of the recipient.</xs:documentation>

      </xs:annotation>

   </xs:element>

   <xs:element name="batchname" type="xs:anyURI" block="#all">

      <xs:annotation>

         <xs:documentation>Unique identifier for the bundle. </xs:documentation>

      </xs:annotation>

   </xs:element>

   <xs:element name="binary_data" type="hml:binary_dataType">

      <xs:annotation>
```

```
          <xs:documentation>Holds data of any sort, base64 encoded. This can be
used for data that is not XML structured.</xs:documentation>

       </xs:annotation>

   </xs:element>

   <xs:element name="bundle" type="hml:bundleType" block="#all">

       <xs:annotation>

          <xs:documentation>Represents a collection of letters and their common
properties. An HML document may contain a number of bundles. A bundle contains a
number    of letters.</xs:documentation>

       </xs:annotation>

   </xs:element>

   <xs:element name="common" type="hml:commonType" block="#all">

       <xs:annotation>

          <xs:documentation>Contains the data and information that is common to
all the letters of a bundle. Similar to a global declaration for the bundle.
Contains descriptions of the bundle as well as common data.</xs:documentation>

       </xs:annotation>

   </xs:element>

   <xs:element name="common_data" type="hml:common_dataType" block="#all">

       <xs:annotation>

          <xs:documentation>Declaration of data common to a set of letters. The
common_data is identified with a unique identifier that may be referred to with
an internal_ref tag somewhere else in the bundle.</xs:documentation>

       </xs:annotation>

   </xs:element>

   <xs:element name="contact" type="hml:contactType" block="#all">

       <xs:annotation>

          <xs:documentation>Contains the list of possible contact points for
either a mailee or addressee. The contact points can be postal_address, email,
fax or any externally defined format that enables delivery.</xs:documentation>

       </xs:annotation>

   </xs:element>

   <xs:element name="declaration" type="hml:declarationType" block="#all">
```

```
       <xs:annotation>

           <xs:documentation>Contains meta-information for theencapsulating
bundle.</xs:documentation>

       </xs:annotation>

   </xs:element>

   <xs:element name="external_ref" type="hml:external_refType">

       <xs:annotation>

           <xs:documentation>External URI reference to a data
segment.</xs:documentation>

       </xs:annotation>

   </xs:element>

   <xs:element name="generation_application" type="xs:string" block="#all">

       <xs:annotation>

           <xs:documentation>The application that processed this bundle.
Generation date and generation application tags are grouped in tag track_history
to enable a list of processing times to be used for tracking.</xs:documentation>

       </xs:annotation>

   </xs:element>

   <xs:element name="generation_date" type="xs:dateTime" block="#all">

       <xs:annotation>

           <xs:documentation>The date and time of the processing of this bundle.
Generation date and generation application tags are grouped in tag track_history
to enable a list of processing times to be used for tracking.</xs:documentation>

       </xs:annotation>

   </xs:element>

   <xs:element name="hml" type="hml:hmlType" block="#all">

       <xs:annotation>

           <xs:documentation>The root level tag defines the
document</xs:documentation>

       </xs:annotation>

   </xs:element>

   <xs:element name="internal_ref" type="hml:internal_refType" block="#all">
```

```
      <xs:annotation>

         <xs:documentation>Internal reference to a unique key id declared in the
common data section.</xs:documentation>

      </xs:annotation>

   </xs:element>

   <xs:element name="letter" type="hml:letterType" block="#all">

      <xs:annotation>

         <xs:documentation>The smallest message unit. A message is constituted
by an instance of a letter. A bundle contains a list of
letters.</xs:documentation>

      </xs:annotation>

   </xs:element>

   <xs:element name="letter_data" type="hml:data_type" block="#all">

      <xs:annotation>

         <xs:documentation>Contains the data of the letter, either indirectly as
internal or external references, or as actual data. Letter data is an open slot
for any external data structure and does not attempt to describe the data itself.
Letter_data replaces tags  resource, envelope, page, enclosure and
data.</xs:documentation>

      </xs:annotation>

   </xs:element>

   <xs:element name="mailee" type="hml:maileeType" block="#all">

      <xs:annotation>

         <xs:documentation>The mailee section contains the possible contact
points of the sender.</xs:documentation>

      </xs:annotation>

   </xs:element>

   <xs:element name="processing" type="hml:processingType" block="#all">

      <xs:annotation>

         <xs:documentation>Defines a property to follow the data contained in
the current encapsulating document fragmentl and fragments inside (and after the
processing tag), except if the type is redefined in an  document subfragment.
Similar to an environment variable for the data. Processing holds and transports
the properties of the data contained in the document fragment. These options may
or may not be relevant for pre-processing, processing and post-processing,
depending on the HML application. </xs:documentation>
```

```
        </xs:annotation>

    </xs:element>

    <xs:element name="sender_id" type="xs:string" block="#all" final="#all">

        <xs:annotation>

            <xs:documentation>The id used by the HML  processor to identify the
originator of the bundle</xs:documentation>

        </xs:annotation>

    </xs:element>

    <xs:element name="track_history" type="hml:track_historyType" block="#all">

        <xs:annotation>

            <xs:documentation>Contains one record of a processing event. A list of
track_history records the processing of the bundle.</xs:documentation>

        </xs:annotation>

    </xs:element>

    <xs:complexType name="acknowledgementType" final="#all">

        <xs:choice>

            <xs:element ref="hml:contact"/>

            <xs:element ref="hml:internal_ref"/>

        </xs:choice>

        <xs:attribute name="preference" type="xs:IDREFS" use="optional">

            <xs:annotation>

                <xs:documentation>preference attribute is specifying what is the
order of preferred delivery channel.

The IDREFS has to be the one defined in the contact.

If none specified then the order in which the channels had been defined will be
used.</xs:documentation>

            </xs:annotation>

        </xs:attribute>

    </xs:complexType>

    <xs:complexType name="addresseeType" final="#all">

        <xs:choice>
```

```
          <xs:element ref="hml:contact"/>

          <xs:element ref="hml:internal_ref"/>

       </xs:choice>

       <xs:attribute name="preference" type="xs:IDREFS" use="optional">

          <xs:annotation>

             <xs:documentation>preference attribute is specifying what is the
order of preferred delivery channel.

The IDREFS has to be the one defined in the contact.

If none specified then the order in which the channels had been defined will be
used.</xs:documentation>

          </xs:annotation>

       </xs:attribute>

    </xs:complexType>

    <xs:complexType name="binary_dataType">

       <xs:simpleContent>

          <xs:extension base="xs:base64Binary">

             <xs:attribute name="type" type="xs:string" use="required"/>

          </xs:extension>

       </xs:simpleContent>

    </xs:complexType>

    <xs:complexType name="bundleType">

       <xs:sequence>

          <xs:element ref="hml:common"/>

          <xs:element ref="hml:letter" maxOccurs="unbounded"/>

       </xs:sequence>

       <xs:attribute name="bundleid" type="xs:ID" use="optional"/>

    </xs:complexType>

    <xs:complexType name="commonType" final="#all">

       <xs:sequence>

          <xs:element ref="hml:declaration"/>
          <xs:element ref="hml:processing" minOccurs="0" maxOccurs="unbounded"/>

          <xs:element ref="hml:common_data" minOccurs="0" maxOccurs="unbounded"/>

          <xs:element ref="hml:contact" minOccurs="0" maxOccurs="unbounded"/>

          <xs:element ref="hml:mailee"/>

          <xs:element ref="hml:acknowledgement" minOccurs="0"/>

       </xs:sequence>

   </xs:complexType>

   <xs:complexType name="common_dataType">

       <xs:complexContent>

          <xs:extension base="hml:restricted_dataType">

             <xs:attribute name="id" type="xs:ID" use="required"/>

          </xs:extension>

       </xs:complexContent>

   </xs:complexType>

   <xs:complexType name="contactType" final="restriction">

       <xs:sequence maxOccurs="unbounded">

          <xs:any namespace="##other" processContents="strict">

             <xs:annotation>

                <xs:documentation>Any tags are allowed here.

Only the HML processor may be able to read these elements from different
namespaces.

If the HML processor is unable to read the element it will be
ignored.</xs:documentation>

             </xs:annotation>

          </xs:any>

       </xs:sequence>

       <xs:attribute name="id" type="xs:ID"/>

   </xs:complexType>

   <xs:complexType name="data_type">

       <xs:complexContent>
```

```
            <xs:extension base="hml:restricted_dataType">

               <xs:choice maxOccurs="unbounded">

                  <xs:element ref="hml:binary_data"/>

                  <xs:element ref="hml:external_ref"/>

                  <xs:any namespace="##other"  processContents="strict"/>

                  <xs:element ref="hml:internal_ref"/>

               </xs:choice>

            </xs:extension>

         </xs:complexContent>

      </xs:complexType>

      <xs:complexType name="declarationType" final="#all">

         <xs:sequence>

            <xs:element ref="hml:sender_id"/>

            <xs:element ref="hml:batchname"/>

            <xs:element ref="hml:track_history"/>

            <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded">

               <xs:annotation>

                  <xs:documentation>A placeholder to any other external elements.

   Very useful to insert tracing information from other namespaces.

   Multiple instances are allowed here to enabled the posibility to insert
   informations from multiple namespaces not only one.</xs:documentation>

               </xs:annotation>

            </xs:any>

         </xs:sequence>

      </xs:complexType>

      <xs:complexType name="external_refType">

         <xs:attribute name="id" type="xs:anyURI" use="required"/>

         <xs:attribute name="type" type="xs:string" use="required"/>

      </xs:complexType>
   <xs:complexType name="hmlType" final="#all">

      <xs:choice>

         <xs:element ref="hml:bundle" maxOccurs="unbounded"/>

      </xs:choice>

      <xs:attribute name="version" type="xs:string" use="required"/>

      <xs:attribute name="id" type="xs:ID" use="optional"/>

   </xs:complexType>

   <xs:complexType name="internal_refType">

      <xs:attribute name="key_id" type="xs:IDREF" use="required"/>

   </xs:complexType>

   <xs:complexType name="letterType" final="#all">

      <xs:sequence>

         <xs:element ref="hml:processing" minOccurs="0" maxOccurs="unbounded"/>

         <xs:element ref="hml:letter_data"/>

         <xs:element ref="hml:mailee" minOccurs="0"/>

         <xs:element ref="hml:addressee"/>

         <xs:element ref="hml:acknowledgement" minOccurs="0"
maxOccurs="unbounded"/>

      </xs:sequence>

      <xs:attribute name="letterclass" type="xs:string" use="required"/>

      <xs:attribute name="letterid" type="xs:ID" use="required"/>

   </xs:complexType>

   <xs:complexType name="maileeType" final="#all">

      <xs:choice>

         <xs:element ref="hml:contact"/>

         <xs:element ref="hml:internal_ref"/>

      </xs:choice>

      <xs:attribute name="preference" type="xs:IDREFS" use="optional">

         <xs:annotation>
```

```
             <xs:documentation>preference attribute is specifying what is the
order of preferred delivery channel.

The IDREFS has to be the one defined in the contact.

If none specified then the order in which the channels had been defined will be
used.</xs:documentation>

          </xs:annotation>

       </xs:attribute>

    </xs:complexType>

    <xs:complexType name="processingType">

       <xs:attribute name="value" type="xs:string" use="optional"/>

       <xs:attribute name="metric" type="xs:string" use="optional"/>

       <xs:attribute name="type" type="xs:string" use="required"/>

    </xs:complexType>

    <xs:complexType name="restricted_dataType">

       <xs:choice>

          <xs:element ref="hml:binary_data"/>

          <xs:element ref="hml:external_ref"/>

          <xs:any namespace="##other"/>

       </xs:choice>

    </xs:complexType>

    <xs:complexType name="track_historyType" final="#all">

       <xs:sequence maxOccurs="unbounded">

          <xs:element ref="hml:generation_date"/>

          <xs:element ref="hml:generation_application"/>

       </xs:sequence>

    </xs:complexType>

</xs:schema>
```

Bibliography

[1] EN ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes (ISO 3166-1)*

[2] ISO 3166-2, *Codes for the representation of names of countries and their subdivisions — Part 2: Country subdivision code*

[3] ISO 8879, *Information processing — Text and office systems — Standard Generalized Markup Language (SGML)*

[4] REC-xml-20060816, Extensible Markup Language (XML) 1.1 (Second Edition), issued by W3C. Referenced in this document as [XML-2006]. See http://www.w3.org/TR/xml11/ for latest version.

[5] REC-xml-names-20091208, *Namespaces in XML issued by W3C. Referenced in this document as [XML-names-2004]. Seefor latest version.*

[6] RFC 3548 (RFC3548), "*The Base16, Base32, and Base64 Data Encodings" by The Internet Society. Referenced in this document as [Base64]. Seefor latest version.*

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

...making excellence a habit.™