![SIST logo]

SLOVENSKI STANDARD

SIST-TS CEN/TS 14014:2007

01-januar-2007

DcýbY˙ghcf]hj Y!˙<]Vf]XbU˙dcýHU!˙8 YZ[b]W]ˇUˇcVˇ]_YˇXc_i a YbHˇLA @nU glfUb_Y#I dcfUVb]_Yˇdf]ˇdcbi Xb]_i .ˇg_i dbcˇi dcfUVb]ˇgYnbUa ˙cnbU V

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

Postalische Dienstleistungen - Hybrid mail - Document Typ Definition (DTD) für Kunden zum Dienstanbieter: Eine allgemein verwendbare Auflistung von vordefinierten Regeln

Services postaux - Courrier hybride - Définition XML de l'encapsulation des lettres pour le traitement automatisé du courrier

Ta slovenski standard je istoveten z: **CEN/TS 14014:2006**

ICS:

03.240	Poštne storitve	Postal services
35.240.99	W] [¦æà} ã z\^Á^zãç^ÁVÁæ å¦˘*ã@Á[å¦[b˜@	IT applications in other fields

SIST-TS CEN/TS 14014:2007 **en**

2003-01.Slovenski inštitut za standardizacijo. Razmnoževanje celote ali delov tega standarda ni dovoljeno.

TECHNICAL SPECIFICATION

SPÉCIFICATION TECHNIQUE

TECHNISCHE SPEZIFIKATION

CEN/TS 14014

May 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 le traitement automatisé du courrier

Postalische Dienstleistungen - Hybrid mail - Document Typ Definition (DTD) für Kunden zum Dienstanbieter: Eine allgemein verwendbare Auflistung von vordefinierten Regeln

This Technical Specification (CEN/TS) was approved by CEN on 4 March 2006 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.

CEN members are the national standards bodies of Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.

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

Management Centre: rue de Stassart, 36   B-1050 Brussels

Ref. No. CEN/TS 14014:2006: E

Contents

Page

iTeh STANDARD PREVIEW

(standards.iteh.ai)

Foreword

This Technical Specification (CEN/TS 14014:2006) has been prepared by Technical Committee CEN/TC 331 "Postal services", the secretariat of which is held by NEN.

This Technical Specification supersedes ENV 14014:2001. An explanation of the differences between this Technical Specification and ENV 14014 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 CEN Technical Specification: Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and 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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

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 a 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 contains 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).

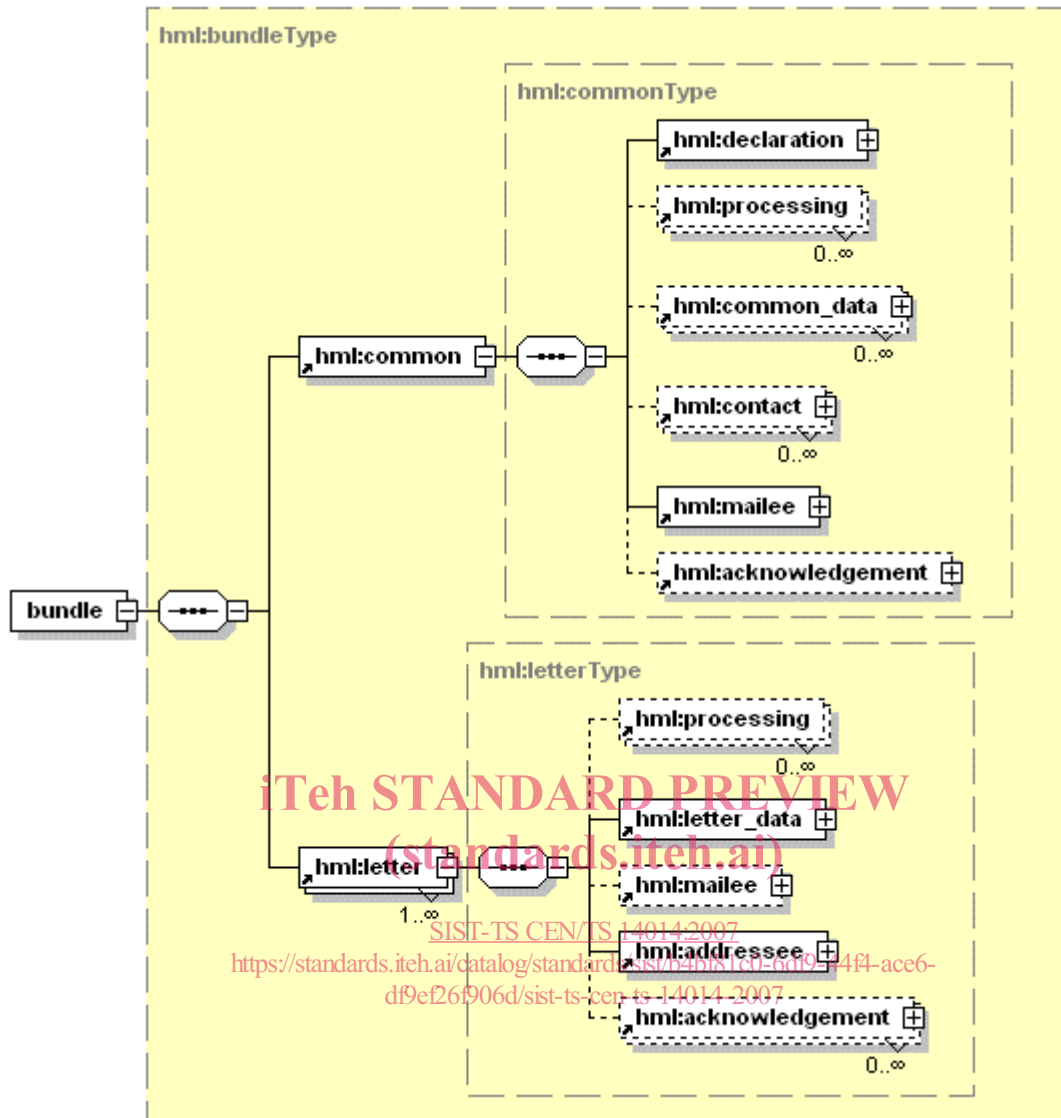**Figure 1 — Structure of a HML (Hybrid Mail Language)**

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 referenced documents are indispensable for the application of this Technical Specification. 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 Multiple-Octet Coded Character Set (UCS)

3   Terms and definitions

For the purposes of this Technical Specification, 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   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   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   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 Technical Specification and to be later used for writing documents, and the result of a possible translation of existing data structure(s). In this Technical Specification the target language is **HML**.

NOTE   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 1.0 as described in [XML-2004]. 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-2004]

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:
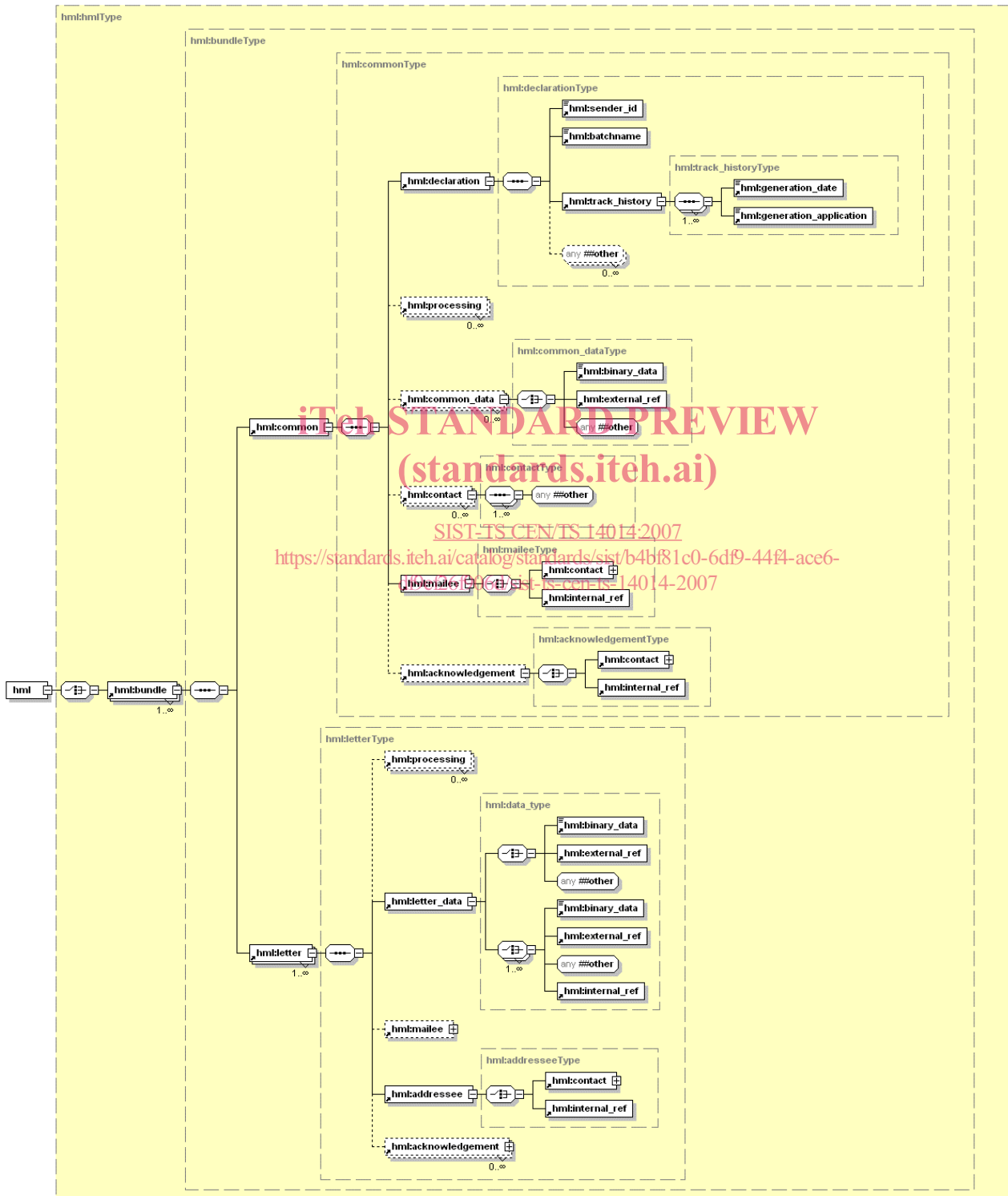

Figure 2 — Illustration of the structure of the HML XSD

6.2 General rule for HML documents

6.2.1   General

HML is based on eXtensible Markup Language (XML) version 1.0 as described in [XML-2004]. 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 is the ISO/IEC 10646-1. 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.