

1 ebXML Transport, Routing & Packaging 2 Messaging Service Specification

3 Working Draft 10-August-2000

4 This version:

5 ebXML Messaging Service Specification v0-1.doc

6 Latest version:

7 N/A

8 Previous version:

9 ebXML Transport, Routing & Packaging Message Envelope Specification 0.6

10 ebXML Message Header Specification v0-63

11 Editors:

12 Dick Brooks <dick@8760.com>

13 David Burdett <david.burdett@commerceone.com>

14 Authors:

15 Dick Brooks <dick@8760.com>

16 David Burdett <david.burdett@commerceone.com>

17 Christopher Ferris <chris.ferris@sun.com>

18 John Ibbotson <john_ibbotson@uk.ibm.com>

19 Nick Kassem <nick.kassem@sun.com>

20 Contributors:

21 The members of the Transport Routing and Packaging Project Team

22 Abstract

23 This document is a draft proposal whose purpose is to solicit additional input and convey the
24 current state of the *ebXML Message* structure recommendations.

25 This document defines the envelope and header structure used to encapsulate data for transport
26 between parties. Every attempt has been made to ensure that ebXML requirements as stated in
27 the ebXML Transport, Routing and Packaging: Overview and Requirements, Version 0.96, are
28 addressed. The current specification is a working draft. Some of the requirements are not yet
29 supported. Adherence to industry standards, consideration of existing business-to-business
30 practices and support for small and medium enterprises were key factors influencing the
31 direction of this specification.

32 Notational Conventions

33 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
34 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be
35 interpreted as described in Key Words for Use in RFC's to Indicate Requirement Levels (RFC
36 2119).



37 Terms in *Italics* are defined in the ebXML Glossary of Terms [Glossary]. Terms listed in **Bold**
38 **Italics** represent the element and/or attribute content of the *ebXML Message Header*.

39 **Status of this Document**

40 This document represents work in progress upon which no reliance should be made.



41 Table of Contents

42	1	Introduction	5
43	1.1	Purpose and Scope	5
44	1.1.1	Goals	5
45	1.2	Related ebXML Specifications	5
46	1.3	Specification Structure	6
47	1.4	General Conventions	6
48	2	Packaging Specification	6
49	2.1	ebXML Message Structure	6
50	2.1.1	MIME usage Conventions	7
51	2.2	ebXML Message Envelope	7
52	2.2.1	Content-Type	8
53	2.2.1.1	type Attribute	8
54	2.2.1.2	boundary Attribute	8
55	2.2.1.3	version Attribute	8
56	2.2.1.4	charset Attribute	8
57	2.2.2	Content-Length	9
58	2.2.3	ebXML Message Envelope Example	9
59	2.3	ebXML Header Container	9
60	2.3.1	Content-ID	9
61	2.3.2	Content-Length	9
62	2.3.3	Content-Type	10
63	2.3.3.1	version Attribute	10
64	2.3.3.2	charset Attribute	10
65	2.3.4	ebXML Header Container Example	10
66	2.4	ebXML Payload Container	10
67	2.4.1	Content-ID	11
68	2.4.2	Content-Length	11
69	2.4.3	Content-Type	11
70	2.4.4	Example of an ebXML Payload Container	11
71	3	ebXML Header Document	12
72	3.1	Root Element	12
73	3.2	Manifest	13
74	3.2.1	DocumentReference	13
75	3.3	Header	13
76	3.3.1	From and To	14
77	3.3.2	TPAInfo	14
78	3.3.3	MessageData	14
79	3.3.4	ReliableMessagingInfo	15
80	4	Normative References	15
81	5	Acknowledgments	15
82	6	Authors' Address	16
83	Appendix A	Schemas and DTD Definitions	18
84	A.1	XML Header DTD	18
85	A.2	XML Header Schema Definition	19
86	Appendix B	Examples	22



87	B.1	Complete Example of an ebXML Message Envelope using multipart/related Content-Type sent via HTTP POST.....	22
88			
89	B.2	Complete Example of an ebXML Message Envelope using multipart/related Content-Type sent via SMTP.....	25
90			
91	Appendix C	Candidate Packaging Technologies and Selection Process.....	30
92	C.1	Selection Process.....	30
93	C.2	MIME.....	30
94	C.3	XML.....	30
95	C.4	Conclusion.....	31
96	Appendix D	MIME Type discussion	32
97	Appendix E	Communication Protocol Envelope Mappings.....	32
98	E.1	HTTP.....	32
99	E.2	SMTP	32
100	E.3	FTP	32
101	Appendix D	Non-Normative References.....	33



1 Introduction

This specification defines an ebXML Messaging Service that describes how to securely and reliably exchange messages between two parties. It includes descriptions of:

- the *ebXML Message* structure used to encapsulate (package) *ebXML Message* payloads for transport between parties, and
- the behavior of the messaging service that sends or receives those messages.

No assumption or dependency is made relative to communication protocol or type of payload. The specifications contained here are both payload and communication protocol neutral.

1.1 Purpose and Scope

This document defines the enveloping and *ebXML Message* header structure used to transfer *ebXML Messages* over a data communication mechanism. This document provides sufficient detail to develop software for the packaging, exchange and processing of *ebXML Messages*.

NOTE: Message security, extensibility, service interface, reliability, and versioning will be addressed in future versions of this document.

1.1.1 Goals

The goals of this specification are to:

- Meet the requirements as specified by the ebXML Transport, Routing and Packaging: Overview and Requirements, Version 0.96 [TRPREQ]
- Be compatible with other ebXML specifications
- Leverage existing industry standards
- Enable parties to "package" very simple to very complex payloads
- Be payload neutral
- Be communication protocol neutral

1.2 Related ebXML Specifications

The following set of related specifications will be delivered in phases:

- **ebXML Messaging Service Specification** (this document) - defines the structure of the messages and the behavior of messaging services software. This will include:
 - definitions of the messages
 - behavior of the messaging service software
 - reliable messaging
 - message security
 - extensibility and versioning



- **ebXML Trading Partner Specification** (under development) - defines how one party can discover and/or agree the information that party needs to know about another party prior to sending them a message that complies with this specification
- **ebXML Messaging Service Interface Specification** (to be developed) - defines an interface that may be used by software to interact with an ebXML Messaging Service

1.3 Specification Structure

This specification is organized around the following main topics:

- **Packaging Specification** - A description of how to package an *ebXML Message* and associated parts. This section includes specifications for the various structures and containers.
- **Message Headers** - A specification of the structure and composition of the information necessary for an ebXML Messaging Service to successfully generate or process an ebXML compliant message.

Appendices to the specification cover:

- Appendix A Schemas and DTD Definitions
- Appendix B Examples
- Appendix C Candidate Packaging Technologies and Selection Process
- Appendix D MIME Type discussion
- Appendix E Communication Protocol Envelope Mappings
- Appendix D Non-Normative References

1.4 General Conventions

For all messages following the ebXML standard, a single message structure is defined, regardless of message type

2 Packaging Specification

2.1 ebXML Message Structure

An *ebXML Message* consists of:

- an outer Communication Protocol Envelope, such as HTTP or SMTP,
- a communication protocol independent *ebXML Message Envelope*, specifically MIME multipart/related, that contains the two main parts of the Message:
 - a *ebXML Header Container* that is used to envelope one *ebXML Header Document*, and
 - an optional single *ebXML Payload Container* that **MUST** be used to envelope the real payload of the Message if payload is present

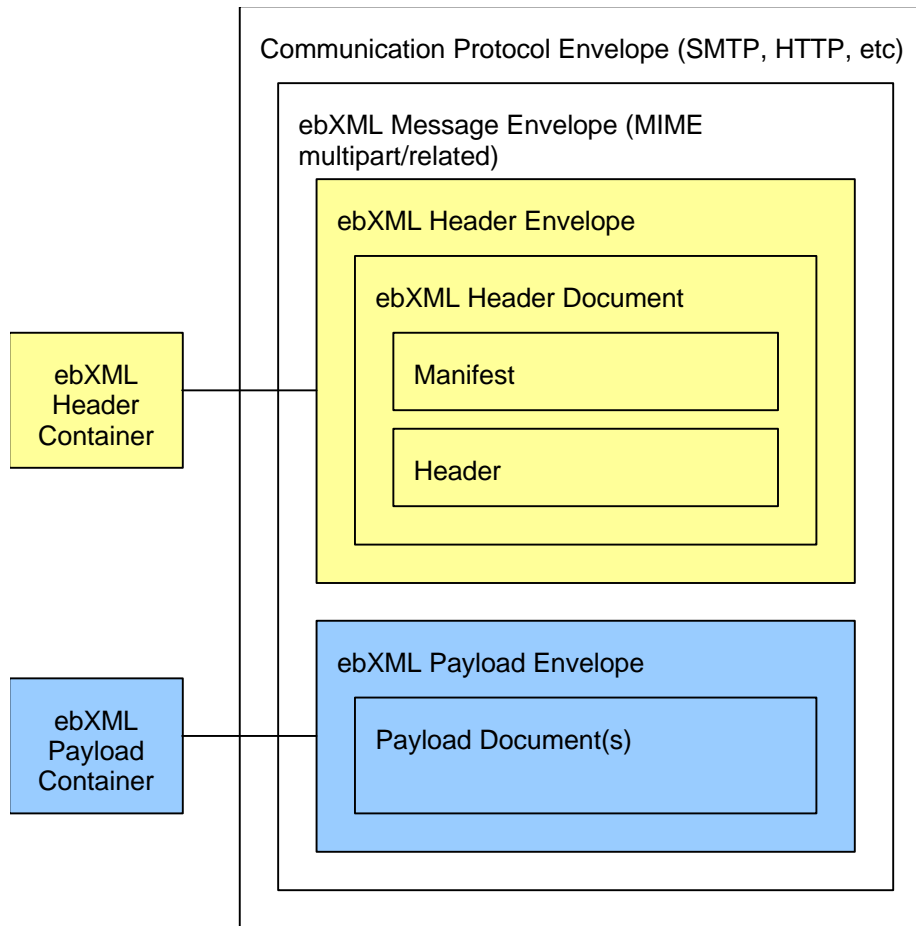


Figure 2-1 ebXML Message Structure

An *ebXML Header (or Payload) Envelope* are the MIME headers that are associated with a MIME part.

An *ebXML Header (or Payload) Document* is the content of the MIME part and is:

- an XML document in an ebXML Header, or
- an XML or some other document for the ebXML Payload

The rules for creating a Communication protocol Envelope are described in Appendix E

2.1.1 MIME usage Conventions

Values associated with MIME header attributes are valid in both quoted and unquoted form. For example, the forms `type="ebxml"` and `type=ebxml` are both valid.

2.2 ebXML Message Envelope

The *ebXML Message Envelope* is used to identify the message as an ebXML compliant structure and encapsulates the header and payload body parts. It MUST conform to [RFC2045] and MUST contain two MIME headers:

- `Content-Type`



- Content-Length

2.2.1 Content-Type

Content-Type MUST be set to `multipart/related` for all *ebXML Message Envelopes*. See Appendix C for selection rationale. For example:

```
Content-Type: multipart/related;
```

The Content-Type header contains four attributes:

- type
- boundary
- version, and
- charset.

2.2.1.1 type Attribute

The type attribute is used to identify the *ebXML Message Envelope* as an ebXML compliant structure. It MUST be set to `"application/vnd.eb+xml"`. For example:

```
type="application/vnd.eb+xml"
```

2.2.1.2 boundary Attribute

The boundary attribute is used to identify the body part separator used to identify the start and end points of each body part contained in the message. The boundary SHOULD be chosen carefully to insure that it does not occur within the content area of a body part see [RFC 2045] for guidance on how to do this. For example:

```
boundary:="-----8760"
```

2.2.1.3 version Attribute

The version attribute is used to identify the particular version of *ebXML Message Envelope* being used. There are currently two valid values for version:

1. "0" indicating a version-less message; ALL ebXML implementations must support version-less messages
2. "0.1" indicating the current version of ebXML.

Currently, there are no version-less *ebXML Message Envelopes* defined, therefore all message headers SHOULD USE "0.1". For example:

```
version="0.1"
```

2.2.1.4 charset Attribute

The charset attribute is used to identify the character set used to create the message. The list of valid values can be found at <http://www.iana.org/>. The default charset value is "iso-8859-1". For example:

```
charset="iso-8859-1"
```




216 2.2.2 Content-Length

217 The Content-Length header is a decimal value used to identify the total number of OCTETS
218 contained in all constituent message body parts, including body part boundaries. Example:

```
219 Content-Length: 9841
```

220 2.2.3 ebXML Message Envelope Example

221 An example of a compliant *ebXML Message Envelope* appears as follows:

```
222 Content-Type: multipart/related; type="application/vnd.eb+xml" "boundary:="-----8760"  
223 charset="iso-8859-1"  
224 Content-Length: 9841
```

225 2.3 ebXML Header Container

226 The *ebXML Header Container* is a MIME body part that MUST consist of:

- 227 • one *ebXML Header Envelope*, and
- 228 • one *ebXML Header Document*

229 The *ebXML Header Document* is described in section 3 of this document.

230 The following rules apply:

- 231 • the *ebXML Header Container* MUST be the first MIME body part in the *ebXML Message*.
- 232 • there MUST be one and only one *ebXML Header Document* associated with every *ebXML*
233 *Message*.

234 The *ebXML Header Envelope* conforms to [RFC 2045] and MUST consist of three MIME
235 headers:

- 236 • Content-ID
- 237 • Content-Length
- 238 • Content-Type

239 The *ebXML Header Document* within the content portion of the container MAY be enhanced
240 during transport, provided it has not been digitally signed. Any change in the size of the *ebXML*
241 *Header Document* MUST be reflected in Content-Length attribute of the *ebXML Message*
242 *Envelope* and *ebXML Header Envelope*.

243 2.3.1 Content-ID

244 The Content-ID MIME header identifies this instance of an ebXML Message header body part.
245 The value for Content-ID SHOULD be a unique identifier, in accordance with RFC 2045. For
246 example:

```
247 Content-ID: <2000-0722-161201-123456789@ebxmlhost.realm>
```

248 2.3.2 Content-Length

249 The Content-Length header is a decimal value used to identify the total number of OCTETS
250 contained in the *ebXML Header Container* MIME body part. For example:



251 Content-Length: 4208

252 2.3.3 Content-Type

253 The Content-Type for an ebXML header is identified with the value
254 "application/vnd.eb+xml". Content-Type MUST contain two attributes:

- 255 • version, and
- 256 • charset

257 2.3.3.1 version Attribute

- 258 • The version attribute indicates the version of the ebXML Messaging Service Specification to
259 which the *ebXML Header Document* conforms. For example:

260 `version="1.0";`

261 2.3.3.2 charset Attribute

262 The charset attribute identifies the character set used to create the message. The list of valid
263 values can be found at <http://www.iana.org/>.

264 The charset attribute SHALL be equivalent to the encoding attribute of the *ebXML Header*
265 *Document* (see section 3). For maximum interoperability it is RECOMMENDED that [UTF-8] be
266 used. Note: this is not the default for MIME.

267 For example:

268 `charset="UTF-8"`

269 2.3.4 ebXML Header Container Example

270 The following represents an example of an *ebXML Header Envelope* and *ebXML Header*
271 *Document*:

272	Content-ID: ebxmlheader-123	-----		
273	Content-Length: 2048		ebXML Header Envelope	
274	Content-Type: application/vnd.eb+xml	-----		
275				ebXML
276	<ebXMLHeaderDocument>	-----		Header
277	<MessageHeader>.....		ebXML Header Document	Container
278	</MessageHeader>			
279	</ebXMLHeaderDocument>	-----		

280 A complete example of an *ebXML Header Container* is presented in Appendix B.

281 2.4 ebXML Payload Container

282 If the *ebXML Message* contains a payload, then a single *ebXML Payload Container* MUST be
283 used to envelop it.

284 If there is no payload within the *ebXML Message* then the *ebXML Payload Container* MUST not
285 be present.

286 The contents of the *ebXML Payload Container* MUST be identified by the *Message Manifest*
287 element within the *ebXML Header Document* (see section 3.2).

288 If the *Message Manifest* is an empty element then an ebXML Payload Container MUST NOT be
289 present in the *ebXML Message*.



290 If an *ebXML Payload Container* is present then it MUST conform to [RFC2045] and MUST
291 consist of:

- 292 • a MIME header portion - the *ebXML Payload Envelope*, and
- 293 • a content portion - the payload itself.

294 The *ebXML Payload Envelope*, MUST consist of three MIME headers:

- 295 • Content-ID
- 296 • Content-Length
- 297 • Content-Type

298 The ebXML Messaging Service Specification makes no provision, nor limits in any way the
299 structure or content of payloads. Payloads MAY be a simple plain text object or complex nested
300 multipart objects. This is the implementer's decision.

301 2.4.1 Content-ID

302 The Content-ID MIME Header is used to uniquely identify an instance of an *ebXML Message*
303 payload body part. The value for Content-ID SHOULD be a unique identifier, in accordance with
304 MIME [RFC 2045]. For example:

305 `Content-ID: <2000-0722-161201-987654321@ebxmlhost.realm>`

306 2.4.2 Content-Length

307 The Content-Length header is a decimal value used to identify the total number of OCTETS
308 contained in the content portion of the *ebXML Payload Container*. For example:

309 `Content-Length: 5012`

310 2.4.3 Content-Type

311 The Content-Type for an ebXML payload is determined by the implementer and is used to
312 identify the type of data contained in the content portion of the *ebXML Payload Container*. For
313 example:

314 `Content-Type: application/xml`

315 2.4.4 Example of an ebXML Payload Container

316 The following represents an example of an *ebXML Payload Envelope* and a payload:

317	Content-ID: ebxmlpayload-123	-----	ebXML Payload Envelope	ebXML Payload Container	
318	Content-Length: 4096				
319	Content-Type: application/xml	-----			
320			Payload		
321	<Invoice>	-----			
322	<Invoicedata>.....				
323	</Invoicedata>				
324	</Invoice>	-----			

325 A complete example of the ebXML Payload Container is presented in Appendix B.



3 ebXML Header Document

The *ebXML Header Document* is a single [XML] document with a number of principal header elements within it where each principal header element is a separate XML element.

In general, separate principal header elements are used where:

- different software is likely to be used to generate that header element,
- the structure of the header element might vary independently of the other header elements, or
- the data contained in the header element MAY need to be digitally signed separately from the other header elements.

3.1 Root Element

The root element of the *ebXML Header Document* is named **ebXMLHeader**. It is comprised of three attributes and two subordinate elements.

The first attribute is the namespace declaration (*xmlns*) (see [XML Namespace] which has a REQUIRED value of "http://www.ebxml.org/namespaces/messageHeader".

The second attribute is the **Version** attribute. This attribute is required. Its purpose is to provide for future versioning capabilities. It has a default value of '1.0'.

The last of the **ebXMLHeader** attributes is the **MessageType** attribute. Its purpose is to enable ebXML-aware software to distinguish between normal and communication protocol-specific messages, such as acknowledgment and error messages.

The **MessageType** is an enumeration consisting of three possible values:

- **Normal** – the ebXML Payload Container contains data that has been provided to the ebXML Messaging Service by the software that called it
- **Acknowledgment** – a ebXML Messaging Service-specific acknowledgment message.
- **Error** – a ebXML Messaging Service-specific error message.

The **ebXMLHeader** element contains the following two elements:

- **MessageManifest** - contains a list of references to the other parts of the Message. This includes references to the documents, which comprise the *Payload* of the *Message*.
- **MessageHeader** - contains the information REQUIRED by the recipient to process the message. The message originator creates this information to which additional information MAY be added.

The **MessageHeader** and **MessageManifest** are REQUIRED elements in every *Message*.

The following is a sample **ebXMLHeader** document fragment demonstrating the overall structure:

```
<?xml version="1.0"?>
<ebXMLMessageHeader xmlns="http://www.ebxml.org/namespaces/messageHeader"
  Version="1.0" MessageType="Normal">
  <Manifest>...</Manifest>
  <Header>...</Header>
</ebXMLMessageHeader>
```



3.2 Manifest

The required **Manifest** element is a composite element consisting of zero or more **DocumentReference** elements. Each **DocumentReference** element identifies data associated with the message, whether included as part of the message, or remote resources accessible via a URL. The **Manifest** SHALL be the first subordinate element in the **ebXMLMessageHeader**. It identifies the payload document(s) contained in the *ebXML Message Container*. The purpose of the **Manifest** is to make it easier to directly extract a particular document associated with the Message. See also section

3.2.1 DocumentReference

The **DocumentReference** element is a composite element consisting of two required subordinate elements as follows:

- **DocumentDescription** - an optional textual description of the document/resource
- **DocumentLabel** - a code that enables the purpose of the referenced document to be determined without retrieving it
- **DocumentId** - a URL of the Content-ID of a MIME body part, as defined in [RFC2111], representing payload data, or a remote URL to some external resource.

The following fragment demonstrates a typical **Manifest** for a message with a single payload MIME body part:

```
<Manifest>
  <DocumentReference>
    <DocumentLabel>PurchaseOrder</DocumentLabel>
    <DocumentId>cid:0987654321</DocumentId>
  </DocumentReference>
</Manifest>
```

3.3 Header

The **Header** element immediately follows the **Manifest** element. It is required in all **ebXMLMessageHeader** documents. The **Header** element is a composite element comprised of the following required subordinate elements:

- **From** – the logical address of the sender of the message.
- **To** – the logical address of the intended recipient of the message.
- **TPAInfo** – a composite set of information which relates to the *Trading Partner Agreement* under which the message is governed
- **MessageData** – a composite set of information which uniquely identifies the *Message*
- **ReliableMessagingInfo** - information which identifies the degree of reliability with which the message SHOULD be delivered

The following fragment demonstrates the structure of the **Header** element of the **ebXMLMessageHeader** document:

```
<Header>
  <From>...</From>
  <To>...</To>
  <TPAInfo>...</TPAInfo>
  <MessageData>...</MessageData>
  <ReliableMessagingInfo>...</ReliableMessagingInfo>
</Header>
```



3.3.1 From and To

The **From** element identifies the *Party* which originated the message. It is a logical identifier, which MAY take the form of a URN. An example of this would be a DUNS number. The **From** element consists of a **PartyId** element.

The **To** element identifies the intended recipient of the message. As with **From**, it is a logical identifier which is comprised of a **PartyId** element.

The **PartyId** element has a single attribute; **context** and a text value. The purpose of the context attribute is to provide a context for the text value of the **PartyId** element. The following fragment demonstrates usage of the **From** and **To** elements of the **ebXMLMessageHeader**.

```
<From>
  <PartyId context="DUNS">12345</PartyId>
</From>
<To>
  <PartyId context="DUNS">54321</PartyId>
</To>
```

3.3.2 TPAINfo

The **TPAINfo** element follows the **From** and **To** elements in the **Header** element structure. The **TPAINfo** element is a composite set of information which relates to the *Trading Partner Agreement* under which the message is governed. The **TPAINfo** element has four subordinate elements as follows:

- **TPAId** – a URI which identifies the *Trading Partner Agreement* which governs the processing of the message
- **ConversationId** – a URI which identifies the set of related messages that make up a conversation between two *Parties*
- **ServiceInterface** – Identifies the Service Interface that SHOULD act on the payload in the message. It is unique within the domain of the *Party* to which the message is being sent. URN's MAY be considered suitable for the element content.
- **Action** – Identifies a process within a Service Interface, which processes the Message. **Action** SHALL be unique within the Service Interface in which it is defined.

The following example fragment demonstrates the usage of the **TPAINfo** element.

```
<TPAINfo>
  <TPAId context = "tpadb">12345678</TPAId>
  <ConversationId context = "tpadb">987654321</ConversationId>
  <ServiceInterface>QuoteToCollect</ServiceInterface>
  <Action>NewPurchaseOrder</Action>
</TPAINfo>
```

3.3.3 MessageData

The required **MessageData** element follows the **TPAINfo** element. The purpose of the **MessageData** element is to provide a means of identifying an *ebXML Message*. It is a composite element which contains the following three elements:

- **MessageId** – a unique identifier for the message conforming to [RFC2392]. The "local part" of the identifier is implementation dependent.
- **TimeStamp** – a value representing the time that the message header was created conforming to [ISO-8601]. The format of CCYYMMDDTHHMMSS.SSSZ is used. This time format is Coordinated Universal Time (UTC).



- **RefToMessageId** – an optional reference to an earlier *ebXML Message*. If there is no earlier message then the element **MUST** be empty. If element is not empty then it **MUST** contain the value of the **MessageId** of the earlier related *ebXML Message*.

The following example demonstrates the usage of the **MessageData** element.

```
<MessageData>
  <MessageId>UUID-2</MessageId>
  <TimeStamp>20000725T121905.000Z</TimeStamp>
  <RefToMessageId>UUID-1</RefToMessageId>
</MessageData>
```

3.3.4 ReliableMessagingInfo

The last element of the **ebXMLMessageHeader** is the **ReliableMessagingInfo** element. This element identifies the degree of reliability with which the message will be delivered. This element has a single attribute, **DeliverySemantics**. This attribute is an enumeration, which may have one of the following values:

- "AtMostOnce" – reliable messaging semantics, which specifies that a given message will be received by the Service Interface handler no more than once.
- "Unspecified" – reliable delivery semantics are not specified.

```
<ReliableMessagingInfo>
  <DeliverySemantics>AtMostOnce</DeliverySemantics>
  <TimeStamp>20000725T121905.000Z</TimeStamp>
  <RefToMessageId>UUID-1</RefToMessageId>
</ReliableMessagingInfo>
```

4 Normative References

- [Glossary] ebXML Glossary, see ebXML Project Team Home Page
- [ISO 8601] International Standards Organization Ref. ISO 8601 Second Edition, Published 1997
- [RFC 2392] IETF Request For Comments 2111. Content-ID and Message-ID Uniform Resource Locators. E. Levinson, Published August 1998
- [RFC2045] IETF RFC 2045. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, N Freed & N Borenstein, Published November 1996
- [TRPREQ] ebXML Transport, Routing and Packaging: Overview and Requirements, Version 0.96, Published 25 May 2000
- [UTF-8] UTF-8 is an encoding that conforms to ISO/IEC 10646. See [XML] for usage conventions.
- [XML Namespace] Recommendation for Namespaces in XML, World Wide Web Consortium, 14 January 1999, "<http://www.w3.org/TR/REC-xml-names>"
- [XML] Extensible Mark Up Language. A W3C recommendation. See <http://www.w3.org/TR/1998/REC-xml-19980210> for the 10 February 1998 version.

5 Acknowledgments

The author's wish to acknowledge the support of the members of the Transport, Routing and Packaging Project Team who contributed ideas to this specification by the group's discussion e-mail list, on conference calls and during face-to-face meetings.



495 Ralph Berwanger – bTrade.com
496 Jonathan Borden - Author of XMTP
497 Jon Bosak - Sun
498 Doug Bunting - Ariba
499 David Burdett - Commerce One
500 Lawrence Ding - WorldSpan
501 Rik Drummond - Drummond Group
502 Christopher Ferris - Sun
503 Ian Jones - BT
504 Henry Lowe - OMG
505 Jim McCarthy - webXI
506 Bob Miller - GEIS
507 Dale Moberg - Sterling Commerce
508 Kathy Spector – Extricity
509 Martha Warfelt - DaimlerChrysler
510 Prasad Yendluri - Vitria

511 **6 Authors' Address**

512 Contact information for the Authors of this specification follow.

513 David Burdett
514 Commerce One Inc
515 4400 Rosewood Drive 3rd Fl, Bldg 4,
516 Pleasanton, CA 94588,
517 USA
518 Telephone: (925) 520-4422 or (650) 623-2888
519 E-mail: david.burdett@commerceone.com

520 John Ibbotson
521 IBM UK Ltd
522 Hursley Park
523 Winchester
524 SO21 2JN
525 United Kingdom
526 Telephone: +44 (1962) 815188
527 E-mail: john_ibbotson@uk.ibm.com

528 Christopher Ferris
529 Sun Microsystems, Inc.
530 One Network Drive
531 Burlington, Ma 01803-0903
532 USA
533 Telephone: (781) 442-3063
534 E-mail: chris.ferris@sun.com



535 Dick Brooks
536 Group 8760
537 110 12th Street North
538 Suite F103
539 Birmingham, Alabama 35203
540 USA
541 Telephone: (205) 250-8053
542 E-mail: dick@8760.com

543 Nicholas Kassem
544 Java Software, Sun Microsystems
545 901 San Antonio Road, MS CUP02-201
546 Palo Alto, CA 94303-4900
547 Telephone: (408) 863-3535
548 E-mail: Nick.Kassem@eng.sun.com



Appendix A Schemas and DTD Definitions

The following are definitions for validation of the *ebXML Message* header structure.

A.1 XML Header DTD

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/1999/XMLSchema">
<ELEMENT ebXMLHeader (Manifest , Header )>
<!ATTLIST ebXMLHeader Version CDATA #FIXED '1.0'
      MessageType CDATA #FIXED 'Normal' >
<ELEMENT Manifest (DocumentReference )+>
<ELEMENT DocumentReference (DocumentDescription?, DocumentLabel , DocumentId )>
<ELEMENT DocumentDescription (#PCDATA )>
<!ATTLIST DocumentDescription e-dtype NMTOKEN #FIXED 'string' >
<ELEMENT DocumentLabel (#PCDATA )>
<!ATTLIST DocumentLabel e-dtype NMTOKEN #FIXED 'string' >
<ELEMENT DocumentId (#PCDATA )>
<!ATTLIST DocumentId e-dtype NMTOKEN #FIXED 'uri' >
<ELEMENT Header (From , To , TPAInfo , MessageData , ReliableMessagingInfo )>
<ELEMENT TPAInfo (TPAId , ConversationId , ServiceInterface , Action )>
<ELEMENT ServiceInterface (#PCDATA )>
<!ATTLIST ServiceInterface e-dtype NMTOKEN #FIXED 'string' >
<ELEMENT Action (#PCDATA )>
<!ATTLIST Action e-dtype NMTOKEN #FIXED 'string' >
<ELEMENT TPAId (#PCDATA )>
<!ATTLIST TPAId context CDATA 'Undefined'
      e-dtype NMTOKEN #FIXED 'uri' >
<ELEMENT ConversationId (#PCDATA )>
<!ATTLIST ConversationId context CDATA 'Undefined'
      e-dtype NMTOKEN #FIXED 'uri' >
<ELEMENT MessageData (MessageId , TimeStamp , RefToMessageId )>
<ELEMENT RefToMessageId (#PCDATA )>
<!ATTLIST RefToMessageId e-dtype NMTOKEN #FIXED 'uuid' >
<ELEMENT TimeStamp (#PCDATA )>
<!ATTLIST TimeStamp e-dtype NMTOKEN #FIXED 'dateTime' >
<ELEMENT MessageId (#PCDATA )>
<!ATTLIST MessageId e-dtype NMTOKEN #FIXED 'uuid' >
<ELEMENT From (PartyId )>
<ELEMENT To (PartyId )>
<ELEMENT PartyId (#PCDATA )>
<!ATTLIST PartyId context CDATA 'Undefined'
      e-dtype NMTOKEN #FIXED 'uri' >
<ELEMENT ReliableMessagingInfo EMPTY>
<!ATTLIST ReliableMessagingInfo DeliverySemantics (AtMostOnce | Unspecified ) #FIXED
'Unspecified' >
```



592 A.2 XML Header Schema Definition

```
593 <?xml version="1.0"?>
594 <schema xmlns="http://www.w3.org/1999/XMLSchema">
595   <element name="ebXMLHeader">
596     <complexType content="elementOnly">
597       <sequence>
598         <element ref="Manifest"/>
599         <element ref="Header"/>
600       </sequence>
601       <attribute name="Version" use="fixed" value="1.0" type="string"/>
602       <attribute name="MessageType" use="fixed" value="Normal" type="string"/>
603     </complexType>
604   </element>
605
606   <element name="Manifest">
607     <complexType content="elementOnly">
608       <sequence minOccurs="0" maxOccurs="unbounded">
609         <element ref="DocumentReference"/>
610       </sequence>
611     </complexType>
612   </element>
613
614   <element name="DocumentReference">
615     <complexType content="elementOnly">
616       <sequence minOccurs="1" maxOccurs="unbounded">
617         <element ref="DocumentDescription"/>
618         <element ref="DocumentLabel"/>
619         <element ref="DocumentId"/>
620       </sequence>
621     </complexType>
622   </element>
623
624   <element name="DocumentLabel" type="string">
625   </element>
626
627   <element name="DocumentId" type="uri">
628   </element>
629
630   <element name="Header">
631     <complexType content="elementOnly">
632       <sequence>
633         <element ref="From"/>
634         <element ref="To"/>
635         <element ref="TPA"/>
636         <element ref="MessageData"/>
637         <element ref="ReliableMessagingInfo"/>
638       </sequence>
639     </complexType>
640   </element>
641
642   <element name="BusinessServiceInterface" type="string">
```



```
643     </element>
644
645     <element name = "Action" type = "string"/>
646     <element name = "TPAId">
647         <complexType base = "uri" content = "textOnly">
648             <attribute name = "context" use = "default" value = "Undefined" type = "string"/>
649         </complexType>
650     </element>
651
652     <element name = "ConversationId">
653         <complexType base = "uri" content = "textOnly">
654             <attribute name = "context" use = "default" value = "Undefined" type = "string"/>
655         </complexType>
656     </element>
657
658     <element name = "MessageData">
659         <complexType content = "elementOnly">
660             <sequence>
661                 <element ref = "MessageId"/>
662                 <element ref = "TimeStamp"/>
663                 <element ref = "RefToMessageId"/>
664             </sequence>
665         </complexType>
666     </element>
667
668     <element name = "RefToMessageId" type = "uuid">
669     </element>
670
671     <element name = "TimeStamp" type = "dateTime">
672     </element>
673
674     <element name = "MessageId" type = "uuid">
675     </element>
676
677     <element name = "From">
678         <complexType content = "elementOnly">
679             <sequence>
680                 <element ref = "PartyId"/>
681             </sequence>
682         </complexType>
683     </element>
684
685     <element name = "To">
686         <complexType content = "elementOnly">
687             <sequence>
688                 <element ref = "PartyId"/>
689             </sequence>
690         </complexType>
691     </element>
692
693     <element name = "PartyId">
694         <complexType base = "uri" content = "textOnly">
695             <attribute name = "context" use = "default" value = "Undefined" type = "string"/>
696         </complexType>
```



```
697 </element>
698
699 <element name = "ReliableMessagingInfo">
700   <complexType content = "empty">
701     <attribute name = "DeliverySemantics" use = "fixed" value = "Unspecified">
702       <simpleType base = "ENUMERATION">
703         <enumeration value = "AtMostOnce"/>
704         <enumeration value = "Unspecified"/>
705       </simpleType>
706     </attribute>
707   </complexType>
708 </element>
709
710 <element name = "TPAInfo">
711   <complexType content = "elementOnly">
712     <sequence>
713       <element ref = "TPAId"/>
714       <element ref = "ConversationId"/>
715       <element ref = "BusinessServiceInterface"/>
716       <element ref = "Action"/>
717     </sequence>
718   </complexType>
719 </element>
720
721 </schema>
```



Appendix B Examples

The following are complete examples of *ebXML Messages* showing the structure as defined in this specification.

B.1 Complete Example of an ebXML Message Envelope using multipart/related Content-Type sent via HTTP POST

```
POST /ebxmlhandler HTTP/1.1
Accept: multipart/related
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Group 8760 InsideAgent
Host: localhost:9090
Connection: Keep-Alive
Content-Type: multipart/related; type=application/vnd.eb+xml; version=0.1;
boundary=-----7d02a82e5f8
Content-Length: 9293

-----7d02a82e5f8
Content-ID: ebxmlheader-9981
Content-Length: 211
Content-Type: application/vnd.eb+xml; charset="UTF-8";

<?xml version="1.0" encoding="UTF-8"?>
<ebXMLMessageHeader xmlns='http://www.xml.org/ebXMLStds/ebXMLMessageHeaderv1'>
  <Version>1.0</Version>
  <MessageType>Request</MessageType>
  <ServiceType>Payroll</ServiceType>
  <Intent>RecordCommission</Intent>
</ebXMLMessageHeader>
-----7d02a82e5f8
Content-ID: ebxmlpayload-9981
Content-Length: 7517
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v2.5 - http://www.xmlspy.com -->
<HITISMessage xmlns="" Version="1.0">
  <Header OriginalBodyRequested="false" ImmediateResponseRequired="true">
    <FromURI>http://www.pms.com/HITISInterface</FromURI>
    <ToURI>http://www.crs.com/HITISInterface</ToURI>
    <ReplyToURI>http://www.pms.com/HITISInterface</ReplyToURI>
    <MessageID>1234567890</MessageID>
    <OriginalMessageID>1234567890</OriginalMessageID>
    <TimeStamp>1999-11-10T10:23:44</TimeStamp>
    <Token>1234-567-8901</Token>
    <!--Token to be assigned in response to HITISRegister-->
  </Header>
  <Body>
    <HITISOperation OperationName="CommissionEventsUpdate">
      <CommissionEvents>
        <CommissionEvent>
          <ConfirmationID>18097YZ</ConfirmationID>
          <ConfirmationOriginatorCode>DBZ223</ConfirmationOriginatorCode>
          <CommissionOriginatorCode>3457YTXV</CommissionOriginatorCode>
          <ReservationID>098787818097YZ</ReservationID>
          <HotelReference>
            <ChainCode>HI234</ChainCode>
            <HotelCode>1234STL</HotelCode>
          </HotelReference>
          <OriginalBookingDate>19991223T17:53:22</OriginalBookingDate>
          <StayDateRange>
            <StartInstant>20000122</StartInstant>
            <Duration>00000003T000000</Duration>
          </StayDateRange>
        </CommissionEvent>
      </CommissionEvents>
    </HITISOperation>
  </Body>
</HITISMessage>
```



```
785      <GuestNames>
786        <NameInfo>
787          <NamePrefix>Mr.</NamePrefix>
788          <NameFirst>John</NameFirst>
789          <NameMiddle>Q.</NameMiddle>
790          <NameSur>jones</NameSur>
791          <NameSuffix>Jr.</NameSuffix>
792          <NameTitle>Professor</NameTitle>
793          <NameOrdered>JohnJones</NameOrdered>
794        </NameInfo>
795        <NameInfo>
796          <NamePrefix>Mrs.</NamePrefix>
797          <NameFirst>Sally</NameFirst>
798          <NameMiddle>T.</NameMiddle>
799          <NameSur>Jones</NameSur>
800          <NameSuffix/>
801          <NameTitle/>
802          <NameOrdered>SallyJones</NameOrdered>
803        </NameInfo>
804      </GuestNames>
805      <ProfileCertification CertificationType="ARC">
806        <CertificationID>67TR901-AZ</CertificationID>
807      </ProfileCertification>
808      <ProfileReference>
809        <!--Profile to be inserted as a reusable component-->
810        <Profile/>
811      </ProfileReference>
812      <Commissions>
813        <Commission CommissionStatusType="Full">
814          <CommissionableAmount>
815            <Currency>
816              <CurrencyCode>USD</CurrencyCode>
817              <Amount>185.00</Amount>
818            </Currency>
819          </CommissionableAmount>
820          <PrepaidAmount>
821            <Currency>
822              <CurrencyCode>USD</CurrencyCode>
823              <Amount>12.00</Amount>
824            </Currency>
825          </PrepaidAmount>
826          <CommissionPercent>0.0525</CommissionPercent>
827          <FlatCommission>not applicable<Currency>
828            <CurrencyCode>USD</CurrencyCode>
829            <Amount>00.00</Amount>
830          </Currency>
831          </FlatCommission>
832          <Comment>Default percentage commission agreement</Comment>
833          <CommissionReasonCode>7930</CommissionReasonCode>
834          <BillToID>HOTEL7890</BillToID>
835          <HotelReference>
836            <ChainCode>HI234</ChainCode>
837            <HotelCode>1234STL</HotelCode>
838          </HotelReference>
839        </Commission>
840        <Commission CommissionStatusType="Partial">
841          <CommissionableAmount>
842            <Currency>
843              <CurrencyCode>USD</CurrencyCode>
844              <Amount>185.00</Amount>
845            </Currency>
846          </CommissionableAmount>
847          <PrepaidAmount>
848            <Currency>
849              <CurrencyCode>USD</CurrencyCode>
850              <Amount>00.00</Amount>
851            </Currency>
852          </PrepaidAmount>
853          <Comment>This commission per agreement with Travel Agents,
854          Inc.</Comment>
855          <CommissionPercent>00.00</CommissionPercent>
856          <FlatCommission>
857            <Currency>
858              <CurrencyCode>USD</CurrencyCode>
859              <Amount>10.00</Amount>
```



```
860         </Currency>
861     </FlatCommission>
862     <CommissionReasonCode>7930</CommissionReasonCode>
863     <BillToID>HOTEL7890</BillToID>
864     <HotelReference>
865         <ChainCode>HI234</ChainCode>
866         <HotelCode>1234STL</HotelCode>
867     </HotelReference>
868     </Commission>
869 </Commissions>
870 </CommissionEvent>
871 <CommissionEvent>
872     <ConfirmationID/>
873     <ConfirmationOriginatorCode/>
874     <CommissionOriginatorCode>3457YTXV</CommissionOriginatorCode>
875     <ReservationID>09878783276XY</ReservationID>
876     <HotelReference>
877         <ChainCode>BASS123</ChainCode>
878         <HotelCode>1234STL</HotelCode>
879     </HotelReference>
880     <OriginalBookingDate>19991223T17:53:22</OriginalBookingDate>
881     <StayDateRange>
882         <StartInstant>20000122</StartInstant>
883         <Duration>00000003T000000</Duration>
884     </StayDateRange>
885     <GuestNames>
886         <NameInfo>
887             <NamePrefix>Mr.</NamePrefix>
888             <NameFirst>Kevin</NameFirst>
889             <NameMiddle>R.</NameMiddle>
890             <NameSur>Smithson</NameSur>
891             <NameSuffix>Jr.</NameSuffix>
892             <NameTitle>Professor</NameTitle>
893             <NameOrdered> Kevin Smithson</NameOrdered>
894         </NameInfo>
895         <NameInfo>
896             <NamePrefix>Miss</NamePrefix>
897             <NameFirst>Mary</NameFirst>
898             <NameMiddle>T.</NameMiddle>
899             <NameSur>Smithson</NameSur>
900             <NameSuffix>esq.</NameSuffix>
901             <NameTitle>Professor</NameTitle>
902             <NameOrdered> MarySmithson</NameOrdered>
903         </NameInfo>
904     </GuestNames>
905     <ProfileCertification CertificationType="ARC">
906         <CertificationID>67TR901-AZ</CertificationID>
907     </ProfileCertification>
908     <ProfileReference>
909         <Profile/>
910     </ProfileReference>
911 </Commissions>
912     <Commission CommissionStatusType="Full">
913         <CommissionableAmount>
914             <Currency>
915                 <CurrencyCode>USD</CurrencyCode>
916                 <Amount>185.00</Amount>
917             </Currency>
918         </CommissionableAmount>
919         <PrepaidAmount>
920             <Currency>
921                 <CurrencyCode>USD</CurrencyCode>
922                 <Amount>12.00</Amount>
923             </Currency>
924         </PrepaidAmount>
925         <CommissionPercent>0.0525</CommissionPercent>
926         <FlatCommission>not applicable<Currency>
927             <CurrencyCode>USD</CurrencyCode>
928             <Amount>00.00</Amount>
929         </Currency>
930     </FlatCommission>
931     <Comment>Default percentage commission agreement</Comment>
932     <CommissionReasonCode>7930</CommissionReasonCode>
933     <BillToID>HOTEL7890</BillToID>
934     <HotelReference>
```




```
935         <ChainCode>HI234</ChainCode>
936         <HotelCode>1234STL</HotelCode>
937     </HotelReference>
938 </Commission>
939 <Commission CommissionStatusType="Partial">
940     <CommissionableAmount>
941         <Currency>
942             <CurrencyCode>USD</CurrencyCode>
943             <Amount>185.00</Amount>
944         </Currency>
945     </CommissionableAmount>
946     <PrepaidAmount>
947         <Currency>
948             <CurrencyCode>USD</CurrencyCode>
949             <Amount>00.00</Amount>
950         </Currency>
951     </PrepaidAmount>
952     <Comment>Flat commission per agreement with TA</Comment>
953     <CommissionPercent>00.00</CommissionPercent>
954     <FlatCommission>
955         <Currency>
956             <CurrencyCode>USD</CurrencyCode>
957             <Amount>10.00</Amount>
958         </Currency>
959     </FlatCommission>
960     <CommissionReasonCode>7930</CommissionReasonCode>
961     <BillToID>HOTEL7890</BillToID>
962     <HotelReference>
963         <ChainCode>HI234</ChainCode>
964         <HotelCode>1234STL</HotelCode>
965     </HotelReference>
966 </Commission>
967 </Commissions>
968 </CommissionEvent>
969 </CommissionEvents>
970 </HITISOperation>
971 </Body>
972 </HITISMessage>
973
974 -----7d02a82e5f8--
```

B.2 Complete Example of an ebXML Message Envelope using multipart/related Content-Type sent via SMTP

The default Content-transfer-encoding type of 7BIT is being used in this message.

```
978
979 From dick@8760.com Sun May 7 17:01:14 2000
980 Received: from granger.mail.mindspring.net by alpha2000.tech-comm.com;
981 (8.8.5/1.1.8.2/05Jun95-1217PM)
982 id RAA32702; Sun, 7 May 2000 17:01:13 -0500 (CDT)
983 Received: from gamma (user-33qt101.dialup.mindspring.com [199.174.132.21])
984 by granger.mail.mindspring.net (8.9.3/8.8.5) with SMTP id SAA11942
985 for <ebxmlhandler@8760.com>; Sun, 7 May 2000 18:11:14 -0400 (EDT)
986 From: "Dick Brooks (E)" <dick@8760.com>
987 To: <ebxmlhandler@8760.com>
988 Subject: OTA Commission Event
989 Date: Sun, 7 May 2000 17:07:38 -0500
990 Message-ID: <NDBBIOBLMLCDOHCHIKMGKEEIDAAA.dick@8760.com>
991 MIME-Version: 1.0
992 X-Priority: 3 (Normal)
993 X-MSMail-Priority: Normal
994 X-Mailer: Microsoft Outlook IMO, Build 9.0.2416 (9.0.2910.0)
995 Importance: Normal
996 X-MimeOLE: Produced By Microsoft MimeOLE V5.00.2314.1300
997 Content-Length: 8081
998 Content-Type: multipart/related; type="application/vnd.eb+xml"; version="0.1";
999 charset="iso-8859-1"; boundary="-----_NextPart_000_0005_01BFB846.BF7FABA0"
1000
1001 -----_NextPart_000_0005_01BFB846.BF7FABA0
1002 Content-Type: application/vnd.eb+xml
```



```
1003 Content-ID: ebxmlheader-9000
1004 Content-Length: 272
1005
1006 <?xml version="1.0" encoding="UTF-8"?>
1007 <ebXMLMessageHeader xmlns='http://www.xml.org/ebXMLStds/ebXMLMessageHeaderv1'>
1008 <Version>1.0</Version>
1009 <MessageType>Request</MessageType>
1010 <ServiceType>Payroll</ServiceType>
1011 <Intent>RecordCommission</Intent>
1012 </ebXMLMessageHeader>
1013 -----_NextPart_000_0005_01BFB846.BF7FABA0
1014 Content-Type: text/xml
1015 Content-ID: ebxmlpayload-9000
1016 Content-Length: 7515
1017
1018 <?xml version="1.0" encoding="UTF-8"?>
1019 <!-- edited with XML Spy v2.5 - http://www.xmlspy.com -->
1020 <HITISMessage xmlns="" Version="1.0">
1021   <Header OriginalBodyRequested="false" ImmediateResponseRequired="true">
1022     <FromURI>http://www.pms.com/HITISInterface</FromURI>
1023     <ToURI>http://www.crs.com/HITISInterface</ToURI>
1024     <ReplyToURI>http://www.pms.com/HITISInterface</ReplyToURI>
1025     <MessageID>1234567890</MessageID>
1026     <OriginalMessageID>1234567890</OriginalMessageID>
1027     <TimeStamp>1999-11-10T10:23:44</TimeStamp>
1028     <Token>1234-567-8901</Token>
1029     <!--Token to be assigned in response to HITISRegister-->
1030   </Header>
1031   <Body>
1032     <HITISOperation OperationName="CommissionEventsUpdate">
1033       <CommissionEvents>
1034         <CommissionEvent>
1035           <ConfirmationID>18097YZ</ConfirmationID>
1036           <ConfirmationOriginatorCode>DBZ223</ConfirmationOriginatorCode>
1037           <CommissionOriginatorCode>3457YTXV</CommissionOriginatorCode>
1038           <ReservationID>098787818097YZ</ReservationID>
1039           <HotelReference>
1040             <ChainCode>HI234</ChainCode>
1041             <HotelCode>1234STL</HotelCode>
1042           </HotelReference>
1043           <OriginalBookingDate>19991223T17:53:22</OriginalBookingDate>
1044           <StayDateRange>
1045             <StartInstant>20000122</StartInstant>
1046             <Duration>00000003T000000</Duration>
1047           </StayDateRange>
1048           <GuestNames>
1049             <NameInfo>
1050               <NamePrefix>Mr.</NamePrefix>
1051               <NameFirst>John</NameFirst>
1052               <NameMiddle>Q.</NameMiddle>
1053               <NameSur>jones</NameSur>
1054               <NameSuffix>Jr.</NameSuffix>
1055               <NameTitle>Professor</NameTitle>
1056               <NameOrdered>JohnJones</NameOrdered>
1057             </NameInfo>
1058             <NameInfo>
1059               <NamePrefix>Mrs.</NamePrefix>
1060               <NameFirst>Sally</NameFirst>
1061               <NameMiddle>T.</NameMiddle>
1062               <NameSur>Jones</NameSur>
1063               <NameSuffix/>
1064               <NameTitle/>
1065               <NameOrdered>SallyJones</NameOrdered>
1066             </NameInfo>
1067           </GuestNames>
1068           <ProfileCertification CertificationType="ARC">
1069             <CertificationID>67TR901-AZ</CertificationID>
1070           </ProfileCertification>
1071           <ProfileReference>
1072             <!--Profile to be inserted as a reusable component-->
1073             <Profile/>
1074           </ProfileReference>
1075           <Commissions>
1076             <Commission CommissionStatusType="Full">
1077               <CommissionableAmount>
```



```
1078         <Currency>
1079             <CurrencyCode>USD</CurrencyCode>
1080             <Amount>185.00</Amount>
1081         </Currency>
1082     </CommissionableAmount>
1083     <PrepaidAmount>
1084         <Currency>
1085             <CurrencyCode>USD</CurrencyCode>
1086             <Amount>12.00</Amount>
1087         </Currency>
1088     </PrepaidAmount>
1089     <CommissionPercent>0.0525</CommissionPercent>
1090     <FlatCommission>not applicable<Currency>
1091         <CurrencyCode>USD</CurrencyCode>
1092         <Amount>00.00</Amount>
1093     </Currency>
1094 </FlatCommission>
1095     <Comment>Default percentage commission agreement</Comment>
1096     <CommissionReasonCode>7930</CommissionReasonCode>
1097     <BillToID>HOTEL7890</BillToID>
1098     <HotelReference>
1099         <ChainCode>HI234</ChainCode>
1100         <HotelCode>1234STL</HotelCode>
1101     </HotelReference>
1102 </Commission>
1103 <Commission CommissionStatusType="Partial">
1104     <CommissionableAmount>
1105         <Currency>
1106             <CurrencyCode>USD</CurrencyCode>
1107             <Amount>185.00</Amount>
1108         </Currency>
1109     </CommissionableAmount>
1110     <PrepaidAmount>
1111         <Currency>
1112             <CurrencyCode>USD</CurrencyCode>
1113             <Amount>00.00</Amount>
1114         </Currency>
1115     </PrepaidAmount>
1116     <Comment>This commission per agreement with Travel Agents,
1117 Inc.</Comment>
1118     <CommissionPercent>00.00</CommissionPercent>
1119     <FlatCommission>
1120         <Currency>
1121             <CurrencyCode>USD</CurrencyCode>
1122             <Amount>10.00</Amount>
1123         </Currency>
1124     </FlatCommission>
1125     <CommissionReasonCode>7930</CommissionReasonCode>
1126     <BillToID>HOTEL7890</BillToID>
1127     <HotelReference>
1128         <ChainCode>HI234</ChainCode>
1129         <HotelCode>1234STL</HotelCode>
1130     </HotelReference>
1131 </Commission>
1132 </Commissions>
1133 </CommissionEvent>
1134 <CommissionEvent>
1135     <ConfirmationID/>
1136     <ConfirmationOriginatorCode/>
1137     <CommissionOriginatorCode>3457YTXV</CommissionOriginatorCode>
1138     <ReservationID>09878783276XY</ReservationID>
1139     <HotelReference>
1140         <ChainCode>BASS123</ChainCode>
1141         <HotelCode>1234STL</HotelCode>
1142     </HotelReference>
1143     <OriginalBookingDate>19991223T17:53:22</OriginalBookingDate>
1144     <StayDateRange>
1145         <StartInstant>20000122</StartInstant>
1146         <Duration>00000003T000000</Duration>
1147     </StayDateRange>
1148     <GuestNames>
1149         <NameInfo>
1150             <NamePrefix>Mr.</NamePrefix>
1151             <NameFirst>Kevin</NameFirst>
1152             <NameMiddle>R.</NameMiddle>
```



```
1153      <NameSur>Smithson</NameSur>
1154      <NameSuffix>Jr.</NameSuffix>
1155      <NameTitle>Professor</NameTitle>
1156      <NameOrdered> Kevin Smithson</NameOrdered>
1157    </NameInfo>
1158    <NameInfo>
1159      <NamePrefix>Miss</NamePrefix>
1160      <NameFirst>Mary</NameFirst>
1161      <NameMiddle>T.</NameMiddle>
1162      <NameSur>Smithson</NameSur>
1163      <NameSuffix>esq.</NameSuffix>
1164      <NameTitle>Professor</NameTitle>
1165      <NameOrdered> MarySmithson</NameOrdered>
1166    </NameInfo>
1167  </GuestNames>
1168  <ProfileCertification CertificationType="ARC">
1169    <CertificationID>67TR901-AZ</CertificationID>
1170  </ProfileCertification>
1171  <ProfileReference>
1172    <Profile/>
1173  </ProfileReference>
1174  <Commissions>
1175    <Commission CommissionStatusType="Full">
1176      <CommissionableAmount>
1177        <Currency>
1178          <CurrencyCode>USD</CurrencyCode>
1179          <Amount>185.00</Amount>
1180        </Currency>
1181      </CommissionableAmount>
1182      <PrepaidAmount>
1183        <Currency>
1184          <CurrencyCode>USD</CurrencyCode>
1185          <Amount>12.00</Amount>
1186        </Currency>
1187      </PrepaidAmount>
1188      <CommissionPercent>0.0525</CommissionPercent>
1189      <FlatCommission>not applicable<Currency>
1190        <CurrencyCode>USD</CurrencyCode>
1191        <Amount>00.00</Amount>
1192      </Currency>
1193    </FlatCommission>
1194    <Comment>Default percentage commission agreement</Comment>
1195    <CommissionReasonCode>7930</CommissionReasonCode>
1196    <BillToID>HOTEL7890</BillToID>
1197    <HotelReference>
1198      <ChainCode>HI234</ChainCode>
1199      <HotelCode>1234STL</HotelCode>
1200    </HotelReference>
1201  </Commission>
1202  <Commission CommissionStatusType="Partial">
1203    <CommissionableAmount>
1204      <Currency>
1205        <CurrencyCode>USD</CurrencyCode>
1206        <Amount>185.00</Amount>
1207      </Currency>
1208    </CommissionableAmount>
1209    <PrepaidAmount>
1210      <Currency>
1211        <CurrencyCode>USD</CurrencyCode>
1212        <Amount>00.00</Amount>
1213      </Currency>
1214    </PrepaidAmount>
1215    <Comment>Flat commission per agreement with TA</Comment>
1216    <CommissionPercent>00.00</CommissionPercent>
1217    <FlatCommission>
1218      <Currency>
1219        <CurrencyCode>USD</CurrencyCode>
1220        <Amount>10.00</Amount>
1221      </Currency>
1222    </FlatCommission>
1223    <CommissionReasonCode>7930</CommissionReasonCode>
1224    <BillToID>HOTEL7890</BillToID>
1225    <HotelReference>
1226      <ChainCode>HI234</ChainCode>
1227      <HotelCode>1234STL</HotelCode>
```



```
1228         </HotelReference>
1229         </Commission>
1230     </Commissions>
1231     </CommissionEvent>
1232 </CommissionEvents>
1233 </HITISOperation>
1234 </Body>
1235 </HITISMessage>
1236 -----=_NextPart_000_0005_01BFB846.BF7FABA0--
```



Appendix C Candidate Packaging Technologies and Selection Process

The packaging sub-group began its investigation of packaging technologies by identifying the technologies currently used for business-to-business message exchange or were being developed for this purpose. The following packaging technologies were identified:

- MIME - currently in use by companies exchanging business transactions using E-mail and HTTP
- XML - currently used by RosettaNet and Microsoft (BizTalk and SOAP) and others

C.1 Selection Process

Each candidate technology was evaluated based on its ability to meet the requirements listed in the section titled "Packaging and other Requirements" in this document. When necessary, specific parties were contacted to provide details describing how a technology was being used to meet specific requirements. The following parties were contacted to provide expert insight:

- Microsoft - David Turner, regarding use of XML packaging in BizTalk
- Develop Mentor - Don Box, regarding use of XML packaging in SOAP
- Vitria - Prasad Yendluri, regarding use of XML packaging in RosettaNet
- Jonathan Borden - author of [XMTP], an XML to MIME transformation tool

The packaging sub-group considered the inputs of people from the ebXML Transport mailing list as well as the parties listed above, before making a selection.

C.2 MIME

Multipurpose Internet Mail Extensions (MIME) is an international standard created by the Internet Engineering Task Force. It has been implemented by numerous software vendors across the globe and has been used to exchange mixed type payloads, including XML, for several years. MIME was designed purely as a packaging (enveloping) solution to allow the transport of mixed payloads using Internet E-mail (SMTP). MIME is also being used by other transport technologies as a packaging technology, most notably HTTP.

C.3 XML

eXtensible Markup Language (XML) version 1.0 is a technical specification holding a RECOMMENDED status created by the World Wide Web Consortium. It has been implemented by numerous software vendors across the globe and has been used to describe a broad spectrum of document structures from very simple to very complex. XML is a very flexible markup language that can be used to represent virtually any type of document. XML can be used solely for packaging (enveloping) documents of any type, providing the data can be "transformed" into "legal" XML.

In some cases, XML documents MUST be placed into transport specific "envelopes" before being transported. For example, XML data MUST be placed in a MIME envelope when being transported via SMTP or HTTP.



C.4 Conclusion

The packaging sub-group examined the capabilities of both XML and MIME relative to the list of packaging requirements above. It's important to note that neither technology met all of the ebXML requirements and in the end it was the packaging sub-groups assessment of which technology came closest to meeting ALL of the ebXML requirements that determined which technology SHOULD be used.

MIME was chosen to serve as the ebXML packaging technology, over XML, based on the information contained in following table:

Reason	Requirement(s) Satisfied
There is no formal packaging recommendation within IETF or W3C, based on XML. If ebXML were to choose XML as a packaging technology it would be required to define an XML packaging specification and submit this to IETF or W3C for adoption as a formal standard.	to not reinvent the wheel - re-use where possible [TRPREQ]
XML requires that binary and other types of payload data including XML documents be base64 encoded in order to be encapsulated within a XML root document. Base64 encoding ensures that no illegal XML characters exist within a document and recursive XML documents are "hidden". Base64 encoding imposes a significant processing overhead and results in larger messages, which affect both transmission and processing times. Base64 encoding of binary data is required of MIME content when being transported by SMTP, but this is a transport level requirement, not a requirement imposed by MIME. Binary data can be packaged and transported without alteration when using MIME over HTTP	Minimize intrusion to payload (special encoding or alteration) Low processing overhead
At the time of defining this specification there is no industry standard way to package an encrypted message, or portion of a message, using XML.	All or part of the documents in a message MAY be encrypted prior to sending [TRPREQ]
MIME could be used in conformance within existing IETF recommendations, no additions or changes are initially required to produce a functional envelope.	to not reinvent the wheel - re-use where possible [TRPREQ]



Appendix D MIME Type discussion

Three MIME media types were considered to serve as Content-Type for the *ebXML Message Envelope*:

- Multipart/related
- Multipart/Mixed
- Multipart/form-data

The group selected the multipart/related media type to serve as the preferred message envelope Content-Type.

Note:

There was some discussion over the similarities of multipart/related and multipart/mixed, both of which appear to offer similar capabilities and both could meet stated requirements. However, the group converged on multipart/related, believing it to be more semantically appropriate for ebXML.

There was significant discussion over whether to support multipart/form-data as an alternate Content-Type for message-envelope, due to the large installed base of web browsers that support this Content-Type.

It was determined that multipart/related was a more generic Content-Type than multipart/form-data and the multipart/related Content-Type is the preferred Content-Type for ebXML Message Envelopes. Multipart/form-data Content-Type is typically associated with HTTP/HTML web forms, whereas multipart/related can be associated with any type of data.

Additionally, due to limitations in their handling of multipart ebXML payloads it was determined that existing web browsers are unable to support the full breadth of functions needed to package complex *ebXML Messages* containing multipart payloads. Therefore browser vendors are encouraged to add support for the ebXML enveloping standard as specified in this document.

Appendix E Communication Protocol Envelope Mappings

This section provides rules and definitions for the completion of the Communication Protocol Envelope Mappings for HTTP [RFC xxxx], SMTP [RFC xxxx] and FTP [RFC xxxx].

E.1 HTTP

To be completed

E.2 SMTP

To be completed

E.3 FTP

To be completed



1315 **Appendix D Non-Normative References**

- 1316 [XMTP] XMTP - Extensible Mail Transport Protocol
1317 <http://www.openhealth.org/documents/xmtp.htm>