



Creating A Single Global Electronic Market

1 ebXML Transport, Routing & Packaging 2 Messaging Service Specification

3 Working Draft 13-September-2000

4 This version:

5 ebXML Messaging Service Specification v0-21.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 Introduction5
43 1.1 Purpose and Scope5
44 1.1.1 Goals5
45 1.2 Related ebXML Specifications5
46 1.3 Specification Structure6
47 1.4 General Conventions6
48 2 Packaging Specification6
49 2.1 ebXML Message Structure6
50 2.1.1 MIME usage Conventions7
51 2.2 ebXML Message Envelope7
52 2.2.1 Content-Type8
53 2.2.1.1 type Attribute8
54 2.2.1.2 boundary Attribute8
55 2.2.1.3 version Attribute8
56 2.2.1.4 charset Attribute8
57 2.2.2 Content-Length8
58 2.2.3 ebXML Message Envelope Example9
59 2.3 ebXML Header Container9
60 2.3.1 Content-ID9
61 2.3.2 Content-Length9
62 2.3.3 Content-Type9
63 2.3.3.1 version Attribute10
64 2.3.3.2 charset Attribute10
65 2.3.4 ebXML Header Container Example10
66 2.4 ebXML Payload Container10
67 2.4.1 Content-ID11
68 2.4.2 Content-Length11
69 2.4.3 Content-Type11
70 2.4.4 Example of an ebXML Payload Container11
71 3 ebXML Header Document12
72 3.1 Root Element12
73 3.2 Manifest13
74 3.2.1 DocumentReference13
75 3.3 Header13
76 3.3.1 From and To14
77 3.3.2 TPAInfo14
78 3.3.3 MessageData14
79 3.3.4 ReliableMessagingInfo15
80 4 Normative References15
81 5 Acknowledgments16
82 6 Authors' Address16
83 Appendix A Schemas and DTD Definitions18
84 A.1 XML Header DTD18
85 A.2 XML Header Schema Definition19
86 Appendix B Examples22



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



102 1 Introduction

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

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

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

110 1.1 Purpose and Scope

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

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

116 1.1.1 Goals

117 The goals of this specification are to:

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

125 1.2 Related ebXML Specifications

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

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



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

139 **1.3 Specification Structure**

140 This specification is organized around the following main topics:

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

147 Appendices to the specification cover:

- 148 • Appendix A Schemas and DTD Definitions
- 149 • Appendix B Examples
- 150 • Appendix C Candidate Packaging Technologies and Selection Process
- 151 • Appendix D MIME Type discussion
- 152 • Appendix E Communication Protocol Envelope Mappings
- 153 • Appendix D Non-Normative References

154 **1.4 General Conventions**

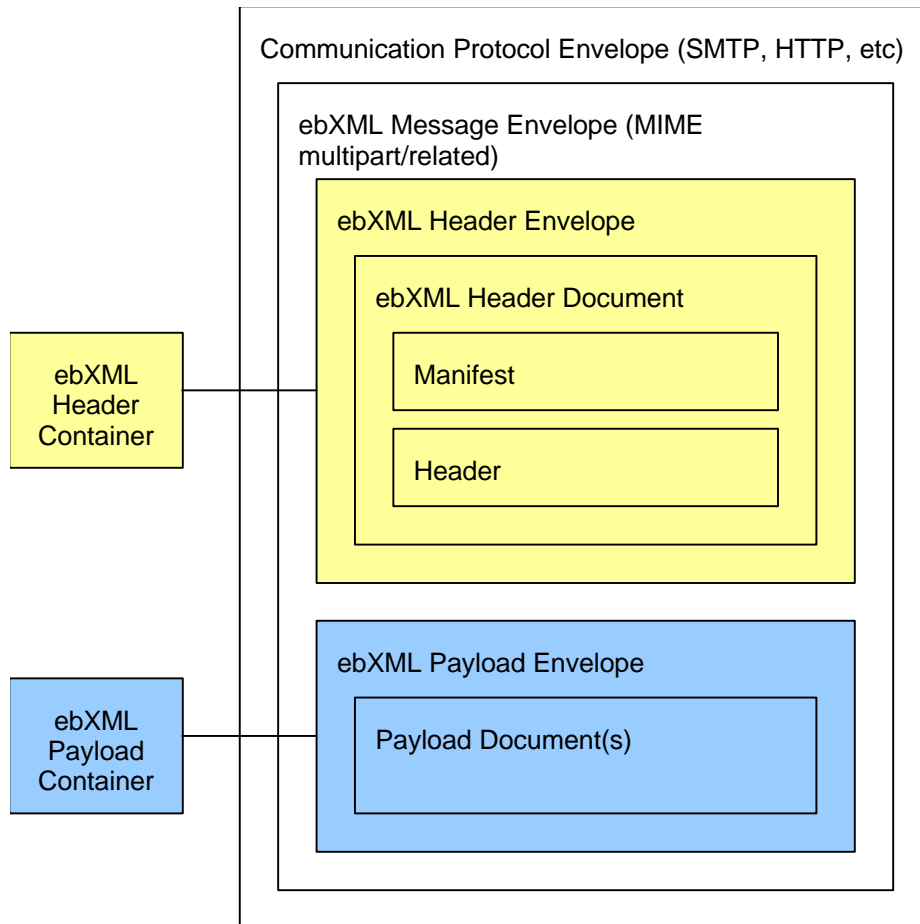
155 For all messages following the ebXML standard, a single message structure is defined,
156 regardless of message type

157 **2 Packaging Specification**

158 **2.1 ebXML Message Structure**

159 An *ebXML Message* consists of:

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



166

167 **Figure 2-1 ebXML Message Structure**

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

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

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

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

174 **2.1.1 MIME usage Conventions**

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

177 **2.2 ebXML Message Envelope**

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

- 181 • `Content-Type`



- 182
- Content-Length

183 **2.2.1 Content-Type**

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

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

187 The Content-Type header contains four attributes:

- 188 • type
- 189 • boundary
- 190 • version, and
- 191 • charset.

192 **2.2.1.1 type Attribute**

193 The type attribute is used to identify the *ebXML Message Envelope* as an ebXML compliant
194 structure. It conforms to an XML Media Type [XMLMedia] and MUST be set to
195 "application/vnd.eb+xml". For example:

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

197 **2.2.1.2 boundary Attribute**

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

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

203 **2.2.1.3 version Attribute**

204 The version attribute is used to identify the particular version of *ebXML Message Envelope* being
205 used. All message headers SHOULD USE "0.21". For example:

```
206 version="0.21"
```

207 **2.2.1.4 charset Attribute**

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

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

212 **2.2.2 Content-Length**

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

```
215 Content-Length: 9841
```




216 2.2.3 ebXML Message Envelope Example

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

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

221 2.3 ebXML Header Container

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

- 223 • one *ebXML Header Envelope*, and
- 224 • one *ebXML Header Document*

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

226 The following rules apply:

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

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

- 232 • Content-ID
- 233 • Content-Length
- 234 • Content-Type

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

239 2.3.1 Content-ID

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

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

244 2.3.2 Content-Length

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

```
247 Content-Length: 4208
```

248 2.3.3 Content-Type

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



- 251 • version, and
- 252 • charset

253 2.3.3.1 version Attribute

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

```
256 version="1.0";
```

257 2.3.3.2 charset Attribute

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

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

263 For example:

```
264 charset="UTF-8"
```

265 2.3.4 ebXML Header Container Example

266 The following represents an example of an *ebXML Header Envelope* and *ebXML Header*
267 *Document*:

268 Content-ID: ebxmlheader-123 -----		
269 Content-Length: 2048	ebXML Header Envelope	
270 Content-Type: application/vnd.eb+xml -----		ebXML
271		Header
272 <ebXMLHeader> -----		Container
273 <Manifest>.....	ebXML Header Document	
274 </Manifest>		
275 <Header>.....		
276 </Header>		
277 </ebXMLHeader> -----		

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

279 2.4 ebXML Payload Container

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

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

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

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

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

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



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

- 293 • Content-ID
- 294 • Content-Length
- 295 • Content-Type

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

299 2.4.1 Content-ID

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

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

304 2.4.2 Content-Length

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

```
307 Content-Length: 5012
```

308 2.4.3 Content-Type

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

```
312 Content-Type: application/xml
```

313 2.4.4 Example of an ebXML Payload Container

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

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

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



324 3 ebXML Header Document

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

327 In general, separate principal header elements are used where:

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

333 3.1 Root Element

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

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

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

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

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

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

348 The **ebXMLHeader** element MUST contain the following two elements:

- 349 • **Manifest** - contains a list of references to the other parts of the Message. This includes
350 references to the documents, which comprise the *Payload* of the *Message*.
- 351 • **Header** - contains the information REQUIRED by the recipient to process the message. The
352 message originator creates this information to which additional information MAY be added.

353 The **Header** and **Manifest** are REQUIRED elements in every *Message*.

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

```
356 <?xml version="1.0"?>  
357 <ebXMLHeader xmlns="http://www.ebxml.org/namespaces/messageHeader"  
358   Version="1.0" MessageType="Normal">  
359   <Manifest>...</Manifest>  
360   <Header>...</Header>  
361 </ebXMLHeader>
```



362 3.2 Manifest

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

370 3.2.1 DocumentReference

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

- 373 • **DocumentDescription** - an optional textual description of the document/resource
- 374 • **DocumentLabel** - a code that enables the purpose of the referenced document to be
375 determined without retrieving it
- 376 • **DocumentId** - a URL of the Content-ID of a MIME body part, as defined in [RFC2392],
377 representing payload data, or a remote URL to some external resource.

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

```
380 <Manifest>  
381   <DocumentReference>  
382     <DocumentLabel>PurchaseOrder</DocumentLabel>  
383     <DocumentId>cid:0987654321</DocumentId>  
384   </DocumentReference>  
385 </Manifest>
```

386 3.3 Header

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

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

397 The following fragment demonstrates the structure of the **Header** element of the **ebXMLHeader**
398 document:

```
399 <Header>  
400   <From>...</From>  
401   <To>...</To>  
402   <TPAInfo>...</TPAInfo>  
403   <MessageData>...</MessageData>  
404   <ReliableMessagingInfo>...</ReliableMessagingInfo>  
405 </Header>
```



406 3.3.1 From and To

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

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

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

```
415 <From>  
416   <PartyId context="DUNS">12345</PartyId>  
417 </From>  
418 <To>  
419   <PartyId context="DUNS">54321</PartyId>  
420 </To>
```

421 3.3.2 TPAInfo

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

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

435 The following example fragment demonstrates the usage of the **TPAInfo** element.

```
436 <TPAInfo>  
437   <TPAId context = "tpadb">12345678</TPAId>  
438   <ConversationId context = "tpadb">987654321</ConversationId>  
439   <ServiceInterface>QuoteToCollect</ServiceInterface>  
440   <Action>NewPurchaseOrder</Action>  
441 </TPAInfo>
```

442 3.3.3 MessageData

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

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



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

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

```
455 <MessageData>  
456 <MessageId>UUID-2</MessageId>  
457 <TimeStamp>20000725T121905.000Z</TimeStamp>  
458 <RefToMessageId>UUID-1</RefToMessageId>  
459 </MessageData>
```

460 3.3.4 ReliableMessagingInfo

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

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

```
468 <ReliableMessagingInfo>  
469 <DeliverySemantics>AtMostOnce</DeliverySemantics>  
470 </ReliableMessagingInfo>
```

471 4 Normative References

- 472 [Glossary] ebXML Glossary, see ebXML Project Team Home Page
- 473 [ISO 8601] International Standards Organization Ref. ISO 8601 Second Edition, Published 1997
- 474 [RFC 2392] IETF Request For Comments 2392. Content-ID and Message-ID Uniform Resource
475 Locators. E. Levinson, Published August 1998
- 476 [RFC2045] IETF RFC 2045. Multipurpose Internet Mail Extensions (MIME) Part One: Format of
477 Internet Message Bodies, N Freed & N Borenstein, Published November 1996
- 478 [TRPREQ] ebXML Transport, Routing and Packaging: Overview and Requirements, Version
479 0.96, Published 25 May 2000
- 480 [UTF-8] UTF-8 is an encoding that conforms to ISO/IEC 10646. See [XML] for usage
481 conventions.
- 482 [XML Namespace] Recommendation for Namespaces in XML, World Wide Web Consortium, 14
483 January 1999, <http://www.w3.org/TR/REC-xml-names>
- 484 [XMLMedia] IETF Internet Draft on XML Media Types. See <http://www.imc.org/draft-murata-xml>
485 Note. It is anticipated that this Internet Draft will soon become a RFC. Final versions
486 of this specification will refer to the equivalent RFC.
- 487 [XML] Extensible Mark Up Language. A W3C recommendation. See
488 <http://www.w3.org/TR/1998/REC-xml-19980210> for the 10 February 1998 version.



489 **5 Acknowledgments**

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

493 Ralph Berwanger – bTrade.com
494 Jonathan Borden - Author of XMTP
495 Jon Bosak - Sun
496 Doug Bunting - Ariba
497 David Burdett - Commerce One
498 Philippe De Smedt - Viquity
499 Lawrence Ding - WorldSpan
500 Rik Drummond - Drummond Group
501 Christopher Ferris - Sun
502 Ian Jones - BT
503 Henry Lowe - OMG
504 Jim McCarthy - webXI
505 Bob Miller - GEIS
506 Dale Moberg - Sterling Commerce
507 Kathy Spector – Extricity
508 Nikola Stojanovic
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



549 **Appendix A Schemas and DTD Definitions**

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

551 **A.1 XML Header DTD**

```
552 <?xml version = "1.0"?>
553 <schema xmlns = "http://www.w3.org/1999/XMLSchema">
554 <!ELEMENT ebXMLHeader (Manifest , Header )>
555 <!ATTLIST ebXMLHeader Version CDATA #FIXED '1.0'
556 MessageTypes CDATA #FIXED 'Normal' >
557 <!ELEMENT Manifest (DocumentReference )+>
558 <!ELEMENT DocumentReference (DocumentDescription?, DocumentLabel , DocumentId )>
559 <!ELEMENT DocumentDescription (#PCDATA )>
560 <!ATTLIST DocumentDescription e-dtype NMTOKEN #FIXED 'string' >
561 <!ELEMENT DocumentLabel (#PCDATA )>
562 <!ATTLIST DocumentLabel e-dtype NMTOKEN #FIXED 'string' >
563 <!ELEMENT DocumentId (#PCDATA )>
564 <!ATTLIST DocumentId e-dtype NMTOKEN #FIXED 'uri' >
565 <!ELEMENT Header (From , To , TPAInfo , MessageData , ReliableMessagingInfo )>
566 <!ELEMENT TPAInfo (TPAId , ConversationId , ServiceInterface , Action )>
567 <!ELEMENT ServiceInterface (#PCDATA )>
568 <!ATTLIST ServiceInterface e-dtype NMTOKEN #FIXED 'string' >
569 <!ELEMENT Action (#PCDATA )>
570 <!ATTLIST Action e-dtype NMTOKEN #FIXED 'string' >
571 <!ELEMENT TPAId (#PCDATA )>
572 <!ATTLIST TPAId context CDATA 'Undefined'
573 e-dtype NMTOKEN #FIXED 'uri' >
574 <!ELEMENT ConversationId (#PCDATA )>
575 <!ATTLIST ConversationId context CDATA 'Undefined'
576 e-dtype NMTOKEN #FIXED 'uri' >
577 <!ELEMENT MessageData (MessageId , TimeStamp , RefToMessageId )>
578 <!ELEMENT RefToMessageId (#PCDATA )>
579 <!ATTLIST RefToMessageId e-dtype NMTOKEN #FIXED 'uuid' >
580 <!ELEMENT TimeStamp (#PCDATA )>
581 <!ATTLIST TimeStamp e-dtype NMTOKEN #FIXED 'dateTime' >
582 <!ELEMENT MessageId (#PCDATA )>
583 <!ATTLIST MessageId e-dtype NMTOKEN #FIXED 'uuid' >
584 <!ELEMENT From (PartyId )>
585 <!ELEMENT To (PartyId )>
586 <!ELEMENT PartyId (#PCDATA )>
587 <!ATTLIST PartyId context CDATA 'Undefined'
588 e-dtype NMTOKEN #FIXED 'uri' >
589 <!ELEMENT ReliableMessagingInfo EMPTY>
590 <!ATTLIST ReliableMessagingInfo DeliverySemantics (AtMostOnce | Unspecified ) #FIXED
591 '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 = "TPAInfo"/>
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>
```



722 Appendix B Examples

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

725 B.1 Complete Example of an ebXML Message Envelope using 726 multipart/related Content-Type sent via HTTP POST

```
727 POST /ebxmlhandler HTTP/1.1
728 Accept: multipart/related
729 Accept-Language: en-us
730 Accept-Encoding: gzip, deflate
731 User-Agent: Group 8760 InsideAgent
732 Host: localhost:9090
733 Connection: Keep-Alive
734 Content-Type: multipart/related; type=application/vnd.eb+xml; version=0.1;
735 boundary=-----7d02a82e5f8
736 Content-Length: 9293
737
738 -----7d02a82e5f8
739 Content-ID: ebxmlheader-9981
740 Content-Length: 211
741 Content-Type: application/vnd.eb+xml; charset="UTF-8";
742
743 <?xml version="1.0" encoding="UTF-8"?>
744 <ebXMLHeader xmlns='http://www.xml.org/ebXMLStds/ebXMLMessageHeaderv1'>
745   <Version>1.0</Version>
746   <MessageType>Request</MessageType>
747   <ServiceType>Payroll</ServiceType>
748   <Intent>RecordCommission</Intent>
749 </ebXMLHeader>
750 -----7d02a82e5f8
751 Content-ID: ebxmlpayload-9981
752 Content-Length: 7517
753 Content-Type: text/xml
754
755 <?xml version="1.0" encoding="UTF-8"?>
756 <!-- edited with XML Spy v2.5 - http://www.xmlspy.com -->
757 <HITISMessage xmlns="" Version="1.0">
758   <Header OriginalBodyRequested="false" ImmediateResponseRequired="true">
759     <FromURI>http://www.pms.com/HITISInterface</FromURI>
760     <ToURI>http://www.crs.com/HITISInterface</ToURI>
761     <ReplyToURI>http://www.pms.com/HITISInterface</ReplyToURI>
762     <MessageID>1234567890</MessageID>
763     <OriginalMessageID>1234567890</OriginalMessageID>
764     <TimeStamp>1999-11-10T10:23:44</TimeStamp>
765     <Token>1234-567-8901</Token>
766     <!--Token to be assigned in response to HITISRegister-->
767   </Header>
768   <Body>
769     <HITISOperation OperationName="CommissionEventsUpdate">
770       <CommissionEvents>
771         <CommissionEvent>
772           <ConfirmationID>18097YZ</ConfirmationID>
773           <ConfirmationOriginatorCode>DBZ223</ConfirmationOriginatorCode>
774           <CommissionOriginatorCode>3457YTXV</CommissionOriginatorCode>
775           <ReservationID>098787818097YZ</ReservationID>
776           <HotelReference>
777             <ChainCode>HI234</ChainCode>
778             <HotelCode>1234STL</HotelCode>
779           </HotelReference>
780           <OriginalBookingDate>19991223T17:53:22</OriginalBookingDate>
781           <StayDateRange>
782             <StartInstant>20000122</StartInstant>
783             <Duration>00000003T000000</Duration>
784           </StayDateRange>

```



785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859

```
<GuestNames>
  <NameInfo>
    <NamePrefix>Mr.</NamePrefix>
    <NameFirst>John</NameFirst>
    <NameMiddle>Q.</NameMiddle>
    <NameSur>jones</NameSur>
    <NameSuffix>Jr.</NameSuffix>
    <NameTitle>Professor</NameTitle>
    <NameOrdered>JohnJones</NameOrdered>
  </NameInfo>
  <NameInfo>
    <NamePrefix>Mrs.</NamePrefix>
    <NameFirst>Sally</NameFirst>
    <NameMiddle>T.</NameMiddle>
    <NameSur>Jones</NameSur>
    <NameSuffix/>
    <NameTitle/>
    <NameOrdered>SallyJones</NameOrdered>
  </NameInfo>
</GuestNames>
<ProfileCertification CertificationType="ARC">
  <CertificationID>67TR901-AZ</CertificationID>
</ProfileCertification>
<ProfileReference>
  <!--Profile to be inserted as a reusable component-->
  <Profile/>
</ProfileReference>
<Commissions>
  <Commission CommissionStatusType="Full">
    <CommissionableAmount>
      <Currency>
        <CurrencyCode>USD</CurrencyCode>
        <Amount>185.00</Amount>
      </Currency>
    </CommissionableAmount>
    <PrepaidAmount>
      <Currency>
        <CurrencyCode>USD</CurrencyCode>
        <Amount>12.00</Amount>
      </Currency>
    </PrepaidAmount>
    <CommissionPercent>0.0525</CommissionPercent>
    <FlatCommission>not applicable<Currency>
      <CurrencyCode>USD</CurrencyCode>
      <Amount>00.00</Amount>
    </Currency>
    </FlatCommission>
    <Comment>Default percentage commission agreement</Comment>
    <CommissionReasonCode>7930</CommissionReasonCode>
    <BillToID>HOTEL7890</BillToID>
    <HotelReference>
      <ChainCode>HI234</ChainCode>
      <HotelCode>1234STL</HotelCode>
    </HotelReference>
  </Commission>
  <Commission CommissionStatusType="Partial">
    <CommissionableAmount>
      <Currency>
        <CurrencyCode>USD</CurrencyCode>
        <Amount>185.00</Amount>
      </Currency>
    </CommissionableAmount>
    <PrepaidAmount>
      <Currency>
        <CurrencyCode>USD</CurrencyCode>
        <Amount>00.00</Amount>
      </Currency>
    </PrepaidAmount>
    <Comment>This commission per agreement with Travel Agents,
Inc.</Comment>
    <CommissionPercent>00.00</CommissionPercent>
    <FlatCommission>
      <Currency>
        <CurrencyCode>USD</CurrencyCode>
        <Amount>10.00</Amount>
      </Currency>
    </FlatCommission>
  </Commission>
</Commissions>
```



```
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
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974

```

    <ChainCode>HI234</ChainCode>
    <HotelCode>1234STL</HotelCode>
  </HotelReference>
</Commission>
<Commission CommissionStatusType="Partial">
  <CommissionableAmount>
    <Currency>
      <CurrencyCode>USD</CurrencyCode>
      <Amount>185.00</Amount>
    </Currency>
  </CommissionableAmount>
  <PrepaidAmount>
    <Currency>
      <CurrencyCode>USD</CurrencyCode>
      <Amount>00.00</Amount>
    </Currency>
  </PrepaidAmount>
  <Comment>Flat commission per agreement with TA</Comment>
  <CommissionPercent>00.00</CommissionPercent>
  <FlatCommission>
    <Currency>
      <CurrencyCode>USD</CurrencyCode>
      <Amount>10.00</Amount>
    </Currency>
  </FlatCommission>
  <CommissionReasonCode>7930</CommissionReasonCode>
  <BillToID>HOTEL7890</BillToID>
  <HotelReference>
    <ChainCode>HI234</ChainCode>
    <HotelCode>1234STL</HotelCode>
  </HotelReference>
</Commission>
</Commissions>
</CommissionEvent>
</CommissionEvents>
</HITISOperation>
</Body>
</HITISMessage>
-----7d02a82e5f8--

```

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

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

978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002

```

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

-----_NextPart_000_0005_01BFB846.BF7FABA0
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 <ebXMLHeader 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 </ebXMLHeader>
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
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151

```
<CommissionableAmount>
  <Currency>
    <CurrencyCode>USD</CurrencyCode>
    <Amount>185.00</Amount>
  </Currency>
</CommissionableAmount>
<PrepaidAmount>
  <Currency>
    <CurrencyCode>USD</CurrencyCode>
    <Amount>12.00</Amount>
  </Currency>
</PrepaidAmount>
<CommissionPercent>0.0525</CommissionPercent>
<FlatCommission>not applicable<Currency>
  <CurrencyCode>USD</CurrencyCode>
  <Amount>00.00</Amount>
</Currency>
</FlatCommission>
<Comment>Default percentage commission agreement</Comment>
<CommissionReasonCode>7930</CommissionReasonCode>
<BillToID>HOTEL7890</BillToID>
<HotelReference>
  <ChainCode>HI234</ChainCode>
  <HotelCode>1234STL</HotelCode>
</HotelReference>
</Commission>
<Commission CommissionStatusType="Partial">
  <CommissionableAmount>
    <Currency>
      <CurrencyCode>USD</CurrencyCode>
      <Amount>185.00</Amount>
    </Currency>
  </CommissionableAmount>
  <PrepaidAmount>
    <Currency>
      <CurrencyCode>USD</CurrencyCode>
      <Amount>00.00</Amount>
    </Currency>
  </PrepaidAmount>
  <Comment>This commission per agreement with Travel Agents,
Inc.</Comment>
  <CommissionPercent>00.00</CommissionPercent>
  <FlatCommission>
    <Currency>
      <CurrencyCode>USD</CurrencyCode>
      <Amount>10.00</Amount>
    </Currency>
  </FlatCommission>
  <CommissionReasonCode>7930</CommissionReasonCode>
  <BillToID>HOTEL7890</BillToID>
  <HotelReference>
    <ChainCode>HI234</ChainCode>
    <HotelCode>1234STL</HotelCode>
  </HotelReference>
</Commission>
</Commissions>
</CommissionEvent>
<CommissionEvent>
  <ConfirmationID/>
  <ConfirmationOriginatorCode/>
  <CommissionOriginatorCode>3457YTXV</CommissionOriginatorCode>
  <ReservationID>09878783276XY</ReservationID>
  <HotelReference>
    <ChainCode>BASS123</ChainCode>
    <HotelCode>1234STL</HotelCode>
  </HotelReference>
  <OriginalBookingDate>19991223T17:53:22</OriginalBookingDate>
  <StayDateRange>
    <StartInstant>20000122</StartInstant>
    <Duration>0000003T000000</Duration>
  </StayDateRange>
  <GuestNames>
    <NameInfo>
      <NamePrefix>Mr.</NamePrefix>
      <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
1228
1229
1230
1231
1232
1233
1234
1235
1236

```
<HotelCode>1234STL</HotelCode>  
  </HotelReference>  
    </Commission>  
      </Commissions>  
        </CommissionEvent>  
          </CommissionEvents>  
            </HITISOperation>  
          </Body>  
        </HITISMessage>  
-----_NextPart_000_0005_01BFB846.BF7FABA0--
```



1237 **Appendix C Candidate Packaging Technologies and** 1238 **Selection Process**

1239 The packaging sub-group began its investigation of packaging technologies by identifying the
1240 technologies currently used for business-to-business message exchange or were being
1241 developed for this purpose. The following packaging technologies were identified:

- 1242 • MIME - currently in use by companies exchanging business transactions using E-mail and
1243 HTTP
- 1244 • XML - currently used by RosettaNet and Microsoft (BizTalk and SOAP) and others

1245 **C.1 Selection Process**

1246 Each candidate technology was evaluated based on its ability to meet the requirements listed in
1247 the section titled "Packaging and other Requirements" in this document. When necessary,
1248 specific parties were contacted to provide details describing how a technology was being used to
1249 meet specific requirements. The following parties were contacted to provide expert insight:

- 1250 • Microsoft - David Turner, regarding use of XML packaging in BizTalk
- 1251 • Develop Mentor - Don Box, regarding use of XML packaging in SOAP
- 1252 • Vitria - Prasad Yendluri, regarding use of XML packaging in RosettaNet
- 1253 • Jonathan Borden - author of [XMTP], an XML to MIME transformation tool

1254 The packaging sub-group considered the inputs of people from the ebXML Transport mailing list
1255 as well as the parties listed above, before making a selection.

1256 **C.2 MIME**

1257 Multipurpose Internet Mail Extensions (MIME) is an international standard created by the Internet
1258 Engineering Task Force. It has been implemented by numerous software vendors across the
1259 globe and has been used to exchange mixed type payloads, including XML, for several years.
1260 MIME was designed purely as a packaging (enveloping) solution to allow the transport of mixed
1261 payloads using Internet E-mail (SMTP). MIME is also being used by other transport technologies
1262 as a packaging technology, most notably HTTP.

1263 **C.3 XML**

1264 eXtensible Markup Language (XML) version 1.0 is a technical specification holding a
1265 RECOMMENDED status created by the World Wide Web Consortium. It has been implemented
1266 by numerous software vendors across the globe and has been used to describe a broad
1267 spectrum of document structures from very simple to very complex. XML is a very flexible
1268 markup language that can be used to represent virtually any type of document. XML can be used
1269 solely for packaging (enveloping) documents of any type, providing the data can be
1270 "transformed" into "legal" XML.

1271 In some cases, XML documents MUST be placed into transport specific "envelopes" before
1272 being transported. For example, XML data MUST be placed in a MIME envelope when being
1273 transported via SMTP or HTTP.



1274

C.4 Conclusion

1275

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.

1276

1277

1278

1279

1280

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

1281

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]



1282 **Appendix D MIME Type discussion**

1283 Three MIME media types were considered to serve as Content-Type for the *ebXML Message*
1284 *Envelope*:

- 1285 • Multipart/related
- 1286 • Multipart/Mixed
- 1287 • Multipart/form-data

1288 **The group selected the multipart/related media type to serve as the preferred message**
1289 **envelope Content-Type.**

1290 *Note:*

1291 *There was some discussion over the similarities of multipart/related and multipart/mixed, both of*
1292 *which appear to offer similar capabilities and both could meet stated requirements. However, the*
1293 *group converged on multipart/related, believing it to be more semantically appropriate for ebXML.*

1294 *There was significant discussion over whether to support multipart/form-data as an alternate*
1295 *Content-Type for message-envelope, due to the large installed base of web browsers that*
1296 *support this Content-Type.*

1297 *It was determined that multipart/related was a more generic Content-Type than multipart/form-*
1298 *data and the multipart/related Content-Type is the preferred Content-Type for ebXML Message*
1299 *Envelopes. Multipart/form-data Content-Type is typically associated with HTTP/HTML web forms,*
1300 *whereas multipart/related can be associated with any type of data.*

1301 Additionally, due to limitations in their handling of multipart ebXML payloads it was determined
1302 that existing web browsers are unable to support the full breadth of functions needed to package
1303 complex *ebXML Messages* containing multipart payloads. Therefore browser vendors are
1304 encouraged to add support for the ebXML enveloping standard as specified in this document.

1305 **Appendix E Communication Protocol Envelope** 1306 **Mappings**

1307 This section provides rules and definitions for the completion of the Communication Protocol
1308 Envelope Mappings for HTTP [RFC xxxx], SMTP [RFC xxxx] and FTP [RFC xxxx].

1309 **E.1 HTTP**

1310 To be completed

1311 **E.2 SMTP**

1312 To be completed

1313 **E.3 FTP**

1314 To be completed



1315

Appendix D Non-Normative References

1316

[XMTP]

XMTP - Extensible Mail Transport Protocol

1317

<http://www.openhealth.org/documents/xmtp.htm>