OASIS ebXML Messaging Services

Working Draft 08, 18 January 2006

Document identifier:

ebms_core-3.0_draft-08

Location:

http://www.oasis-open.org/committees/documents.php?wg_abbrev=ebxml-msg

Editor:

Matthew MacKenzie, Adobe Systems Incorporated mattm@adobe.com>
Jeff Turpin, Cyclone Commerce <jturpin@cyclonecommerce.com>
Pete Wenzel, Sun Microsystems <pete.wenzel@sun.com>

Contributors:

Doug Bunting, Sun Microsystems <doug.bunting@sun.com>
Jacques Durand, Fujitsu Software <jdurand@us.fujitsu.com>
Ric Emery, Cyclone Commerce <remery@cyclonecommerce.com>
Kazunori Iwasa, Fujitsu Limited <kiwasa@jp.fujitsu.com>
Hamid Ben Malek, Fujitsu Software <hmalek@us.fujitsu.com>
Dale Moberg, Cyclone Commerce <dmoberg@cyclonecommerce.com>
Sacha Schlegel, Cyclone Commerce <sschlegel@cyclonecommerce.com>

Abstract:

This specification focuses on defining a communications-protocol neutral method for exchanging electronic business messages. It defines specific enveloping constructs supporting reliable, secure delivery of business information. Furthermore, the specification defines a flexible enveloping technique, permitting messages to contain payloads of any format type. This versatility ensures legacy electronic business systems employing traditional syntaxes (i.e. UN/EDIFACT, ASC X12, or HL7) can leverage the advantages of the ebXML infrastructure along with users of emerging technologies.

Status:

This draft reflects the TC's consensus on the general feature set expressed herein; however, the technical details are subject to change.

This document was last revised or approved by the TC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the ebxmlmsg@lists.oasis-open.org list. Others should use the comment form at http://www.oasisopen.org/committees/comments/form.php?wg_abbrev=ebxml-msg.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer

to the Intellectual Property Rights section of the OASIS ebXML Messaging Services TC web page (http://www.oasis-open.org/committees/ebxml-msg/ipr.php).

Appendix A. SOAP Format and Bindings

This appendix specifies the SOAP format (SOAP versions, packaging of attachments and/or binary data) used in ebMS-3, as well as how this SOAP format is transported over HTTP and SMTP.

ebMS-3 does not require the usage of SOAP-1.1 and/or SwA (SOAP-1.1 With Attachments). We consider the attachments specification of SwA as being orthogonal to the SOAP version. In other words, attachments could well be used for SOAP 1.2 in the same way they are used for SOAP 1.1. Similarly, we also consider MTOM being orthogonal to the SOAP version (however, MTOM will not be addressed in this core specification).

A conformant implementation of ebMS-3 may well choose to use SOAP-1.2 instead of SOAP-1.1. Since SwA is orthogonal to the SOAP version, there are two possibilities:

- An implementation of ebMS-3 may choose SOAP-1.1 with Attachments
- An implementation of ebMS-3 may choose SOAP-1.2 with Attachments

By allowing the use of SOAP 1.2 with Attachments, it may seem that we are playing here a role that W3C should have done, but this is not the case. We are simply being conformant to standards (See the note below, a quotation from David Chappell):

Note: What are "standards" exactly?

> When talking about the use and adoption of standards, it is hard to tell just exactly what the word means. We live in a world of multiple overlapping efforts from different standards bodies to define standard specifications. Vendor alliances are producing web services specifications outside the domain of any standards body or consortium.
> […]There are also "de facto standards" such as open source implementations, which either invented their own ways of doing things or conform to industry-accepted "standard" specifications. The Apache SOAP and Apache Axis toolkits are examples of such standards.
> […] In short, the word "standard," in terms of standards-based integration, refers to either a specification or an implementation that has gained enough traction in the industry to have long-lasting staying power, and is open enough for multiple vendors to implement or repackage it.

We consider the word "standards" in a wide sense as outlined above by Chappell's quotation. From the implementation point of view, all libraries that support SOAP 1.2 are able to add attachments to a SOAP message in the same way it is done for SOAP 1.1. In other words, adding attachments to SOAP 1.2 in the same way it is done to SOAP 1.1 is really a standard since all implementations (without exception) supporting SOAP 1.2 are doing it and have been doing it for a long time without waiting a W3C recommendation on attachments for SOAP 1.2.

SwA uses the multipart/related MIME encapsulation. This encapsulation is independent of the version of SOAP being used (in fact it can encapsulate any XML document, not just SOAP), and also independent of the transport protocol (the encapsulation could be transported via HTTP, SMTP, etc…).

1.1 Using SwA with SOAP-1.1

The following example shows an ebMS-3 message using SOAP 1.1 with attachment. The ebMS-3 message in this example contains two payloads:

1. The first payload is the picture of a car. This picture is in binary form as an attachment with a Content-ID equal to “car-photo”.

2. The second payload is an XML fragment within the SOAP body. This XML fragment has id attribute equal to “carData”

The XML fragment in the SOAP body contains a reference to another binary data, namely the picture of the car owner):

```
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
        start="<car-data@toyoya.com>"

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <car-data@toyoya.com>

<?xml version='1.0' ?>
<S11:Envelope xmlns:S11="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-3_0.xsd">
  <S11:Header>
     <eb:Messaging eb:version="3.0" S11:mustUnderstand="1">
         …
           <eb:PayloadInfo>
              <eb:PartInfo href="cid:car-photo" />
              <eb:PartInfo href="#carData" />
           </eb:PayloadInfo>
     </eb:Messaging>
  </S11:Header>

  <S11:Body>
     <t:Data id="carData" xmlns:t="http://toyota.com">
        <t:Mileage>20000</t:Mileage>
        <t:OwnerPicture href="cid:picture-of-owner"/>
     </t:Data>
  </S11:Body>
</S11:Envelope>

--MIME_boundary
Content-Type: image/tiff
Content-Transfer-Encoding: binary
Content-ID: <car-photo>

...binary TIFF image of the car...

--MIME_boundary–
Content-Type: image/tiff
Content-Transfer-Encoding: binary
Content-ID: <picture-of-owner>

...binary TIFF image of the car’s owner...
--MIME_boundary–
```

Example 1: SOAP-1.1 with Attachment

1.2 Using SwA with SOAP-1.2

The following (Example 2) shows the same message given in example 1, except that SOAP-1.2 is being used instead of SOAP-1.1:

```
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
              start="<car-data@toyoya.com>"


--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <car-data@toyoya.com>

<?xml version='1.0' ?>
<S12:Envelope xmlns:S12="http://www.w3.org/2003/05/soap-envelope"
    xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-3_0.xsd">
  <S12:Header>
    <eb:Messaging eb:version="3.0" S12:mustUnderstand="true">
        …
          <eb:PayloadInfo>
             <eb:PartInfo href="cid:car-photo" />
             <eb:PartInfo href="#carData" />
          </eb:PayloadInfo>
    </eb:Messaging>
  </S12:Header>

  <S12:Body>
    <t:Data id="carData" xmlns:t="http://toyota.com">
        <t:Mileage>20000</t:Mileage>
        <t:OwnerPicture href="cid:picture-of-owner"/>
    </t:Data>
  </S12:Body>
</S12:Envelope>

--MIME_boundary
Content-Type: image/tiff
Content-Transfer-Encoding: binary
Content-ID: <car-photo>

...binary TIFF image of the car...

--MIME_boundary–
Content-Type: image/tiff
Content-Transfer-Encoding: binary
Content-ID: <picture-of-owner>

...binary TIFF image of the car's owner...
--MIME_boundary–
```

Example 2: SOAP-1.2 with Attachments

What were the differences between Example 1 and Example 2 (SOAP 1.1/SOAP 1.2 with attachments)? The differences are the following:

- In SOAP 1.1, the namespace of the SOAP elements (Envelope, Header, and Body) is http://schemas.xmlsoap.org/soap/envelope/ versus the namespace http://www.w3.org/2003/05/soap-envelope for SOAP 1.2
- In SOAP 1.1, the attribute mustUnderstand takes 0 or 1 as values, whereas in SOAP 1.2, the values for the attribute mustUnderstand are true and false.

That's it **[We are not explaining here the differences in syntax between SOAP 1.1 and SOAP 1.2. See section 6 of the recommendation http://www.w3.org/TR/soap12-part0/ for more details on such differences. We are simply explaining here the differences between SOAP 1.1 and SOAP 1.2 in terms of Mime packaging headers only and in particular how the sample listing given in Example 1 is migrated to become a sample listing for SOAP 1.2 with attachments as given by Example 2]**. Another difference between SOAP 1.1 and SOAP 1.2 would be in the SOAPAction header. When using HTTP as the transport protocol, there will be an HTTP header called SOAPAction if SOAP 1.1 is being used. If SOAP 1.2 is used, instead of the SOAPAction header there will be an action parameter, as illustrated in the following listings:

```
SOAPAction: leasing
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
              start="<car-data@toyoya.com>"
```

HTTP headers when using SOAP 1.1 with attachments

```
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
              start="<car-data@toyoya.com>"; action=leasing
```

HTTP headers when using SOAP 1.2 with attachments

When using SMTP transport, the only additional requirement is that the Mime-Version header must be present (among other SMTP related headers such as To, From, Date, etc…). The following listings show the headers for both SOAP 1.1 and SOAP 1.2 over SMTP:

```
From: hamid@us.fujitsu.com
To: leasing-office@toyota.com
Date: Mon, 23 Jan 2006 17:33:00 CST
Mime-Version: 1.0
SOAPAction: leasing
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
              start="<car-data@toyoya.com>"
```

SMTP headers when using SOAP 1.1 with attachments

```
From: hamid@us.fujitsu.com
To: leasing-office@toyota.com
Date: Mon, 23 Jan 2006 17:33:00 CST
Mime-Version: 1.0
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
              start="<car-data@toyoya.com>"; action=leasing
```

SMTP headers when using SOAP 1.2 with attachments