

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

|                       |                |                         |           |                         |
|-----------------------|----------------|-------------------------|-----------|-------------------------|
| <b>IssueID</b>        | 1              | <b>Request Date</b>     | 7/16/2001 |                         |
| <b>Request Source</b> | Face to Face   | <b>Source Reference</b> |           |                         |
| <b>Change Type</b>    | MinorTechnical | <b>Disposition</b>      |           | <b>Disposition Date</b> |
| <b>Requested By</b>   | Iwasa          | <b>Email</b>            |           |                         |

### ***Source Reference:***

**Issue** Define how encryption works

Current Message Service specification does not define how to apply encryption function  
The CPP/CPA specification enforces to use SSL as transport level's security function with HTTP, but it doesn't define neither SSL version nor algorithm.  
This might causes security problem, because SSL security level will be the lowest level when the negotiation was failed Proposal  
For HTTP binding, define detailed conditions of SSL such as:  
SSL 3.0 or TLS 1.0 ?  
Certification Requirement  
Both sender side and receiver side? Or only receiver side?  
What algorithms are used for key, encryption and hash?  
For SMTP binding, some other definition would be required  
\*Should these definitions be done by CPP/CPA?

### ***Notes***

|                     |              |            |                |
|---------------------|--------------|------------|----------------|
| <b>Line Number:</b> | <b>Start</b> | <b>End</b> | <b>Section</b> |
|                     | 0            | 0          | TBD            |

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

***IssueID*** 2 ***Request Date*** 7/16/2001

***Request Source*** Face to Face ***Source Reference***

***Change Type*** MinorTechnical ***Disposition***

***Disposition Date***

***Requested By*** Iwasa ***Email***

### ***Source Reference:***

***Issue*** Support multiple digital signatures

In the current Message Service specification, a Message Service Handler is required to support only one digital signature  
However real business systems require multiple digital signatures  
Independent signatures for header (SOAP Env.) and payload  
Independent signatures for each payload in multiple payloads  
Proposal  
Spec should describe that multiple digital signatures should be supported

### ***Notes***

***Line Number: Start*** ***End*** ***Section***  
0 0 TBD

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

|                              |                |                                |           |                                |
|------------------------------|----------------|--------------------------------|-----------|--------------------------------|
| <b><i>IssueID</i></b>        | 3              | <b><i>Request Date</i></b>     | 7/16/2001 |                                |
| <b><i>Request Source</i></b> | Face to Face   | <b><i>Source Reference</i></b> |           |                                |
| <b><i>Change Type</i></b>    | MinorTechnical | <b><i>Disposition</i></b>      |           | <b><i>Disposition Date</i></b> |
| <b><i>Requested By</i></b>   | Iwasa          | <b><i>Email</i></b>            |           |                                |

### ***Source Reference:***

***Issue*** Define ebXML Messaging Profiles for compliance purposes

Current Message Service specification defines many functions (e.g. Security, Synchronous/Asynchronous, FTP ...)  
But the specification does not define what function must be supported to conform with the specification  
So, it is difficult to decide what function should be implemented  
This cause interoperability problems between different vendor implementations  
Proposal  
Define what functions must be supported at least in the implementation to conform with the specificatio

### ***Notes***

|                                  |                   |                       |
|----------------------------------|-------------------|-----------------------|
| <b><i>Line Number: Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
| 0                                | 0                 | TBD                   |

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

|                              |                |                                |           |                                |
|------------------------------|----------------|--------------------------------|-----------|--------------------------------|
| <b><i>IssueID</i></b>        | 4              | <b><i>Request Date</i></b>     | 7/16/2001 |                                |
| <b><i>Request Source</i></b> | Face to Face   | <b><i>Source Reference</i></b> |           |                                |
| <b><i>Change Type</i></b>    | MinorTechnical | <b><i>Disposition</i></b>      |           | <b><i>Disposition Date</i></b> |
| <b><i>Requested By</i></b>   | Iwasa          | <b><i>Email</i></b>            |           |                                |

### ***Source Reference:***

***Issue*** Make it easier to use ebXML Messaging without CPAs

Current Message Service depends on CPP/CPA to obtain communication parameters  
However it is complex process to generate a CPA from two CPPs. The process needs access to the Registry, comparison of CPP, etc.  
Enforcement of the CPA generation process might prevent spreading of the ebXML Message Service in the market

Proposal  
Permits Message Service to work without the CPA generation process. For example,  
Define default communication parameters in Message Service specification or CPP/CPA specification, or  
Add simple negotiation protocol to Messaging Service specification

### ***Notes***

|                            |                     |                   |                       |
|----------------------------|---------------------|-------------------|-----------------------|
| <b><i>Line Number:</i></b> | <b><i>Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
|                            | 0                   | 0                 | TBD                   |

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

***IssueID*** 5 ***Request Date*** 7/16/2001

***Request Source*** Face to Face ***Source Reference***

***Change Type*** MajorTechnical ***Disposition***

***Disposition Date***

***Requested By*** Iwasa ***Email***

### ***Source Reference:***

***Issue*** Support for Large Messages using HTTP compress function

Current Message Service specification lacks consideration for large messages

ebXML application can compress its payload before message transfer, but this causes interoperability problem

Proposal

HTTP has compress function (Content-encoding header). So define how to use the HTTP's compress function

Some other consideration is required for SMTP

### ***Notes***

***Line Number: Start*** ***End*** ***Section***

0 0 TBD

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

|                              |                |                                |           |                                |
|------------------------------|----------------|--------------------------------|-----------|--------------------------------|
| <b><i>IssueID</i></b>        | 6              | <b><i>Request Date</i></b>     | 7/16/2001 |                                |
| <b><i>Request Source</i></b> | Face to Face   | <b><i>Source Reference</i></b> |           |                                |
| <b><i>Change Type</i></b>    | MinorTechnical | <b><i>Disposition</i></b>      |           | <b><i>Disposition Date</i></b> |
| <b><i>Requested By</i></b>   | Iwasa          | <b><i>Email</i></b>            |           |                                |

### ***Source Reference:***

***Issue*** Clearer explanation of multi-hop

Current Message Service specification does not describe clearly about message sequence of multi-hop  
Proposal  
Define multi-hop sequence  
Keep application level's multi-hop control as is, because :  
MSH level's multi-hop control complexes the specification  
It is not clear the whether MSH level's multi-hop control is really needed or not in the market  
Candidate for the multi-hop sequence is :  
Permit Multi-hop only when SyncReplyRequest is set to False  
Use Store and Forward sequence for Muiiti-ho

### ***Notes***

|                                  |                   |                       |
|----------------------------------|-------------------|-----------------------|
| <b><i>Line Number: Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
| 0                                | 0                 | TBD                   |

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

***IssueID*** 7 ***Request Date*** 7/16/2001

***Request Source*** Face to Face ***Source Reference***

***Change Type*** MajorTechnical ***Disposition***

***Disposition Date***

***Requested By*** Iwasa ***Email***

### ***Source Reference:***

***Issue*** Service Interface Specification Required

There is no MSH's standard API specification

Current Message Service specification does not define MSH's API

JCP's JAXM (Java API for XML Messaging) supports only SOAP level functions

This causes portability problem of ebXML application

It is difficult to develop standard conformance check program without MSH's standard API specification

Proposal

Define MSH's standard API

***Notes*** Suggest defer as not 1.1 issue

***Line Number: Start*** ***End*** ***Section***

0 0 TBD

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

|                              |                |                                |           |                                |
|------------------------------|----------------|--------------------------------|-----------|--------------------------------|
| <b><i>IssueID</i></b>        | 8              | <b><i>Request Date</i></b>     | 7/16/2001 |                                |
| <b><i>Request Source</i></b> | Face to Face   | <b><i>Source Reference</i></b> |           |                                |
| <b><i>Change Type</i></b>    | MajorTechnical | <b><i>Disposition</i></b>      |           | <b><i>Disposition Date</i></b> |
| <b><i>Requested By</i></b>   | Iwasa          | <b><i>Email</i></b>            |           |                                |

### ***Source Reference:***

***Issue*** Provide MultiCast support

Multicast is useful function to distribute same message to many destinations  
Standard MOM specifications (ex. JMS, OMG Notification) has the multicast function (publish/subscribe)  
But current Message Service lacks this function  
Solution  
Define MSH API with multicast message function

### ***Notes***

|                            |                     |                   |                       |
|----------------------------|---------------------|-------------------|-----------------------|
| <b><i>Line Number:</i></b> | <b><i>Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
|                            | 0                   | 0                 | TBD                   |



## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

|                              |                |                                |           |                                |
|------------------------------|----------------|--------------------------------|-----------|--------------------------------|
| <b><i>IssueID</i></b>        | 9              | <b><i>Request Date</i></b>     | 7/16/2001 |                                |
| <b><i>Request Source</i></b> | Face to Face   | <b><i>Source Reference</i></b> |           |                                |
| <b><i>Change Type</i></b>    | MinorTechnical | <b><i>Disposition</i></b>      |           | <b><i>Disposition Date</i></b> |
| <b><i>Requested By</i></b>   | Iwasa          | <b><i>Email</i></b>            |           |                                |

### ***Source Reference:***

***Issue*** Ack message sequence is ambiguous when SyncReply is used

Proposal

To add following description to make it clear :

When SyncReply is set to True, Message Service level's Ack Message is carried by HTTP response message

When SyncReply is set to False, Message Service level's Ack Message is carried by independent HTTP request message (POST method)

### ***Notes***

|                                  |                   |                       |
|----------------------------------|-------------------|-----------------------|
| <b><i>Line Number: Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
| 0                                | 0                 | TBD                   |

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

|                              |                |                                |           |                                |
|------------------------------|----------------|--------------------------------|-----------|--------------------------------|
| <b><i>IssueID</i></b>        | 10             | <b><i>Request Date</i></b>     | 7/16/2001 |                                |
| <b><i>Request Source</i></b> | Face to Face   | <b><i>Source Reference</i></b> |           |                                |
| <b><i>Change Type</i></b>    | MinorTechnical | <b><i>Disposition</i></b>      |           | <b><i>Disposition Date</i></b> |
| <b><i>Requested By</i></b>   | Iwasa          | <b><i>Email</i></b>            |           |                                |

### ***Source Reference:***

***Issue*** Synchronous Messaging with Best Effort

In the specification, it is unclear that whether following semantics is possible or not:  
deliverySemantics is set to BestEffort and  
SyncReply is set to True  
This semantics is same as SOAP RPC (synchronous and non-reliable)  
Proposal  
Add description of the semantics above clearly if the semantics is allowed

### ***Notes***

|                                  |                   |                       |
|----------------------------------|-------------------|-----------------------|
| <b><i>Line Number: Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
| 0                                | 0                 | TBD                   |

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

|                              |                |                                |           |                                |
|------------------------------|----------------|--------------------------------|-----------|--------------------------------|
| <b><i>IssueID</i></b>        | 11             | <b><i>Request Date</i></b>     | 7/16/2001 |                                |
| <b><i>Request Source</i></b> | Face to Face   | <b><i>Source Reference</i></b> |           |                                |
| <b><i>Change Type</i></b>    | MinorTechnical | <b><i>Disposition</i></b>      |           | <b><i>Disposition Date</i></b> |
| <b><i>Requested By</i></b>   | Iwasa          | <b><i>Email</i></b>            |           |                                |

### ***Source Reference:***

***Issue*** MessageId Format

Receiving MSH uses MessageId to detect duplication of received messages

The MessageId format must conform to RFC2392, however this RFC does not define detailed format (Even MessageId length is not decided)

So it makes difficult to develop efficient duplication check mechanism in the receiving MSH

Proposal

Define detailed format of MessageId

### ***Notes***

|                                  |                   |                       |
|----------------------------------|-------------------|-----------------------|
| <b><i>Line Number: Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
| 0                                | 0                 | TBD                   |

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

|                       |                |                         |                             |
|-----------------------|----------------|-------------------------|-----------------------------|
| <b>IssueID</b>        | 12             | <b>Request Date</b>     | 7/25/2001                   |
| <b>Request Source</b> | e-mail         | <b>Source Reference</b> |                             |
| <b>Change Type</b>    | Editorial      | <b>Disposition</b>      | <b>Disposition Date</b>     |
| <b>Requested By</b>   | Sanjay Cherian | <b>Email</b>            | Sanjay_Chерian@stercomm.com |

### ***Source Reference:***

**Issue** Clarify use of UTC

These are the places in the specification where <TimeStamp/> appears:

8.4.6.2, 8.5.2.3, 8.6.1, 8.13.2 and 8.14.1.

One of the descriptions has been included below for reference:

|            |   |
|------------|---|
| [Line 801] | 8.4.6.2 Timestamp element   |
| [Line 802] | The REQUIRED Timestamp is a value representing the time that the message header was created |
| [Line 803] | conforming to an [XMLSchema] timeInstant.   |

This statement does not mandate that UTC be used when date and time is specified with <Timestamp/>. On the other hand, there is no facility in the CPP to define the time zone that one party operates within and no facility in the CPA to capture the timezone that two parties must agree upon.

The use of UTC is suggested through all the examples in the specification that use <Timestamp/>. For example:

[Line 953]      <eb:Timestamp>2000-07-25T12:19:05Z</eb:Timestamp>

I understand that the Z at the end indicates that the time specified here is relative to UTC (the same as GMT).

In my opinion, it is a small matter to require the use of UTC in the ebMS specification

### ***Notes***

|                     |              |            |                |
|---------------------|--------------|------------|----------------|
| <b>Line Number:</b> | <b>Start</b> | <b>End</b> | <b>Section</b> |
|                     | 0            | 0          | 8.13.2         |
|                     | 0            | 0          | 8.5.2.3        |

**Friday, August 24, 2001**

**Page 12 of 38**

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

0 0 8.14.1  
0 0 8.4.6.2

***IssueID*** 13 ***Request Date*** 7/30/2001

***Request Source*** e-mail ***Source Reference***

***Change Type*** MinorTechnical ***Disposition*** ***Disposition Date***

***Requested By*** Arvola Chan ***Email***

***Source Reference:***

***Issue*** Support Full XML Schema Recommendation

***Notes***

***Line Number: Start*** ***End*** ***Section***

0 0 TBD

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

|                       |             |                         |           |                         |
|-----------------------|-------------|-------------------------|-----------|-------------------------|
| <b>IssueID</b>        | 14          | <b>Request Date</b>     | 7/30/2001 |                         |
| <b>Request Source</b> | e-mail      | <b>Source Reference</b> |           |                         |
| <b>Change Type</b>    | Editorial   | <b>Disposition</b>      |           | <b>Disposition Date</b> |
| <b>Requested By</b>   | Arvola Chan | <b>Email</b>            |           |                         |

### ***Source Reference:***

**Issue** Clarification of use of Message Ack

Arvola -

Section 8.7 does not clearly indicate the circumstances under which the ackRequested attribute should be set (to Signed or Unsigned). Is this governed by the ReliableMessaging and NonRepudiation element for the DocExchange associated with the DeliveryChannel that is being used? In particular, when an error is encountered in processing a message, what should be the strategy for setting the ackRequested attribute in the error message? In other words, under what circumstances, if any, are error messages to be sent reliably? Thanks,-Arvola

David Burdett:

Setting the ackRequested to Signed or Unsigned is a decision that the designer (and/or implementer) of the business process makes when they design or build a business process collaboration or business process transaction. Factors that need to be considered include (IMO):

The nature of the business process/transaction - e.g. payments probably need to be secure

The requirements of the individual trading partners

I think what would be really useful is to have a guide that describes how to design a business process/transaction using ebXML Messaging. Do you agree? If so should it be in the 1.1 spec or something separate. I think that if an error is discovered then including the ackRequested set to true on the error message runs the risk of a never ending series of messages. The only use cases to consider are where a message is being sent reliably in which case ...

If the message that was in error has ackRequested set to Signed/Unsigned and the error message sent in return is lost, then the sender of the original message will resend it which will cause the error message to be resent - see example 1 below

If the message that was in error has ackRequested set to None (e.g. it is a synchronous response) then sending the error message with Ack Requested set to Signed/Unsigned makes sense otherwise the sender of the error message will not know if the message was delivered - see example 2 below

EXAMPLE 1 Message (with error)(AckRequested=S/U)-----><----- ErrorMessage (Includes Acknowledgment element) EXAMPLE 2 Message (no errors)(AckRequested=S/U)-----><----- Message (with error) (Includes Acknowledgment element) Error Message (AckRequested=S/U)-----><----- Message (Includes Acknowledgment element only) A general rule (it's somewhere in the spec but I can't immediately find where) says that if you find an error in an error message then you don't respond with another error message and sort out the problem by some other means.

Michael Wang

Take your example further by taking 'deliverySemantics=OnceAndOnlyOnce'

**Friday, August 24, 2001**

**Page 14 of 38**

## ***ebXML Messaging Specification Change/Issues Log***

**Document Messaging Service Spec 1.0 5/11/2001**

for both examples, then should the Initiator finally respond with a plain acknowledgement message to the Responder?

### **EXAMPLE 1**

Message (with error)(AckRequested=S/U)----->  
<-----ErrorMessage (Includes Acknowledgment element)  
Acknowledgement message (without Acknowledgement element) ----->

### **EXAMPLE 2**

Message (no errors)(AckRequested=S/U)----->  
<-----Message (with error) (Includes Acknowledgment element)  
Error Message (AckRequested=S/U)----->  
<-----Message (Includes Acknowledgment element only)  
Acknowledgement message (without Acknowledgement element) ----->

According to the spec (section 10.3.3 line 1811-1813) that acknowledgement message **MUST** be generated when 'deliverySemantics=OnceAndOnlyOnce'. If this is the case then as you have also pointed out that one runs the risk of a never ending series of messages (or unnecessary ack messages).

Now, if the final round of ack messages are not required then I think it would help by clarifying what kind of messages need to be explicitly acked when 'deliverySemantics=OnceAndOnlyOnce'. e.g. only Request and Response messages require acks and errors, signals and acks themselves do not require to be acknowledged.

### ***Notes***

David ...





## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**Line Number:** Start      End      Section  
0                      0 8.7

**IssueID**            15    **Request Date**            7/30/2001

**Request Source** e-mail                      **Source Reference**

**Change Type**    MinorTechnical    **Disposition**                      **Disposition Date**

**Requested By**    David Fischer                      **Email** david@drummondgroup.com

You make a good point. I think that the last ack (without the acknowledgment elements) should NOT be sent as they do not add anything.

**Source Reference:**      However I agree what you say about lines 1811-13. The spec needs clarification in this area.

**Issue**      Default value for SequenceNumber should be continue

David Fischer

Status attribute on SequenceNumber should have a default of "Continue" rather than require a value. BTW, I don't see this attribute in the schema (AppendixA).

David Burdett

I agree with you. This is a bug - no definition of Status and it should be in the Schema

### ***Notes***

**Line Number:** Start      End      Section  
0                      0 TBD

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

|                       |                |                         |                         |
|-----------------------|----------------|-------------------------|-------------------------|
| <b>IssueID</b>        | 16             | <b>Request Date</b>     | 7/30/2001               |
| <b>Request Source</b> | e-mail         | <b>Source Reference</b> |                         |
| <b>Change Type</b>    | MajorTechnical | <b>Disposition</b>      | <b>Disposition Date</b> |
| <b>Requested By</b>   | David Fischer  | <b>Email</b>            | david@drummondgroup.com |

### ***Source Reference:***

**Issue** Encapsulation of ebXML Messages inside another ebXML Message

David Fischer:

It seems that there will be no way to avoid occasional recursion within ebXML-MS such as encrypting a message with SMIME and putting the result in a bodypart wrapped by a minimal ebXML header structure. At the receiving end, the bodypart would then be decrypted and the resulting message resubmitted to the ebXML parser (recursive).

This behavior would also solve the concern over potential in-route additions to the Manifest (put the old message in a bodypart, create a new set of headers with additions to the Manifest, then once the new message is parsed then the old message should be resubmitted to the parser (recursive)).

While none of this behavior is prohibited by the current spec, neither is it specifically allowed. Can we add one sentence somewhere to specifically allow this?

David Burdett:

I think the process (encapsulation) is a useful one. However I think that this should be a separate spec that describes how to do it - it could be an OASIS TC spec. There are several other useful examples, for example using sequencing to transport very large messages by chopping them into chunks. They are both additional good ideas that can be layered on top of what is already there.

### ***Notes***

|                     |              |            |                |
|---------------------|--------------|------------|----------------|
| <b>Line Number:</b> | <b>Start</b> | <b>End</b> | <b>Section</b> |
|                     | 0            | 0          | TBD            |

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

|                              |             |                                |           |                                |
|------------------------------|-------------|--------------------------------|-----------|--------------------------------|
| <b><i>IssueID</i></b>        | 17          | <b><i>Request Date</i></b>     | 7/26/2001 |                                |
| <b><i>Request Source</i></b> | e-mail      | <b><i>Source Reference</i></b> |           |                                |
| <b><i>Change Type</i></b>    | Editorial   | <b><i>Disposition</i></b>      |           | <b><i>Disposition Date</i></b> |
| <b><i>Requested By</i></b>   | Arvola Chan | <b><i>Email</i></b>            |           |                                |

### ***Source Reference:***

***Issue*** Minor error in transforms for digital signatures

Lines 2027-2029 indicate that there are two mandatory transforms, one of which is <http://www.w3.org/2000/09/xmldsig#enveloped-signature>. This transform is missing from the example message on lines 2045-2092.

### ***Notes***

|                            |                     |                   |                       |
|----------------------------|---------------------|-------------------|-----------------------|
| <b><i>Line Number:</i></b> | <b><i>Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
|                            | 2027                | 2029              |                       |
|                            | 2045                | 2092              |                       |

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

|                       |             |                         |           |                         |
|-----------------------|-------------|-------------------------|-----------|-------------------------|
| <b>IssueID</b>        | 18          | <b>Request Date</b>     | 7/26/2001 |                         |
| <b>Request Source</b> | e-mail      | <b>Source Reference</b> |           |                         |
| <b>Change Type</b>    | Editorial   | <b>Disposition</b>      |           | <b>Disposition Date</b> |
| <b>Requested By</b>   | Arvola Chan | <b>Email</b>            |           |                         |

### ***Source Reference:***

**Issue** Status Reuest & Status Response errors

Since the StatusRequest element resides in the SOAP Body rather than the SOAP header, the statement on lines 1580-1581 is technically incorrect. It should say "...elements in the SOAP Header and Body."

The StatusResponse element described in section 8.13 needs to be updated. The schema in Appendix A shows:  
<attribute name="messageStatus" type="tns:messageStatus.type"/>

```
<simpleType name="messageStatus.type"> <restriction base="NMTOKEN">
<enumeration value="Unauthorized"/>
<enumeration value="NotRecognized"/>
<enumeration value="Received"/>
<enumeration value="Processed"/>
<enumeration value="Forwarded"/>
</restriction>
</simpleType>
```

The values Processed and Forwarded are missing from section 8.13.3 messageStatus attribute.

### ***Notes***

|                     |              |            |                |
|---------------------|--------------|------------|----------------|
| <b>Line Number:</b> | <b>Start</b> | <b>End</b> | <b>Section</b> |
|                     | 1580         | 1581       |                |
|                     | 0            | 0          | 8.13.3         |
|                     | 0            | 0          | 8.13           |

**Friday, August 24, 2001**

**Page 19 of 38**

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

|                              |             |                                |           |                                |
|------------------------------|-------------|--------------------------------|-----------|--------------------------------|
| <b><i>IssueID</i></b>        | 19          | <b><i>Request Date</i></b>     | 7/30/2001 |                                |
| <b><i>Request Source</i></b> | e-mail      | <b><i>Source Reference</i></b> |           |                                |
| <b><i>Change Type</i></b>    | Editorial   | <b><i>Disposition</i></b>      |           | <b><i>Disposition Date</i></b> |
| <b><i>Requested By</i></b>   | Arvola Chan | <b><i>Email</i></b>            |           |                                |

### ***Source Reference:***

***Issue*** Schema error - via includes deliveryReceiptRequested when it should not.

Line 2581 in the ebXML SOAP Extension Elements Schema shows

```
<attribute name="deliveryReceiptRequested" type="tns:signedUnsigned.type" use="default" value="None"/>
```

under the Via element. This is inconsistent with the description of the Via element in Section 8.7 which does not include the deliveryReceiptRequested attribute.

Section 8.4.7.1 shows that the deliveryReceiptRequested attribute is part of the QualityOfServiceInfo element.

### ***Notes***

|                            |                     |                   |                       |
|----------------------------|---------------------|-------------------|-----------------------|
| <b><i>Line Number:</i></b> | <b><i>Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
|                            | 0                   | 0                 | 8.7                   |
|                            | 2581                | 2581              | 8.7                   |
|                            | 2581                | 2581              |                       |
|                            | 0                   | 0                 | 8.4.7.1               |

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

|                       |                |                         |           |                         |
|-----------------------|----------------|-------------------------|-----------|-------------------------|
| <b>IssueID</b>        | 20             | <b>Request Date</b>     | 7/15/2001 |                         |
| <b>Request Source</b> | e-mail         | <b>Source Reference</b> |           |                         |
| <b>Change Type</b>    | MinorTechnical | <b>Disposition</b>      |           | <b>Disposition Date</b> |
| <b>Requested By</b>   | Arvola Chan    | <b>Email</b>            |           |                         |

### ***Source Reference:***

**Issue** Clarify what parameters are in the Message Header, the CPA, the CPP and how they all inter-relate.  
Section 10.2 in the ebMS spec describes the reliable messaging parameters: deliverySemantics, mshTimeAccuracy, TimeToLive, reliableMessagingMethod, ackRequested, retries, retryInterval, persistDuration. Only deliverySemantics, TimeToLive, reliableMessagingMethod, ackRequested can be found in Appendix A (ebXML SOAP Extension Elements Schema); retries, retryInterval, and persistDuration can only be found in Appendix D of the ebCPP spec.

I find the statement on line 1695 "This parameter information can be specified in the CPA or in the MessageHeader (section 8.4.2)." imprecise in the following sense:

mshTimeAccuracy is neither in the MessageHeader nor in the CPA.

TimeToLive is not mentioned anywhere in the CPA. It is not clear how the sending MSH should pick a value for this parameter.

The sub-sections describing retries, retryInterval, and persistDuration do not clearly indicate that these parameters are to be obtained from the CPA. Furthermore, their spellings do not match those in the CPA (case difference). It will be helpful to specify how elements in the MessageHeader can be used to look up these values from the CPA. The CPAId alone is not sufficient. The Service and Action elements will also have to be used to locate the relevant DocExchange, ebXMLBinding, and ReliableMessaging elements from the CPA.

It is not clear if the current Reliable Messaging specification works over multiple hops. Line 1774 prescribes that a TraceHeader be created in accordance with Section 8.5.2. The latter section however does not say anything about how to determine the next intermediary, in those cases where one or more intermediaries are to be involved. The descriptions on lines 1825 and 1829 on how to populate the From and To element in the Acknowledgement Message also do not clearly explain the circumstances under which sub-elements under the last TraceHeader in the incoming message are to be used.

I also find the following issues with Section 10.3.2 on Receiving Message Behavior:

Line 1783 uses the URI <http://www.ebxml.org/namespaces/messageService/MessageAcknowledgement>. This is not consistent with the URI specified on line 1823: [uri:www.ebxml.org/messageService](http://www.ebxml.org/messageService).

Steps 2.d)i) on line 1800 and 2.d)ii) on line 1802 are confusing. The phrase "and resend it" on line 1800 should be struck out. Otherwise, the message would be resent twice.

Step 2.c) omits the crucial action of passing on the message which has been found not to be a duplicate to the application.

Step 2.d)iii) is incomplete. The action to be taken when syncReply is set to True and the CPA indicates no application response is included is left unspecified. I believe in this case an Acknowledgement Message should be generated and returned synchronously.

**Friday, August 24, 2001**

**Page 21 of 38**

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

### ***Notes***

Under Section 10.3.5 Duplicate Message Handling, I find the description of an "identical message" puzzling. Why is it possible for a duplicate

| <b><i>Line Number:</i></b> | <b><i>Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
|----------------------------|---------------------|-------------------|-----------------------|
|                            | 0                   | 0                 | 10.2                  |
| 1695                       | 0                   | 0                 | 8.4.2                 |
|                            | 0                   | 0                 | ++                    |
|                            | 0                   | 0                 | 8.5.2                 |

***IssueID*** 21 ***Request Date*** 7/19/2001

***Request Source*** e-mail ***Source Reference***

***Change Type*** Editorial ***Disposition*** ***Disposition Date***

***Requested By*** Himagiri Mukkamala ***Email*** himagiri@sybase.com

### ***Source Reference:***

***Issue*** Update timestamp when resending message.

According to ebXML Reliable Messaging proposal, on failure to deliver, message should be resent.

In this case should the <Timestamp> element in MessageData be updated to refer to the new timestamp or the whole message be sent as it is without changing anything.

If so, MSH should'nt recreate the digital envelope if requested and if non-repudation no re-signing of the document.

### ***Notes***

| <b><i>Line Number:</i></b> | <b><i>Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
|----------------------------|---------------------|-------------------|-----------------------|
|                            | 0                   | 0                 | TBD                   |

***Friday, August 24, 2001***

***Page 22 of 38***

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 22 **Request Date**

**Request Source** **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00017.html>

**Change Type** Editorial **Disposition** **Disposition Date**

**Requested By** Michael Wang **Email** [mwang@tibco.com](mailto:mwang@tibco.com)

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00017.html>

**Issue** SimplePart element

Section 7.7.3 SimplePart element should probabaly be numbered as 7.7.2.1 and its title should be "NamespaceSupported element". (Section 7.7.2 already discusses SimplePart element.)

**Notes** Not sure this refers to the ebXML Messaging spec

**Line Number: Start** **End** **Section**  
0 0 7.7.3



## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 23 **Request Date** 8/6/2001

**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00042.html>

**Change Type** Major Technical **Disposition**

**Requested By** David Burdett

**Disposition Date**

**Email** david.burdett@commerceone.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00042.html>

**Issue** Add a "from service" to each message to indicate the service to which acknowledgements, deliver receipts, errors, message status etc should be returned.

Without this the sending MSH has to persist a message in order to determine which Service/Application sent it.

### ***Notes***

**Line Number: Start** **End** **Section**  
0 0 Various

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 24 **Request Date** 8/2/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00009.html>

**Change Type** MinorTechnical **Disposition**  
**Requested By** David Fischer **Email** david@drummondgroup.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00009.html>

**Issue** Always send an acknowledgment when sending a message asynchronously.

Rather than spend all this time trying to figure out when to send an Acknowledgement, what if we ALWAYS send an Acknowledgement to the sender/previous hop? I know this might consume some bandwidth but maybe it would solve some other problems? Someone told me RosettaNet works this way but I have not yet been able to confirm.

### ***Notes***

**Line Number: Start** **End** **Section**  
0 0 tbd

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 25 **Request Date** 8/2/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00006.html>

**Change Type** MinorTechnical **Disposition**  
**Requested By** Martin Sachs **Email** mwsachs@us.ibm.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00006.html>

**Issue** Remove reference to mshTimeAccuracy and instead suggest that good practice is followed such as using NTP is used instead.

**Notes**

**Line Number: Start** **End** **Section**  
0 0 tbd

## ***ebXML Messaging Specification Change/Issues Log***

***Document*** Messaging Service Spec 1.0 5/11/2001

|                              |               |                                |                                |
|------------------------------|---------------|--------------------------------|--------------------------------|
| <b><i>IssueID</i></b>        | 28            | <b><i>Request Date</i></b>     | 8/2/2001                       |
| <b><i>Request Source</i></b> | e-mail        | <b><i>Source Reference</i></b> |                                |
| <b><i>Change Type</i></b>    | Editorial     | <b><i>Disposition</i></b>      | <b><i>Disposition Date</i></b> |
| <b><i>Requested By</i></b>   | David Fischer | <b><i>Email</i></b>            | david@drummondgroup.com        |

### ***Source Reference:***

***Issue*** Inconsistency in description of id attributes  
Section 8.2.5 id attributes  
"Each of the ebXML SOAP extension elements listed above has an optional id attribute..."  
Section 8.11.1 id attribute  
"The Manifest element MUST have an id attribute that is an XML ID (See section 8.2.5)."

8.11.1 needs to be changed from MUST to MAY.

### ***Notes***

|                            |                     |                   |                       |
|----------------------------|---------------------|-------------------|-----------------------|
| <b><i>Line Number:</i></b> | <b><i>Start</i></b> | <b><i>End</i></b> | <b><i>Section</i></b> |
|                            | 0                   | 0                 | 8.2.5                 |

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 31 **Request Date** 8/2/2001  
**Request Source** **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00013.html>

**Change Type** Editorial **Disposition** **Disposition Date**  
**Requested By** David Fischer **Email** david@drummondgroup.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00013.html>

**Issue** Trace Header Required elements

Need to specify that Sender, Receiver and Timestamp are REQUIRED in the  
TraceHeader element

### ***Notes***

**Line Number: Start End Section**  
0 0 tbd

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 32 **Request Date** 8/2/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00014.html>  
<http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00015.html>

**Change Type** MinorTechnical **Disposition**  
**Requested By** David Fischer **Email** david@drummondgroup.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00014.html>

**Issue** The Sender and Receiver elements (8.5.2.1 & 8.5.2.2) each need a "Type" attribute to match the To and From elements. This is necessary since there may be multiple PartyId elements in a single TraceHeader. As with To and From, the "Type" should default to URI.

Add another paragraph to 8.5.2.1.1:

-----

Sorry, that should have been:

Add another paragraph to 8.5.2.1.1:

The PartyId element MAY have a single type attribute (see section 8.4.1.1).

The PartyId MAY have a single type attribute (see section 8.4.4.1)

### ***Notes***

**Line Number: Start**    **End**    **Section**  
0                      0    8.5.2

**Friday, August 24, 2001**

**Page 29 of 38**

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 33 **Request Date** 8/3/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Change Type** Editorial **Disposition**  
**Requested By** David Fischer **Email** david@drummondgroup.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Issue** Error in example

The Example in section 8.5.3 (TraceHeaderList) does not have a SOAP:actor=next.  
This is a single hop so perhaps the example is correct; however, section 8.5  
says SOAP:actor=next is REQUIRED.

### ***Notes***

**Line Number: Start End Section**  
0 0 8.5.3

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 34 **Request Date** 8/3/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Change Type** Editorial **Disposition** **Disposition Date**  
**Requested By** David Fischer **Email** david@drummondgroup.com  
**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>  
**Issue** Error in example

In the Example in section 8.5.4, Receiver needs a PartyId

### ***Notes***

| <b>Line Number:</b> | <b>Start</b> | <b>End</b> | <b>Section</b> |
|---------------------|--------------|------------|----------------|
|                     | 0            | 0          | 8.5.4          |



## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 35 **Request Date** 8/3/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Change Type** Editorial **Disposition**  
**Requested By** David Fischer **Email** david@drummondgroup.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Issue** Inconsistency in use of ID attribute

Section 8.7 Via element, the id attribute should be optional but the third paragraph says MUST contain. Change "MUST contain" to "has" since the description sections already say whether they are REQUIRED

### ***Notes***

**Line Number: Start End Section**  
0 0 8.7

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 36 **Request Date** 8/3/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Change Type** Editorial **Disposition**  
**Requested By** David Fischer **Email** david@drummondgroup.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Issue** Section 8.15.8 Delivery Receipt element

In the title DeliveryReceipt should be one word.

### ***Notes***

| <b>Line Number:</b> | <b>Start</b> | <b>End</b> | <b>Section</b> |
|---------------------|--------------|------------|----------------|
|                     | 0            | 0          | 8.15.8         |

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 37 **Request Date** 8/3/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Change Type** Editorial **Disposition**  
**Requested By** David Fischer **Email** david@drummondgroup.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Issue** 8.15.4 StatusRequest element

I don't think StatusRequest should be present with StatusResponse (add to the list)

### ***Notes***

**Line Number:** **Start** **End** **Section**  
0 0 8.15.4

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 38 **Request Date** 8/3/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Change Type** MinorTechnical **Disposition**  
**Requested By** David Fischer **Email** david@drummondgroup.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Issue** Handling payloads in the SOAP Body

Section 8.11 Manifest element says:

It is RECOMMENDED that no payload data be present in the SOAP Body.

If it was, how would it be identified in the Manifest? BizTalk uses a Document element (vs. an Attachment element) for this purpose. We don't have anything equivalent that I can see. I would rather take out this statement and possibly replace it with something like "All payload data MUST be carried by Reference (URI) or in an Attachment (CID)."

### ***Notes***

| <b>Line Number:</b> | <b>Start</b> | <b>End</b> | <b>Section</b> |
|---------------------|--------------|------------|----------------|
|                     | 0            | 0          | 8.11           |

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 39 **Request Date** 8/3/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Change Type** MinorTechnical **Disposition**  
**Requested By** David Fischer **Email** david@drummondgroup.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Issue** Make messageStatus required

In section 8.13 StatusResponse element, what does it mean if messageStatus is not present? Does this need to be REQUIRED?

### ***Notes***

**Line Number:** Start End Section  
0 0 8.13

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 40 **Request Date** 8/3/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Change Type** MinorTechnical **Disposition**  
**Requested By** David Fischer **Email** david@drummondgroup.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00026.html>

**Issue** Include RefToMessageId in the Delivery Receipt

Don't you need RefToMessageId in the DeliveryReceipt? How do you know which message this is a Delivery Receipt for? Need to add this sub-element. (This could be fixed by adding DeliveryReceipt to the list in MessageData, section 8.4.6.3, but I would prefer it as a REQUIRED sub-element of DeliveryReceipt like it is with StatusRequest & StatusResponse).

BTW... nothing seems to be required in the DeliveryReceipt. What does it mean if it is empty? Shouldn't we require at least a RefToMessageId and a Timestamp?

### ***Notes***

**Line Number: Start** **End** **Section**  
0 0 8.4.6.3

## ***ebXML Messaging Specification Change/Issues Log***

**Document** Messaging Service Spec 1.0 5/11/2001

**IssueID** 41 **Request Date** 8/8/2001  
**Request Source** e-mail **Source Reference** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00075.html>

**Change Type** MinorTechnical **Disposition**  
**Requested By** Martin Sachs **Email** mwsachs@us.ibm.com

**Source Reference:** <http://lists.oasis-open.org/archives/ebxml-msg/200108/msg00075.html>

**Issue** Clarify use of parameters in the CPA or message header.

10.2 Reliable Messaging Parameters

Lines 1694-1759: Lines 1695-1696 state that this information "can be specified in the CPA or in the message header." However it does not state where each item of information shall be specified. For each item, it must be said where the item shall be found. This is not a user choice matter. The implementation must know where to look. Please correct each item under section 10.2

### ***Notes***

**Line Number: Start**    **End**    **Section**  
0                      0    10.2