# Table of Contents

1. Introduction ................................................................................................................................................ 3  
   1.1 Acknowledgments................................................................................................................................. 3  
   1.2 Message Service Document Set ......................................................................................................... 3  
   1.3 ebXML Standards Documents Set ................................................................................................. 4  
   1.4 Key Words ......................................................................................................................................... 4  
2. Transport Protocols .................................................................................................................................... 6  
   2.1. HTTPS Protocol ............................................................................................................................... 6  
   2.2. SMTP Protocol ............................................................................................................................... 7  
   2.3. Additional Protocols ....................................................................................................................... 7  
3. ebMS 2.0 Message Structure ...................................................................................................................... 8  
   3.1 Introduction....................................................................................................................................... 8  
   3.2 Message Packaging ........................................................................................................................... 8  
   3.2.1 The Envelope / Message Package ............................................................................................... 8  
   3.2.2 The Header Container ................................................................................................................ 9  
   3.2.2.1 The Soap Envelope .............................................................................................................. 10  
   3.2.2.2 The SOAP Header .............................................................................................................. 10  
   3.2.2.3 The SOAP Body ................................................................................................................. 13  
4. Security, Reliable Messaging, and Error Reporting ................................................................................. 15  
   4.2. ebMS Reliable Messaging ............................................................................................................... 16  
   4.2.1 Guaranteed Delivery Assurance: Once-And-Only-Once ............................................................ 17  
   4.2.2 Acknowledgments ....................................................................................................................... 17  
   4.2.3 Error Handling ............................................................................................................................ 18  

Appendix A - Service, Action & CPAID Tables ............................................................................................. 20  
   AEC Project Implementation : .............................................................................................................. 20  
   Crop Nutrient Connectivity (CNC & XR) Projects Implementation Service, Action & CPAID ......... 21  
   Seed Connectivity (SC) Project Implementation Service, Action & CPAID .................................. 23  
   Product Cost Information (PCI) Project Implementation Service, Action & CPAID .................... 25  

Appendix B. Sample Envelopes .................................................................................................................. 24  
   Incoming Order Create /OrderResponse HTTPS Example: ............................................................. 25  
   Order Response Sent for OrderCreate: ......................................................................................... 27  
   Alternate Version of Order Response Sent for an OrderCreate: .................................................... 28  
   Incoming Order Create /OrderResponse SMTP Examples: ........................................................... 29  
   Outgoing PriceSheet: ......................................................................................................................... 35
1. Introduction

The AgGateway Ag eStandards were developed to meet Agricultural Industry needs for Internet based Business-To-Business interactions between enterprises utilizing XML-based standards. Ag eStandards utilized the Chem eStandards XML documents as the message payload and initially leveraged the messaging aspects of ebMS 1.0. ebXML is a joint effort from the UN (UNCEFACT) and the Organization for Structured Information Standards (OASIS) that provides an XML-based framework to define, discover, and execute business processes via the Internet. It is considered a horizontal standard that is not tied to industry vertical standards or payloads, and provides the potential for cross-industry convergence.

The ebXML messaging service specification (ebMS) deals with enabling secure and reliable transport, routing, and packaging of business messages across the Internet. Messaging Services sit above the core Internet data transfer protocols (HTTP, SMTP, FTP) and below the business application level software that understands and processes the message. At this time, ebMS 1.0 is the only ebXML specification that has been implemented at AGGATEWAY. The ebMS messaging service specification has evolved to a more recent version (currently ebMS 2.0) and other messaging service specifications have gained in popularity and robustness. There is a need to provide additional methods to satisfy the messaging service requirements.

This document interprets the ebXML Message Service Specification version 2.0 into an implementation specification for the AgGateway community. The focus is to create a map of the XML Element data requirements, and the ebMS functionality usage in order to promote interoperability between AgGateway trading partners.

1.1 Acknowledgments

Many sources of information were utilized during the production of this document. The construction of this standards document is based upon the work of many preceding groups. The sources include:

- The Chem eStandards Envelope and Security version 3.0
- STAR ebMS Implementation Guidelines
- A TRP Protocol Stack – A Tiered Approach to the PIDX Requirements
- AEC ebMS 1.0 Implementation
- OpenTravel™ Alliance 2001C Infrastructure Specification

1.2 Message Service Document Set

This document presents a set of implementation specifications that are necessary to fulfill in order to utilize ebMS 2.0 interoperably between various Message Service Handlers (MSH) within the
AgGateway eCommerce trading community. This is a specification document that is specific to ebMS 2.0, and is based on the high level generic requirements defined in the AgGateway Message Service Guidelines document.

A Requirement in this document is defined as an item or process that is required for interoperability. An item/process is determined to be a Requirement if either a system failure or interoperability failure will occur upon its removal. A Recommendation is a preferred method for implementation or an optional element. An item/process will be assigned a Recommendation status if its removal will not cause a system failure or interoperability failure to occur. The ebMS and Web Services documents are considered Specification documents. Specifications are companion documents to this document that describe specific implementation details that are necessary for completeness. The Specification documents may include both required and recommended items necessary to implement AgGateway B2B interoperable applications. If additional Message Service Specifications are required to be utilized, additional specification documents could be added to define their use.

1.3 ebXML Standards Documents Set

In addition to the ebXML Message Service Specification version 2.0, the following set of related specifications are included as part of the ebXML initiative:

• ebXML Technical Architecture Specification [ebTA] – defines the overall technical architecture for ebXML
• ebXML Technical Architecture Risk Assessment Technical Report [secRISK] – defines the security mechanisms necessary to negate anticipated, selected threats
• ebXML Collaboration Protocol Profile and Agreement Specification [ebCPP] – defines how one party can discover and/or agree upon the information the party needs to know about another party prior to sending them a message that complies with this specification
• ebXML Business Process Specification Schema [ebBPSS] – defines the decomposition of a business process to the transactional level in terms of XML.
• ebXML Registry/Repository Services Specification [ebRS] – defines a registry service for the ebXML environment

AgGateway has not currently implemented the additional ebXML specifications. These specifications have not been widely adopted throughout the eCommerce community at this time. The implementation of these specifications requires a good deal of effort and further definition of business processes and tools that have not sufficiently evolved to warrant adoption. Interpretation of certain elements within the message service specification that are related to these specifications was necessary and is noted within this document.

1.4 Key Words

Several words used within this document are intended to be interpreted in a particular way. The following list outlines the specific words and their meanings as they pertain to the AgGateway Ag eStandards:
MUST, REQUIRED, or SHALL - These words mean that the definition is an absolute requirement of the specification.

MUST NOT or SHALL NOT - These phrases mean that the definition is an absolute prohibition of the specification.
NOT REQUIRED – This phrase means that the definition is optional of the specification.

SHOULD or RECOMMENDED - These words mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT or NOT RECOMMENDED - These phrases mean that there may exist valid reasons in particular circumstances, when the particular behavior is acceptable or even useful. However, the full implication should be understood and the case carefully weighed before implementing any behavior described with this label.
2. Transport Protocols

AgGateway supports the use of two application level transport protocols to transport ebMS messages over the internet. HTTPS and SMTP protocols are acceptable for use with ebMS 2.0. Either HTTPS or SMTP may be used between AgGateway trading partners. Internal middleware partner configuration currently includes the partner’s protocol usage choice, and a modification of the protocol choice would require prior partner notification and re-configuration.

2.1. HTTPS Protocol

The HyperText Transfer Protocol (HTTP) was defined in an Internet Engineering Task Force (IETF) Document (ref: RFC 2616) in 1999 and remains the defacto web application transfer protocol. HTTPS was chosen by ebMS because it works well with today's Internet infrastructure, such as network firewalls. HTTPS is also known as SSL over HTTP.

EbMS AgGateway implementers MUST support the HTTP protocol specifications defined in appendix B.2 of the ebXML Message Service Specification version 2.0 document. The document defines the content type of Multipart/Related MIME, the SOAPAction as ebXML, HTTP Response code usage, use of HTTP POST messages as the message type, etc.

Additionally, ebMS AgGateway implementers MUST support Basic Authentication (ref: RFC 2617) for access control, operating over a secure channel, using SSL Version 3.0 or TLS (ref: RFC 2246) with 128-bit key sizes for symmetric encryption algorithms. For bi-directional communication, a separate mutually authenticated HTTPS connection is configured in each direction. Although the same concept as browser to server HTTPS is being used, for Ag eStandards, a higher level of security is involved when the connections are server to server. Mutual Authentication may be accomplished during the SSL handshake. The SSL handshake requires that the Receiver pass a Digital Certificate to the Sender. The Sender can verify that the Receiver is a known party and that the Receivers Digital Certificate has been signed by a Trusted Party, such as a Certificate Authority. Optionally, SSL can be used by the Receiver to require the Sender pass a Digital Certificate, enabling a mutual bi-directional authentication. Mutual Authentication is recommended for AGGATEWAY Ag eStandard connections.

EbMS AgGateway implementers MUST accept certificates from well-known Certificate Authorities (e.g. Verisign, Entrust, Thawte, et al), during the establishment of an SSL session for production servers. Optionally, self signed digital certificates SHOULD be accepted for test servers. More information pertaining to using Certificates within the AgGateway community can be found in Best Practice Guidelines For Digital Certificate Handling.

EbMS AgGateway implementers utilizing HTTPS MUST maintain ebMS systems that are available 24 hours per day, 7 days per week (allowing for maintenance windows), to receive and process electronic business transactions from trading partners.
2.2. SMTP Protocol

The Simple Mail Transfer Protocol (SMTP) was defined in an IETF Document (ref: RFC 22821) in 2001 and remains the defacto web email transfer protocol. The SMTP protocol was chosen to be used as a solution for Non-Addressable endpoints. Non-Addressable endpoints provide temporary cost effective connections to the internet, such as dial up and imply asynchronous messaging will take place. SMTP is the Internet standard host-to-host mail transport protocol. It is the defacto standard protocol used by all major internet email applications. SMTP is utilized as an internet protocol that will pass messages to a separately provided mailbox. Messaging software will have to be able to read messages from an inbox and write messages to an outbox. An email address for the mailbox will need to be provided as part of the message address.

EbMS AgGateway implementers MUST support the SMTP protocol specifications defined in appendix B.3 of the ebXML Message Service Specification version 2.0 document by OASIS. The document defines the content type of Multipart/Related MIME, the SOAPAction as ebXML, the use of a “To” and “From” valid email address, and the use of a valid “Date”.

EbMS AgGateway implementers MUST format the ebMS message according to the rules specified within the ebMS specification (in other words, a valid payload encapsulated within a valid ebMS 2.0 multipart MIME envelope containing a valid ebMS Header and Body).

2.3. Additional Protocols

HTTPS communication is supported for AgGateway Ag eStandards compliance, because it is open and non-proprietary, yet provides a high degree of security and reliability as used here. SMTP is supported for trading partners with non-addressable endpoints that desire to utilize non permanent dial up connections in order to communicate via XML.

A pair of trading partners may select another approach to communication that better meets their mutual needs. While the AgGateway Ag eStandards recognize the use of additional transfer protocols, such as Secure FTP and asynchronous message queuing transports, there will not be any specific implementation guidance beyond the HTTPS and SMTP transports. If trading partners decide to utilize any of the protocols other than HTTPS or SMTP, every effort should be made to provide the same level of security that is provided when transporting messages via the supported protocols.
3. ebMS 2.0 Message Structure

3.1 Introduction

The EbXML Message Service, which is based on Simple Object Access Protocol (SOAP) version 1.1 and the SOAP with Attachments informational document, provides the functionality needed for two or more parties to engage in an “electronic business transaction”. Products that implement the ebMS specification should be capable of the reliable and secure exchange of business data associated with a “business transaction”.

A high quality ebMS implementation would also provide facilities for encrypting and digitally signing data, access control, authentication and authorization, real-time error notifications, logging, auditing and administrative tasks based upon the needs of the applications.

Regardless of which model is used AgGateway ebMS implementers SHOULD provide the same service levels, access control, reliability, availability and security required by the electronic business transactions defined by AgGateway.

3.2 Message Packaging

3.2.1 The Envelope / Message Package

In general terms, XML Messages transmitted via the Internet are enclosed in a container that is known as the message envelope. The envelope must provide a boundary mechanism to insure that the transmission of the entire message is accomplished and must provide a means of separation for the different components. A mechanism must be provided that will identify the contents of the message. The mechanism must provide information pertaining to each payload that is packaged within the message.

An ebXML Message following the ebMS 2.0 conventions MUST contain one header container and zero to many payload containers. Both containers are enveloped by a single MIME/Multipart envelope with a type of "text/xml", and this entire package is referred to as a Message Package. An example of the Message Package MIME Part is:

```
Content-Type: multipart/related; type="text/xml"; boundary="boundaryValue";
start=messagepackage-123@example.com

--boundaryValue
Content-ID: <messagepackage-123@example.com>
```

[Source: ebXML Message Service Specification Version 2.0]
The header container contains service information relating to the message. The payload container(s) contain the “business specific” information. The payload message may or may not be in an XML format according to the ebMS 2.0 specification.

AgGateway ebMS implementers MUST include either zero payloads (for a technical acknowledgment message), or one and only one payload in an acceptable XML format (for a business message) within a Message Package. Multiple payloads and non-XML payloads are not currently supported.

The general structure and composition of an AgGateway compliant ebXML Message is described in Figure 1:

![Message Structure](image)

Figure 1 : Message Structure  [Source: ebXML Message Service Specification Version 2.0]

### 3.2.2 The Header Container

The Header container is a MIME PART that contains the SOAP Envelope. The MIME Content-Type header for the **Header Container** MUST have the value "text/xml" in accordance with the [SOAP] specification. The Content-Type header MAY contain a "charset" attribute. For example: Content-Type: text/xml; charset="UTF-8". AgGateway recommends that implementations specify a content-type of application/xml with the optional character set attribute set to “utf-8”.
3.2.2.1 The Soap Envelope

The SOAP Envelope is defined as an XML document, structured according to SOAP version 1.1. The SOAP Envelope XML document consists of one Envelope Element, which contains one mandatory body element and one optional header element. The ebMS specification ver 2.0 mandates that all ebXML header documents MUST contain both a header and a body element. The following example depicts a skeleton SOAP Envelope structure:

```
 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
 http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd">
  <SOAP-ENV:Header>
    .........
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    .........
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The Namespace for the SOAP envelope is defined on the Envelope Element. The value of “http://schemas.xmlsoap.org/soap/envelope/” is a REQUIRED value for the namespace. The prefix used for the Namespace in this example is SOAP-ENV. The two most common prefixes are SOAP and SOAP-ENV. These prefixes MUST be encoded consistently throughout the Envelope document. The namespace and schemaLocation definitions MUST conform to the ebMS 2.0 specifications (see section 2.3 of ebMS ver 2.0 Specification).

3.2.2.2 The SOAP Header

The SOAP Header element is the first child element of the SOAP Envelope Element. The Header namespace and schemaLocation definitions MUST conform to the ebMS 2.0 specifications (see section 2.3 of ebMS ver 2.0 Specification). The header structure is constructed as follows:

```
<SOAP-ENV:Header
 xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
  <eb:MessageHeader ...
  .........
</eb:MessageHeader>
</SOAP-ENV:Header>
```
(digital signatures are currently NOT USED by AgGateway), and the **ErrorList** element that is used for acknowledgment errors (see Section 4).

The **MessageHeader element**: is a composite element comprised of the following subordinate elements:

- an optional **ID** attribute that is NOT Utilized by AgGateway
- a **VERSION** attribute is set to 2.0 for the ebMS version being used.
- a SOAP **mustUnderstand** attribute with a value of "1" (indicates whether the contents of the element MUST be understood by a receiving process or else the message MUST be rejected in accordance with SOAP).
- **From and To element**: These are REQUIRED Elements that identify the sending and receiving party. They each include the **PartyId** element that contains the value, and the value is defined by the **PartyId type** attribute. For the AgGateway implementers the value MUST be either “DUNS” with 9 digit number or “DUNS+4” for a 13 digit numbers or “GLN” with 13 digit numbers will be sent. An example is below:

```xml
<eb:MessageHeader eb:version="2.0" SOAP-ENV:mustUnderstand="1">
  <eb:From>
    <eb:PartyId eb:type="DUNS">878542765</eb:PartyId>
  </eb:From>
  <eb:To>
    <eb:PartyId eb:type="DUNS+4">0503651210000</eb:PartyId>
  </eb:To>
</eb:MessageHeader>
```

- **CPAId element**: The **CPAId** element is a REQUIRED ebXML element. AgGateway does not utilize the ebXML CPA specification to form an agreement between Trading Partners, so the convention to identify the from and to party in the CPAId element is not necessary. The AgGateway convention for naming the CPAId is to use the Document message name unless the business message is used to close a conversation, in which case it would use the name of the message to which the response is being made (See Table 1 below). An example is below:

```xml
<eb:CPAId>OrderCreate</eb:CPAId>
```
**ConversationId element**: The REQUIRED ConversationId element is a string identifying the set of related messages that make up a conversation between two Parties. It MUST be unique within the From and To party pair. The Party initiating a conversation determines the initial value of the ConversationId element and populates that value in the initial message of the conversation. All subsequent messages pertaining to that conversation will be populated with the same conversation id value. The usage of Conversations to define the AgGateway OrderCreate / OrderResponse process is Optional because of middleware software constraints. If a Conversation is used, the middleware will be configured to close the conversation after the business response document (OrderResponse) with a matching conversation Id is received (matching the OrderCreate). The value of the Action Element determines whether a conversation is being utilized or not (See Table 1 Below). If a Conversation is not used, the middleware will be configured to close the conversation after the Reliable Messaging Acknowledgment is received. A Conversation Id Element will be sent on the business response document (OrderResponse), but will not match the OrderCreate Conversation Id.

**Service element**: The REQUIRED Service element identifies the service that acts on the message and was meant to be related to a business process defined in terms of an ebXML Business Process Specification Schema. AgGateway does not utilize ebXML BPSS. The Service is always the same as the CPAId value but preceded by “uri:”. For Example:

```
<eb:Service>uri:OrderCreate</eb:Service>
```

**Action element**: The REQUIRED Action element identifies a process within a Service that processes the Message. Action SHALL be unique within the Service in which it is defined. The value of the Action element is specified by the designer of the service. An example of the Action element follows:

```
<eb:Action>SendResponseEnd</eb:Action>
```

The CPA ID, Action, Service, and Conversation ID Elements are defined in AgGateway in APPENDIX A.
CPAID, SERVICE, ACTION and CONVERSATION ID Element Usage

- **MessageData element**: The REQUIRED MessageData element provides a means of uniquely identifying an ebXML Message. It contains the following:
  
  a. **MessageId element**: The REQUIRED Message ID MUST be Globally unique and conformant to ebMS specifications which require that the value is conformant to RFC2392. For example:

  ```xml
  <eb:MessageId>nexus/435f3d22:10018f2f184:-7c74</eb:MessageId>
  ```

  b. **Timestamp element**: The REQUIRED Timestamp element is a value representing the creation time of the message header and MUST be in UTC format (Universal Time Code as defined by ISO 8601). For Example:

  ```xml
  <eb:Timestamp>2005-11-11T15:08:50.975Z</eb:Timestamp>
  ```

  c. **RefToMessageId element**: If Present (based on the message type – such as a Response MUST have the RefToMessageID), the RefToMessageId MUST contain the MessageId value of an earlier ebXML Message to which this message relates. This element is only present in Reliable Messaging Acknowledgments. See Reliable Messaging Chapter.

  d. **TimeToLive element**: If Present, the TimeToLive element indicates the time at which this message expires and cannot be processed. The TimeToLive element is discussed further in the Reliable Messaging chapter. This is NOT Used in AgGateway.

- **DuplicateElimination element**: The DuplicateElimination element identifies a request by the sender for the receiving Partner to check for duplicate messages. Checking for Duplicates is REQUIRED by AgGateway. See Reliable Messaging section 4.2.1.

- **Description element**: This is an OPTIONAL element that is not utilized by AgGateway.

The **Acknowledgment Element**: The Acknowledgment element is used by the To Party that received a message, to send a reliable messaging acknowledgment to the From Party that sent the message to notify of receipt. This element is only present in Reliable Messaging Acknowledgments. The RefToMessageId in a message containing an Acknowledgment element identifies the message for which the receipt is being generated. The RefToMessageId is the MessageId of the original message. See Reliable Messaging Chapter.

3.2.2.3 The SOAP Body
The SOAP **Body** element is the second child element of the SOAP **Envelope** element. It MUST have a namespace qualifier that matches the SOAP **Envelope** namespace declaration for the namespace "http://schemas.xmlsoap.org/soap/envelope/". The Body namespace and schemaLocation definitions MUST conform to the ebMS 2.0 specifications (see section 2.3 of ebMS ver 2.0 Specification). The body structure is constructed as follows:

```
<SOAP-ENV:Body
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<eb:Manifest eb:version="2.0">
   .......
</eb:Manifest>
</SOAP-ENV:Body>
```

The Element added by ebXML to extend SOAP in the SOAP Body is the Manifest Element.

**The Manifest Element:** The Manifest is a composite element that summarizes message payloads. A Manifest element MUST be present if one or more Payloads exist, and all Payloads MUST be referenced in the Manifest. The manifest facilitates payload extraction and provides payload information to an application without requiring that a payload be parsed. The structure and content of the Manifest element MUST conform to the ebMS version 2.0 specifications. An example is illustrated below:

```
<eb:Manifest soap-env:mustUnderstand="1" eb:version="2.0">
   <eb:Reference xmlns:xlink="http://www.w3.org/1999/xlink" eb:id="Payload-01"
xlink:href="nexus/435f3d22:10018f2f184:-7d8f-body1"/>
</eb:Manifest>
```

AgGateway does REQUIRE a Manifest element be used. For those Partners that cannot populate the manifest to indicate the current Payload that is contained in the message, the value “Payload-01” will be used. For Partners able to provide detailed information about the Payload, the Manifest will appear as follows:

```
<eb:Manifest eb:version="2.0">
   <eb:Reference xlink:type="simple" xlink:href="cid:OrderResponse"
xlink:role="http://www.elemica.com/Chem/">
      <eb:Schema
         eb:location="http://www.elemica.com/Chem202/OrderResponsev2.0.2.dtd"
eb:version="2.0.2" />
      <eb:Description xml:lang="en-us">chemXMLOrderResponse2.0.2</eb:Description>
   </eb:Reference>
</eb:Manifest>
```
4. Security, Reliable Messaging, and Error Reporting

4.1. Security Requirements Summary

Security is concerned with the protection of information and systems from the risk of failure or inaccuracy.

1. AgGateway REQUIRES that the message is encrypted using Digital Certificates and the Mutual Authentication is accomplished. Mutual authentication requires both the sender and receiver to verify their identities before transmission of a message proceeds. This provides both partners with the assurance that they are sharing information only with trading partners whose identities have been verified. Authentication requires trading partners to mutually validate their electronic transmissions, usually by an independent third party. The method used to accomplish this is SSL via HTTP.

The SSL Handshake initiates an SSL session between two partners. The SSL handshake requires that the receiver pass a Digital Certificate to the sender. The sender can verify that the receiver is a known party and that the receiver’s Digital Certificate has been signed by a Trusted Party, such as a Certificate Authority. In this manner, a sender authenticates the identity of the receiver. The Digital Certificates for production servers are REQUIRED to be purchased from a Certificate Authority (Verisign, Thawte, etc.) and cannot be self signed.

Confidentiality can be managed at the session level and/or the message level. Session level security ensures that the connection between the sender and the receiver is secure. Secure Sockets Layer (SSL) is used to enable security requirements at the session level. SSL is the defacto method for transmitting private documents over the Internet. SSL is a cryptographic system, which is based on the use of Digital Certificates. Digital Certificates use a public key known to everyone and a private or secret key known only to the recipient of the message. The encryption occurs after the certificate-based authentication is performed.

2. Authorization ensures that an authenticated trading partner is permitted to send the type of data or transaction that is contained in the message. Once the trading partner’s identity is authenticated and a connection is established, the receiving partner’s application software must determine what capabilities are granted to the partner. This would include allowing specific message types (i.e. ShipNotice) and particular data requirements that further qualify a message (i.e. particular order types). AgGateway RECOMMENDS that each receiving partner will need to design internal mechanisms to control the authorization process for incoming messages. There is a possibility that some of these mechanisms may be controlled manually.

3. Message level security is the ability to encrypt part or all of a payload applied to the message payload itself and offers an additional layer of security during transportation. AgGateway does not have a requirement for the use of message-based encryption for the message.
4. Digital Signatures are used to enable Integrity. To create a signature, the message to be sent is used to create a hash, which is known as the message digest that represents the message data. Any change to the message would result in a change to the digest. The sender digitally signs the digest with a key. The receiver can un-encrypt the sender's digest, then calculate a new digest for the received message, if the two digests match, the sender can be assured that the message has not been changed in transit. AgGateway does not require the use of Digital signatures.

5. Message Ordering is the ability to present messages in a particular order and to identify and validate the sequence of messages. In other words, message 3 follows message 2, which follows message 1. A sender expects that the first message will be processed before the second message. Message Ordering guarantees that messages are processed in a sequence defined by the message sender. AgGateway does not require the use of message ordering.

6. TimeToLive defines a timestamp that is associated with the creation of a message that defines its useful processing life. If the receiver receives a message who’s TimeToLive has expired, the message should be ignored. In order to implement the TimeToLive process, a policy must be defined that sets the time parameter for the expiration of the lifespan of the message. This parameter would be specific to the business process / business message that is executed. AgGateway does not require the use of a TimeToLive process.

7. Non-repudiation means that it can be verified that the sender and the recipient were, in fact, the parties who claimed to send or receive the message, respectively. Non-repudiation provides protection against data being renounced or a transaction being denied. All forms of non-repudiation are enabled by persistently storing information received as part of the transaction. AgGateway does not require that its messaging systems enable support for Non-Repudiation of Receipt, Non-Repudiation of Origin, or Non-Repudiation of Content. An acknowledgment of receipt is required, but it currently does not contain a message digest.

4.2. ebMS Reliable Messaging

In the most basic level, reliable messaging refers to the ability of a sender to deliver a message once and only once to its intended receiver and a receiver to receive and process a message once and only once. Additionally, more robust requirements for reliable messaging would include that each message sent be received exactly once (once and only once), at most once, at least once, and so on; Messages be received in the same order in which they were sent; Failure to deliver a message will be made known to both the sender and receiver.

AgGateway requires the support of Once-And-Only-Once guaranteed delivery. To Implement this, a Receiving Partner MUST reply to the Sending Partner with an Acknowledgment. An Acknowledgment Message is any ebXML message containing an Acknowledgment element. Failure to receive an Acknowledgment from the Receiving Partner MAY trigger a re-sending of the original message until an Acknowledgment is received or the number of maximum retries is met.
Whenever an identical message may be received more than once, some method of duplicate detection and elimination is indicated, usually through the storage of message IDs.

4.2.1 Guaranteed Delivery Assurance: Once-And-Only-Once

Implementers for the Sending Partner MUST provide and use the following features within the SOAP Header of the message sent:

• MUST include an AckRequested element. This is an element in the SOAP Header that requests the return of an acknowledgment from the receiving partner. A sample is as follows:

```
<eb:AckRequested eb:version="2.0" eb:signed="false" SOAP-ENV:mustUnderstand="1" />
```

• MUST include a DuplicateElimination element. This element signals to the Receiving partner that duplicates must be checked for. An example is as follows:

```
<DuplicateElimination/>
```

• The TimeToLive element is NOT REQUIRED
• Parties MUST agree prior to transmission to a value for Retry Interval
• Parties MUST agree prior to transmission to a value for NumberOfRetries

4.2.2 Acknowledgments

AgGateway REQUIRES the use of a Reliable Messaging Acknowledgment to acknowledge the receipt (not validation) of a message. EbMS version 2.0 sends Acknowledgments as a SOAP Header with a null SOAP Body and no Payload attachments. The SOAP Header will contain an Acknowledgment Element that conforms to ebMS 2.0 and AgGateway requirements. The sending of the acknowledgment MUST conform to the ebXML Reliable Messaging Protocol defined in section 6.5 of the ebMS 2.0 specification.

The Acknowledgment element for AgGateway will contain the following elements and attributes:

• a REQUIRED version attribute identifying ebMS 2.0

• a REQUIRED SOAP mustUnderstand attribute with a value of "1"

• an Optional SOAP actor attribute (Default target is the To Party if the Actor is not defined)
• a REQUIRED Timestamp element

• a REQUIRED RefToMessageId element that contains the MessageId of the message whose delivery is being reported.
• a REQUIRED From element identifying the party that is sending the Acknowledgment.

An example of the Acknowledgment element is as follows:

```xml
<eb:Acknowledgment SOAP:mustUnderstand="1" eb:version="2.0">
  <eb:Timestamp>2005-03-09T12:22:30</eb:Timestamp>
  <eb:RefToMessageId>0ah5gm0v00kuengi00000d27</eb:RefToMessageId>
  <eb:From>
    <eb:PartyId eb:type="DUNS+4">0503651210000</eb:PartyId>
  </eb:From>
</eb:Acknowledgment>
```

### 4.2.3 Error Handling

In AgGateway an exception message MAY be sent for problems that occur up until the time when the receipt process is completed with an Acknowledgment message. Validation and business process exceptions that arise will be dealt with manually between partners.

AgGateway conforms with the Error Processing that is described in the ebXML Message Service Specification, Version 2.0 in section 4.2. This processing capability is inherent in the middleware products supporting ebMS 2.0, if properly configured.

Data communications protocols errors are detected and reported using the standard mechanisms supported by that data communications protocol and do not use the error reporting mechanism described by ebMS2.0.

Errors are reported by use of an ErrorList element within the SOAP Header element that indicates the message identified by the RefToMessageId in the MessageHeader element has an error.

The ErrorList element consists of:

• a version attribute identifying ebMS 2.0

• a SOAP mustUnderstand attribute with a value of "1"

• highestSeverity attribute:
  
  The highestSeverity attribute contains the highest severity of any of the Error elements. Specifically, if any of the Error elements have a severity of Error, highestSeverity MUST be set to Error; otherwise, highestSeverity MUST be set to Warning (See ebMS ver2.0 Section 4.2.3).
One or more **Error** elements An **Error** element consists of:

- **codeContext** attribute identifying the namespace or schema of the error messages.

- **errorCode** attribute indicating the nature of the error whose valid values are listed in ebMS 2.0 section 4.2.3

- **severity** attribute indicating either a warning or an error (See ebMS 2.0 section 4.2.3)

- **location** attribute pointing to the part of the message containing the problem.

- **Description** element that provides a narrative description of the error.

An example of the ErrorList element is below:

```xml
<eb:ErrorList eb:version="2.0" soap:mustUnderstand="1" eb:highestSeverity="Error">
  <eb:Error eb:codeContext="http://www.ebxml.org/messageServiceErrors"
    eb:errorCode="ValueNotRecognized"
    eb:severity="Error"
    eb:location="/soap:Envelope/soap:Body/eb:Manifest/eb:Reference">
    <eb:Description xml:lang="us-en">
      VV-001:[B2BCORE.0082.9025] Missing Object
    </eb:Description>
  </eb:Error>
</eb:ErrorList>
```
### Appendix A - Service, Action & CPAID Tables

**Accelerated Electronic Connectivity (AEC) Project Implementation:**

<table>
<thead>
<tr>
<th>Document and Direction</th>
<th>Receiving party is able to provide identical ConversationID in the response</th>
<th>Receiving party unable to provide identical ConversationID in the response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OrderCreate - From Buyer to Seller</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:OrderCreate</td>
<td>uri:OrderCreate</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFile</td>
<td>SendResponseEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>OrderCreate</td>
<td>OrderCreate</td>
</tr>
<tr>
<td><strong>OrderResponse - From Seller to Buyer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:OrderCreate</td>
<td>uri:OrderCreate</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFileResponse</td>
<td>SendResponseEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>OrderCreate</td>
<td>OrderCreate</td>
</tr>
<tr>
<td>OrderResponse?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>ShipNotice - From Buyer to Seller</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:ShipNotice</td>
<td>uri:ShipNotice</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFile</td>
<td>SendFile</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>ShipNotice</td>
<td>ShipNotice</td>
</tr>
<tr>
<td><strong>Invoice - From Buyer to Seller</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:Invoice</td>
<td>uri:Invoice</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFile</td>
<td>SendFile</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>Invoice</td>
<td>Invoice</td>
</tr>
<tr>
<td><strong>ProductMovement - From Buyer to Seller</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:ProductMovement</td>
<td>uri:ProductMovement</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFile</td>
<td>SendFile</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>ProductMovementReport</td>
<td>ProductMovementReport</td>
</tr>
<tr>
<td><strong>IAU - From Buyer to Seller</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:InventoryActualUsage</td>
<td>uri:InventoryActualUsage</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFile</td>
<td>SendFile</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>InventoryActualUsage</td>
<td>InventoryActualUsage</td>
</tr>
</tbody>
</table>
## Crop Nutrient Connectivity Projects Implementation Service, Action & CPAID

<table>
<thead>
<tr>
<th>Document and Direction</th>
<th>Receiving party is able to provide identical ConversationID in the response</th>
<th>Receiving party unable to provide identical Conversation ID in the response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ContractCreate - To a Distributor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNContract</td>
<td>uri:CNContract</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>ContractCreate</td>
<td>ContractCreateEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CNContract</td>
<td>CNContract</td>
</tr>
<tr>
<td><strong>ContractResponse - To a Manufacturer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNContract</td>
<td>uri:CNContract</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>ContractResponse</td>
<td>ContractResponseEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CNContract</td>
<td>CNContract</td>
</tr>
<tr>
<td>Must return ContractCreate ConversationID in ContractResponse?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>OrderCreate - To a Manufacturer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNOrder</td>
<td>uri:CNOrder</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>OrderCreate</td>
<td>OrderCreateEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CNOrder</td>
<td>CNOrder</td>
</tr>
<tr>
<td><strong>OrderResponse - To a Distributor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNOrder</td>
<td>uri:CNOrder</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>OrderResponse</td>
<td>OrderResponseEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CNOrder</td>
<td>CNOrder</td>
</tr>
<tr>
<td>Must return OrderCreate ConversationID in OrderResponse?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>OrderChange - To the Seller (Buyer initiated)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNOrderChangeBuyer</td>
<td>uri:CNOrderChangeBuyer</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>OrderChange</td>
<td>OrderChangeEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CNOrderChangeBuyer</td>
<td>CNOrderChangeBuyer</td>
</tr>
<tr>
<td><strong>OrderResponse (for OrderChange) - To the Buyer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNOrderChangeBuyer</td>
<td>uri:CNOrderChangeBuyer</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>OrderChangeResponse</td>
<td>OrderChangeResponseEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CNOrderChangeBuyer</td>
<td>CNOrderChangeBuyer</td>
</tr>
<tr>
<td>Must return OrderChange ConversationID in OrderResponse?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>OrderResponse (Unsolicited by Seller) - To the Buyer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNOrderChangeSeller</td>
<td>uri:CNOrderChangeSeller</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>OrderChange</td>
<td>OrderChangeEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CNOrderChangeSeller</td>
<td>CNOrderChangeSeller</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td><strong>OrderChange Response (for unsolicited change initiated by Seller) - To the Seller</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Must return OrderChange ConversationID in OrderResponse?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNOrderChangeSeller</td>
<td>uri:CNOrderChangeSeller</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>OrderChangeResponse</td>
<td>OrderChangeResponseEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CNOrderChangeSeller</td>
<td>CNOrderChangeSeller</td>
</tr>
<tr>
<td>Must return OrderChange ConversationID in OrderResponse?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>ShipNotice - to a Distributor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNShipment</td>
<td></td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>ShipNotice</td>
<td></td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CNShipment</td>
<td></td>
</tr>
<tr>
<td><strong>ShipmentStatus - to a Distributor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNShipmentStatus</td>
<td></td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>ShipmentStatus</td>
<td></td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CnShipmentStatus</td>
<td></td>
</tr>
<tr>
<td><strong>Invoice - to a Distributor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:CNInvoice</td>
<td></td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>Invoice</td>
<td></td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>CNInvoice</td>
<td></td>
</tr>
<tr>
<td><strong>PaymentDetail – to a Manufacturer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:PaymentDetail</td>
<td>uri:PaymentDetail</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>PaymentDetail</td>
<td>PaymentDetailEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>PaymentDetail</td>
<td>PaymentDetail</td>
</tr>
</tbody>
</table>
# Seed Connectivity (SC) Project Implementation Service, Action & CPAID

<table>
<thead>
<tr>
<th>Document and Direction</th>
<th>Receiving party is able to provide identical ConversationID in the response</th>
<th>Receiving party unable to provide identical Conversation ID in the response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeliveryConfirmation - To a Trading Partner</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:SeedDeliveryConfirmation</td>
<td>uri: SeedDeliveryConfirmation</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFile</td>
<td>SendFileEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>SeedDeliveryConfirmation</td>
<td>SeedDeliveryConfirmation</td>
</tr>
<tr>
<td><strong>DeliveryConfirmationResponse - To a trading partner</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:SeedDeliveryConfirmationResponse</td>
<td>uri: SeedDeliveryConfirmationResponse</td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFileResponse</td>
<td>SendFileResponseEnd</td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>SeedDeliveryConfirmationResponse</td>
<td>SeedDeliveryConfirmationResponse</td>
</tr>
<tr>
<td>Must return OrderCreate ConversationID in OrderResponse?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>ShipNotice - to a trading partner</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:SeedShipNotice</td>
<td></td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFile</td>
<td></td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>SeedShipNotice</td>
<td></td>
</tr>
<tr>
<td><strong>Invoice - to a trading partner</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:SeedInvoice</td>
<td></td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFile</td>
<td></td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>SeedInvoice</td>
<td></td>
</tr>
<tr>
<td><strong>ProductMovement - to a Trading Partner</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebMS_Service</td>
<td>uri:SeedProductMovementReport</td>
<td></td>
</tr>
<tr>
<td>ebMS_Action</td>
<td>SendFile</td>
<td></td>
</tr>
<tr>
<td>ebMS_CPAID/Choreography Name</td>
<td>SeedProductMovementReport</td>
<td></td>
</tr>
</tbody>
</table>
## Product Cost Information (PCI) Project Implementation
### Service, Action & CPAID

<table>
<thead>
<tr>
<th>Document and Direction</th>
<th>Receiving party is able to provide identical ConversationID in the response</th>
<th>Receiving party unable to provide identical Conversation ID in the response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Cost Information (sent by the Buyer to the Seller)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>ebMS_Service</code></td>
<td><code>uri:PriceSheet</code></td>
<td><code>uri:PriceSheet</code></td>
</tr>
<tr>
<td><code>ebMS_Action</code></td>
<td><code>PriceSheet</code></td>
<td><code>PriceSheetEnd</code></td>
</tr>
<tr>
<td><code>ebMS_CPAID/Choreography Name</code></td>
<td><code>PriceSheet</code></td>
<td><code>PriceSheet</code></td>
</tr>
</tbody>
</table>
Appendix B. Sample Envelopes

Incoming Order Create /OrderResponse HTTPS Example:

Incoming OrderCreate Envelope:

```xml
<?xml version="1.0" encoding="UTF-8"?>
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Header>
    <eb:MessageHeader eb:version="2.0" soapenv:mustUnderstand="1" xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
      <eb:From>
        <eb:PartyId eb:type="GLN">0576403930000</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="GLN">9666166330000</eb:PartyId>
      </eb:To>
      <eb:CPAId>Order</eb:CPAId>
      <eb:ConversationId>srv252_1386000_2007091512243106</eb:ConversationId>
      <eb:Service>uri:Order</eb:Service>
      <eb:Action>OrderCreate</eb:Action>
      <eb:MessageData>
        <eb:MessageId>srv252/1386000:1090cd58170:-7fb8</eb:MessageId>
      </eb:MessageData>
      <eb:DuplicateElimination/>
    </eb:MessageHeader>
    <eb:AckRequested eb:version="2.0" eb:signed="false" soapenv:mustUnderstand="1"
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>
  </soapenv:Header>
    <eb:Manifest soapenv:mustUnderstand="1" eb:version="2.0" xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>
  </soapenv:Body>
</soapenv:Envelope>
```
Technical Response Sent For OrderCreate:

<?xml version="1.0" encoding="UTF-8"?>
  <soapenv:Header/>
  <eb:MessageHeader eb:version="2.0" soapenv:mustUnderstand="1" xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
    <eb:From>
      <eb:PartyId eb:type="GLN">9666166330000</eb:PartyId>
    </eb:From>
    <eb:To>
      <eb:PartyId eb:type="GLN">0576403930000</eb:PartyId>
    </eb:To>
    <eb:CPAId>Order</eb:CPAId>
    <eb:ConversationId>srv252_1386000_2007091512243206</eb:ConversationId>
    <eb:Service>uri:www.ebxml.org/messageService</eb:Service>
    <eb:Action>Acknowledgment</eb:Action>
    <eb:MessageData>
      <eb:MessageId>ALFTXML1/7bd9f2:1090cd59ca:7fb8</eb:MessageId>
      <eb:Timestamp>2007-09-15T12:32:00Z</eb:Timestamp>
      <eb:RefToMessageId>srv252/1386000:1090cd58170:7fb8</eb:RefToMessageId>
    </eb:MessageData>
  </eb:MessageHeader>
  <eb:Acknowledgment eb:version="2.0" soapenv:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH" soapenv:mustUnderstand="1" xmlns="">
    <eb:Timestamp>2006-01-27T12:32:00Z</eb:Timestamp>
    <eb:RefToMessageId>srv252/1386000:1090cd58170:7fb8</eb:RefToMessageId>
    <eb:From>
      <eb:PartyId eb:type="GLN">9666166330000</eb:PartyId>
    </eb:From>
  </eb:Acknowledgment>
</soapenv:Envelope>
Order Response Sent for OrderCreate:

**Note: This document contains the same ConvID and contains an Action of OrderResponse. Other values are possible (see Table 1 on page 13 and second example below).**

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <soapenv:Header>
    <eb:MessageHeader eb:version="2.0" soapenv:mustUnderstand="1" xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
      <eb:From>
        <eb:PartyId eb:type="GLN">9666166330000</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="GLN">0576403930000</eb:PartyId>
      </eb:To>
      <eb:CPAId>Order</eb:CPAId>
      <eb:ConversationId>srv252_1386000_2007091512243106</eb:ConversationId>
      <eb:Service>uri:Order</eb:Service>
      <eb:Action>OrderResponse</eb:Action>
      <eb:MessageData>
        <eb:MessageId>ALFTXML1/7bd9f2:1090cd59caec:-7f9c</eb:MessageId>
      </eb:MessageData>
    </eb:MessageHeader>
    <eb:AckRequested eb:version="2.0" eb:signed="false" soapenv:mustUnderstand="1" xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>
  </soapenv:Header>
  <soapenv:Body">
      <eb:Reference xsi:href="ALFTXML1/7bd9f2:1090cd59caec:-7f9c-body1"/>
    </eb:Manifest>
  </soapenv:Body>
</soapenv:Envelope>
```
Alternate Version of Order Response Sent for an OrderCreate:

**Note: This document contains a different value for the ConvID and contains an alternate version of the timestamp (with miliseconds).

```xml
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
<soap:Header>
<eb:MessageHeader eb:version="2.0" soap:mustUnderstand="1">
<eb:From>
<eb:PartyId eb:type="GLN">878542765</eb:PartyId>
</eb:From>
<eb:To>
<eb:PartyId eb:type="GLN">0576403930000</eb:PartyId>
</eb:To>
<eb:CPAId>Order</eb:CPAId>
<eb:ConversationId>srv252_1386000_2007091512243106</eb:ConversationId>
<eb:Service>uri:Order</eb:Service>
<eb:Action>OrderResponseEnd</eb:Action>
<eb:MessageData>
<eb:MessageId>0ah5gm0v17frrier0000050cq</eb:MessageId>
<eb:Timestamp>2006-03-09T15:56:31.975Z</eb:Timestamp>
</eb:MessageData>
<eb:DuplicateElimination/>
<eb:AckRequested eb:version="2.0" soap:mustUnderstand="1" soap:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH" eb:signed="false"/>
</eb:MessageHeader>
<soap:Header>
<soap:Body>
<eb:Manifest eb:version="2.0">
<eb:Reference eb:id="OrderResponse-0ah5gm0v17frrier0000050cp" xlink:type="simple" xlink:href="cid:OrderResponse-0ah5gm0v17frrier0000050cp"/>
</eb:Manifest>
</soap:Body>
</soap:Envelope>
```
Technical Response Sent After Receiving Order Response:

<?xml version="1.0" encoding="UTF-8"?>
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Header>
    <eb:MessageHeader eb:version="2.0" soapenv:mustUnderstand="1" xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
      <eb:From>
        <eb:PartyId eb:type="GLN">0576403930000</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="GLN">9666166330000</eb:PartyId>
      </eb:To>
      <eb:CPAId>Order</eb:CPAId>
      <eb:ConversationId>srv252_1386000_2007091512243106</eb:ConversationId>
      <eb:Service>uri:www.ebxml.org/messageService/</eb:Service>
      <eb:Action>Acknowledgment</eb:Action>
      <eb:MessageData>
        <eb:MessageId>srv252/1386000:1090cd58170:-7f97</eb:MessageId>
        <eb:RefToMessageId>ALFTXML1/7bd9f2:1090cd59cae:-7f9c</eb:RefToMessageId>
      </eb:MessageData>
    </eb:MessageHeader>
    <eb:Acknowledgment eb:version="2.0" soapenv:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH" soapenv:mustUnderstand="1" xmlns="">
      <eb:From>
        <eb:PartyId eb:type="GLN">0576403930000</eb:PartyId>
      </eb:From>
      <eb:RefToMessageId>ALFTXML1/7bd9f2:1090cd59cae:-7f9c</eb:RefToMessageId>
    </eb:Acknowledgment>
  </soapenv:Header>
</soapenv:Envelope>

Incoming Order Create /OrderResponse SMTP Examples:

OrderCreate Envelope:

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
OrderResponse Envelope:

<?xml version="1.0" encoding="UTF-8"?>
  <soapenv:Header>
    <eb:MessageHeader eb:version="2.0" soapenv:mustUnderstand="1">
      <eb:From>
        <eb:PartyId eb:type="GLN">057640393</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="GLN">878542765</eb:PartyId>
      </eb:To>
      <eb:CPAId>Order</eb:CPAId>
      <eb:ConversationId>srv252_1386000_2007091512243106</eb:ConversationId>
      <eb:Service>uri:Order</eb:Service>
      <eb:Action>OrderResponse</eb:Action>
      <eb:MessageData>
        <eb:MessageId>dads-computer/1b31fd9:10a2266484a:-7ff4</eb:MessageId>
        <eb:Timestamp>2006-03-22T09:41:57Z</eb:Timestamp>
      </eb:MessageData>
    </eb:MessageHeader>
    <eb:AckRequested eb:version="2.0" eb:signed="false" soapenv:mustUnderstand="1">
    </eb:AckRequested>
  </soapenv:Header>
    <eb:Manifest>
      <eb:Reference xmlns:xlink="http://www.w3.org/1999/xlink" eb:id="Payload-1" xlink:href="dads-computer/1b31fd9:10a2266484a:-7ff4-body1"/>
    </eb:Manifest>
  </soapenv:Body>
</soapenv:Envelope>
xml:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xml:xsd="http://www.w3.org/2001/XMLSchema"
xml:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <eb:MessageHeader eb:version="2.0" soapenv:mustUnderstand="1" xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
      <eb:From>
        <eb:PartyId eb:type="DUNS">878542765</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="DUNS+4">057640393</eb:PartyId>
      </eb:To>
      <eb:CPAId>OrderCreate</eb:CPAId>
      <eb:ConversationId>0ah5gm0v18h72f200003nlv</eb:ConversationId>
      <eb:Service>uri:OrderCreate</eb:Service>
      <eb:Action>SendFileResponse</eb:Action>
      <eb:MessageData>
        <eb:MessageId>0ah5gm0v18h72hk200003nm6</eb:MessageId>
        <eb:Timestamp>2006-03-22T14:52:29Z</eb:Timestamp>
      </eb:MessageData>
      <eb:MessageHeader>
        <eb:AckRequested eb:version="2.0" eb:signed="false" soapenv:mustUnderstand="1" xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>
      </eb:MessageHeader>
    </eb:MessageHeader>
  </soapenv:Header>
    <eb:Manifest soapenv:mustUnderstand="1" eb:version="2.0" xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>
  </soapenv:Body>
</soapenv:Envelope>

Full Sample With Mime Parts (Unencrypted):

Message-ID: <2175801.1142277400870.JavaMail.markus.breilmann@smtp.1und1.com>
Date: Mon, 13 Mar 2006 20:16:40 +0100 (CET)
From: nexus.test@xioma.de
To: nexus.test@xioma.de
Subject: como/1fef80a:109f4eb8b41:-7ffe
Mime-Version: 1.0
Content-Type: multipart/related;
  boundary="----=_Part_0_10964996.1142277400410"
SOAPAction: ebXML

-------- Part_0_10964996.1142277400410
Content-Type: text/xml; charset=UTF-8
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:eb="http://www.ebxml.org/namespaces/messageHeader">
  <soapenv:Header>
    <eb:MessageHeader eb:version="1.0" soapenv:mustUnderstand="1"
      xmlns:eb="http://www.ebxml.org/namespaces/messageHeader">
      <eb:From>
        <eb:PartyId eb:type="Custom">LocalPartner</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId>uri:Xioma</eb:PartyId>
      </eb:To>
      <eb:CPAId>GenericFile</eb:CPAId>
      <eb:ConversationId>como/1fef80a:109f4eb8b41:-7ffe</eb:ConversationId>
      <eb:Service>uri:GenericFile</eb:Service>
      <eb:Action>SendFile</eb:Action>
      <eb:MessageData>
        <eb:MessageId>como/1fef80a:109f4eb8b41:-7ffc</eb:MessageId>
        <eb:MessageData>
          <eb:QualityOfServiceInfo eb:deliverySemantics="OnceAndOnlyOnce"/>
        </eb:MessageData>
        <eb:TraceHeaderList eb:version="1.0" soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
          soapenv:mustUnderstand="1" xmlns:eb="http://www.ebxml.org/namespaces/messageHeader">
          <eb:TraceHeader>
            <eb:Sender>
              <eb:PartyId eb:type="Custom">LocalPartner</eb:PartyId>
            </eb:Sender>
            <eb:Receiver>
              <eb:PartyId>uri:Xioma</eb:PartyId>
            </eb:Receiver>
            <eb:Location>nexus.test@xioma.de</eb:Location>
          </eb:TraceHeader>
          <eb:TraceHeaderList>
            <eb:Via eb:version="1.0" soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
              eb:ackRequested="Unsigned" eb:reliableMessagingMethod="ebXML" soapenv:mustUnderstand="1"
              xmlns:eb="http://www.ebxml.org/namespaces/messageHeader"/>
          </eb:TraceHeaderList>
        </eb:TraceHeaderList>
      </eb:MessageData>
    </eb:MessageHeader>
  </soapenv:Header>
  <soapenv:Body>
    <eb:Manifest soapenv:mustUnderstand="1" eb:version="1.0"
      xmlns:eb="http://www.ebxml.org/namespaces/messageHeader">
      <eb:Reference eb:id="Payload-0" xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:href="como/1fef80a:109f4eb8b41:-7ffc-body1"
        xmlns:eb="http://www.ebxml.org/namespaces/messageHeader"/>
    </eb:Manifest>
  </soapenv:Body>
</soapenv:Envelope>
"POST /NexusE2E/xmlhandler HTTP/1.1"
"SOAPAction: "ebXML"
"Content-Type: multipart/related; type="text/xml"; boundary=MIME_boundary; "
"[0x9]start=a2fb984-d1f8-4387-bfc6-7ccc99425fde_xml-Header"
"User-Agent: Jakarta Commons-HttpClient/3.0.1"
"Host: agm.growmark.com"
"Content-Length: 2937"
Mime-Version: 1.0
Content-Type: multipart/related; type="text/xml"; start="<srv258/30e280:11503c8d2b0:-7f7aebXML-Header>"; boundary="MIME_boundary"

--MIME_boundary
Content-ID: <srv258/30e280:11503c8d2b0:-7f7aebXML-Header>
Content-Type: text/xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:eb="http://www.ebxml.org/namespaces/messageHeader">
<soapenv:Header>
<eb:MessageHeader eb:version="1.0" soapenv:mustUnderstand="1"
xmlns:eb="http://www.ebxml.org/namespaces/messageHeader">
<eb:From>
<eb:PartyId eb:type="GLN">1100011002518</eb:PartyId>
</eb:From>
<eb:To>
<eb:PartyId eb:type="GLN">062924500011</eb:PartyId>
</eb:To>
<eb:CPAId>Invoice</eb:CPAId>
<eb:ConversationId>3619201_030753_20070914123745</eb:ConversationId>
<eb:Service>uri:Invoice</eb:Service>
<eb:Action>Invoice</eb:Action>
<eb:MessageData>
<eb:MessageId>srv258/30e280:11503c8d2b0:-7f7a</eb:MessageId>
<eb:Timestamp>2007-09-14T12:37:45Z</eb:Timestamp>
</eb:MessageData>
<eb:QualityOfServiceInfo eb:deliverySemantics="OnceAndOnlyOnce"/>
</eb:MessageHeader>
<eb:TraceHeaderList eb:version="1.0" soapenv:mustUnderstand="1"
xmlns:eb="http://www.ebxml.org/namespaces/messageHeader">
<eb:TraceHeader>
<eb:SenderId>
<eb:PartyId eb:type="GLN">1100011002518</eb:PartyId>
</eb:SenderId>
<eb:Location>https://agm.growmark.com:443/NexusE2E/xmlhandler</eb:Location>
</eb:SenderId>
<eb:SenderId>
<eb:PartyId eb:type="GLN">062924500011</eb:PartyId>
</eb:SenderId>
<eb:Location>https://agmtest.growmark.com:443/NEXUSe2e/handler/ebxml10</eb:Location>
<eb:Receiver/>
<eb:Timestamp>2007-09-14T12:37:45Z</eb:Timestamp>
<eb:TraceHeader/>
<eb:TraceHeaderList/>
<eb:Via eb:version="1.0" soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
    eb:ackRequested="Unsigned" eb:reliableMessagingMethod="ebXML" soapenv:mustUnderstand="1"
    xmlns:eb="http://www.ebxml.org/namespaces/messageHeader"/>
</soapenv:Header>
<soapenv:Body>
    <eb:Manifest soapenv:mustUnderstand="1" eb:version="1.0"
        xmlns:eb="http://www.ebxml.org/namespaces/messageHeader">
        <eb:Reference eb:id="Payload-0" xmlns:xlink="http://www.w3.org/1999/xlink"
            xlink:href="srv258/30e280:11503c8d2b0:-7f7a-body1"
            xmlns:eb="http://www.ebxml.org/namespaces/messageHeader"/>
    </eb:Manifest>
</soapenv:Body>
</soapenv:Envelope>
Outgoing PriceSheet:

```xml
    xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
  <SOAP-ENV:Header>
    <eb:MessageHeader SOAP-ENV:mustUnderstand="1" eb:version="2.0">
      <eb:From>
        <eb:PartyId eb:type="DUNS+4">9876543210000</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="DUNS+4">1234567890000</eb:PartyId>
      </eb:To>
      <eb:CPAId>PriceSheet</eb:CPAId>
      <eb:ConversationId>qw12356yt-bb23-4537-8e8c-35643efg9232</eb:ConversationId>
      <eb:Service>uri:PriceSheet</eb:Service>
      <eb:Action>PriceSheet</eb:Action>
      <eb:MessageData>
        <eb:MessageId>998a8b0e-xc34-456c-9408-013253624368</eb:MessageId>
        <eb:Timestamp>2010-01-30T17:00:25Z</eb:Timestamp>
      </eb:MessageData>
    </eb:MessageHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <eb:Manifest eb:id="Manifest" eb:version="2.0">
      <eb:Reference eb:id="Payload-1" xlink:href="998a8b0e-xc34-456c-9408-013253624368-body1"
        xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink" />
    </eb:Manifest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```