ebMS Interoperability Test-3Q03 Report for the OASIS ebXML Messaging Services Committee

December 18, 2003

Sponsored by:

Uniform **C**ode **C**ouncil, **I**nc. (UCC)
www.uc-council.org

HIPAA Conformance **C**ertification **O**rganization (HCCO)
www.hcco.us

Prepared & Facilitated By:
DRUMMOND GROUP INC.
www.drummondgroup.com

Drummond group

Table of Contents

Introduction

Drummond Group Inc. (DGI) recently completed its third round of Interoperability testing for the OASIS ebXML Message Service (ebMS) specification version 2.

This report is a formal communication to the OASIS committee, intended to inform the members about features of the standard that are being implemented, the level of interoperability that has been gained, issues found with the specification and consensus items that have been reached during the testing that were required to enable and prove interoperability between the participants.

We hope that this vendor-neutral information is useful in understanding how version 2 is being implemented, and may serve as helpful background during the definition of version 3 of the standard.

The majority of the information in this report is drawn from the Final Report on the testing, which can be viewed on the eBusinessReady website, www.ebusinessready.org.

Brief History of Testing

The first round of ebMS testing (4Q01) covered all ebMS required features, including message packaging and digital signatures. Additional testing included reliable messaging features, multiple attachments, messaging over http with ssl, optional testing of SMIME encryption, optional testing of extended features and informal testing of error scenarios.

The second round of ebMS testing (3Q02) expanded error testing and added SMTP testing. All participants tested directly with the Centers for Disease Control and Prevention (CDC) PHIN implementation of ebMS, and two participants executed optional testing of XMLEncryption and Client Authentication with the CDC.

The Basic Profile and Industry Specific Profiles

In this third round of ebMS testing (3Q03), DGI has defined an approach to accelerate adoption of ebMS by multiple industries. A Basic ebMS Profile has been defined, which includes all the required ebMS features and several optional features including reliable messaging, ping/pong and message status. This Basic Profile is intended to reflect a common base of interoperability, horizontal over any industry, which includes the features most commonly implemented by vendors and end users. Participants are required to successfully execute all Basic Profile tests. Please see Appendix A for a listing of the Basic Profile test suite.

Where there is market demand, DGI will describe industry-specific ebMS profiles, above and beyond the Basic Profile, that comprise subsets or supersets of the ebMS standard, allowing vendors to prove interoperability over feature sets important to specific industries. Participants may choose to optionally participate in industry-specific profile tests; the results of all industry-specific tests will be formally reported.

The CDC PHIN Profile

During the ebMS-3Q03 test round, six participants executed an industry profile of ebMS as prescribed by the CDC for its PHIN (Public Health Information Network). The PHIN architecture is comprised of several standards, including HL7 data formats, and relies on ebMS as the standard message service transport layer. The two key features of the ebMS PHIN Profile are:

- The ability to enable SSL Client Authentication
- Encryption of payloads using the XMLEncryption standard

Please see Appendix B for a listing of the CDC PHIN Profile test suite.

Test Requirements

Trading Partner Requirements

All participants were required to establish trading partner relationships with each other. Each participant provided X.509 digital certificates. Some participants generated their own certificates and others acquired them from well-known third-party Certificate Authorities. Some participants chose to use separate certificates for Digital Signature, XMLEncryption and SSL and others used one certificate for all forms of security.

Participants were responsible for distributing network information and configuring firewalls for access.

Technical Requirements – Basic Profile

Each participant successfully sent and received all tests cases in the Basic Profile with each and every other participant, with the exception of the single Large Message test and the Error Tests. These Basic Profile test cases, which can be found in the Appendix A, cover the core requirements of the ebMS standard and include some optional features of ebMS that are widely implemented and or desired by end users. In summary, these features were tested:

- Message Packaging
- Digital Signature
- Error Handling
- Synchronous and Asynchronous messaging
- Synchronous and Asynchronous Acknowledgments of Receipt
- HTTP and HTTP/S Transfer Protocol
- Single and Multiple Payloads, including XML, EDIFACT, X12 and JPEG
- Large Messages of approximately 50 megabytes
- Reliable Messaging
  - Acknowledgment of receipt
  - Senders ability to retry failed messages
  - Receivers ability to detect and ignore duplicate messages
  - Receivers ability to store messages
- Message Status
- Ping/Pong

Technical Requirements – CDC PHIN Profile

During the round, six participants successfully executed three tests designed to test interoperability of features required by the CDC for its PHIN architecture. For additional technical information regarding the relationship between ebMS, please see:

http://www.cdc.gov/phin/messaging/index.htm

http://www.cdc.gov/phin/components

The three key features tested were:

- SSL Client Authentication
- XML Encryption combined with SSL Client Authentication
- Digital signature combined with XML Encryption with Client Authentication

Consensus Items

During the first three ebMS interoperability rounds, several issues arose that required consensus to achieve interoperability. Some of these items are outside the scope of the ebMS version 2 specifications and may be related to underlying technical specifications such as MIME. Also, some of these issues address features that are unclear in the ebMS version 2 specifications.

DGI will follow these conventions and expects participants to read, understand and provide input and feedback on these issues as appropriate.

—Synchronous Messages and the SyncReply element
The majority of tests in the Basic Profile are asynchronous, reflecting DGI's view that the market perceives asynchronous messaging as scalable and appropriate for B2B messaging. The tests that require synchronous messages utilize the ebMS SyncReply element and assume an ebMS SyncReplyMode of mshSignalsOnly.

—SOAP Action HTTP Header in Sync Reply responses
Message Handlers should tolerate synchronous replies that contain a SOAPAction header and synchronous replies that do not contain a SOAPAction header. ebMS 2 is unclear on which is preferred.

—The SOAPAction value must be "ebXML" with the quotation marks
ebMS version 2 recommends the value be "ebXML," but does not directly address if it must be surrounded by double quotes.

—Format for Messages without payloads
Message Handlers should tolerate messages that have no payloads sent as multipart/related, and tolerate messages that have no payloads which are sent as plain SOAP format or text/xml. The ebMS 2 specification is not completely clear on this issue, but does provide examples of both.

—SOAP Faults
Message Handlers should tolerate SOAP Faults sent as HTTP 500 responses or sent as HTTP 200 responses, or sent as separate posts. It has been found that many products in the field generate SOAP Faults in the form of HTTP 500 responses which causes some application server or web server products to ignore data in the HTTP body, resulting in a special case that needs to be understood and taken into account.

—CPA Exchange
Individual participants may optionally exchange CPAs. The exchange of CPAs will not be tested.

—Role Element value
If the Role element is present, as contained in either ToParty or FromParty elements, its value must be agreed upon by the two testing parties. The ebMS version 2.0 specifications are unclear on how to set this value. Recently, an addendum to ebXML CPA version 2.0 has specified that the value for the Role element should be the same as the Role/@name value in the CPA.

—cid based start parameter in the MIME Content-Type header
The start parameter of the MIME Content-Type header may be a cid (content-id) style reference. It may contain a prefix of “cid:” which should be stripped to obtain the Content-ID value. For example,

SOAPAction: "ebXML"
Content-Type: Multipart/Related; type="text/xml";
boundary=”MIMEBoundary”; start="cid:myContentIDHeader"
--MIMEBoundary
Content-Type: text/xml
Content-ID: <myContentIDHeader>

—MIME Multipart/Related header, case sensitivity
ebMS version 2.0 specification requires a MIME Content-Type: Multipart/Related header to appear as an HTTP header. Consensus is that Message Handlers will tolerate mixed case in the phrase Multipart/Related. In other words, these 2 phrases are both valid “Multipart/Related” and “multipart/related.”

ebMS 2 does not directly address this issue, but the consensus was that underlying MIME specifications require case insensitivity for this header.

—Service and Action values
All tests use the same value for Service and different values for Action. This consensus is intended to reflect changes in CPA version 2 and to allow ease of use and interoperability between participants using CPA 1, participants using CPA 2 and participants not using CPA.

—XML DSIG KeyInfo
The KeyInfo element related to Digital Signatures is optional, if it is present it may be ignored. In other words, the assumption is that Digital Certificates used to sign messages will be exchanged out of band and will be known beforehand by all participants.

—XML DSIG Namespace
The attribute that declares the XML Digital Signature namespace must be at the Signature element level, for example:

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    .
</ds:Signature>
```

—Payloads are not canonicalized during the digital signing process.
ebMS 2 states that transformations applied to payloads are “implementation dependent.” This consensus allows for interoperable validation of signature digests. Some past participants have pointed out that C14 canonicalization removes comments, which could result in security or integrity issues.

—Payloads are not canonicalized for Digital Signature / XMLEncryption
When combining XMLEncryption with Digital Signature, payloads will be treated at all times during processing as simple bytestreams, even if the payload is XML being encrypted with XMLEncryption. This follows the above consensus that Payloads are not canonicalized for Signature. This consensus has implications on the way security toolkits are configured and used. Some security toolkits will attempt to canonicalize XML data by default, and must be configured to treat an XML Payload as a simple byte stream.

—Sign first, Encrypt second
For the purposes of the CDC PHIN profile, participants are required to apply Digital Signature first and XMLEncryption of the XML Payload second.

—References to original message MUST be included in a signed Acknowledgment
This consensus was gained by a majority of past participants and is supported by ebMS 2 specification which states “if you support signed acknowledgments, it is required that you include references to the original message digests.” A further consensus was reached that signed acknowledgments should include these References, even if the original message itself was not signed.

—Content-ID and MessageID MUST conform to MIME and include @
ebMS 2 does not address this issue directly, but the consensus is that the underlying MIME specifications do. The exact consensus is that these ID elements should be formatted in this fashion to comply with MIME specifications, but that receivers should act liberally, and not reject a message solely based on Content-ID or MessageID not containing an @.

—Error Messages should not be signed
Error Messages should be sent in the clear, to avoid the possibility that the Error is related to signing processes. Specifically, when a signed acknowledgment is requested, and an Error message is generated in reply, that Error message should not be signed.

—SyncReply element may appear in a synchronous reply
ebMS 2 does not forbid the SyncReply element from appearing in a synchronous reply. General discussion has been that there may be some use case scenarios where this is useful and a receiving Message Handler (the MSH receiving the response) should allow SyncReply element in an HTTP reply.

—Timestamp may differ in acknowledgments to duplicate requests
ebMS 2 is not completely clear that acknowledgments to duplicate requests should be exact copies of the original acknowledgement message. This consensus was reached after finding that many vendor implementations allow the Timestamp value to differ in the acknowledgments.

—Timestamp may differ in retry messages
ebMS 2 states that a message retry should be a resend of “the original message” It is not clear if ebMS 2 requires that the message Timestamps cannot change. Discussion around this issue is that MessageID is the primary element used to detect duplicate messages, and using real-time timestamps (as opposed to repeating an earlier timestamp) is useful for auditing and identifying message replay attacks.
Consensus is that the Timestamp element of a message sent as a retry may differ from the Timestamp element contained in the original message.

—Content-type for XML Encryption
The Content-Type MIME header value for XML Attachments, for the CDC PHIN Profile can be either text/xml or application/xml. In other words, the Content-Type should have these values, even if the payload is XML encrypted with the XMLEncryption standard. Products should gracefully handle a message that uses one of these two values.

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

or
--MIMEBoundary
Content-type: text/xml

During the ebMS-3Q03 test round, this topic was discussed. An alternative suggestion was to use the value application/xenc+xml. Investigation into this MIME type revealed that it has been registered, but not yet accepted formally as a valid MIME type.

—XMLDSIG Namespace declaration must be at Signature element level
The attribute that declares the XML Digital Signature namespace must be at the Signature element level, for example:

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
…
</ds:Signature>
```

It is not unreasonable to include this namespace declaration at the SOAP envelope level, but there appears to be an overwhelming best practice to specify it at the Signature element level. During the ebMS-3Q03 test round, at least one security toolkit was found not to process XML Digital Signature without its presence.
The XML Digital Signature specification lists a non-normative DTD that describes the declaration at the Signature level. The XML Digital Signature normative XML Schema however, does not describe this format.
All Digital Signature examples from ebMS version 2 specifications have the declaration at signature level, an informal survey of examples from the web revealed every available example placing the declaration at the signature level.

—ConversationIDs must be unique over CPAId
ConversationID must be unique for non-long running conversations under a specific CPAid. If the ConversationID is duplicated, the implementation MAY respond with an error.

—KeyName must be X509 Distinguished Name format
For the purposes of the CDC PHIN profile, when using XMLEncryption the value of the KeyName element must be a valid X509 Distinguished name.
This is simply a best practice convention used during the test to enable the lookup of the related key, and is the format prescribed by CDC PHIN architecture. In practice, in the field systems may sometimes choose to use “alias” values other than Distinguished Name that are defined and agreed upon by the two parties.

Frequently Found Problems

The most frequently found problems with products new to the test process are related to the format of MIME and HTTP headers. Please review the consensus items related to these issues:

- SOAP Action HTTP Header in Sync Reply responses
- The SOAPAction value must be "ebXML" with the quotation marks
- SOAP Faults
- cid based start parameter in the MIME Content-Type header
- MIME Multipart/Related header, case sensitivity
- Content-ID and MessageID MUST conform to MIME and include @
- Content-type for XML Encryption

Another frequently found problem is documented in consensus items under:

- Role Element value

Interoperability Issues during the test

Below are additional discussions of interoperability issues found during the ebMS-3Q03 test round. Some of these discussions are intended to provide more background information about a Consensus Item, some of these are issues that were resolved without the need to document a consensus item and some issues were not completely resolved and are documented here to continue discussion. These details are also provided in a formal report to the OASIS ebXML Message Services committee and the CDC.

XMLEncryption Default RSA Padding mode

During the 3Q03 ebMS tests of the CDC PHIN profile, the group encountered problems when combining the XSS4j security toolkit with the BouncyCastle JCE provider. The issue turned out to be that when XSS4j instantiates an encryption cipher, it asks for the default padding mode. BouncyCastle appears to use a different default padding mode than other JCE providers. The result is that data encrypted with BouncyCastle as the JCE provider using the default mode cannot be decrypted by other toolkits using the default mode.

During the test round, one participant coded a workaround for XSS4j. It effectively changes the instantiation so that a specific padding mode is required. For example:

```
Cipher cipher2 = Cipher.getInstance("RSA/ECB/PKCS1Padding", "BC");
```

As opposed to using the default padding mode with this style of invocation

```
Cipher cipher2 = Cipher.getInstance("RSA ", "BC");
```

Contacts from both the XSS4j team and the BouncyCastle team have been informed of this finding.

XMLSchema Instance declaration missing

At least one participant was formatting messages without including an XMLSchema instance declaration as below

```
<SOAP:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
…
```

Under section 2.2.2 of ebMS version 2, it is strongly recommended that the declaration is present. It is possible that its absence will cause receivers to be unable to parse and validate an inbound message. Participants voluntarily checked and updated implementations to follow the specification recommendation and include the declaration.

More on the consensus on XML DSIG KeyInfo

Currently DGI ebMS test has a consensus that “The KeyInfo element related to Digital Signatures is optional, if it is present it may be ignored. In other words, the assumption is that Digital Certificates used to sign messages will be exchanged out of band and will be known beforehand by all participants.”

During the ebMS-3Q03 test round, additional complications were found with this consensus. At least one security toolkit forced the use of the Digital Certificate in the KeyInfo element if it was present. In this case, the sender must either not format the KeyInfo element, or make sure that the Digital Certificate passed within the KeyInfo element is the correct Digital Certificate to be used for Digital Signature by the receiving partner.

Different interpretations on the use of ConversationID

During the ebMS-3Q03 test round, several discussions revealed a difference of interpretation on the meaning and use of the ConversationID element.

The ebMS specification requires that ConversationID be present in all messages, and requires that if you implement the optional MessageOrdering feature (which is not tested by DGI) that ConversationID must stay the same over all ordered messages.

The issues were raised during attempts to define and test long-running conversations, some participants’ interpretation were that all messages from the original initiator should contain the same ConversationID, others interpretation was that ConversationIDs cannot be repeated by the original initiator, but may be repeated in replies from the original receiver. Also, one interpretation was that the ConversationID of a MessageStatus request should match the ConversationID of the original message whose status is being queried, while others did not recognize this interpretation or, at the least, did not enforce this behavior.

A consensus item was added that reiterates a statement in the ebMS version 2.0 specifications that for non-long running conversations, the ConversationID should not be repeated, and that ConversationIDs are intended to be unique over CPAId.

Different interpretations of duplicate messages and acknowledgments

There continues to be discussion on interpretation of the ebMS version 2.0 specifications related to the correct format for retried or duplicate messages and their associated acknowledgments. The specification states that a retransmitted message should be “a resend of the original message with the same ebXML SOAP Header, Body and Payload containers” but then defines from the receivers’ point of view that a duplicate message is determined only by the fact that the MessageID is the same as a previously received MessageID. Some implementations are taking a liberal interpretation of these requirements and allowing key fields such as Timestamp, MIME Headers and possibly other fields to be different in a retransmitted message versus the original message.

The same discussions have taken place related to retransmitted acknowledgements; some implementations are allowing timestamp to differ, in some implementations only MIME headers differ, in some implementations the MessageID and Timestamps differ.

The DGI ebMS test rounds have tended to define liberal consensus items around these issues.

We would like the ebMS committee to discuss these issues and keep them in mind as ebMS version 3 is being defined, and these same issues will be rehashed in relation to the possible adoption of WS-Reliability. There may be additional complications that have not been shared, including how these requirements affect Signed messages and the interpretation of Timestamp elements. We believe that these specific issues have a great effect on critical architecture decisions, which are often made early in a product cycle, and that clarity on these requirements is essential.

Sign first, Encrypt second

The ebMS test suite has a consensus item stating, “For the purposes of the CDC PHIN profile, participants are required to apply Digital Signature first and XMLEncryption of the XML Payload second.”

This consensus was described because this is the method prescribed by the CDC PHIN architecture and an agreement is required so that participants can configure their systems in the same manner.

There was discussion during the round that as a general principle there may be situations, other than the CDC PHIN Profile, where it makes sense to encrypt first and sign second.

Canonicalization of Payloads

There was in-depth discussion and troubleshooting related to this issue during the ebMS-3Q03 test round. A consensus was formed during the 3Q02 test round that XML Payloads should not be canonicalized during the Digital Signature process. The gist of the argument was that C14N canonicalization will remove comments and other white space like characters, which may result in unintended consequences, including possible security or auditing issues.

The discussion during the 4Q02 round centered on whether or not to employ canonicalization when XMLEncryption is combined with Digital Signature. A consensus was built that payloads will not be canonicalized. This consensus follows the logic of the earlier convention and also is the manner prescribed by CDC PHIN architecture.

There were problems implementing this consensus. When processing Encryption and Signature, some security toolkits will by default attempt to canonicalize or de-canonicalize data that it recognizes as XML with XMLEncryption applied. To overcome these issues, some participants had to physically configure their security toolkits (at both send and receiver implementations) to treat the payloads as simple binary byte streams. In other words, the data was not encoded via DOM or other similar mechanisms, but was treated at all times as a simple byte array.

One specific problem that was found was that if on the receiving side the payload is considered to be XML, the security toolkit will pass the XML data to the end application with the XML prelude declaration

(i.e., "<?xml version="1.0" encoding="UTF-8"?>")

missing, causing the application to view the payload as invalid XML.

Additional Discussion

- Use of CPA (Collaboration Protocol Agreement)

The DGI ebMS Interoperability tests do not require the use of ebXML CPA and allows for testing between implementations where:

- Both partners employ CPA
- Neither partner employs CPA
- One of two partners employs CPA

Although this test is not intended to formally test CPA features, by default the tests exercise the above configurations and some of the tests exercise behavior that must be implemented by CPA or by a CPA like architecture. For example, Test I3 exercises the ability to return an ebMS error declaring a requested feature is “Not Supported,” the ability to recognize this error may be implemented via a CPA based system. Participants may or may not physically use a CPA based system to support these types of tests.

- HTML formatted errors

Per ebMS version 2, message handlers may return SOAP Faults or ebMS error list messages when errors are encountered. Some implementations return html format errors in specific situations. This may be due to the fact that message handlers are often implemented in servlet containers, which by default may return html formatted errors. It is not clear if this is a violation of the specifications or if it was considered during the design of ebMS error handling.

Appendix A: Basic Profile Test Suite

The Table below summarizes the entire Basic Profile Test Suite which was executed by all participants during the Debug phase. Participants executed a representative subset of these tests during the Final Test.

A1	Exchange Certificates			
B1	Simple Transfer	http	async	Small XML
B2	Simple Transfer SSL	https	async	Small XML
C1	Large Message	http	async	Very Large X12
D1	Signed Data	http	async	Small XML
D2	Signed Data SSL	https	async	Small XML
E1	UnSigned with Ack	http	async	Small XML
E2	UnSigned with ACK sync	http	sync	Small XML
E3	Signed Data/UnSigned Ack	http	async	Small XML
E4	Signed Data/ Signed Ack Sync	http	sync	Small XML
E5	Signed Data/Signed Ack SSL	https	async	Small XML
F1	Two Payloads	http	async	Small XML Medium binary jpeg
F2	Five Payloads	http	async	Medium X12 HCCO HIPPA Small EDIFACT Small XML Large XML Medium binary jpeg
F3	Two Payloads Signed Data	http	sync	Small EDIFACT Medium binary jpeg
F4	Five Payloads Signed Data/Ack SSL	https	async	Medium X12 HCCO HIPPA Small EDIFACT Small XML Large XML Medium binary jpeg
G1	Ping Pong	http	sync	none
G2	Ping Pong SSL	https	async	none
G3	Message Status SSL	https	async	none
H1	Once and only once	https	async	Small XML
H2	Duplicate Detection	https	async	Small XML Medium binary jpeg
H3	Long Running Conversation	https	async	Medium X12 HCCO HIPPA Small EDIFACT Medium binary jpeg
I1	SOAP Fault	http	async	Small XML
I2	Value not recognized	http	async	Small XML
I3	Not Supported	http	async	Small XML
I4	Inconsistent sync	http	sync	Small XML
I5	Security Failure	http	async	Small XML
I6	Time to Live expired	http	async	Small XML
I7	Sequence Numbers out of sequence	http	async	Small XML
I8	Message Header format	http	async	Small XML
I9	Missing Payload	http	async	none
I10	Delivery Failure	http	async	Small XML

Appendix B: The CDC PHIN Profile Tests

The Table below summarizes the CDC PHIN Profile Tests which were executed by six participants.

Test	Description	Transfer	Sync/Async	Payload
J1	Client Authentication	https	sync	Small XML
J2	Client Authentication & XML Encryption	https	sync	Small XML
J3	Client Authentication, Digital Signature & XML Encryption	https	sync	Small XML

Drummond group

About Drummond Group Inc.

Drummond Group Inc. (DGI) is an independent, privately held company that works with software vendors, vertical industries and the standards community to drive adoption for standards by conducting interoperability and conformance testing, publishing related strategic research and developing vertical industry strategies. Founded in 1999, DGI represents best-of-breed in the industry on linking horizontal infrastructure technologies, standards and interoperability issues with the needs of vertical industries such as retail, grocery, health care, transportation, government and automotive. For more information, please visit www.drummondgroup.com or email: info@drummondgroup.com.