

ebXML Message Service Handler Implementation Conformance Checklist

Message Service Specification Version 2.0

April 2002

Part I: Core Functionality

All mandatory syntax, features and behavior must be supported.

<i>MSSv2.0 Section Ref.</i>	<i>MSH Behaviour</i>	<i>Level</i>	<i>GoXML-MS</i>
Packaging Specification			
2.1.1	All generated ebXML messages conform to SOAP 1.1 and the SOAP Messages with Attachments specification.	MUST	
2.1.2	All MIME headers generated for each <i>Message Package</i> are in conformance with the SOAP Messages with Attachments Specification.	MUST	
2.1.2	The Content-Type MIME header in the <i>Message Package</i> contains a <code>type</code> attribute of "text/xml"	MUST	
2.1.2	The Content-ID MIME header in any generated <i>Message Package</i> contains a <code>start</code> parameter identifying the first MIME part.	RECOMMENDED	
2.1.2	Non-Multipart SOAP messages are supported in instances where there are no payload sections.	MUST	
2.1.3.1	The MIME Content-Type header for each generated <i>SOAP Message</i> has the value "text/xml"	MUST	
2.1.3.1	The MIME Content-Type header each generated <i>SOAP Message</i> specifies the character set used to generate the message.	MAY	
2.1.3.2	The UTF-8 character set is used by default when encoding each <i>SOAP Message</i> .	RECOMMENDED	
2.1.4	If there is no application payload then there are zero payload MIME parts.	MUST	
2.1.4	The contents of each payload MIME part are identified in the <i>Manifest</i> element within any generated SOAP body.	MUST	
2.1.5	Unrecognised MIME headers in any incoming MIME part are ignored.	MUST	
2.1.6	If a MIME error is present in an incoming ebXML message then it is reported as per the SOAP with Attachments specification.	MUST	
2.2.1	If a generated <i>SOAP Message</i> includes an XML prolog then the prolog specifies the version of XML applied.	MUST	
2.2.2	If a generated <i>SOAP Message</i> includes an XML prolog and the prolog contains an encoding declaration then the encoding declaration matches the <code>charset</code> attribute of the Content-Type MIME header in the <i>Header Container</i> .	MUST	
2.3	All ebXML extension elements included within generated SOAP <i>Envelope</i> , <i>Header</i> and <i>Body</i> elements are namespace qualified to: "http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"	MUST	

MSSv2.0 Section Ref.	MSH Behaviour	Level	GoXML-MS
2.3.2	Generated SOAP <i>Envelope</i> elements include the XMLSchema-instance namespace qualified <i>schemaLocation</i> attribute indicating the extended ebXML envelope schema: "http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd"	RECOMMENDED	
2.3.2	Generated SOAP <i>Header</i> and <i>Body</i> attributes both include a XMLSchema-instance namespace qualified <i>schemaLocation</i> attribute indicating the extended ebXML Msg Header schema "http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"	RECOMMENDED	
2.3.3	A generated SOAP <i>Header</i> element is namespace qualified as per the SOAP namespace declaration in the SOAP <i>Envelope</i> element.	MUST	
2.3.4	A generated SOAP <i>Body</i> element is namespace qualified as per the SOAP namespace declaration in the SOAP <i>Envelope</i> element.	MUST	
2.3.5.1	A generated SOAP <i>Header</i> element always contains one ebXML <i>MessageHeader</i> element.	REQUIRED	
2.3.6	Any foreign namespace qualified elements present within generated ebXML extension elements are namespace qualified with a namespace that is not "http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd".	MUST	
2.3.7	An XML <i>ID</i> attribute is supplied for each generated ebXML element (to assist with such tasks as specifying elements included in a digital signature).	MAY	
2.3.8	A generated ebXML <i>MessageHeader</i> element always contains a <i>version</i> attribute with a value of "2.0". If a message is received with a version that is not recognised by the MSH then the MSH responds with an error.	MUST	
2.3.9	All ebXML extensions of the SOAP <i>Header</i> element (<i>MessageHeader</i> , <i>SyncReply</i> , <i>MessageOrder</i> , ...) contain the <i>mustUnderstand</i> attribute namespace qualified to the SOAP namespace (http://schemas.xmlsoap.org/soap/envelope).	MUST	
2.3.9	Any SOAP <i>Header</i> extension with a <i>mustUnderstand</i> attribute set to "1" that is not understood by the MSH will result in the message being rejected.	MUST	
Core Extension Elements			
3.1.1	<i>From</i> and <i>To</i> elements are always present as the children of a generated <i>MessageHeader</i> element and identify the parties that originated/are-intended-to-receive the message respectively.	MUST	
3.1.1	Generated <i>From</i> and <i>To</i> elements each contain one or more <i>PartyId</i> elements and zero or one <i>Role</i> element.	MUST	
3.1.1	When generated <i>From</i> or <i>To</i> elements contain multiple <i>PartyId</i> elements the <i>PartyId</i> values are treated as identifying the same organisation.	MUST	
3.1.1	If the <i>Role</i> element is generated it always follows the last <i>PartyId</i> element	MUST	
3.1.1.1	If content of the <i>PartyId</i> element on an inbound	SHOULD	

<i>MSSv2.0 Section Ref.</i>	<i>MSH Behaviour</i>	<i>Level</i>	<i>GoXML-MS</i>
	message is not a URI and the element does not contain a <i>type</i> attribute then the MSH responds with an error (Inconsistent/Error).		
3.1.2	If value of the <i>CPAId</i> element on an inbound message cannot be resolved then the MSH responds with an error (NotRecognized/Error).	MUST	
3.1.2	If the receiving MSH detects an inconsistency between an incoming message and the relevant CPA then it responds with an error (Inconsistent/Error)	MUST	
3.1.3	Generated <i>ConversationId</i> element values are strings that are unique within the content of a the specified CPAId.	MUST	
3.1.3	The generated <i>ConversationId</i> will be present in all messages pertaining to the given conversation.	MUST	
3.1.4.1	If the value of the <i>Service</i> element is not a URI and the element's <i>type</i> attribute is not present then the MSH responds with an error (Inconsistent/Error)	MUST	
3.1.5	If the receiving MSH does not recognise both the <i>Service</i> and <i>Action</i> values of an incoming message then it responds with an error (NotRecognised/Error).	MUST	
3.1.6.1	The <i>MessageId</i> element within the <i>MessageData</i> element is always present and its value is generated as a globally unique identifier.	MUST	
3.1.6.2	The <i>Timestamp</i> element within the <i>MessageData</i> element is always present and indicates the time at which the message header was generated. The element value is expressed according to the dateTime XMLSchema datatype in the UTC timezone.	MUST	
3.1.6.3	The <i>RefToMessageId</i> element within the <i>MessageData</i> element, if present, contains the <i>MessageId</i> value of an earlier ebXML Message to which this message relates.	MUST	
3.1.6.3	If there is no earlier related message then the <i>RefToMessageId</i> element is never present.	MUST	
3.1.6.3	For a generated error message the <i>RefToMessageId</i> element is always present with a value indicating the message in error.	MUST	
3.1.6.4	If a <i>TimeToLive</i> element is present within the <i>MessageData</i> element then it represents the time by which the message should be delivered to the <i>To Party MSH</i> . The element value is expressed according to the XML Schema dateTime datatype in the UTC timezone.	MUST	
3.1.6.4	If the MSH receives a message for which it is the <i>To Party MSH</i> and the <i>TimeToLive</i> is greater than the time of the internal clock (adjusted for UTC) then an error message is returned to the <i>To Party MSH</i> (TimeToLiveExpired/Error).	MUST	
3.1.7	If the <i>DuplicateElimination</i> element is present within the <i>MessageHeader</i> of an inbound message then the MSH eliminates duplicate messages.	MUST	
3.1.7	If the relevant CPA specifies a <i>duplicateElimination</i> value of <i>never</i> then the <i>DuplicateElimination</i> element must not be present.	MUST	
3.1.8	When a <i>Description</i> element is generated the	MUST	

<i>MSSv2.0 Section Ref.</i>	<i>MSH Behaviour</i>	<i>Level</i>	<i>GoXML-MS</i>
	language of the free form description is specified via the <i>xml:lang</i> attribute.		
3.2	No payload/application data is present in generated SOAP <i>Body</i> / ebXML <i>Manifest</i> elements.	RECOMMENDED	
3.2.2	If the <i>xlink:href</i> element of a generated <i>Manifest/Reference</i> element specifies a URI via content id ("cid:") then a MIME part with a matching <i>content-id</i> exists in the payload container of the message. If there is no matching payload then an error message is directed to the <i>From Party MSH</i> (MimeProblem/Error).	MUST	
3.2.2	If the <i>xlink:href</i> element of a <i>Manifest/Reference</i> element on an inbound message specifies a URI that is not a content id (not "cid:"), and that cannot be resolved, then the MSH reports an error to the <i>From Party MSH</i> (MimeProblem/Error)	MAY	
3.2.2	If a MIME payload part exists on an incoming message that is not referenced by a <i>Manifest/Reference</i> element then it is discarded.	SHOULD	
Core Modules			
4.1	Zero or more <i>Signature</i> elements belonging to the XML Signature namespace may be included to digitally sign an outbound message.	MUST	
4.1	If there is more than one <i>Signature</i> element within the SOAP <i>Header</i> then it is the first signature that represents digital signing of the message by the From Party MSH.	MUST	
4.1.2.1	The MSH offers security utility services with regard to message payload items according to the Document Exchange section of the relevant CPA.	MAY	
4.1.2.1	The MSH applies security measures to an ebXML message (as a whole) based upon the Transport section of the relevant CPA.	MUST	
4.1.3	Digital signatures are generated and rendered according the XML Signature specification (XMLDSIG). The <i>SignedInfo</i> element has a <i>CanonicalizationMethod</i> , <i>SignatureMethod</i> and one or more <i>Reference</i> elements.	MUST	
4.1.3	The canonicalization method applied to the data to be signed is Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"	RECOMMENDED	
4.1.3	The <i>SignatureMethod</i> element is present and has an <i>Algorithm</i> attribute on any generated digitally signed message.	MUST	
4.1.3	The value of the <i>Algorithm</i> attribute is Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"	RECOMMENDED	
4.1.3	The MSH supports the signature algorithm DSA-SHA1.	MUST	
4.1.3	The generated XML Signature <i>Reference</i> element has a URI attribute value of "" (indicating that the signature is to be applied to the document that contains the <i>Signature</i> element).	SHALL	
4.1.3	The MSH supports the optional addition of the informative <i>Type</i> attribute with value "http://www.w3.org/2000/09/xmldsig#Object" on the XML Signature <i>Reference</i> element.	SHALL	

MSSv2.0 Section Ref.	MSH Behaviour	Level	GoXML-MS
4.1.3	Any generated XML Signature <i>Reference</i> element includes a child <i>Transforms</i> element which in turn includes a first <i>Transform</i> element with an <i>Algorithm</i> attribute of value " http://www.w3.org/2000/09/xmldsig#enveloped-signature ".	SHALL	
4.1.3	A second <i>Transform</i> element is generated with the requisite <i>XPath</i> element excluding all elements with SOAP actor attributes targeting the <i>nextMSH</i> or next SOAP node.	MUST	
4.1.3	The MSH when acting in an intermediary role does not change, format or in any way modify any element not targeted at that intermediary MSH. The MSH does not add or delete white space.	MUST	
4.1.3	The last generated <i>Transform</i> element has an <i>Algorithm</i> attribute with a value of " http://www.w3.org/TR/2001/REC-xml-c14n-20010315 ".	SHOULD	
4.1.3	Any payload data requiring digital signature is identified by an XML Signature <i>Reference</i> element that has a <i>URI</i> attribute resolving to the location of that data.	MUST	
4.1.3	The value of the <i>URI</i> attribute of a generated XML Signature <i>Reference</i> element matches the <i>xlink:href</i> URI value present in the <i>Manifest/Reference</i> element corresponding to that same payload.	RECOMMENDED	
4.1.3	Signature generation takes place before any transfer encoding (eg base64) is applied to the SOAP Envelope or payload MIME parts.	MUST	
4.1.3.2	A digitally signed inbound message may be acknowledged with a digitally signed acknowledgement. Any such acknowledgement message contains an XML Signature <i>Reference</i> element list corresponding to the <i>Reference</i> elements contained in the original message.	MUST	
4.1.4.3	The communication channel used to transport the ebXML message can be used to provide uni or bi-directional party authentication (eg TLS over TCP/IP).	MAY	
4.1.4.4	The communication channel used to transport the ebXML message can be used to provide data integrity of the message content (eg TLS over TCP/IP).	MAY	
4.1.4.5	If signature and encryption of a message component is requested of the MSH, signing takes place prior to encryption.	MUST	
4.1.4.6	The communication channel used to transport the ebXML message can be used to provide data confidentiality for the message content (eg TLS over TCP/IP).	MUST	
4.1.4.8	The source of an ebXML message can be authorised by using a secure network protocol for bilateral authentication of certificates prior to establishing a session (eg TLS over TCP/IP).	MAY	
4.2	When sending messages the MSH can accept and process SOAP Fault values from a downstream SOAP processor.	MUST	
4.2	If an MSH returns a SOAP Fault message to the	MUST	

<i>MSSv2.0 Section Ref.</i>	<i>MSH Behaviour</i>	<i>Level</i>	<i>GoXML-MS</i>
	sender of a SOAP message then this returned message conforms to the SOAP specification guidelines for SOAP Fault values.		
4.2	When an ebXML message is reporting an error with a <i>highestSeverity</i> value of 'Warning' it is not reported or returned as a SOAP Fault.	SHALL NOT	
4.2.2	Errors associated with data communications protocols are detected and reported using the standard mechanisms supported by that protocol and do not use ebXML reporting mechanisms.	SHALL NOT	
4.2.3	The <i>ErrorList</i> extension element of the SOAP <i>Header</i> element is never present if there are no errors to be reported.	MUST NOT	
4.2.3.1	The <i>highestSeverity</i> attribute contains the highest severity of any <i>Error</i> elements generated in an outbound message.	MUST	
4.2.3.2.2	The <i>codeContext</i> attribute of any generated <i>Error</i> element is always a URI.	MUST	
4.2.3.2.2	The namespace/scheme specified by <i>codeContext</i> for identifying errorCodes is the default value of urn:oasis:names:tc:ebxml-msg:service:errors .	RECOMMENDED	
4.2.3.2.3	For each <i>Error</i> element the <i>errorCode</i> attribute is present and indicates the nature of the error.	REQUIRED	
4.2.3.2.4	For each <i>Error</i> element generated the <i>severity</i> attribute has the value of Warning or Error indicating the severity of the error.	REQUIRED	
4.2.3.2.5	If an error exists in an ebXML element and the containing document is well-formed then the <i>location</i> attribute of the Error element is an XPointer to the erroneous element.	MUST	
4.2.3.2.5	If an error exists in a payload MIME part then the <i>location</i> attribute of the generated <i>Error</i> element contains the <code>content-id</code> (via " cid: ") of the erroneous MIME part.	MUST	
4.2.3.4	The "Short Description" text for each error code provided by the Message Service Specification does not appear in any relevant <i>Error</i> element.	MUST NOT	
4.2.4.1	When an MSH detects an error in a message the error is reported to the MSH that sent the original message in error.	RECOMMENDED	
4.2.4.1	If the error reporting location cannot be found or the message in error has an <i>ErrorList</i> element with <i>highestSeverity</i> set to Error the error is: * Logged; * Resolved by other means; and, * No further action is taken.	RECOMMENDED	
4.2.4.2	If the <i>ErrorURI</i> is implied in the relevant CPA then this is used as the Error Reporting Location.	SHOULD	
4.2.4.2	If the <i>ErrorURI</i> is unavailable in the relevant CPA then a URI specified in the <i>From Party</i> of the message is used as the Error Reporting Location.	MAY	
4.2.4.3	If an <i>ErrorList</i> is included as part of a message being sent as a result of processing an earlier message and the <i>Service</i> and <i>Action</i> values are set as specified by the service enacted, then <i>highestSeverity</i> will not be Error.	MUST	
4.2.4.3	If an <i>ErrorList</i> is included as part of an independent message then the values of <i>Service</i> and <i>Action</i> are;	MUST	

<i>MSSv2.0 Section Ref.</i>	<i>MSH Behaviour</i>	<i>Level</i>	<i>GoXML-MS</i>
	Service: urn:oasis:names:tc:ebxml-msg:service Action: MessageError		
4.3.1	If a <i>SyncReply</i> element is present in a message received over a synchronous communications protocol then that connection is kept open in expectation of the response message using the same connection.	SHOULD	
4.3.1	The <i>SyncReply</i> element may not be used to override the <i>syncReplyMode</i> in the CPA.	MUST NOT	
4.3.1	If the CPA <i>syncReplyMode</i> is set to none and a <i>SyncReply</i> element is present in an inbound message then the MSH issues an error (Inconsistent/Error).	SHOULD	

Part II: Additional Features

All mandatory syntax, features and behavior must be supported for any modules that the MSH implementation has chosen to implement.

<i>MSSv2.0 Section Ref.</i>	<i>MSH Behaviour</i>	<i>Level</i>	<i>GoXML-MS</i>
Reliable Messaging Module			
6	Failure to receive an Acknowledgment message from a receiving MSH by the sending MSH results in successive retries until an Acknowledgment is received or a predetermined number of retries is exceeded.	MAY	
6	If the predetermined number of retries is exceeded the sending MSH notifies the <i>From Party</i> of the probable delivery failure.	MUST	
6.1	All messages sent or received reliably are kept in persistent storage.	MUST	
6.1	After a system failure the MSH ensures that all messages in persistent storage are processed as if the system failure or interruption had not occurred.	MUST	
6.1	The <i>MessageId</i> of any received message is recorded in persistent storage.	MUST	
6.1	A received message is recorded in its entirety at least until the information in the message has been passed to the application needing to process it.	RECOMMENDED	
6.1	The time at which a message is received is recorded in persistent storage.	RECOMMENDED	
6.1	Each response message is stored in its entirety in persistent storage.	RECOMMENDED	
6.3.1	An acknowledgment can be requested from a downstream MSH by inserting the <i>AckRequested</i> element in the SOAP <i>Header</i> of an outbound message.	MUST	
6.3.1	If there are two <i>AckRequested</i> elements in a generated message <i>Header</i> then they do not specify the same value for their respective SOAP <i>actor</i> attributes.	MUST	
6.3.1	At most one <i>AckRequested</i> element may be targeted at the actor URI for the <i>Next MSH</i> and at most one <i>AckRequested</i> element may be targeted at the actor URI for the <i>To Party MSH</i> in a given message.	MUST	

MSSv2.0 Section Ref.	MSH Behaviour	Level	GoXML-MS
6.3.1.1	The <i>AckRequested</i> element can only be targeted at either the <i>Next MSH</i> or the <i>To Party MSH</i> .	MUST	
6.3.1.2	The <i>From Party MSH</i> may explicitly request that an acknowledgment message be either signed or unsigned by setting the <i>signed</i> attribute of the <i>AckRequested</i> element appropriately.	MUST	
6.3.1.2	Before setting the value of the <i>signed</i> attribute on an outgoing message the <i>Sending MSH</i> checks if the <i>Receiving MSH</i> supports acknowledgment messages of the type requested.	SHOULD	
6.3.1.2	If an <i>Acknowledgment Message</i> of the type requested on an inbound message can be produced then a message containing an <i>Acknowledgment</i> element is returned to the <i>Sending MSH</i> .	MUST	
6.3.1.2	If an <i>Acknowledgment Message</i> of the type requested cannot be produced then an error is reported to the <i>Sending MSH</i> . The error is Inconsistent/Error if the request is inconsistent with the relevant CPA, or Inconsistent/Warning if the mode is not supported.	MUST	
6.3.1.3	If an <i>Acknowledgment Message</i> is requested of the MSH node acting in the role of <i>To Party</i> then the <i>Acknowledgment</i> element generated is targeted to the MSH node acting in the role of <i>From Party</i> .	MUST	
6.3.1.4	A pure <i>Acknowledgment Message</i> (with no payload) does not include an <i>AckRequested</i> element.	MUST	
6.3.1.4	Any generated error message will not include an <i>AckRequested</i> element.	MUST	
6.3.2.1	The SOAP <i>actor</i> attribute in a generated <i>Acknowledgment</i> element is either not specified or has a value corresponding to the <i>AckRequested</i> element of the message being acknowledged.	SHALL	
6.3.2.2	The <i>Timestamp</i> element is present within any generated <i>Acknowledgment</i> element. The value is in XML Schema dateTime format in the UTC timezone and represents the time at which the message being acknowledged was received by the MSH generating the <i>Acknowledgment Message</i> .	REQUIRED	
6.3.2.3	The <i>RefToMessageId</i> element contains the <i>MessageId</i> of the message whose delivery is being acknowledged.	REQUIRED	
6.3.2.4	If present, the <i>From</i> element in a generated <i>Acknowledgment</i> element identifies the party sending the <i>Acknowledgment Message</i> .	MUST	
6.3.2.4	If the <i>From</i> element is omitted from the <i>Acknowledgment</i> element on an inbound message then the value of the <i>From</i> element in the <i>MessageHeader</i> is used to identify the party sending the acknowledgment.	MUST	
6.3.2.5	If the message being acknowledged contains an <i>AckRequested</i> element with the <i>signed</i> attribute set to true then one or more <i>Reference</i> elements are included in the generated <i>Acknowledgment</i> element.	MUST	
6.3.2.5	Any <i>Reference</i> elements included in a generated <i>Acknowledgment</i> element are namespace qualified to the XML Signature namespace and conform to the XML Signature specification.	MUST	

MSSv2.0 Section Ref.	MSH Behaviour	Level	GoXML-MS
6.3.2.5	Upon receipt of an end-to-end acknowledgment message the <i>From Party MSH</i> notifies the client application of successful delivery of the referenced message.	MAY	
6.2.2.5	Any subsequent inbound <i>Error</i> or <i>Acknowledgment</i> messages with a <i>RefToMessageId</i> value equal to an already received <i>Acknowledgment Message</i> are ignored.	SHOULD	
6.3.2.7	If no errors were detected in the message received and the <i>Acknowledgment Message</i> is being sent with no payload data then the <i>Service</i> and <i>Action</i> values are: Service - urn:oasis:names:tc:ebxml-msg:service Action - Acknowledgment	MUST	
6.4.1	The <i>DuplicateElimination</i> element can be included in an outbound message to indicate to a <i>Receiving MSH</i> that it must eliminate duplicates.	MUST	
6.4.1	If the value of <i>duplicateElimination</i> in the CPA is never the <i>DuplicateElimination</i> element will not be included in outbound messages.	MUST	
6.4.1	If <i>DuplicateElimination</i> element is present on an inbound message then that message is persisted in a persistent store and presented to the <i>To Party Application</i> at-most-once.	MUST	
6.4.1	If duplicate elimination is not supported or if the value of <i>duplicateElimination</i> in the CPA does not match the implied value of the <i>DuplicateElimination</i> on an inbound message then an error is reported to the <i>From Party</i> (Inconsistent/Error).	SHOULD	
6.4.3	If a message that requested acknowledgment is not acknowledged within the <i>RetryInterval</i> then the message is redelivered up to a maximum number of retries as specified by the <i>Retries</i> parameter in the relevant CPA.	SHOULD	
6.4.4	The minimum time elapsed between re-sends of the same message is dictated by the <i>RetryInterval</i> parameter from the relevant CPA.	SHOULD	
6.4.5	For a reliably delivered message the <i>TimeToLive</i> value specified adheres to the following formula: $TimeToLive > Timestamp + ((Retries + 1) * RetryInterval)$	MUST	
6.4.6	A received reliably sent message is kept in persistent storage for at least the length of time specified by the <i>PersistDuration</i> parameter in the relevant CPA.	MUST	
6.4.6	If the length of time specified by the <i>PersistDuration</i> parameter in the relevant CPA has passed since a message was first sent then a message with the same <i>MessageId</i> will not be sent again.	SHOULD	
6.4.6	If a message cannot be successfully delivered before expiry of the <i>PersistDuration</i> period then a delivery failure is reported.	SHOULD	
6.4.4	The <i>TimeToLive</i> of a reliably sent message is always less than the sum of the message <i>TimeStamp</i> and the <i>PersistDuration</i> period (as defined in the relevant CPA).	MUST	
6.4.7	If the communications protocol is not synchronous	MUST	

<i>MSSv2.0</i> <i>Section Ref.</i>	<i>MSH Behaviour</i>	<i>Level</i>	<i>GoXML-MS</i>
	then the value of the <i>syncReplyMode</i> in the relevant CPA is ignored.		
6.4.7	If the <i>syncReplyMode</i> in the relevant CPA is not none then a <i>SyncReply</i> element is required in an inbound message and the MSH returns any response from the application or business process in the payload of the synchronous reply message.	MUST	
6.5.2	When acting as an intermediary node the MSH does <i>not</i> filter out perceived duplicate messages.	MUST	
6.5.3	An <i>Acknowledgment Message</i> is generated whenever a message is received with an <i>AckRequested</i> element that has a SOAP actor URI targeting the MSH.	MUST	
6.5.3	Any <i>Acknowledgment Message</i> is placed in persistent storage with the same <i>PersistDuration</i> as the original message.	MUST	
6.5.3	An <i>Acknowledgment Message</i> can be delivered as part of the normal response to the received message.	MUST	
6.5.4	If there is a communications protocol error during a message send then the message is resent as if the MSH had not received an <i>Acknowledgment Message</i> .	MUST	
6.5.5	If a duplicate message is received and the original acknowledgment is still present in the persistent store then this original <i>Acknowledgment Message</i> is resent.	SHOULD	
6.5.5	If a duplicate message is received and the original acknowledgment is not present in the persistent store, and the <i>syncReplyMode</i> is set to none , and the CPA indicates that an application response is included, then a response from the application is gathered by the MSH and returned synchronously.	SHOULD	
6.5.5	If a duplicate message is received and the original acknowledgment is not present in the persistent store, and the <i>syncReplyMode</i> is not none , then a new <i>Acknowledgment Message</i> is generated and sent.	SHOULD	
6.5.7	If a message sent with <i>AckRequested</i> element cannot be delivered then an error message is sent to the <i>From Party</i> . If the delivery failure arose because the message could not be transmitted then the reported error is DeliveryFailure/Error. If the message was transmitted however no acknowledgment was received then the reported error is DeliveryFailure/Warning.	SHALL	
6.5.7	If an <i>Error Message</i> with an error code set to DeliveryFailure cannot be delivered successfully then the ultimate destination of the error message is informed of the failure by some undefined means.	MUST	
Message Status Service Module			
7	When a <i>Message Status</i> request message is received that references a previously received message that had been sent reliably and is present in persistent storage then a <i>Message Status Response Message</i> is sent.	SHOULD	
7	When a <i>Message Status Request Message</i> is	MAY	

<i>MSSv2.0 Section Ref.</i>	<i>MSH Behaviour</i>	<i>Level</i>	<i>GoXML-MS</i>
	received that references a previously received message that had not been sent reliably a <i>Message Status Response</i> is sent.		
7	The <i>Message Status Request Service</i> is not being used to implement reliable messaging.	SHOULD	
7	If a <i>Message Status Request Message</i> is received for a service that is not supported then an <i>Error Message</i> is returned (NotSupported/Error).	SHOULD	
7.1.1	A generated <i>Message Status Request Message</i> consists of no payload and the <i>MessageHeader/StatusRequest</i> elements configured as specified in the Message Service Specification.	MUST	
7.1.2	A generated <i>Message Status Response Message</i> consists of no payload and <i>MessageHeader/StatusResponse</i> elements configured as specified in the Message Service Specification.	MUST	
7.1.3	When a <i>Message Status Request Message</i> is received from an party deemed to be unauthorised then a response is sent with the <i>messageStatus</i> attribute set to UnAuthorized .	MAY	
7.2.3	A <i>StatusRequest</i> element is not included along with any of the <i>Manifest</i> , <i>StatusResponse</i> , or <i>ErrorList</i> elements.	MUST	
7.3.1	In a generated <i>Status Response Message</i> the <i>RefToMessageId</i> element child of the <i>MessageData</i> element specifies the <i>MessageId</i> of the message containing the associated <i>StatusRequest</i> element. Conversely, the <i>RefToMessageId</i> element child of the <i>StatusRequest/StatusResponse</i> elements always contains the <i>MessageId</i> of the message whose status is being queried.	MUST	
7.3.2	The <i>Timestamp</i> element child of a <i>StatusResponse</i> element contains the time at which the message being reported on was originally received. This element is omitted in the case where the original message was NotRecognised , or the status request is UnAuthorised .	MUST	
7.3.3	The <i>messageStatus</i> attribute is only set to one of the following values as dicated by the Message Service Specification: UnAuthorized, NotRecognized, Received, Processed, or Forwarded .	MUST	
7.3.5	A <i>StatusResponse</i> element does not get included along with any of the <i>Manifest</i> , <i>StatusRequest</i> , or <i>ErrorList</i> (with highest severity set to Error) elements.	MUST	
Message Service Handler Ping Service Module			
8	If a <i>Message Service Handler Ping Message</i> is received but not supported an <i>Error Message</i> is returned (NotSupported/Error)	SHOULD	
8.1	Any generated <i>Message Service Handler Ping Message</i> consists of no payload and the <i>MessageHeader & Signature</i> elements are configured as specified in the Message Service Specification.	MUST	
8.2	When a <i>Message Service Handler Ping Message</i> is received a <i>Message Service Handler Pong Message</i> is returned.	MAY	
8.2	Any generated <i>Message Service Handler Pong</i>	MUST	

<i>MSSv2.0</i> <i>Section Ref.</i>	<i>MSH Behaviour</i>	<i>Level</i>	<i>GoXML-MS</i>
	<i>Message</i> consists of no payload and the <i>MessageHeader & Signature</i> elements are configured as specified in the Message Service Specification.		
MessageOrder Module			
9	<i>MessageOrder</i> functionality is only used in conjunction with the Reliable Messaging Module configured for Once-And-Only-Once delivery (thereby requiring at least <i>DuplicateElimination</i> and <i>AckRequested</i> directed to the <i>To Party MSH</i>).	MUST	
9.1	When the <i>MessageOrder</i> element is present the <i>DuplicateElimination</i> element is always present and the <i>SyncReply</i> element is never present.	MUST	
9.1.1	When receiving ordered messages the MSH processes messages only in the sequence indicated by the <i>SequenceNumber</i> element.	MUST	
9.1.1	A received ordered message is not passed to the destination application until all messages with a lower (earlier) <i>SequenceNumber</i> have previously been passed.	MUST	
9.1.1	If the maximum number of out-of-sequence ordered messages have been received then the <i>Sending MSH</i> is sent an error (<i>DeliveryFailure/Error</i>).	MUST	
9.1.1	The <i>SequenceNumber</i> element has value of 0 in each of the following cases: * It is the first ordered message from the <i>Sending MSH</i> within the conversation. * It is the first ordered message after a reset instruction is sent by the <i>Sending MSH</i> . * It is the first ordered message after the sequence wrapped at value 99999999.	MUST	
9.1.1	When the <i>SequenceNumber</i> element has been set to a value of 0 for either of the first two reasons noted in the previous requirement the <i>status</i> attribute of the message is set to Reset .	MUST	
9.1.1	In all case where the <i>status</i> attribute is not set to Reset it is set to Continue .	MUST	
9.1.1	When acting as the <i>Sending MSH</i> the <i>SequenceNumber</i> for a conversation is only reset when all previously sent messages have been accounted for.	MUST	
9.2	A <i>MessageOrder</i> element is never included in the same message as a <i>SyncReply</i> element.	MUST	
9.2	If a message is received in which the <i>MessageOrder</i> element is included with a <i>SyncReply</i> element an error is reported (<i>Inconsistent/Error</i>).	SHOULD	
Multi-Hop Module			
10.1	Multi-hop reliable messaging can be used between intermediary MSH nodes by applying the <i>AckRequested</i> and <i>Acknowledgment</i> elements with the SOAP actor attribute set to <i>NextMSH</i> (urn:oasis:names:tc:ebxml-msg:actor:nextMSH).	MAY	
10.1.1	When acting as an intermediary the node removes any <i>AckRequested</i> element with a SOAP actor attribute of <i>NextMSH</i> .	MUST	
10.1.1	When acting as an intermediary the node can insert a single <i>AckRequested</i> element with a SOAP actor	MAY	

<i>MSSv2.0 Section Ref.</i>	<i>MSH Behaviour</i>	<i>Level</i>	<i>GoXML-MS</i>
	attribute of <i>NextMSH</i> .		
10.1.1	There are no situations in which two <i>AckRequested</i> elements are generated in the same message with a SOAP <i>actor</i> attribute value targetting the <i>NextMSH</i> .	MUST	
10.1.1	If a <i>SyncReply</i> element is present in a message then an <i>AckRequested</i> element with SOAP <i>actor</i> attribute targetting the <i>NextMSH</i> is never included.	MUST	
10.1.1	If the <i>SyncReply</i> and <i>AckRequested</i> elements are received in the same message then an error is reported (Inconsistent/Error).	MUST	
10.1.1	When acting in the role of intermediary a node may synchronously return an intermediate <i>Acknowledgment Message</i> to the <i>Sending MSH</i> if no <i>SyncReply</i> element is specified.	MAY	
10.1.3	If an inbound message contains two <i>AckRequested</i> elements (one addressing <i>NextMSH</i> , one addressing <i>ToPartyMSH</i>), and the MSH node is in the combined role of Next and To Party MSH, then that node is able to differentiate the acknowledgment requests based upon the <i>actor</i> attribute and then send acknowledgments as applicable.	MUST	
10.1.3	A reliable message received by an MSH node in the role of intermediary is not acknowledged until the message is both persisted and delivered to the <i>Next MSH</i> .	MUST	
10.1.4	When a signed <i>Acknowledgment Message</i> is requested by an intermediate node it is only generated as a standalone message and is not bundled with any other data (payload).	MUST	
10.2	When acting in the role of intermediary the MSH does not attempt to participate in Message Order processing.	MUST	