



1 **Web Services Reliable Messaging**
2 **(WS-Reliable Messaging)**

3 **Working Draft 01, August 16th 2005**

4 **Document identifier:**

5 WS-ReliableMessaging-1.0draft-01.doc

6 **Location:**

7 <http://docs.oasis-open.org/ws-rx/2005/07/WS-ReliableMessaging-1.0-draft-01.doc>

8 **Editors:**

9 Doug Davis, IBM <dug@us.ibm.com>

10 TBD

11 **Abstract:**

12 This specification (WS-ReliableMessaging) describes a protocol that allows messages
13 to be delivered reliably between distributed applications in the presence of software
14 component, system, or network failures. The protocol is described in this
15 specification in a transport-independent manner allowing it to be implemented using
16 different network technologies. To support interoperable Web services, a SOAP
17 binding is defined within this specification.

18 The protocol defined in this specification depends upon other Web services
19 specifications for the identification of service endpoint addresses and policies. How
20 these are identified and retrieved are detailed within those specifications and are out
21 of scope for this document.

22 **Composable Architecture:**

23 By using the SOAP [[SOAP](#)] and WSDL [[WSDL](#)] extensibility model, SOAP-based and
24 WSDL-based specifications are designed to be composed with each other to define a
25 rich Web services environment. As such, WS-ReliableMessaging by itself does not
26 define all the features required for a complete messaging solution. WS-
27 ReliableMessaging is a building block that is used in conjunction with other
28 specifications and application-specific protocols to accommodate a wide variety of
29 protocols related to the operation of distributed Web services.

30 **Status:**

31 TBD

Page : 2 Line : 22 Author : AK 08/16/2005

This seems out of place. I would like to suggest that we add a new subsection under introduction called 'Relation to other specification'. We can include this para as well as stuff about conformance to WS-Addressing in it.

32 Table of Contents

33	1INTRODUCTION.....	6
34	1.1GOALS AND REQUIREMENTS.....	6
35	1.1.1Requirements.....	6
36	1.2NOTATIONAL CONVENTIONS.....	6
37	1.3NAMESPACE.....	7
38	1.4COMPLIANCE.....	8
39	2RELIABLE MESSAGING MODEL.....	9
40	2.1GLOSSARY.....	10
41	2.2PROTOCOL PRECONDITIONS.....	11
42	2.3PROTOCOL INVARIANTS.....	11
43	2.4EXAMPLE MESSAGE EXCHANGE.....	11
44	3RM PROTOCOL ELEMENTS.....	14
45	3.1SEQUENCES.....	14
46	3.2SEQUENCE ACKNOWLEDGEMENT.....	16
47	3.3REQUEST ACKNOWLEDGEMENT.....	18
48	3.4SEQUENCE CREATION.....	19
49	3.5SEQUENCE TERMINATION.....	23
50	4FAULTS.....	24
51	4.1SEQUENCEFAULT ELEMENT.....	26
52	4.2SEQUENCE TERMINATED.....	27
53	4.3UNKNOWN SEQUENCE.....	27
54	4.4INVALID ACKNOWLEDGEMENT.....	27
55	4.5MESSAGE NUMBER ROLLOVER.....	28
56	4.6LAST MESSAGE NUMBER EXCEEDED.....	28
57	4.7CREATE SEQUENCE REFUSED.....	29
58	5SECURITY CONSIDERATIONS.....	30
59	6REFERENCES.....	32
60	6.1NORMATIVE.....	32
61	6.2NON-NORMATIVE.....	33
62	APPENDIX A.SCHEMA	34

63	APPENDIX B.MESSAGE EXAMPLES.....	39
64	B.1.CREATE SEQUENCE.....	40
65	B.2. INITIAL TRANSMISSION.....	42
66	B.3.FIRST ACKNOWLEDGEMENT.....	44
67	B.4.RETRANSMISSION.....	45
68	B.5.TERMINATION.....	46
69	APPENDIX C.WSDL.....	48
70	APPENDIX D.ACKNOWLEDGMENTS.....	50
71	APPENDIX E.REVISION HISTORY.....	51
72	APPENDIX F.NOTICES.....	52

73 **1 Introduction**

74 It is often a requirement for two Web services that wish to communicate to do so
75 reliably in the presence of software component, system, or network failures. The
76 primary goal of this specification is to create a modular mechanism for reliable
77 message delivery. It defines a messaging protocol to identify, track, and manage the
78 reliable delivery of messages between exactly two parties, a source and a
79 destination. It also defines a SOAP binding that is required for interoperability.
80 Additional bindings may be defined.

81 This mechanism is extensible allowing additional functionality, such as security, to be
82 tightly integrated. This specification integrates with and complements the WS-
83 Security, WS-Policy, and other Web services specifications. Combined, these allow
84 for a broad range of reliable, secure messaging options.

85 **1.1 Goals and Requirements**

86 **1.1.1 Requirements**

87 **1.2 Notational Conventions**

88 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
89 "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
90 document are to be interpreted as described in RFC 2119 [[KEYWORDS](#)].

91 This specification uses the following syntax to define normative outlines for
92 messages:

- 93 • The syntax appears as an XML instance, but values in italics indicate data types instead
94 of values.
- 95 • Characters are appended to elements and attributes to indicate cardinality:
 - 96 ○ "?" (0 or 1)
 - 97 ○ "*" (0 or more)
 - 98 ○ "+" (1 or more)
- 99 • The character "|" is used to indicate a choice between alternatives.
- 100 • The characters "[" and "]" are used to indicate that contained items are to be treated as a
101 group with respect to cardinality or choice.

Page : 5 Line : 92 Author : AK 08/16/2005

It would be nice to use language similar or same as the WS-Addressing/WSDL 2.0 spec which use the same notation (modulo copyright concerns). This is of course not critical, just consistency across ws-* specs. Regardless, we do need to add statements about what {any} and @{any} mean

- 102 • An ellipsis (i.e. "...") indicates a point of extensibility that allows other child, or attribute,
103 content. Additional children and/or attributes MAY be added at the indicated extension
104 points but MUST NOT contradict the semantics of the parent and/or owner, respectively.
105 If an extension is not recognized it SHOULD be ignored.
- 106 • XML namespace prefixes (See Section [Namespace](#)) are used to indicate the namespace
107 of the element being defined.

108 1.3 Namespace

109 The XML namespace [[XML-ns](#)] URI that MUST be used by implementations of this
110 specification is:

111 <http://schemas.xmlsoap.org/ws/2005/02/rm>

112 Table 1 lists XML namespaces that are used in this specification. The choice of any
113 namespace prefix is arbitrary and not semantically significant.

114 The following namespaces are used in this document:

115 *Table 1*

Prefix	Namespace
S	http://www.w3.org/2003/05/soap-envelope
S11	http://schemas.xmlsoap.org/soap/envelope/
wstrm	http://schemas.xmlsoap.org/ws/2005/02/rm
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
xs	http://www.w3.org/2001/XMLSchema

116 The normative schema for WS-Reliable Messaging can be found at:

117 <http://schemas.xmlsoap.org/ws/2005/02/rm/wstrm.xsd>

118 All sections explicitly noted as examples are informational and are not to be
119 considered normative.

120 If an action URI is used, and one is not already defined per the rules of the WS-
121 Addressing specification [[WS-Addressing](#)], then the action URI MUST consist of the
122 reliable messaging namespace URI concatenated with the "/" character and the
123 element name. For example:

Page : 6 Line : 109 Author : AK 08/16/2005

This is ambiguous. How about replacing this stmt by something like:

"Elements defined by this specification below to the following namespace ..."

Page : 6 Line : 118 Author : AK 08/16/2005

Add the following:

Namespace names of the general form "http://example.org/..." and

"http://example.com/..." represent application or context-dependent URIs (see RFC 2396 [RFC 2396]).

Page : 6 Line : 120 Author : AK 08/16/2005

This para shouldn't be under the 'namespace' subsection. How about moving this to the 'relationship with other spec' subsection?

124 <http://schemas.xmlsoap.org/ws/2005/02/rm/SequenceAcknowledgement>

125 **1.4 Compliance**

126 An implementation is not compliant with this specification if it fails to satisfy one or
127 more of the MUST or REQUIRED level requirements defined herein. A SOAP Node
128 MUST NOT use the XML namespace identifier for this specification (listed in
129 Section [Namespace](#)) within SOAP Envelopes unless it is compliant with this
130 specification.

131 Normative text within this specification takes precedence over normative outlines,
132 which in turn take precedence over the XML Schema [[XML Schema Part 1](#), [Part 2](#)]
133 descriptions.

134 2 Reliable Messaging Model

135 Many errors may interrupt a conversation. Messages may be lost, duplicated or
136 reordered. Further the host systems may experience failures and lose volatile state.

137 WS-ReliableMessaging provides an interoperable protocol that a Reliable Messaging
138 (RM) Source and Reliable Messaging (RM) Destination use to provide Application
139 Source and Destination a guarantee that a message that is sent will be delivered.
140 The guarantee is specified as a delivery assurance. The protocol supports the
141 endpoints in providing these delivery assurances. It is the responsibility of the RM
142 Source and RM Destination to fulfill the delivery assurances, or raise an error. The
143 protocol defined here allows endpoints to meet this guarantee for the delivery
144 assurances defined below.

145 Persistence considerations related to an endpoint's ability to satisfy the delivery
146 assurances defined below are the responsibility of the implementation and do not
147 affect the wire protocol. As such, they are out of scope of this specification.

148 There are four basic delivery assurances that endpoints can provide:

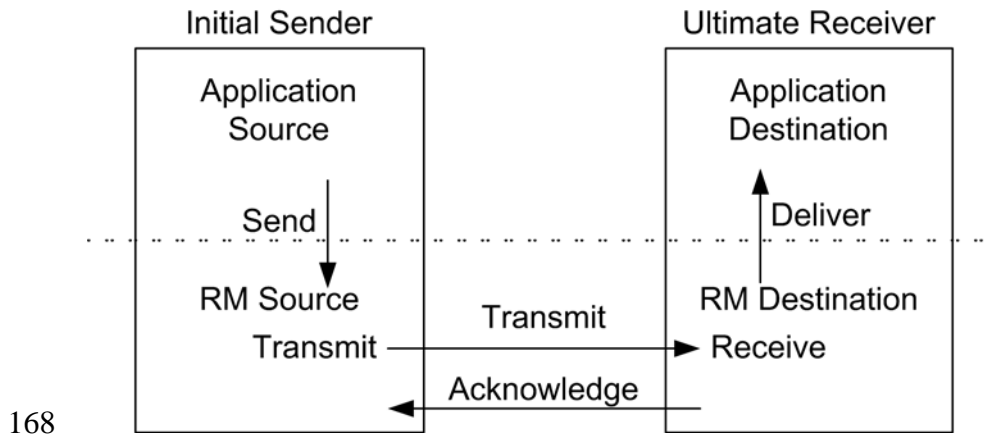
149 **AtMostOnce** Messages will be delivered at most once without duplication or an error
150 will be raised on at least one endpoint. It is possible that some messages in a
151 sequence may not be delivered.

152 **AtLeastOnce** Every message sent will be delivered or an error will be raised on at
153 least one endpoint. Some messages may be delivered more than once.

154 **ExactlyOnce** Every message sent will be delivered without duplication or an error
155 will be raised on at least one endpoint. This delivery assurance is the logical "and" of
156 the two prior delivery assurances.

157 **InOrder** Messages will be delivered in the order that they were sent. This delivery
158 assurance may be combined with any of the above delivery assurances. It requires
159 that the sequence observed by the ultimate receiver be non-decreasing. It says
160 nothing about duplications or omissions.

161 Figure 1 below illustrates the entities and events in a simple reliable message
162 exchange. First, the Application Source Sends a message for reliable delivery. The
163 Reliable Messaging (RM) Source accepts the message and Transmits it one or more
164 times. After receiving the message, the RM Destination Acknowledges it. Finally,
165 the RM Destination delivers the message to the Application Destination. The exact
166 roles the entities play and the complete meaning of the events will be defined
167 throughout this specification.



169 Figure 1: Reliable Messaging Model

170 2.1 Glossary

171 The following definitions are used throughout this specification:

172 **Endpoint:** A referencable entity, processor, or resource where Web service messages
173 are originated or targeted.

174 **Application Source:** The endpoint that Sends a message.

175 **Application Destination:** The endpoint to which a message is Delivered.

176 **Delivery Assurance:** The guarantee that the messaging infrastructure provides on
177 the delivery of a message.

178 **RM Source:** The endpoint that transmits the message.

179 **RM Destination:** The endpoint that receives the message.

180 **Send:** The act of submitting a message to the RM Source for reliable delivery. The
181 reliability guarantee begins at this point.

182 **Deliver:** The act of transferring a message from the RM Destination to the
183 Application Destination. The reliability guarantee is fulfilled at this point.

184 **Transmit:** The act of writing a message to a network connection.

185 **Receive:** The act of reading a message from a network connection.

186 **Acknowledgement:** The communication from the RM Destination to the RM Source
187 indicating the successful receipt of a message.

Page : 9 Line : 172 Author : AK 08/16/2005

This is the same definition as ws-addressing. I would like to suggest that we just point to the ws-addr spec for this.

188 **2.2 Protocol Preconditions**

189 The correct operation of the protocol requires that a number of preconditions **MUST**
190 be established prior to the processing of the initial sequenced message:

191 • The RM Source **MUST** have an endpoint reference that uniquely identifies the RM Destination
192 endpoint; correlations across messages addressed to the unique endpoint **MUST** be
193 meaningful.

194 • The RM Source **MUST** have knowledge of the destination's policies, if any, and the RM
195 Source **MUST** be capable of formulating messages that adhere to this policy.

196 If a secure exchange of messages is required, then the RM Source and RM
197 Destination **MUST** have a security context.

198 **2.3 Protocol Invariants**

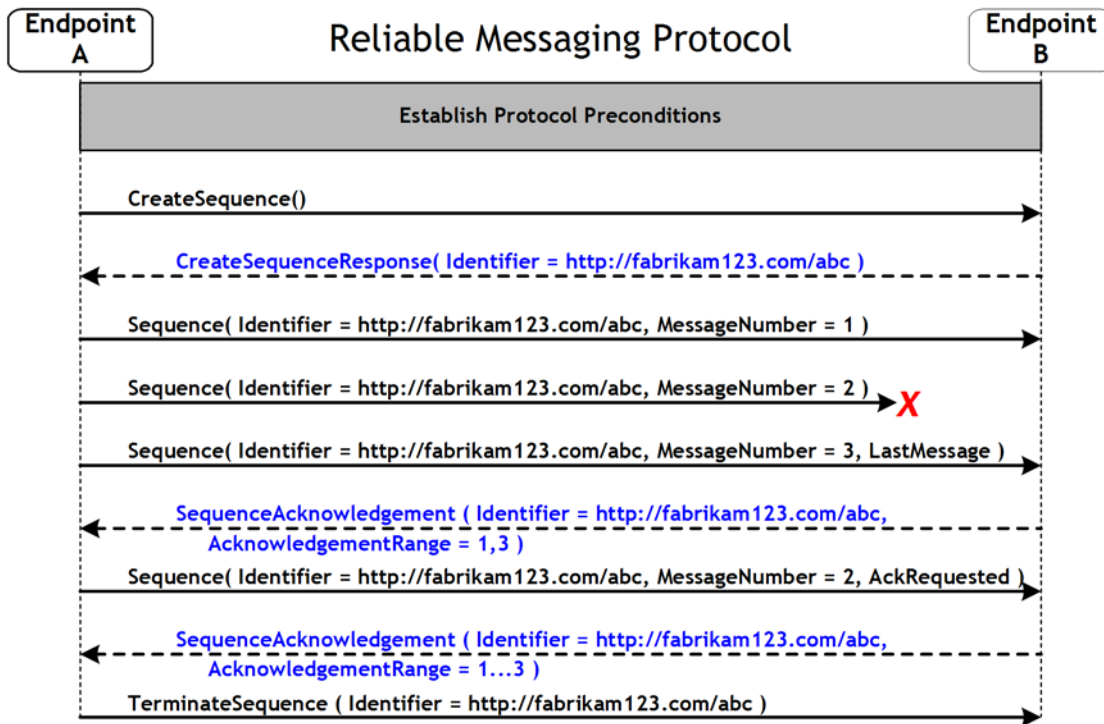
199 During the lifetime of the protocol, two invariants are **REQUIRED** for correctness:

200 • The RM Source **MUST** assign each reliable message a sequence number (defined below)
201 beginning at 1 and increasing by exactly 1 for each subsequent reliable message.

202 Every acknowledgement issued by the RM Destination **MUST** include within an
203 acknowledgement range or ranges the sequence number of every message
204 successfully received by the RM Destination and **MUST** exclude sequence numbers of
205 any messages not yet received.

206 **2.4 Example Message Exchange**

207 Figure 2 illustrates a possible message exchange between two reliable messaging
208 endpoints A and B.



209 Figure 2: The WS-ReliableMessaging Protocol

- 210 1. The protocol preconditions are established. These include policy exchange,
211 endpoint resolution, establishing trust.
- 212 2. The RM Source requests creation of a new Sequence.
- 213 3. The RM Destination creates a Sequence by returning a globally unique identifier.
- 214 4. The RM Source begins sending messages beginning with MessageNumber 1. In
215 the figure the RM Source sends 3 messages.
- 216 5. Since the 3rd message is the last in this exchange, the RM Source includes a
217 <wsrm:LastMessage> token.
- 218 6. The 2nd message is lost in transit.
- 219 7. The RM Destination acknowledges receipt of message numbers 1 and 3 in
220 response to the RM Source's <wsrm:LastMessage> token.
- 221 8. The RM Source retransmits the 2nd message. This is a new message on the
222 underlying transport, but since it has the same sequence identifier and message
223 number so the RM Destination can recognize it as equivalent to the earlier
224 message, in case both are received.

- 225 9. The RM Source includes an `<wsrm:AckRequested>` element so the RM Destination
226 will expedite an acknowledgement.
- 227 10. The RM Destination receives the second transmission of the message with
228 MessageNumber 2 and acknowledges receipt of message numbers 1, 2, and 3
229 which carried the `<wsrm:LastMessage>` token.
- 230 11. The RM Source receives this acknowledgement and sends a `TerminateSequence`
231 message to the RM Destination indicating that the sequence is completed and
232 reclaims any resources associated with the Sequence.
- 233 12. The RM Destination receives the `TerminateSequence` message indicating that the
234 RM Source will not be sending any more messages, and reclaims any resources
235 associated with the Sequence.
- 236 Now that the basic model has been outlined, the details of the elements used in this
237 protocol are now provided in Section 3.

238 3 RM Protocol Elements

239 The protocol elements define extensibility points at various places. Additional
240 children elements and/or attributes MAY be added at the indicated extension points
241 but MUST NOT contradict the semantics of the parent and/or owner, respectively. If a
242 receiver does not recognize an extension, the receiver SHOULD ignore the extension.

243 3.1 Sequences

244 The RM protocol uses a `<wsrm:Sequence>` header block to track and manage the
245 reliable delivery of messages. Messages for which the delivery assurance applies
246 MUST contain a `<wsrm:Sequence>` header block. Each Sequence MUST have a
247 unique `<wsrm:Identifier>` element and each message within a Sequence MUST
248 have a `<wsrm:MessageNumber>` element that increments by 1 from an initial value of
249 1. These values are contained within a `<wsrm:Sequence>` header block accompanying
250 each message being delivered in the context of a Sequence. In addition to mandatory
251 `<wsrm:Identifier>` and `<wsrm:MessageNumber>` elements, the header MAY include a
252 `<wsrm:LastMessage>` element.

253 There MUST be no more than one `<wsrm:Sequence>` header block in any message.

254 The purpose of the `<wsrm:LastMessage>` element is to signal to the RM Destination
255 that the message represents the last message in the Sequence.

256 A following exemplar defines its syntax:

```
257 <wsrm:Sequence ...>  
258   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
259   <wsrm:MessageNumber> xs:unsignedLong </wsrm:MessageNumber>  
260   <wsrm:LastMessage/?>  
261   ...  
262 </wsrm:Sequence>
```

263 The following describes the content model of the Sequence header block.

264 `/wsrm:Sequence`

265 This is the element containing Sequence information for WS-ReliableMessaging. The
266 `<wsrm:Sequence>` element MUST be understood by the RM Destination. The `<wsrm:Sequence>`
267 element MUST have a `mustUnderstand` attribute from the namespace corresponding to the
268 version of SOAP to which the `<wsrm:Sequence>` SOAP header block is bound.

269 `/wsrm:Sequence/wsrm:Identifier`

270 This required element MUST contain an absolute URI conformant with RFC2396 that uniquely
271 identifies the Sequence.

Page : 13 Line : 239 Author : AK 08/16/2005

This paragraph is applicable to section 3 as well as section 4. Suggest that we move this to section 1.

Page : 13 Line : 267 Author : AK 08/16/2005

Change it to say : "... mustUnderstand attribute with a value of 1/true ...'

Page : 13 Line : 270 Author : AK 08/16/2005

there are several reference to RFC2396. 2396 is obsoluted by 3986.

Or like ws-addressing we could move to IRIs (RFC 3987)

272 /wsrm:Sequence/wsrm:Identifier/{any}

273 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
274 to the element.

275 /wsrm:Sequence/wsrm:MessageNumber

276 This required element MUST contain an xs:unsignedLong representing the ordinal position of the
277 message within a Sequence. Sequence MessageNumbers start at 1 and monotonically increase
278 throughout the Sequence. If the message number exceeds the internal limitations of an RM
279 Source or RM Destination or reaches the maximum value of an xs:unsignedLong
280 (18,446,744,073,709,551,615), the RM Source or Destination MUST issue a
281 MessageNumberRollover fault.

282 /wsrm:Sequence/wsrm:LastMessage

283 This element MAY be included by the RM Source endpoint. The <wsrm:LastMessage> element
284 has no content.

285 /wsrm:Sequence/{any}

286 This is an extensibility mechanism to allow different types of information, based on a schema, to
287 be passed.

288 /wsrm:Sequence/{any}

289 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
290 to the element.

291 A RM Source endpoint MUST include a <wsrm:LastMessage> element in the
292 <wsrm:Sequence> element for the last message in a Sequence. An RM Destination
293 endpoint MUST respond with a <wsrm:SequenceAcknowledgement> upon receipt of a
294 <wsrm:LastMessage> element. A Sequence MUST NOT use a <wsrm:MessageNumber>
295 value greater than that which accompanies a <wsrm:LastMessage> element. An RM
296 Destination MUST generate a LastMessageNumberExceeded (See Section 4.6) fault
297 upon receipt of such a message. In the event that an RM Source needs to close a
298 Sequence and there is no application message, the RM Source MAY send a message
299 with an empty body containing <wsrm:Sequence> header with the
300 <wsrm:LastMessage> element. In this usage, the action URI MUST be:

301 `http://schemas.xmlsoap.org/ws/2005/02/rm/LastMessage`

302 in preference to the pattern defined in Section 1.2.

303 The following example illustrates a Sequence header block.

304 `<wsrm:Sequence>`
305 `<wsrm:Identifier>http://example.com/abc</wsrm:Identifier>`
306 `<wsrm:MessageNumber>10</wsrm:MessageNumber>`

Page : 14 Line : 273 Author : AK 08/16/2005

The phrase 'based on schemas' (or 'based on schema to be passed') is used anywhere extensibility is defined. I don't understand this phrase. If the intention is to say that the extensibility attributes/elements must not be from the WSRM schema ("##other") then we should say exactly that.

Page : 14 Line : 294 Author : AK 08/16/2005

not quite accurate. I would like to suggest that we use the XPATH notation used above. I.e. /wsm:Sequence/wsm:LastMessage

307
308

```
<wsrm:LastMessage/>  
</wsrm:Sequence>
```

309 3.2 Sequence Acknowledgement

310 The RM Destination informs the RM Source of successful message receipt using a
311 <wsrm:SequenceAcknowledgement> header block. The
312 <wsrm:SequenceAcknowledgement> header block MAY be transmitted independently
313 or included on return messages. The RM Destination MAY send a
314 <wsrm:SequenceAcknowledgement> header block at any point during which the
315 sequence is valid. The timing of acknowledgements can be advertised using policy
316 and acknowledgements can be explicitly requested using the <wsrm:AckRequested>
317 directive (see Section 3.3).

318 The following exemplar defines its syntax:

```
319 <wsrm:SequenceAcknowledgement ...>  
320   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
321   [ <wsrm:AcknowledgementRange ...  
322     Upper="xs:unsignedLong"  
323     Lower="xs:unsignedLong"/> +  
324     | <wsrm:Nack> xs:unsignedLong </wsrm:Nack> + ]  
325   ...  
326 </wsrm:SequenceAcknowledgement>
```

327 The following describes the content model of the <wsrm:SequenceAcknowledgement>
328 header block.

329 /wsrm:SequenceAcknowledgement

330 This element contains the Sequence acknowledgement information.

331 /wsrm:SequenceAcknowledgement/wsrm:Identifier

332 This required element MUST contain an absolute URI conformant with RFC2396 that uniquely
333 identifies the Sequence.

334 /wsrm:SequenceAcknowledgement/wsrm:Identifier/@{any}

335 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
336 to the element.

337 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange

338 This optional element, if present, can occur 1 or more times. It contains a range of message
339 Sequence MessageNumbers successfully received by the receiving endpoint manager. The
340 ranges SHOULD NOT overlap. This element MUST NOT be present if <wsrm:Nack> is also
341 present as a child of <wsrm:SequenceAcknowledgement>.

Page : 15 Line : 313 Author : AK 08/16/2005

A better way to say 'return messages' is to say:

'... included in a response message, in the case of a request-response pattern'.

Page : 15 Line : 338 Author : AK 08/16/2005

should 'optional' and 'required' words in the spec be converted to RFC 2119 OPTIONAL and REQUIRED. The occurrences seem to indicate the same meaning as the RFC

Page : 15 Line : 341 Author : AK 08/16/2005

Given that there can be multiple SeqAck headers in a message, an accurate way of saying this is:

"... MUST NOT be present if a sibling <wsrm:Nack> element is also present ..."

342 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Upper
343 This required attribute contains an xs:unsignedLong representing the <wsrm:MessageNumber>
344 of the highest contiguous message in a Sequence range received by the RM Destination.

345 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Lower
346 This required attribute contains an xs:unsignedLong representing the <wsrm:MessageNumber>
347 of the lowest contiguous message in a Sequence range received by the RM Destination.

348 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@{any}
349 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
350 to the element.

351 /wsrm:SequenceAcknowledgement/wsrm:Nack
352 This optional element, if present, MUST contain an xs:unsignedLong representing the
353 <wsrm:MessageNumber> of an unreceived message in a Sequence. This element MUST NOT
354 be present if the <wsrm:AcknowledgementRange> is also present as a child of
355 <wsrm:SequenceAcknowledgement>. The <wsrm:Nack> element permits the gap analysis of
356 the <wsrm:AcknowledgementRange> elements to be performed at the RM Destination rather
357 than at the RM Source which may yield performance benefits in certain environments.

358 /wsrm:SequenceAcknowledgement/{any}
359 This is an extensibility mechanism to allow different (extensible) types of information, based on a
360 schema, to be passed.

361 /wsrm:SequenceAcknowledgement/@{any}
362 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
363 to the element.

364 The following examples illustrate <wsrm:SequenceAcknowledgement> elements:

- 365 • Message numbers 1..10 inclusive in a Sequence have been received by the RM Destination.

```
366 <wsrm:SequenceAcknowledgement>  
367     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
368     <wsrm:AcknowledgementRange Upper="10" Lower="1"/>  
369 </wsrm:SequenceAcknowledgement>
```

- 370 • Message numbers 1..2, 4..6, and 8..10 inclusive in a Sequence have been received by the
371 RM Destination, messages 3 and 7 have not been received.

```
372 <wsrm:SequenceAcknowledgement>  
373     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
374     <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
375     <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
376     <wsrm:AcknowledgementRange Upper="10" Lower="8"/>
```

377

```
</wsrm:SequenceAcknowledgement>
```

- 378 • Message number 3 in a Sequence has not been received by the RM Destination.

379

```
<wsrm:SequenceAcknowledgement>
```

380

```
  <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
```

381

```
  <wsrm:Nack>3</wsrm:Nack>
```

382

```
</wsrm:SequenceAcknowledgement>
```

383 3.3 Request Acknowledgement

384 The purpose of the `<wsrm:AckRequested>` header block is to signal to the RM

385 Destination that the RM Source is requesting that a

386 `<wsrm:SequenceAcknowledgement>` be returned.

387 At any time, the RM Source may request an acknowledgement message from the RM

388 Destination endpoint using an `<wsrm:AckRequested>` header block.

389 The RM Source endpoint requests this acknowledgement by including an

390 `<wsrm:AckRequested>` header block in the message. An RM Destination that receives

391 a message that contains an `<wsrm:AckRequested>` header block MUST respond with

392 a message containing a `<wsrm:SequenceAcknowledgement>` header block.

393 The following exemplar defines its syntax:

394

```
<wsrm:AckRequested ...>
```

395

```
  <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
```

396

```
  <wsrm:MessageNumber> xs:unsignedLong </wsrm:MessageNumber> ?
```

397

```
  ...
```

398

```
</wsrm:AckRequested>
```

399 `/wsrm:AckRequested`

400 This element requests an acknowledgement for the identified sequence.

401 `/wsrm:AckRequested/wsrm:Identifier`

402 This required element MUST contain an absolute URI, conformant with RFC2396, that uniquely

403 identifies the Sequence to which the request applies.

404 `/wsrm:AckRequested/wsrm:Identifier/@{any}`

405 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added

406 to the element.

407 `/wsrm:AckRequested/wsrm:MessageNumber`

408 This optional element, if present, MUST contain an `xs:unsignedLong` representing the highest

409 `<wsrm:MessageNumber>` sent by the RM Source within the Sequence. If present, it MAY be

410 treated as a hint to the RM Destination as an optimization to the process of preparing to transmit a
411 <wsrm:SequenceAcknowledgement>.

412 /wsrm:AckRequested/{any}

413 This is an extensibility mechanism to allow different (extensible) types of information, based on a
414 schema, to be passed.

415 /wsrm:AckRequested/@{any}

416 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
417 to the element.

418 3.4 Sequence Creation

419 The RM Source MUST request creation of an outbound Sequence by sending a
420 <wsrm:CreateSequence> element in the body of a message to the RM Destination
421 which in turn responds either with a <wsrm:CreateSequenceResponse> or a
422 CreateSequenceRefused fault in the body of the response message.
423 <wsrm:CreateSequence> MAY carry an offer to create an inbound sequence which is
424 either accepted or rejected in the <wsrm:CreateSequenceResponse>.

425 The RM Destination of the outbound sequence is the WS-Addressing
426 EndpointReference [WS-Addressing] to which <wsrm:CreateSequence> is sent. The
427 RM Destination of the inbound sequence is the WS-Addressing <wsa:ReplyTo> of the
428 <wsrm:CreateSequence>.

429 The following exemplar defines the <wsrm:CreateSequence> syntax:

```
430 <wsrm:CreateSequence ...>  
431   <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>  
432   <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
433   <wsrm:Offer ...>  
434     <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
435     <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
436     ...  
437   </wsrm:Offer> ?  
438   ...  
439   <wsse:SecurityTokenReference>  
440     ...  
441   </wsse:SecurityTokenReference> ?  
442   ...  
443 </wsrm:CreateSequence>
```

444 /wsrm:CreateSequence

445 This element requests creation of a new Sequence between the RM Source that sends it, and the
446 RM Destination to which it is sent. This element MUST NOT be sent as a header block. The RM
447 Destination MUST respond either with a <wsrm:CreateSequenceResponse> response
448 message or a CreateSequenceRefused fault.

449 /wsrm:CreateSequence/wsrm:AcksTo

450 This required element, of type wsa:EndpointReferenceType as specified by WS-Addressing [WS-
451 Addressing] specifies the endpoint reference to which <wsrm:SequenceAcknowledgement>
452 messages and faults related to the created Sequence are to be sent.

453 /wsrm:CreateSequence/wsrm:Expires

454 This element, if present, of type xs:duration specifies the RM Source's requested duration for
455 the Sequence. The RM Destination MAY either accept the requested duration or assign a lesser
456 value of its choosing. A value of 'PT0S' indicates that the Sequence will never expire. Absence of
457 the element indicates an implied value of 'PT0S'.

458 /wsrm:CreateSequence/wsrm:Expires/@{any}

459 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
460 to the element.

461 /wsrm:CreateSequence/wsrm:Offer

462 This element, if present, enables an RM Source to offer a corresponding Sequence for the reliable
463 exchange of messages transmitted from RM Destination to RM Source.

464 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier

465 This required element MUST contain an absolute URI conformant with RFC2396 that uniquely
466 identifies the offered Sequence.

467 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier/@{any}

468 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
469 to the element.

470 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires

471 This element, if present, of type xs:duration specifies the duration for the Sequence. A value
472 of 'PT0S' indicates that the Sequence will never expire. Absence of the element indicates an
473 implied value of 'PT0S'.

474 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires/@{any}

475 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
476 to the element.

477 /wsrm:CreateSequence/wsrm:Offer/{any}

478 This is an extensibility mechanism to allow different (extensible) types of information, based on a
479 schema, to be passed.

480 /wsrm:CreateSequence/wsrm:Offer/@{any}

481 This is an extensibility mechanism to allow different (extensible) types of information, based on a
482 schema, to be passed.

483 /wsrm:CreateSequence/wsse:SecurityTokenReference

484 This optional element uses the extensibility mechanism defined next to communicate an explicit
485 reference to the security token to be used to authorize messages for the created outbound
486 Sequence and if offered the inbound Sequence, using a <wsse:SecurityTokenReference>
487 as documented in WS-Security [WSSecurity]. All subsequent messages in the outbound
488 Sequence and if offered the inbound Sequence MUST demonstrate proof-of-possession of the
489 referenced key.

490 /wsrm:CreateSequence/{any}

491 This is an extensibility mechanism to allow different (extensible) types of information, based on a
492 schema, to be passed.

493 /wsrm:CreateSequence/@{any}

494 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
495 to the element.

496 A <wsrm:CreateSequenceResponse> is sent in the body of a response message by an
497 RM Destination in response to receipt of a <wsrm:CreateSequence> request
498 message. It carries the <wsrm:Identifier> of the created Sequence and indicates
499 that the RM Source may begin sending messages in the context of the identified
500 Sequence.

501 The following exemplar defines the <wsrm:CreateSequenceResponse> syntax:

```
502 <wsrm:CreateSequenceResponse ...>  
503   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
504   <wsrm:Expires> xs:duration </wsrm:Expires> ?  
505   <wsrm:Accept ...>  
506     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>  
507     ...  
508   </wsrm:Accept> ?  
509   ...  
510 </wsrm:CreateSequenceResponse>
```

511 /wsrm:CreateSequenceResponse

512 This element is sent in the body of the response message in response to a
513 <wsrm:CreateSequence> request message. It indicates that the RM Destination has created

514 a new Sequence at the request of the RM Source. This element MUST NOT be sent as a header
515 block.

516 /wsrm:CreateSequenceResponse/wsrm:Identifier
517 This required element MUST contain an absolute URI conformant with RFC2396 of the Sequence
518 that has been created by the RM Destination.

519 /wsrm:CreateSequenceResponse/wsrm:Identifier/{any}
520 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
521 to the element.

522 /wsrm:CreateSequenceResponse/wsrm:Expires
523 This element, if present, of type `xs:duration` accepts or refines the RM Source's requested
524 duration for the Sequence. A value of 'PT0S' indicates that the Sequence will never expire.
525 Absence of the element indicates an implied value of 'PT0S'. This value MUST be equal or lesser
526 than the value requested by the RM Source in the corresponding `<wsrm:CreateSequence>`
527 message.

528 /wsrm:CreateSequenceResponse/wsrm:Expires/{any}
529 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
530 to the element.

531 /wsrm:CreateSequenceResponse/wsrm:Accept
532 This element, if present, enables an RM Destination to accept the offer of a corresponding
533 Sequence for the reliable exchange of messages transmitted from RM Destination to RM Source.
534 This element MUST be present if the corresponding `<wsrm:CreateSequence>` message
535 contained an `<wsrm:Offer>` element.

536 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo
537 This required element, of type `wsa:EndpointReferenceType` as specified by WS-Addressing [WS-
538 Addressing], specifies the endpoint reference to which `<wsrm:SequenceAcknowledgement>`
539 messages related to the accepted Sequence are to be sent.

540 /wsrm:CreateSequenceResponse/wsrm:Accept/{any}
541 This is an extensibility mechanism to allow different (extensible) types of information, based on a
542 schema, to be passed.

543 /wsrm:CreateSequenceResponse/wsrm:Accept/{any}
544 This is an extensibility mechanism to allow different (extensible) types of information, based on a
545 schema, to be passed.

546 /wsrm:CreateSequenceResponse/{any}

Page : 21 Line : 534 Author : AK 08/16/2005

too vague. A wsrn:Offer element can be in the extensibility point. A better way would be to use the xpath like syntax that is already being used.

547 This is an extensibility mechanism to allow different (extensible) types of information, based on a
548 schema, to be passed.

549 /wsrm:CreateSequenceResponse/{any}

550 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
551 to the element.

552 3.5 Sequence Termination

553 After an RM Source receives the <wsrm:SequenceAcknowledgement> acknowledging
554 the complete range of messages in a Sequence, it sends a
555 <wsrm:TerminateSequence> element, in the body of a message to the RM
556 Destination to indicate that the Sequence is complete, and that it will not be sending
557 any further messages related to the Sequence. The RM Destination can safely reclaim
558 any resources associated with the Sequence upon receipt of the
559 <wsrm:TerminateSequence> message.

560 The following exemplar defines the TerminateSequence syntax:

```
561 <wsrm:TerminateSequence ...>  
562   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
563   ...  
564 </wsrm:TerminateSequence>
```

565 /wsrm:TerminateSequence

566 This element is sent by an RM Source after it has received the final
567 <wsrm:SequenceAcknowledgement> covering the full range of a Sequence. It indicates that
568 the RM Destination can safely reclaim any resources related to the identified Sequence. This
569 element MUST NOT be sent as a header block.

570 /wsrm:TerminateSequence/wsrm:Identifier

571 This required element MUST contain an absolute URI conformant with RFC2396 of the Sequence
572 that is being terminated.

573 /wsrm:TerminateSequence/wsrm:Identifier/{any}

574 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
575 to the element.

576 /wsrm:TerminateSequence/{any}

577 This is an extensibility mechanism to allow different (extensible) types of information, based on a
578 schema, to be passed.

579 /wsrm:TerminateSequence/{any}

580 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
581 to the element.

582 4 Faults

583 The fault definitions defined in this section reference certain abstract properties, such
584 as [fault endpoint], that are defined in section 3 of the WS-Addressing [WS-
585 Addressing] specification. Endpoints compliant with this specification MUST include
586 required Message Addressing Properties on all fault messages.

587 Sequence creation uses a CreateSequence, CreateSequenceResponse request-
588 response pattern. Faults for this operation are treated as defined in WS-Addressing.
589 CreateSequenceRefused is a possible fault reply for this operation.

590 UnknownSequence is a fault generated by endpoints when messages carrying RM
591 header blocks targeted at unrecognized sequences are detected, these faults are also
592 treated as defined in WS-Addressing. All other faults in this section relate to the
593 processing of RM header blocks targeted at known sequences and are collectively
594 referred to as sequence faults. Sequence faults SHOULD be sent to the same
595 [destination] as <wsrm:SequenceAcknowledgement> messages. These faults are
596 correlated using the Sequence identifier carried in the detail.

597 WS-ReliableMessaging faults MUST include as the [action] property the default fault
598 action URI defined in the version of WS-Addressing used in the message. The value
599 from the current version is below for informational purposes:

600 `http://schemas.xmlsoap.org/ws/2004/08/addressing/fault`

601 The faults defined in this section are generated if the condition stated in the
602 preamble is met. Fault handling rules are defined in section 4 of WS-Addressing.

603 The definitions of faults use the following properties:

604 [Code] The fault code.

605 [Subcode] The fault subcode.

606 [Reason] The English language reason element.

607 [Detail] The detail element. If absent, no detail element is defined for the fault.

608 The [Code] property MUST be either "Sender" or "Receiver". These properties are
609 serialized into text XML as follows:

SOAP Version	Sender	Receiver
SOAP 1.1	S11:Client	S11:Server
SOAP 1.2	S:Sender	S:Receiver

610 The properties above bind to a SOAP 1.2 fault as follows:

WS-ReliableMessaging-v1%5B1%5D.0-wd-01.sxw

8/16/2005

Copyright © OASIS Open 2005. All Rights Reserved.

Page 24 of 52

Page : 24 Line : 597 Author : AK 08/16/2005

The default fault action URI is defined only for the SOAP binding and it is meant only for ws-addressing related faults. This para should be deleted OR specific action(s) should be defined for WSRM faults.

Page : 24 Line : 598 Author : AK 08/16/2005

this means any version of ws-addressing that is used in the message.

If that is not the intend (which I don't think it is), we need to tie it down to a specific version of WS-Addressing (W3C one)

```

611 <S:Envelope>
612   <S:Header>
613     <wsa:Action>
614       http://schemas.xmlsoap.org/ws/2004/08/addressing/fault
615     </wsa:Action>
616     <!-- Headers elided for clarity. -->
617   </S:Header>
618   <S:Body>
619     <S:Fault>
620       <S:Code>
621         <S:Value> [Code] </S:Value>
622         <S:Subcode>
623           <S:Value> [Subcode] </S:Value>
624         </S:Subcode>
625       </S:Code>
626       <S:Reason>
627         <S:Text xml:lang="en"> [Reason] </S:Text>
628       </S:Reason>
629       <S:Detail>
630         [Detail]
631         ...
632       </S:Detail>
633     </S:Fault>
634   </S:Body>
635 </S:Envelope>

```

636 The properties above bind to a SOAP 1.1 fault as follows when the fault is triggered
637 by processing an RM header block:

```

638 <S11:Envelope>
639   <S11:Header>
640     <wsrm:SequenceFault>
641       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
642       ...
643     </wsrm:SequenceFault>
644     <!-- Headers elided for clarity. -->
645   </S11:Header>
646   <S11:Body>
647     <S11:Fault>
648       <faultcode> [Code] </faultcode>
649       <faultstring> [Reason] </faultstring>
650     </S11:Fault>
651   </S11:Body>

```

652 `</S11:Envelope>`

653 The properties bind to a SOAP 1.1 fault as follows when the fault is generated as a
654 result of processing a `<wsrm:CreateSequence>` request message:

```
655 <S11:Envelope>
656   <S11:Body>
657     <S11:Fault>
658       <faultcode> [Subcode] </faultcode>
659       <faultstring xml:lang="en"> [Reason] </faultstring>
660     </S11:Fault>
661   </S11:Body>
662 </S11:Envelope>
```

663 4.1 SequenceFault Element

664 The purpose of the `<wsrm:SequenceFault>` element is to carry the specific details of
665 a fault generated during the reliable messaging specific processing of a message
666 belonging to a Sequence. The `<wsrm:SequenceFault>` container MUST only be used
667 in conjunction with the SOAP1.1 fault mechanism. It MUST NOT be used in
668 conjunction with the SOAP1.2 binding.

669 The following exemplar defines its syntax:

```
670 <wsrm:SequenceFault ...>
671   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
672   ...
673 </wsrm:SequenceFault>
```

674 The following describes the content model of the `SequenceFault` element.

675 `/wsrm:SequenceFault`

676 This is the element containing Sequence information for WS-ReliableMessaging

677 `/wsrm:SequenceFault/wsrm:FaultCode`

678 This element, if present, MUST contain a qualified name from the set of fault codes defined
679 below.

680 `/wsrm:SequenceFault/{any}`

681 This is an extensibility mechanism to allow different (extensible) types of information, based on a
682 schema, to be passed.

683 `/wsrm:SequenceFault/@{any}`

Page : 26 Line : 678 Author : AK 08/16/2005

I assume this is intended to say fault [subcode]. Is that correct?

684 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
685 to the element.

686 **4.2 Sequence Terminated**

687 This fault is sent by either the RM Source or the RM Destination to indicate that the
688 endpoint that generated the fault has either encountered an unrecoverable condition,
689 or has detected a violation of the protocol and as a consequence, has chosen to
690 terminate the sequence. The endpoint that generates this fault should make every
691 reasonable effort to notify the corresponding endpoint of this decision.

692 Properties:

693 [Code] Sender or Receiver

694 [Subcode] wsrn:SequenceTerminated

695 [Reason] The Sequence has been terminated due to an unrecoverable error.

696 [Detail]

697 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`

698 **4.3 Unknown Sequence**

699 This fault is sent by either the RM Source or the RM Destination in response to a
700 message containing an unknown sequence identifier.

701 Properties:

702 [Code] Sender

703 [Subcode] wsrn:UnknownSequence

704 [Reason] The value of wsrn:Identifier is not a known Sequence identifier.

705 [Detail]

706 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`

707 **4.4 Invalid Acknowledgement**

708 This fault is sent by the RM Source in response to a
709 `<wsrm:SequenceAcknowledgement>` that violates the cumulative acknowledgement
710 invariant. An example of such a violation would be a `SequenceAcknowledgement`
711 covering messages that have not been sent.

712 [Code] Sender

713 [Subcode] wsrn:InvalidAcknowledgement

714 [Reason] The SequenceAcknowledgement violates the cumulative acknowledgement
715 invariant.

716 [Detail]

717 `<wsrm:SequenceAcknowledgement ...> ... </wsrm:SequenceAcknowledgement>`

718 **4.5 Message Number Rollover**

719 This fault is sent to indicate that message numbers for a sequence have been
720 exhausted. It is an unrecoverable error and terminates the Sequence.

721 Properties:

722 [Code] Sender

723 [Subcode] wsrn:MessageNumberRollover

724 [Reason] The maximum value for wsrn:MessageNumber has been exceeded.

725 [Detail]

726 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`

727 **4.6 Last Message Number Exceeded**

728 This fault is sent by an RM Destination to indicate that it has received a message that
729 has a `<wsrm:MessageNumber>` within a Sequence that exceeds the value of the
730 `<wsrm:MessageNumber>` element that accompanied a `<wsrm>LastMessage>` element
731 for the Sequence. This is an unrecoverable error and terminates the Sequence.

732 Properties:

733 [Code] Sender

734 [Subcode] wsrn:LastMessageNumberExceeded

735 [Reason] The value for wsrn:MessageNumber exceeds the value of the
736 MessageNumber accompanying a LastMessage element in this Sequence.

737 [Detail]

738 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`

739 **4.7 Create Sequence Refused**

740 This fault is sent in response to a create sequence request that cannot be satisfied.

741 Properties:

742 [Code] Sender

743 [Subcode] wsrn:CreateSequenceRefused

744 [Reason] The create sequence request has been refused by the RM Destination.

745 [Detail] empty

746 **5 Security Considerations**

747 It is strongly recommended that the communication between services be secured
748 using the mechanisms described in WS-Security. In order to properly secure
749 messages, the body and all relevant headers need to be included in the signature.
750 Specifically, the `<wsrm:Sequence>` header needs to be signed with the body in order
751 to "bind" the two together. The `<wsrm:SequenceAcknowledgement>` header may be
752 signed independently because a reply independent of the message is not a security
753 concern.

754 Because Sequences are expected to exchange a number of messages, it is
755 recommended that a security context be established using the mechanisms described
756 in WS-Trust and WS-SecureConversation. If a Sequence is bound to a specific
757 endpoint, then the security context needs to be established or shared with the
758 endpoint servicing the Sequence. While the context can be established at any time,
759 it is critical that the messages establishing the Sequence be secured even if they
760 precede security context establishment. However, it is recommended that the
761 security context be established first. Security contexts are independent of reliable
762 messaging Sequences. Consequently, security contexts can come and go
763 independent of the lifetime of the Sequence. In fact, it is recommended that the
764 lifetime of a security context be less than the lifetime of the Sequence unless the
765 Sequence is very short-lived.

766 It is common for message Sequences to exchange a number of messages (or a large
767 amount of data). As a result, the usage profile of a Sequence is such that it is
768 susceptible to key attacks. For this reason it is strongly recommended that the keys
769 be changed frequently. This "re-keying" can be effected a number of ways. The
770 following list outlines four common techniques:

- 771 • Closing and re-establishing a security context
- 772 • Exchanging new secrets between the parties
- 773 • Using a derived key sequence and switch "generations"
- 774 • Attaching a nonce to each message and using it in a derived key function with the shared
775 secret

776 The security context may be re-established using the mechanisms described in WS-
777 Trust and WS-SecureConversation. Similarly, secrets can be exchanged using the
778 mechanisms described in WS-Trust. Note, however, that the current shared secret
779 should not be used to encrypt the new shared secret. Derived keys, the preferred
780 solution from this list, can be specified using the mechanisms described in WS-
781 SecureConversation.

782 There is a core tension between security and reliable messaging that can be
783 problematic if not considered in implementations. That is, one aspect of security is
784 to prevent message replay and the core tenet of reliable messaging is to replay
785 messages until they are acknowledged. Consequently, if the security sub-system
786 processes a message but a failure occurs before the reliable messaging sub-system
787 records the message (or the message is considered "processed"), then it is possible
788 (and likely) that the security sub-system will treat subsequent copies as replays and
789 discard them. At the same time, the reliable messaging sub-system will likely
790 continue to expect and even solicit the missing message(s). Care should be taken to
791 avoid and prevent this rare condition.

792 The following list summarizes common classes of attacks that apply to this protocol
793 and identifies the mechanism to prevent/mitigate the attacks:

- 794 • **Message alteration** – Alteration is prevented by including signatures of the message
795 information using WS-Security.
- 796 • **Message disclosure** – Confidentiality is preserved by encrypting sensitive data using WS-
797 Security.
- 798 • **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by
799 comparing secured policies – see WS-Policy and WS-SecurityPolicy).
- 800 • **Authentication** – Authentication is established using the mechanisms described in WS-
801 Security and WS-Trust. Each message is authenticated using the mechanisms described in
802 WS-Security.
- 803 • **Accountability** – Accountability is a function of the type of and string of the key and
804 algorithms being used. In many cases, a strong symmetric key provides sufficient
805 accountability. However, in some environments, strong PKI signatures are required.
- 806 • **Availability** – All reliable messaging services are subject to a variety of availability attacks.
807 Replay detection is a common attack and it is recommended that this be addressed by the
808 mechanisms described in WS-Security. (Note that because of legitimate message replays,
809 detection should include a differentiator besides message id such as a timestamp). Other
810 attacks, such as network-level denial of service attacks are harder to avoid and are outside
811 the scope of this specification. That said, care should be taken to ensure that minimal state is
812 saved prior to any authenticating sequences.

813 **6 References**

814 **6.1 Normative**

815 **[KEYWORDS]**

816 S. Bradner, "[Key words for use in RFCs to Indicate Requirement Levels](#)," RFC 2119, Harvard
817 University, March 1997

818 **[SOAP]**

819 W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May 2000.

820 **[URI]**

821 T. Berners-Lee, R. Fielding, L. Masinter, "[Uniform Resource Identifiers \(URI\): Generic Syntax](#),"
822 RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.

823 **[XML-ns]**

824 W3C Recommendation, "[Namespaces in XML](#)," 14 January 1999.

825 **[XML-Schema1]**

826 W3C Recommendation, "[XML Schema Part 1: Structures](#)," 2 May 2001.

827 **[XML-Schema2]**

828 W3C Recommendation, "[XML Schema Part 2: Datatypes](#)," 2 May 2001.

829 **[WSSecurity]**

830 "[OASIS Web Services Security: SOAP Message Security 1.0 \(WS-Security 2004\)](#)",
831 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds, OASIS
832 Standard 200401, March 2004.

833 **[SecureConversation]**

834 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#),"
835 May 2004.

836 **[Tanenbaum]**

837 "Computer Networks," Andrew S. Tanenbaum, Prentice Hall PTR, 2003.

838 **[WSDL]**

839 W3C Note, "[Web Services Description Language \(WSDL 1.1\)](#)," 15 March 2001.

840 **[WS-Addressing]**

841 D. Box, et al, "[Web Services Addressing \(WS-Addressing\)](#)," August 2004.

Page : 32 Line : 813 Author : AK 08/16/2005

The reference style is inconsistent. Sometimes the author name is listed first, sometimes it is title first.

Page : 32 Line : 818 Author : AK 08/16/2005

need a soap 1.2 ref too

Page : 32 Line : 820 Author : AK 08/16/2005

never used. need to include this (or IRI) ref where ever 2396 is used

Page : 32 Line : 825 Author : AK 08/16/2005

never used. this could be reference where we talk about schema

Page : 32 Line : 827 Author : AK 08/16/2005

never used. could be used where we talk about schema types

Page : 32 Line : 833 Author : AK 08/16/2005

this isn't used either. Why is this a normative reference?

Page : 32 Line : 836 Author : AK 08/16/2005

should be removed, never used

842 **6.2 Non-Normative**

843 **[WS-Policy]**

844 D. Box, et al, "[Web Services Policy Framework \(WS-Policy\)](#)," September 2004.

845 **[WS-PolicyAttachment]**

846 D. Box, et al, "[Web Services Policy Attachment \(WS-PolicyAttachment\)](#)," September 2004.

847 **[SecurityPolicy]**

848 G. Della-Libra, "[Web Services Security Policy Language \(WS-SecurityPolicy\)](#)," December 2002.

849

850

Page : 33 Line : 842 Author : AK 08/16/2005
none of these references are used

851 Appendix A.Schema

852 The normative schema for WS-ReliableMessaging is located at:

853 `http://schemas.xmlsoap.org/ws/2005/02/rm/wsrn.xsd`

854 The following copy is provided for reference.

```
855 <xs:schema targetNamespace="http://schemas.xmlsoap.org/ws/2005/02/rm"
856 xmlns:xs="http://www.w3.org/2001/XMLSchema"
857 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
858 xmlns:wsm="http://schemas.xmlsoap.org/ws/2005/02/rm"
859 elementFormDefault="qualified" attributeFormDefault="unqualified">
860   <xs:import
861 namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
862 schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
863   <!-- Protocol Elements -->
864   <xs:complexType name="SequenceType">
865     <xs:sequence>
866       <xs:element ref="wsm:Identifier"/>
867       <xs:element name="MessageNumber" type="xs:unsignedLong"/>
868       <xs:element name="LastMessage" minOccurs="0">
869         <xs:complexType>
870           <xs:sequence/>
871         </xs:complexType>
872       </xs:element>
873       <xs:any namespace="##other" processContents="lax" minOccurs="0"
874 maxOccurs="unbounded"/>
875     </xs:sequence>
876     <xs:anyAttribute namespace="##other" processContents="lax"/>
877   </xs:complexType>
878   <xs:element name="Sequence" type="wsm:SequenceType"/>
879   <xs:element name="SequenceAcknowledgement">
880     <xs:complexType>
881       <xs:sequence>
882         <xs:element ref="wsm:Identifier"/>
883         <xs:choice>
884           <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
885             <xs:complexType>
886               <xs:sequence/>
```

Page : 34 Line : 860 Author : AK 08/16/2005
why is this import needed?

Page : 34 Line : 880 Author : AK 08/16/2005
all other types are non-anon types. Why is this an exception? for consistency I would suggest making this a non-anon type

```

887         <xs:attribute name="Upper" type="xs:unsignedLong"
888 use="required"/>
889         <xs:attribute name="Lower" type="xs:unsignedLong"
890 use="required"/>
891         <xs:anyAttribute namespace="##other"
892 processContents="lax"/>
893     </xs:complexType>
894 </xs:element>
895     <xs:element name="Nack" type="xs:unsignedLong"
896 minOccurs="unbounded"/>
897 </xs:choice>
898     <xs:any namespace="##other" processContents="lax" minOccurs="0"
899 minOccurs="unbounded"/>
900 </xs:sequence>
901     <xs:anyAttribute namespace="##other" processContents="lax"/>
902 </xs:complexType>
903 </xs:element>
904 <xs:complexType name="AckRequestedType">
905     <xs:sequence>
906         <xs:element ref="wsrm:Identifier"/>
907         <xs:element name="MaxMessageNumberUsed" type="xs:unsignedLong"
908 minOccurs="0"/>
909         <xs:any namespace="##other" processContents="lax" minOccurs="0"
910 minOccurs="unbounded"/>
911     </xs:sequence>
912     <xs:anyAttribute namespace="##other" processContents="lax"/>
913 </xs:complexType>
914 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
915 <xs:element name="Identifier">
916     <xs:complexType>
917         <xs:annotation>
918             <xs:documentation>
919 This type is for elements whose [children] is an anyURI and can have
920 arbitrary attributes.
921             </xs:documentation>
922         </xs:annotation>
923     <xs:simpleContent>
924         <xs:extension base="xs:anyURI">
925             <xs:anyAttribute namespace="##other" processContents="lax"/>
926         </xs:extension>
927     </xs:simpleContent>

```


Page : 35 Line : 887 Author : AK 08/16/2005
should we restrict the unsignedLongs to be > 0?

Page : 35 Line : 907 Author : AK 08/16/2005
The spec uses wsrn:MessageNumber not wsrn:MaxMessageNumberUsed. The spec also says that if there is a diff between the schema and the spec then the spec wins. But I'm not sure if this is true in this case.

```

928     </xs:complexType>
929 </xs:element>
930 <!-- Fault Container and Codes -->
931 <xs:simpleType name="FaultCodes">
932     <xs:restriction base="xs:QName">
933         <xs:enumeration value="wsrm:UnknownSequence"/>
934         <xs:enumeration value="wsrm:SequenceTerminated"/>
935         <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
936         <xs:enumeration value="wsrm:MessageNumberRollover"/>
937         <xs:enumeration value="wsrm:CreateSequenceRefused"/>
938         <xs:enumeration value="wsrm:LastMessageNumberExceeded"/>
939     </xs:restriction>
940 </xs:simpleType>
941 <xs:complexType name="SequenceFaultType">
942     <xs:sequence>
943         <xs:element name="FaultCode" type="xs:QName"/>
944         <xs:any namespace="##any" processContents="lax" minOccurs="0"
945 maxOccurs="unbounded"/>
946     </xs:sequence>
947     <xs:anyAttribute namespace="##any" processContents="lax"/>
948 </xs:complexType>
949 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
950 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
951 <xs:element name="CreateSequenceResponse"
952 type="wsrm:CreateSequenceResponseType"/>
953 <xs:element name="TerminateSequence"
954 type="wsrm:TerminateSequenceType"/>
955 <xs:complexType name="CreateSequenceType">
956     <xs:sequence>
957         <xs:element ref="wsrm:AcksTo"/>
958         <xs:element ref="wsrm:Expires" minOccurs="0"/>
959         <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
960         <xs:any namespace="##other" processContents="lax" minOccurs="0"
961 maxOccurs="unbounded">
962             <xs:annotation>
963                 <xs:documentation>
964 It is the authors intent that this extensibility be used to transfer a
965 Security Token Reference as defined in WS-Security.
966 </xs:documentation>
967             </xs:annotation>
968         </xs:any>

```

Page : 36 Line : 931 Author : AK 08/16/2005
for consistency should we call this FaultCodeType

Page : 36 Line : 943 Author : AK 08/16/2005
should this be tns:FaultCodes instead of xs:QName?

Page : 36 Line : 963 Author : AK 08/16/2005
Schema does not contain SecurityTokenReference but the spec does

```

969     </xs:sequence>
970     <xs:anyAttribute namespace="##other" processContents="lax"/>
971 </xs:complexType>
972 <xs:complexType name="CreateSequenceResponseType">
973     <xs:sequence>
974         <xs:element ref="wsrm:Identifier"/>
975         <xs:element ref="wsrm:Expires" minOccurs="0"/>
976         <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
977         <xs:any namespace="##other" processContents="lax" minOccurs="0"
978 maxOccurs="unbounded"/>
979     </xs:sequence>
980     <xs:anyAttribute namespace="##other" processContents="lax"/>
981 </xs:complexType>
982 <xs:complexType name="TerminateSequenceType">
983     <xs:sequence>
984         <xs:element ref="wsrm:Identifier"/>
985         <xs:any namespace="##other" processContents="lax" minOccurs="0"
986 maxOccurs="unbounded"/>
987     </xs:sequence>
988     <xs:anyAttribute namespace="##other" processContents="lax"/>
989 </xs:complexType>
990 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
991 <xs:complexType name="OfferType">
992     <xs:sequence>
993         <xs:element ref="wsrm:Identifier"/>
994         <xs:element ref="wsrm:Expires" minOccurs="0"/>
995         <xs:any namespace="##other" processContents="lax" minOccurs="0"
996 maxOccurs="unbounded"/>
997     </xs:sequence>
998     <xs:anyAttribute namespace="##other" processContents="lax"/>
999 </xs:complexType>
1000 <xs:complexType name="AcceptType">
1001     <xs:sequence>
1002         <xs:element ref="wsrm:AcksTo"/>
1003         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1004 maxOccurs="unbounded"/>
1005     </xs:sequence>
1006     <xs:anyAttribute namespace="##other" processContents="lax"/>
1007 </xs:complexType>
1008 <xs:element name="Expires">
1009     <xs:complexType>

```

```
1010     <xs:simpleContent>
1011         <xs:extension base="xs:duration">
1012             <xs:anyAttribute namespace="##other" processContents="lax"/>
1013         </xs:extension>
1014     </xs:simpleContent>
1015 </xs:complexType>
1016 </xs:element>
1017 </xs:schema>
```

1018 **Appendix B.Message Examples**

1019 B.1.Create Sequence

1020 Create Sequence

```
1021 <?xml version="1.0" encoding="UTF-8"?>
1022 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1023 xmlns:wsm="http://schemas.xmlsoap.org/ws/2005/02/rm"
1024 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1025   <S:Header>
1026     <wsa:MessageID>
1027       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
1028     </wsa:MessageID>
1029     <wsa:To>http://example.com/serviceB/123</wsa:To>
1030     <wsa:Action>http://schemas.xmlsoap.org/ws/2005/02/rm/CreateSequence
1031   </wsa:Action>
1032     <wsa:ReplyTo>
1033       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1034     </wsa:ReplyTo>
1035   </S:Header>
1036   <S:Body>
1037     <wsm:CreateSequence>
1038       <wsm:AcksTo>
1039         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1040       </wsm:AcksTo>
1041     </wsm:CreateSequence>
1042   </S:Body>
1043 </S:Envelope>
```

1044 Create Sequence Response

```
1045 <?xml version="1.0" encoding="UTF-8"?>
1046 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1047 xmlns:wsm="http://schemas.xmlsoap.org/ws/2005/02/rm"
1048 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1049   <S:Header>
1050     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1051     <wsa:RelatesTo>
1052       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
1053     </wsa:RelatesTo>
1054     <wsa:Action>
```

```
1055     http://schemas.xmlsoap.org/ws/2005/02/rm/CreateSequenceResponse
1056     </wsa:Action>
1057 </S:Header>
1058 <S:Body>
1059     <wsrm:CreateSequenceResponse>
1060         <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1061     </wsrm:CreateSequenceResponse>
1062 </S:Body>
1063 </S:Envelope>
```


1064 B.2. Initial Transmission

1065 The following example WS-ReliableMessaging headers illustrate the message
1066 exchange in the above figure. The three messages have the following headers; the
1067 third message is identified as the last message in the sequence:

1068 Message 1

```
1069 <?xml version="1.0" encoding="UTF-8"?>
1070 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1071 xmlns:wsm="http://schemas.xmlsoap.org/ws/2005/02/rm"
1072 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1073   <S:Header>
1074     <wsa:MessageID>
1075       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
1076     </wsa:MessageID>
1077     <wsa:To>http://example.com/serviceB/123</wsa:To>
1078     <wsa:From>
1079       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1080     </wsa:From>
1081     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1082     <wsm:Sequence>
1083       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1084       <wsm:MessageNumber>1</wsm:MessageNumber>
1085     </wsm:Sequence>
1086   </S:Header>
1087   <S:Body>
1088     <!-- Some Application Data -->
1089   </S:Body>
1090 </S:Envelope>
```

1091 Message 2

```
1092 <?xml version="1.0" encoding="UTF-8"?>
1093 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1094 xmlns:wsm="http://schemas.xmlsoap.org/ws/2005/02/rm"
1095 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1096   <S:Header>
1097     <wsa:MessageID>
1098       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1099     </wsa:MessageID>
1100     <wsa:To>http://example.com/serviceB/123</wsa:To>
```

```
1101     <wsa:From>
1102         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1103     </wsa:From>
1104     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1105     <wsrm:Sequence>
1106         <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1107         <wsrm:MessageNumber>2</wsrm:MessageNumber>
1108     </wsrm:Sequence>
1109 </S:Header>
1110 <S:Body>
1111     <!-- Some Application Data -->
1112 </S:Body>
1113 </S:Envelope>
```

1114 Message 3

```
1115 <?xml version="1.0" encoding="UTF-8"?>
1116 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1117     xmlns:wsrm="http://schemas.xmlsoap.org/ws/2005/02/rm"
1118     xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1119     <S:Header>
1120         <wsa:MessageID>
1121             http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1122         </wsa:MessageID>
1123         <wsa:To>http://example.com/serviceB/123</wsa:To>
1124         <wsa:From>
1125             <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1126         </wsa:From>
1127         <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1128         <wsrm:Sequence>
1129             <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1130             <wsrm:MessageNumber>3</wsrm:MessageNumber>
1131             <wsrm:LastMessage/>
1132         </wsrm:Sequence>
1133     </S:Header>
1134     <S:Body>
1135         <!-- Some Application Data -->
1136     </S:Body>
1137 </S:Envelope>
```

1138 B.3.First Acknowledgement

1139 Message number 2 has not been received by the RM Destination due to some
1140 transmission error so it responds with an acknowledgement for messages 1 and 3:

```
1141 <?xml version="1.0" encoding="UTF-8"?>
1142 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1143 xmlns:wsmr="http://schemas.xmlsoap.org/ws/2005/02/rm"
1144 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1145   <S:Header>
1146     <wsa:MessageID>
1147       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1148     </wsa:MessageID>
1149     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1150     <wsa:From>
1151       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1152     </wsa:From>
1153     <wsa:Action>
1154       http://schemas.xmlsoap.org/ws/2005/02/rm/SequenceAcknowledgement
1155     </wsa:Action>
1156     <wsmr:SequenceAcknowledgement>
1157       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1158       <wsmr:AcknowledgementRange Upper="1" Lower="1"/>
1159       <wsmr:AcknowledgementRange Upper="3" Lower="3"/>
1160     </wsmr:SequenceAcknowledgement>
1161   </S:Header>
1162   <S:Body/>
1163 </S:Envelope>
```

1164 B.4.Retransmission

1165 The sending endpoint discovers that message number 2 was not received so it
1166 resends the message and requests an acknowledgement:

```
1167 <?xml version="1.0" encoding="UTF-8"?>
1168 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1169 xmlns:wsmr="http://schemas.xmlsoap.org/ws/2005/02/rm"
1170 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1171   <S:Header>
1172     <wsa:MessageID>
1173       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1174     </wsa:MessageID>
1175     <wsa:To>http://example.com/serviceB/123</wsa:To>
1176     <wsa:From>
1177       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1178     </wsa:From>
1179     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1180     <wsmr:Sequence>
1181       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1182       <wsmr:MessageNumber>2</wsmr:MessageNumber>
1183     </wsmr:Sequence>
1184     <wsmr:AckRequested>
1185       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1186     </wsmr:AckRequested>
1187   </S:Header>
1188   <S:Body>
1189     <!-- Some Application Data -->
1190   </S:Body>
1191 </S:Envelope>
```

1192 B.5.Termination

1193 The RM Destination now responds with an acknowledgement for the complete
1194 sequence which can then be terminated:

```
1195 <?xml version="1.0" encoding="UTF-8"?>
1196 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1197 xmlns:wsmr="http://schemas.xmlsoap.org/ws/2005/02/rm"
1198 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1199   <S:Header>
1200     <wsa:MessageID>
1201       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1202     </wsa:MessageID>
1203     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1204     <wsa:From>
1205       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1206     </wsa:From>
1207     <wsa:Action>
1208       http://schemas.xmlsoap.org/ws/2005/02/rm/SequenceAcknowledgement
1209     </wsa:Action>
1210     <wsmr:SequenceAcknowledgement>
1211       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1212       <wsmr:AcknowledgementRange Upper="3" Lower="1"/>
1213     </wsmr:SequenceAcknowledgement>
1214   </S:Header>
1215   <S:Body/>
1216 </S:Envelope>
```

1217 Terminate Sequence

```
1218 <?xml version="1.0" encoding="UTF-8"?>
1219 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1220 xmlns:wsmr="http://schemas.xmlsoap.org/ws/2005/02/rm"
1221 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1222   <S:Header>
1223     <wsa:MessageID>
1224       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1225     </wsa:MessageID>
1226     <wsa:To>http://example.com/serviceB/123</wsa:To>
1227     <wsa:Action>
1228       http://schemas.xmlsoap.org/ws/2005/02/rm/TerminateSequence
1229     </wsa:Action>
```

```
1230     <wsa:From>
1231         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1232     </wsa:From>
1233 </S:Header>
1234 <S:Body>
1235     <wsrm:TerminateSequence>
1236         <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1237     </wsrm:TerminateSequence>
1238 </S:Body>
1239 </S:Envelope>
```

1240 Appendix C.WSDL

1241 The non-normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

1242 <http://schemas.xmlsoap.org/ws/2005/02/rm/wsd1/wsrn.wsd1>

1243 The following non-normative copy is provided for reference.

```
1244 <wSDL:definitions xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
1245 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1246 xmlns:wSA="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1247 xmlns:rm="http://schemas.xmlsoap.org/ws/2005/02/rm"
1248 xmlns:tns="http://schemas.xmlsoap.org/ws/2005/02/rm/wsd1"
1249 targetNamespace="http://schemas.xmlsoap.org/ws/2005/02/rm/wsd1">
1250 <wSDL:types>
1251   <xs:schema>
1252     <xs:import namespace="http://schemas.xmlsoap.org/ws/2005/02/rm"
1253     schemaLocation="http://schemas.xmlsoap.org/ws/2005/02/rm/wsrn.xsd"/>
1254     <xs:import
1255     namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1256     schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
1257   </xs:schema>
1258 </wSDL:types>
1259 <wSDL:message name="CreateSequence">
1260   <wSDL:part name="create" element="rm:CreateSequence"/>
1261 </wSDL:message>
1262 <wSDL:message name="CreateSequenceResponse">
1263   <wSDL:part name="createResponse"
1264   element="rm:CreateSequenceResponse"/>
1265 </wSDL:message>
1266 <wSDL:message name="TerminateSequence">
1267   <wSDL:part name="terminate" element="rm:TerminateSequence"/>
1268 </wSDL:message>
1269 <wSDL:portType name="SequenceAbstractPortType">
1270   <wSDL:operation name="CreateSequence">
1271     <wSDL:input message="tns:CreateSequence"
1272     wSA:Action="http://schemas.xmlsoap.org/ws/2005/02/rm/CreateSequence"/>
1273     <wSDL:output message="tns:CreateSequenceResponse"
1274     wSA:Action="http://schemas.xmlsoap.org/ws/2005/02/rm/CreateSequenceResponse"/>
1275   </wSDL:operation>
```

Page : 48 Line : 1254 Author : AK 08/16/2005
why is this imported? It is never used.


```
1277     <wsdl:operation name="TerminateSequence">
1278         <wsdl:input message="tns:TerminateSequence"
1279 wsa:Action="http://schemas.xmlsoap.org/ws/2005/02/rm/CreateSequenceResp
1280 onse"/>
1281     </wsdl:operation>
1282 </wsdl:portType>
1283 </wsdl:definitions>
```

1284 **Appendix D.Acknowledgments**

1285 This document is based on initial contribution to OASIS WS-RX Technical Committee by the
1286 following authors: Ruslan Bilorusets, BEA, Don Box, Microsoft, Luis Felipe Cabrera, Microsoft,
1287 Doug Davis, IBM, Donald Ferguson, IBM, Christopher Ferris, IBM (Editor), Tom Freund, IBM,
1288 Mary Ann Hondo, IBM, John Ibbotson, IBM, Lei Jin, BEA, Chris Kaler, Microsoft, David
1289 Langworthy, Microsoft (Editor), Amelia Lewis, TIBCO Software, Rodney Limprecht, Microsoft,
1290 Steve Lucco, Microsoft, Don Mullen, TIBCO Software, Anthony Nadalin, IBM, Mark Nottingham,
1291 BEA, David Orchard, BEA, Jamie Roots, IBM, Shivajee Samdarshi, TIBCO Software, John
1292 Shewchuk, Microsoft, Tony Storey, IBM

1293 The following individuals have provided invaluable input into the initial contribution:

1294 Keith Ballinger, Microsoft, Stefan Batres, Microsoft, Allen Brown, Microsoft, Michael Conner, IBM,
1295 George Copeland, Microsoft, Francisco Curbera, IBM, Paul Fremantle, IBM, Steve Graham, IBM,
1296 Pat Helland, Microsoft, Rick Hill, Microsoft, Scott Hinkelman, IBM, Tim Holloway, IBM, Efim Hudis,
1297 Microsoft, Gopal Kakivaya, Microsoft, Johannes Klein, Microsoft, Frank Leymann, IBM, Martin
1298 Nally, IBM, Peter Niblett, IBM, Jeffrey Schlimmer, Microsoft, James Snell, IBM, Keith Stobie,
1299 Microsoft, Satish Thatte, Microsoft, Stephen Todd, IBM, Sanjiva Weerawarana, IBM, Roger
1300 Wolter, Microsoft

1301 The following individuals were members of the committee during the development of this
1302 specification:

1303 TBD

1304 **Appendix E.Revision History**

Rev	Date	By Whom	What
wd-01	2005-07-07	Christopher Ferris	Initial version created based on submission by the authors.

1305 **Appendix F.Notices**

1306 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
1307 that might be claimed to pertain to the implementation or use of the technology described in this
1308 document or the extent to which any license under such rights might or might not be available;
1309 neither does it represent that it has made any effort to identify any such rights. Information on
1310 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
1311 website. Copies of claims of rights made available for publication and any assurances of licenses
1312 to be made available, or the result of an attempt made to obtain a general license or permission
1313 for the use of such proprietary rights by implementors or users of this specification, can be
1314 obtained from the OASIS Executive Director.

1315 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
1316 applications, or other proprietary rights which may cover technology that may be required to
1317 implement this specification. Please address the information to the OASIS Executive Director.

1318 Copyright (C) OASIS Open (2005). All Rights Reserved.

1319 This document and translations of it may be copied and furnished to others, and derivative works
1320 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
1321 published and distributed, in whole or in part, without restriction of any kind, provided that the
1322 above copyright notice and this paragraph are included on all such copies and derivative works.
1323 However, this document itself may not be modified in any way, such as by removing the copyright
1324 notice or references to OASIS, except as needed for the purpose of developing OASIS
1325 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
1326 Property Rights document must be followed, or as required to translate it into languages other
1327 than English.

1328 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
1329 successors or assigns.

1330 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1331 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
1332 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
1333 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
1334 PARTICULAR PURPOSE.