



1 Web Services ReliableMessaging 2 (WS-Reliable Messaging)

3 Committee Draft 02, January 23, 2006

4 **Document identifier:**

5 wsrn-1.1-spec-cd-02

6 **Location:**

7 <http://docs.oasis-open.org/ws-rx/wsrn/200510/wsrn-1.1-spec-cd-02.pdf>

8 **Editors:**

9 Gilbert Pilz, BEA <gilbert.pilz@bea.com>
10 Doug Davis, IBM <dug@us.ibm.com>
11 Anish Karmarkar, Oracle <Anish.Karmarkar@oracle.com>
12 Steve Winkler, SAP <steve.winkler@sap.com>
13 Ümit Yalçınalp, SAP <umit.yalcinalp@sap.com>

14 **Contributors:**

15 TBD

16 **Abstract:**

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

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

25 By using the XML [[XML](#)], SOAP [[SOAP 1.1](#)], [[SOAP 1.2](#)] and WSDL [[WSDL 1.1](#)] extensibility model,
26 SOAP-based and WSDL-based specifications are designed to be composed with each other to define a
27 rich Web services environment. As such, WS-ReliableMessaging by itself does not define all the features
28 required for a complete messaging solution. WS-ReliableMessaging is a building block that is used in
29 conjunction with other specifications and application-specific protocols to accommodate a wide variety of
30 protocols related to the operation of distributed Web services.

31 **Status:**

32 This document is a Committee Draft.

33 This document was last revised or approved by the OASIS WS-RX Technical Committee on the above
34 date. The level of approval is also listed above. Check the current location noted above for possible later
35 revisions of this document.

36 For information on whether any patents have been disclosed that may be essential to implementing this
37 specification and any offers of patent licensing terms please refer to the Intellectual Property Rights
38 section of the Technical Committee web page (<http://www.oasis-open.org/committees/ws-rx/ipr.php>).

39 **Table of Contents**

40	1 Introduction.....	4
41	1.1 Goals and Requirements.....	4
42	1.1.1 Requirements.....	4
43	1.2 Notational Conventions.....	4
44	1.3 Namespace.....	4
45	1.4 Compliance.....	5
46	2 Reliable Messaging Model.....	6
47	2.1 Glossary.....	6
48	2.2 Protocol Preconditions.....	7
49	2.3 Protocol Invariants.....	7
50	2.4 Example Message Exchange.....	7
51	3 RM Protocol Elements.....	10
52	3.1 Sequence Creation.....	10
53	3.2 Closing A Sequence.....	13
54	3.3 Sequence Termination.....	14
55	3.4 Sequences.....	15
56	3.5 Request Acknowledgement.....	16
57	3.6 Sequence Acknowledgement.....	16
58	4 Faults.....	20
59	4.1 SequenceFault Element.....	21
60	4.2 Sequence Terminated.....	22
61	4.3 Unknown Sequence.....	22
62	4.4 Invalid Acknowledgement.....	22
63	4.5 Message Number Rollover.....	23
64	4.6 Create Sequence Refused.....	23
65	4.7 Sequence Closed.....	23
66	5 Security Considerations.....	24
67	6 References.....	26
68	6.1 Normative.....	26
69	6.2 Non-Normative.....	26
70	A. Schema.....	28
71	B. Message Examples.....	33
72	B.1 Create Sequence.....	33
73	B.2 Initial Transmission.....	33
74	B.3 First Acknowledgement.....	35
75	B.4 Retransmission.....	35
76	B.5 Termination.....	36

77	C. WSDL.....	38
78	D. Acknowledgments.....	40
79	E. Revision History.....	41
80	F. Notices.....	44

81 1 Introduction

82 It is often a requirement for two Web services that wish to communicate to do so reliably in the presence
83 of software component, system, or network failures. The primary goal of this specification is to create a
84 modular mechanism for reliable delivery of messages. It defines a messaging protocol to identify, track,
85 and manage the reliable delivery of messages between a source and a destination. It also defines a
86 SOAP binding that is required for interoperability. Additional bindings may be defined.

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

90 1.1 Goals and Requirements

91 1.1.1 Requirements

92 1.2 Notational Conventions

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

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

- 97 • The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- 98 • Characters are appended to elements and attributes to indicate cardinality:
 - 99 ○ "?" (0 or 1)
 - 100 ○ "*" (0 or more)
 - 101 ○ "+" (1 or more)
- 102 • The character "|" is used to indicate a choice between alternatives.
- 103 • The characters "[" and "]" are used to indicate that contained items are to be treated as a group
104 with respect to cardinality or choice.
- 105 • An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content
106 specified in this document. Additional children elements and/or attributes MAY be added at the
107 indicated extension points but they MUST NOT contradict the semantics of the parent and/or
108 owner, respectively. If an extension is not recognized it SHOULD be ignored.
- 109 • XML namespace prefixes (See Section [Namespace](#)) are used to indicate the namespace of the
110 element being defined.

111 1.3 Namespace

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

113 `http://docs.oasis-open.org/ws-rx/wsrn/200510`

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

116 The following namespaces are used in this document:

117 *Table 1*

Prefix	Namespace
S	http://www.w3.org/2003/05/soap-envelope
S11	http://schemas.xmlsoap.org/soap/envelope/
wsrn	http://docs.oasis-open.org/ws-rx/wsrn/200510
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing
xs	http://www.w3.org/2001/XMLSchema

118 The normative schema for WS-ReliableMessaging can be found at:

119 <http://docs.oasis-open.org/ws-rx/wsrn/200510/wsrn-1.1.xsd>

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

121 If an action IRI is used, and one is not already defined per the rules of the WS-Addressing specification
122 [WS-Addressing], then the action IRI MUST consist of the WS-RM namespace URI concatenated with a
123 '/', followed by the message element name. For example:

124 `http://docs.oasis-open.org/ws-rx/wsrn/200510/SequenceAcknowledgement`

125 **1.4 Compliance**

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

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

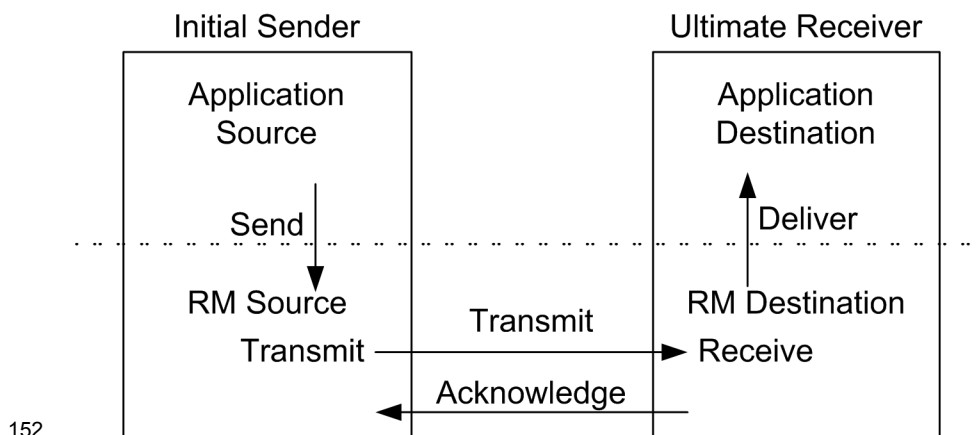
132 2 Reliable Messaging Model

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

135 The WS-ReliableMessaging specification defines an interoperable protocol that requires a Reliable
136 Messaging (RM) Source and Reliable Messaging (RM) Destination to ensure that each message
137 transmitted by the RM Source is successfully received by an RM Destination, or barring successful
138 receipt, that an RM Source can, except in the most extreme circumstances, accurately determine the
139 disposition of each message transmitted as perceived by the RM Destination, so as to resolve any in-
140 doubt status. Note that this specification makes no restriction on the scope of the RM Source or RM
141 Destination entities. For example, either may span multiple WSDL Ports or endpoints.

142 The protocol supports reliability features which include ordered delivery, duplicate elimination, and
143 guaranteed receipt for the RMD. It is expected that the AD and RMD will implement as many of these or
144 as few of these characteristics as necessary to implement the AD. In any case the wire protocol does not
145 change.

146 Figure 1 below illustrates the entities and events in a simple reliable exchange of messages. First, the
147 Application Source Sends a message for reliable delivery. The Reliable Messaging (RM) Source accepts
148 the message and Transmits it one or more times. After receiving the message, the RM Destination
149 Acknowledges it. Finally, the RM Destination delivers the message to the Application Destination. The
150 exact roles the entities play and the complete meaning of the events will be defined throughout this
151 specification.



152
153 Figure 1: Reliable Messaging Model

154 2.1 Glossary

155 The following definitions are used throughout this specification:

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

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

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

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

162 **Endpoint:** As defined in the WS-Addressing specification [WS-Addressing]; a Web service endpoint is a
163 (referenceable) entity, processor, or resource to which Web service messages can be addressed.
164 Endpoint references convey the information needed to address a Web service endpoint.

165 **Receive:** The act of reading a message from a network connection and qualifying it as relevant to RM
166 Destination functions.

167 **RM Destination:** For any one reliable sent message the endpoint that receives the message.

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

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

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

172 **2.2 Protocol Preconditions**

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

- 175 • For any single message exchange the RM Source **MUST** have an endpoint reference that uniquely
176 identifies the RM Destination endpoint.
- 177 • The RM Source **MUST** have knowledge of the destination's policies, if any, and the RM Source
178 **MUST** be capable of formulating messages that adhere to this policy.

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

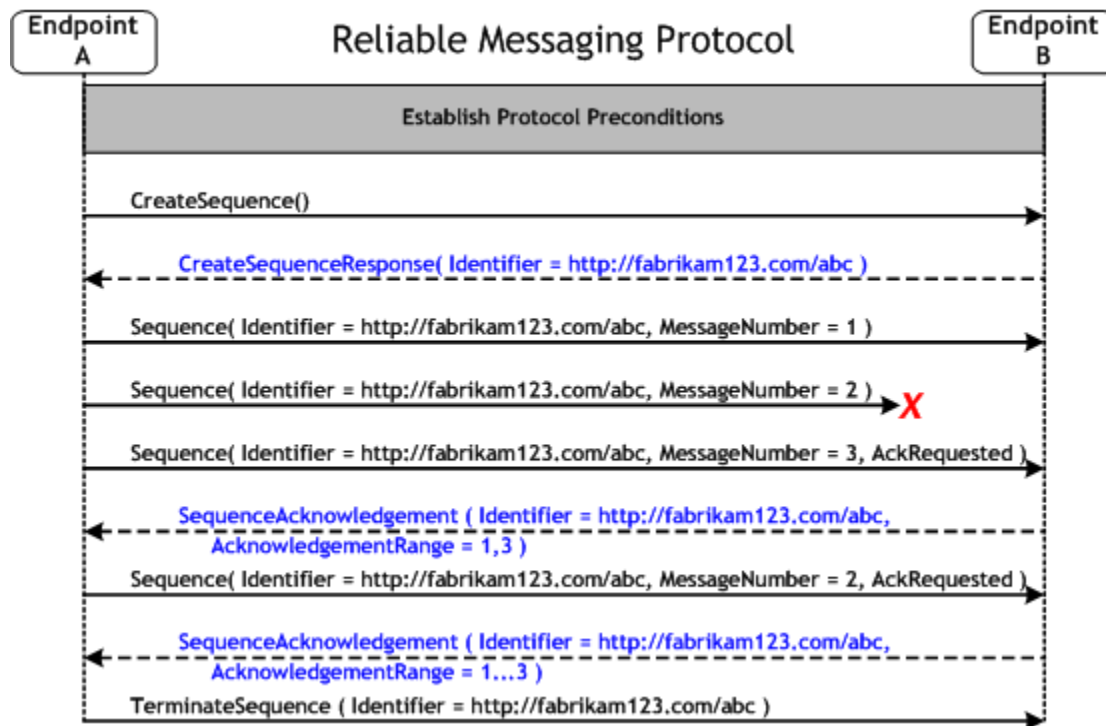
181 **2.3 Protocol Invariants**

182 During the lifetime of a Sequence, two invariants are **REQUIRED** for correctness:

- 183 • The RM Source **MUST** assign each message within a Sequence a message number (defined
184 below) beginning at 1 and increasing by exactly 1 for each subsequent message. These numbers
185 **MUST** be assigned in the same order in which messages are sent by the Application Source.
- 186 • Every acknowledgement issued by the RM Destination **MUST** include within an acknowledgement
187 range or ranges the sequence number of every message successfully received by the RM
188 Destination and **MUST** exclude sequence numbers of any messages not yet received.

189 **2.4 Example Message Exchange**

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



191 Figure 2: The WS-ReliableMessaging Protocol

- 192 1. The protocol preconditions are established. These include policy exchange, endpoint resolution,
193 establishing trust.
- 194 2. The RM Source requests creation of a new Sequence.
- 195 3. The RM Destination creates a Sequence by returning a globally unique identifier.
- 196 4. The RM Source begins sending messages beginning with MessageNumber 1. In the figure above,
197 the RM Source sends 3 messages.
- 198 5. Since the 3rd message is the last in this exchange, the RM Source includes a
199 `<wsrm:AckRequested>` Header.
- 200 6. The 2nd message is lost in transit.
- 201 7. The RM Destination acknowledges receipt of message numbers 1 and 3 as a result of receiving the
202 RM Source's `<wsrm:AckRequested>` Header.
- 203 8. The RM Source retransmits the 2nd message. This is a new message on the underlying transport,
204 but it has the same sequence identifier and message number so the RM Destination can recognize
205 it as equivalent to the earlier message, in case both are received.
- 206 9. The RM Source includes an `<wsrm:AckRequested>` element so the RM Destination will expedite
207 an acknowledgement.
- 208 10. The RM Destination receives the second transmission of the message with MessageNumber 2 and
209 acknowledges receipt of message numbers 1, 2, and 3.
- 210 11. The RM Source receives this acknowledgement and sends a `TerminateSequence` message to the
211 RM Destination indicating that the sequence is completed and reclaims any resources associated
212 with the Sequence.
- 213 12. The RM Destination receives the `TerminateSequence` message indicating that the RM Source will
214 not be sending any more messages, and reclaims any resources associated with the Sequence.
- 215 The RM Source will expect to receive acknowledgements from the RM Destination during the course of a
216 message exchange at occasions described in Section 3 below. Should the acknowledgement not be

217 received in a timely fashion, the RM Source MUST re-transmit the request since either the request or the
218 associated acknowledgement may have been lost. Since the nature and dynamic characteristics of the
219 underlying transport and potential intermediaries are unknown in the general case, the timing of re-
220 transmissions cannot be specified. Additionally, over-aggressive re-transmissions have been
221 demonstrated to cause transport or intermediary flooding which are counterproductive to the intention of
222 providing a reliable exchange of messages. Consequently, implementers are encouraged to utilize
223 adaptive mechanisms that dynamically adjust re-transmission time and the back-off intervals that are
224 appropriate to the nature of the transports and intermediaries envisioned. For the case of TCP/IP
225 transports, a mechanism similar to that described as RTTM in RFC 1323 [RTTM] should be considered.

226 Now that the basic model has been outlined, the details of the elements used in this protocol are now
227 provided in Section 3.

228 3 RM Protocol Elements

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

233 3.1 Sequence Creation

234 The RM Source MUST request creation of an outbound Sequence by sending a
235 `<wsrm:CreateSequence>` element in the body of a message to the RM Destination which in turn
236 responds either with a `<wsrm:CreateSequenceResponse>` or a `CreateSequenceRefused` fault in
237 the body of the response message. `<wsrm:CreateSequence>` MAY carry an offer to create an inbound
238 sequence which is either accepted or rejected in the `<wsrm:CreateSequenceResponse>`. Note,
239 offering a Sequence within the `<wsrm:CreateSequence>` element is simply a protocol optimization.
240 There is no semantic difference between offering a Sequence, and choosing not to offer one and
241 subsequently creating a new Sequence to carry messages from the RM Destination to the RM Source.

242 The following exemplar defines the `<wsrm:CreateSequence>` syntax:

```
243 <wsrm:CreateSequence ...>  
244   <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>  
245   <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
246   <wsrm:Offer ...>  
247     <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
248     <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
249     ...  
250   </wsrm:Offer> ?  
251   ...  
252 </wsrm:CreateSequence>
```

253 `/wsrm:CreateSequence`

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

258 `/wsrm:CreateSequence/wsrm:AcksTo`

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

262 Implementations MUST NOT use an endpoint reference in the AcksTo element that would prevent the
263 sending of Sequence Acknowledgements back to the RM Source. For example, using the WS-Addressing
264 "none" IRI would make it impossible for the RM Destination to ever send Sequence Acknowledgements.

265 `/wsrm:CreateSequence/wsrm:Expires`

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

270 `/wsrm:CreateSequence/wsrm:Expires/@{any}`

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

273 /wsmr:CreateSequence/wsmr:Offer

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

276 /wsmr:CreateSequence/wsmr:Offer/wsmr:Identifier

277 This REQUIRED element MUST contain an absolute URI conformant with RFC3986 that uniquely
278 identifies the offered Sequence.

279 /wsmr:CreateSequence/wsmr:Offer/wsmr:Identifier/@{any}

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

282 /wsmr:CreateSequence/wsmr:Offer/wsmr:Expires

283 This element, if present, of type *xs:duration* specifies the duration for the Sequence. A value of 'PT0S'
284 indicates that the Sequence will never expire. Absence of the element indicates an implied value of
285 'PT0S'.

286 /wsmr:CreateSequence/wsmr:Offer/wsmr:Expires/@{any}

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

289 /wsmr:CreateSequence/wsmr:Offer/{any}

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

292 /wsmr:CreateSequence/wsmr:Offer/@{any}

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

295 /wsmr:CreateSequence/{any}

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

298 /wsmr:CreateSequence/@{any}

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

301 A <wsmr:CreateSequenceResponse> is sent in the body of a response message by an RM
302 Destination in response to receipt of a <wsmr:CreateSequence> request message. It carries the
303 <wsmr:Identifier> of the created Sequence and indicates that the RM Source may begin sending
304 messages in the context of the identified Sequence.

305 The following exemplar defines the <wsmr:CreateSequenceResponse> syntax:

```
306 <wsmr:CreateSequenceResponse ...>  
307   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
308   <wsmr:Expires> xs:duration </wsmr:Expires> ?  
309   <wsmr:Accept ...>  
310     <wsmr:AcksTo ...> wsa:EndpointReferenceType </wsmr:AcksTo>
```

311
312
313
314

```
...
</wsrm:Accept> ?
...
</wsrm:CreateSequenceResponse>
```

315 /wsrm:CreateSequenceResponse

316 This element is sent in the body of the response message in response to a <wsrm:CreateSequence>
317 request message. It indicates that the RM Destination has created a new Sequence at the request of the
318 RM Source. This element MUST NOT be sent as a header block.

319 /wsrm:CreateSequenceResponse/wsrm:Identifier

320 This REQUIRED element MUST contain an absolute URI conformant with RFC3986 of the Sequence that
321 has been created by the RM Destination.

322 /wsrm:CreateSequenceResponse/wsrm:Identifier/@{any}

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

325 /wsrm:CreateSequenceResponse/wsrm:Expires

326 This element, if present, of type `xs:duration` accepts or refines the RM Source's requested duration for
327 the Sequence. A value of 'PT0S' indicates that the Sequence will never expire. Absence of the element
328 indicates an implied value of 'PT0S'. This value MUST be equal to or less than the value requested by the
329 RM Source in the corresponding <wsrm:CreateSequence> message.

330 /wsrm:CreateSequenceResponse/wsrm:Expires/@{any}

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

333 /wsrm:CreateSequenceResponse/wsrm:Accept

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

336 **Note:** If a <wsrm:CreateSequenceResponse> is returned without a child <wsrm:Accept> in response
337 to a <wsrm:CreateSequence> that did contain a child <wsrm:Offer>, then the RM Source MAY
338 immediately reclaim any resources associated with the unused offered Sequence.

339 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo

340 This REQUIRED element, of type `wsa:EndpointReferenceType` as specified by WS-Addressing [WS-
341 Addressing], specifies the endpoint reference to which <wsrm:SequenceAcknowledgement>
342 messages related to the accepted Sequence are to be sent.

343 /wsrm:CreateSequenceResponse/wsrm:Accept/{any}

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

346 /wsrm:CreateSequenceResponse/wsrm:Accept/@{any}

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

349 /wsrm:CreateSequenceResponse/{any}

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

352 /wsrm:CreateSequenceResponse/@{any}

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

355 **3.2 Closing A Sequence**

356 There may be times during the use of an RM Sequence that the RM Source or RM Destination will wish to
357 discontinue using a Sequence. Simply terminating the Sequence discards the state managed by the RM
358 Destination, leaving the RM Source unaware of the final ranges of messages that were successfully
359 delivered to the RM Destination. To ensure that the Sequence ends with a known final state both the RM
360 Source and RM Destination may choose to 'close' the Sequence before terminating it.

361 If the RM Source wishes to close the Sequence then it sends a `<wsrm:CloseSequence>` element, in the
362 body of a message, to the RM Destination. This message indicates that the RM Destination MUST NOT
363 receive any new messages for the specified sequence, other than those already received at the time the
364 `<wsrm:CloseSequence>` element is interpreted by the RMD. Upon receipt of this message, or
365 subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST
366 include a final SequenceAcknowledgement (that MUST include the `<wsrm:Final>` element) header block
367 on each message destined to the RM Source, including the CloseSequenceResponse message and on
368 any Sequence Fault transmitted to the RMS.

369 While the RM Destination MUST NOT receive any new messages for the specified sequence it MUST still
370 process RM protocol messages. For example, it MUST respond to AckRequested, TerminateSequence
371 as well as CloseSequence messages. Note, subsequent CloseSequence messages have no effect on the
372 state of the sequence.

373 In the case where the RM Destination wishes to discontinue use of a sequence it may 'close' the
374 sequence itself. Please see `<wsrm:Final>` above and the SequenceClosed fault below. Note, the
375 SequenceClosed Fault SHOULD be used in place of the SequenceTerminated Fault, whenever possible,
376 to allow the RM Source to still receive Acknowledgements.

377 The following exemplar defines the CloseSequence syntax:

```
378 <wsrm:CloseSequence wsrm:Identifier="xs:anyURI" ...>  
379   ...  
380 </wsrm:CloseSequence>
```

381 /wsrm:CloseSequence

382 This element is sent by an RM Source to indicate that the RM Destination MUST NOT receive any new
383 messages for this sequence. A SequenceClosed fault MUST be generated by the RM Destination when it
384 receives a message for a sequence that is closed.

385 /wsrm:CloseSequence@Identifier

386 This REQUIRED attribute contains an absolute URI conformant with RFC3986 that uniquely identifies the
387 sequence.

388 /wsrm:CloseSequence/{any}

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

391 /wsrm:CloseSequence@{any}

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

394 A `<wsrm:CloseSequenceResponse>` is sent in the body of a response message by an RM Destination
395 in response to receipt of a `<wsrm:CloseSequence>` request message. It indicates that the RM
396 Destination has closed the sequence.

397 The following exemplar defines the `<wsrm:CloseSequenceResponse>` syntax:

```
398 <wsrm:CloseSequenceResponse ...>  
399   ...  
400 </wsrm:CloseSequenceResponse>
```

401 `/wsrm:CloseSequenceResponse`

402 This element is sent in the body of a response message by an RM Destination in response to receipt of a
403 `<wsrm:CloseSequence>` request message. It indicates that the RM Destination has closed the
404 sequence.

405 `/wsrm:CloseSequenceResponse/{any}`

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

408 `/wsrm:CloseSequenceResponse@{any}`

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

411 **3.3 Sequence Termination**

412 When the RM Source has completed its use of the Sequence, it sends a `<wsrm:TerminateSequence>`
413 element, in the body of a message to the RM Destination to indicate that the Sequence is complete, and
414 that it will not be sending any further messages related to the Sequence. The RM Destination can safely
415 reclaim any resources associated with the Sequence upon receipt of the `<wsrm:TerminateSequence>`
416 message. Note, under normal usage the RM source will complete its use of the sequence when all of the
417 messages in the Sequence have been acknowledged. However, the RM Source is free to Terminate or
418 Close a Sequence at any time regardless of the acknowledgement state of the messages.

419 The following exemplar defines the TerminateSequence syntax:

```
420 <wsrm:TerminateSequence ...>  
421   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
422   ...  
423 </wsrm:TerminateSequence>
```

424 `/wsrm:TerminateSequence`

425 This element is sent by an RM Source to indicate it has completed its use of the Sequence, i.e. it MUST
426 NOT send any additional message to the RM Destination referencing this sequence. It indicates that the
427 RM Destination can safely reclaim any resources related to the identified Sequence. This element MUST
428 NOT be sent as a header block.

429 `/wsrm:TerminateSequence/wsrm:Identifier`

430 This REQUIRED element MUST contain an absolute URI conformant with RFC3986 of the Sequence that
431 is being terminated.

432 `/wsrm:TerminateSequence/wsrm:Identifier/@{any}`

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

435 /wsrm:TerminateSequence/{any}

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

438 /wsrm:TerminateSequence/@{any}

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

441 3.4 Sequences

442 The RM protocol uses a <wsrm:Sequence> header block to track and manage the reliable delivery of
443 messages. Messages for which a reliable delivery is required MUST contain a <wsrm:Sequence>
444 header block. Each Sequence MUST have a unique <wsrm:Identifier> element and each message
445 within a Sequence MUST have a <wsrm:MessageNumber> element that increments by 1 from an initial
446 value of 1. These values are contained within a <wsrm:Sequence> header block accompanying each
447 message being delivered in the context of a Sequence.

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

449 A following exemplar defines its syntax:

```
450 <wsrm:Sequence ...>  
451   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
452   <wsrm:MessageNumber> xs:unsignedLong </wsrm:MessageNumber>  
453   ...  
454 </wsrm:Sequence>
```

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

456 /wsrm:Sequence

457 This is the element containing Sequence information for WS-ReliableMessaging. The <wsrm:Sequence>
458 element MUST be understood by the RM Destination. The <wsrm:Sequence> element MUST have a
459 `mustUnderstand` attribute with a value 1/true from the namespace corresponding to the version of
460 SOAP to which the <wsrm:Sequence> SOAP header block is bound.

461 /wsrm:Sequence/wsrm:Identifier

462 This REQUIRED element MUST contain an absolute URI conformant with RFC3986 that uniquely
463 identifies the Sequence.

464 /wsrm:Sequence/wsrm:Identifier/@{any}

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

467 /wsrm:Sequence/wsrm:MessageNumber

468 This REQUIRED element MUST contain an `xs:unsignedLong` representing the ordinal position of the
469 message within a Sequence. Sequence MessageNumbers start at 1 and monotonically increase
470 throughout the Sequence. If the message number exceeds the internal limitations of an RM Source or RM
471 Destination or reaches the maximum value of an `xs:unsignedLong` (18,446,744,073,709,551,615), the RM
472 Source or Destination MUST issue a `MessageNumberRollover` fault.

473 /wsrm:Sequence/{any}

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

476 /wsm:Sequence/@{any}

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

479 The following example illustrates a Sequence header block.

```
480 <wsm:Sequence>  
481   <wsm:Identifier>http://example.com/abc</wsm:Identifier>  
482   <wsm:MessageNumber>10</wsm:MessageNumber>  
483 </wsm:Sequence>
```

484 3.5 Request Acknowledgement

485 The purpose of the <wsm:AckRequested> header block is to signal to the RM Destination that the RM
486 Source is requesting that a <wsm:SequenceAcknowledgement> be returned.

487 The RM Source may request an acknowledgement message from the RM Destination at any time by
488 including an <wsm:AckRequested> header block in the message. An RM Destination that receives a
489 message that contains an <wsm:AckRequested> header block MUST respond with a message
490 containing a <wsm:SequenceAcknowledgement> header block. If a non-mustUnderstand fault occurs
491 when processing an RM Header that was piggy-backed on another message, a fault MUST be generated,
492 but the processing of the original message MUST NOT be affected.

493 The following exemplar defines its syntax:

```
494 <wsm:AckRequested ...>  
495   <wsm:Identifier ...> xs:anyURI </wsm:Identifier>  
  
496   ...  
497 </wsm:AckRequested>
```

498 /wsm:AckRequested

499 This element requests an acknowledgement for the identified sequence.

500 /wsm:AckRequested/wsm:Identifier

501 This REQUIRED element MUST contain an absolute URI, conformant with RFC3986, that uniquely
502 identifies the Sequence to which the request applies.

503 /wsm:AckRequested/wsm:Identifier/@{any}

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

506 /wsm:AckRequested/{any}

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

509 /wsm:AckRequested/@{any}

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

512 3.6 Sequence Acknowledgement

513 The RM Destination informs the RM Source of successful message receipt using a

514 <wsm:SequenceAcknowledgement> header block. The <wsm:SequenceAcknowledgement>

515 header block MAY be transmitted independently or included on return messages. The RM Destination
516 MAY send a `<wsrm:SequenceAcknowledgement>` header block at any point during which the
517 sequence is valid. The timing of acknowledgements can be advertised using policy and
518 acknowledgements can be explicitly requested using the `<wsrm:AckRequested>` directive (see Section
519 [Request Acknowledgement](#)). If a non-mustUnderstand fault occurs when processing an RM Header that
520 was piggy-backed on another message, a fault MUST be generated, but the processing of the original
521 message MUST NOT be affected.

522 The following exemplar defines its syntax:

```
523 <wsrm:SequenceAcknowledgement ...>  
524   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
525   [ [ <wsrm:AcknowledgementRange ...  
526     Upper="xs:unsignedLong"  
527     Lower="xs:unsignedLong"/> +  
  
528     | <wsrm:None/> ]  
529     <wsrm:Final/> ?  
530     | <wsrm:Nack> xs:unsignedLong </wsrm:Nack> + ]  
531  
532   ...  
533 </wsrm:SequenceAcknowledgement>
```

534 The following describes the content model of the `<wsrm:SequenceAcknowledgement>` header block.

535 `/wsrm:SequenceAcknowledgement`

536 This element contains the Sequence acknowledgement information.

537 `/wsrm:SequenceAcknowledgement/wsrm:Identifier`

538 This REQUIRED element MUST contain an absolute URI conformant with RFC3986 that uniquely
539 identifies the Sequence. A message MUST NOT contain multiple `<SequenceAcknowledgement>` header
540 blocks that share the same value for `<Identifier>`.

541 `/wsrm:SequenceAcknowledgement/wsrm:Identifier/@{any}`

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

544 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

545 This OPTIONAL element, if present, can occur 1 or more times. It contains a range of Sequence
546 MessageNumbers successfully received by the RM Destination. The ranges SHOULD NOT overlap. This
547 element MUST NOT be present if a sibling `<wsrm:Nack>` or `<wsrm:None>` element is also present as a
548 child of `<wsrm:SequenceAcknowledgement>`.

549 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Upper`

550 This REQUIRED attribute contains an `xs:unsignedLong` representing the `<wsrm:MessageNumber>` of
551 the highest contiguous message in a Sequence range received by the RM Destination.

552 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Lower`

553 This REQUIRED attribute contains an `xs:unsignedLong` representing the `<wsrm:MessageNumber>` of
554 the lowest contiguous message in a Sequence range received by the RM Destination.

555 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@{any}`

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

558 /wsrm:SequenceAcknowledgement/wsrm:Final

559 This OPTIONAL element, if present, indicates that the RM Destination is not receiving new messages for
560 the specified Sequence. The RM Source can be assured that the ranges of messages acknowledged by
561 this SequenceAcknowledgement header block will not change in the future. This element MUST be
562 present when the Sequence is no longer receiving new message for the specified sequence. Note: this
563 element MUST NOT be used when sending a Nack, it can only be used when sending
564 AcknowledgementRanges or <wsrm:None>.

565 /wsrm:SequenceAcknowledgement/wsrm:Nack

566 This OPTIONAL element, if present, MUST contain an xs:unsignedLong representing the
567 <wsrm:MessageNumber> of an unreceived message in a Sequence. This element permits the gap
568 analysis of the <wsrm:AcknowledgementRange> elements to be performed at the RM Destination
569 rather than at the RM Source which may yield performance benefits in certain environments. The
570 <wsrm:Nack> element MUST NOT be present if a sibling <wsrm:AcknowledgementRange> or
571 <wsrm:None> element is also present as a child of <wsrm:SequenceAcknowledgement>. Upon the
572 receipt of a Nack, an RM Source SHOULD retransmit the message identified by the Nack. The RM
573 Destination MUST NOT issue a <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for
574 a message that it has previously acknowledged within a <wsrm:AcknowledgementRange>. The RM
575 Source SHOULD ignore a <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a
576 message that has previously been acknowledged within a <wsrm:AcknowledgementRange>.

577 /wsrm:SequenceAcknowledgement/wsrm:None

578 This OPTIONAL element, if present, MUST be used when the RM Destination has not received any
579 messages for the specified sequence. The <wsrm:None> element MUST NOT be present if a sibling
580 <wsrm:AcknowledgementRange> or <wsrm:Nack> element is also present as a child of the
581 <wsrm:SequenceAcknowledgement>.

582 /wsrm:SequenceAcknowledgement/{any}

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

585 /wsrm:SequenceAcknowledgement/@{any}

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

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

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

```
590 <wsrm:SequenceAcknowledgement>  
591   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
592   <wsrm:AcknowledgementRange Upper="10" Lower="1"/>  
593 </wsrm:SequenceAcknowledgement>
```

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

```
596 <wsrm:SequenceAcknowledgement>  
597   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
598   <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
599   <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
600   <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
601 </wsrm:SequenceAcknowledgement>
```

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

```
603 <wsrm:SequenceAcknowledgement>  
604     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
605     <wsrm:Nack>3</wsrm:Nack>  
606 </wsrm:SequenceAcknowledgement>
```

607 4 Faults

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

611 Sequence creation uses a CreateSequence, CreateSequenceResponse request-response pattern. Faults
612 for this operation are treated as defined in WS-Addressing. CreateSequenceRefused is a possible fault
613 reply for this operation. UnknownSequence is a fault generated by endpoints when messages carrying RM
614 header blocks targeted at unrecognized sequences are detected, these faults are also treated as defined
615 in WS-Addressing. All other faults in this section relate to the processing of RM header blocks targeted at
616 known sequences and are collectively referred to as sequence faults. Sequence faults SHOULD be sent
617 to the same [destination] as `<wsrm:SequenceAcknowledgement>` messages. These faults are
618 correlated using the Sequence identifier carried in the detail.

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

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

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

625 The definitions of faults use the following properties:

626 [Code] The fault code.

627 [Subcode] The fault subcode.

628 [Reason] The English language reason element.

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

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

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

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

```
633 <S:Envelope>  
634   <S:Header>  
635     <wsa:Action>  
636       http://schemas.xmlsoap.org/ws/2004/08/addressing/fault  
637     </wsa:Action>  
638     <!-- Headers elided for clarity. -->  
639   </S:Header>  
640   <S:Body>  
641     <S:Fault>  
642       <S:Code>  
643         <S:Value> [Code] </S:Value>  
644         <S:Subcode>  
645           <S:Value> [Subcode] </S:Value>  
646         </S:Subcode>  
647       </S:Code>  
648       <S:Reason>
```

```

649     <S:Text xml:lang="en"> [Reason] </S:Text>
650   </S:Reason>
651   <S:Detail>
652     [Detail]
653     ...
654   </S:Detail>
655 </S:Fault>
656 </S:Body>
657 </S:Envelope>

```

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

```

660 <S11:Envelope>
661   <S11:Header>
662     <wsrm:SequenceFault>
663       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
664       ...
665     </wsrm:SequenceFault>
666     <!-- Headers elided for clarity. -->
667   </S11:Header>
668   <S11:Body>
669     <S11:Fault>
670       <faultcode> [Code] </faultcode>
671       <faultstring> [Reason] </faultstring>
672     </S11:Fault>
673   </S11:Body>
674 </S11:Envelope>

```

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

```

677 <S11:Envelope>
678   <S11:Body>
679     <S11:Fault>
680       <faultcode> [Subcode] </faultcode>
681       <faultstring xml:lang="en"> [Reason] </faultstring>
682     </S11:Fault>
683   </S11:Body>
684 </S11:Envelope>

```

685 4.1 SequenceFault Element

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

690 The following exemplar defines its syntax:

```

691 <wsrm:SequenceFault ...>
692   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
693   ...
694 </wsrm:SequenceFault>

```

695 The following describes the content model of the SequenceFault element.

696 /wsrm:SequenceFault

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

698 /wsmr:SequenceFault/wsmr:FaultCode

699 This element, if present, MUST contain a qualified name from the set of fault [Subcodes] defined below.

700 /wsmr:SequenceFault/{any}

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

703 /wsmr:SequenceFault/@{any}

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

706 4.2 Sequence Terminated

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

711 Properties:

712 [Code] Sender or Receiver

713 [Subcode] wsmr:SequenceTerminated

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

715 [Detail]

```
716 <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>
```

717 4.3 Unknown Sequence

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

720 Properties:

721 [Code] Sender

722 [Subcode] wsmr:UnknownSequence

723 [Reason] The value of wsmr:Identifier is not a known Sequence identifier.

724 [Detail]

```
725 <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>
```

726 4.4 Invalid Acknowledgement

727 This fault is sent by the RM Source in response to a <wsmr:SequenceAcknowledgement> that violates
728 the cumulative acknowledgement invariant. An example of such a violation would be a
729 SequenceAcknowledgement covering messages that have not been sent.

730 [Code] Sender

731 [Subcode] wsmr:InvalidAcknowledgement

732 [Reason] The SequenceAcknowledgement violates the cumulative acknowledgement invariant.

733 [Detail]

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

735 **4.5 Message Number Rollover**

736 This fault is sent to indicate that message numbers for a sequence have been exhausted.

737 Properties:

738 [Code] Sender

739 [Subcode] wsrm:MessageNumberRollover

740 [Reason] The maximum value for wsrm:MessageNumber has been exceeded.

741 [Detail]

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

743 **4.6 Create Sequence Refused**

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

745 Properties:

746 [Code] Sender

747 [Subcode] wsrm:CreateSequenceRefused

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

749 [Detail]

```
750 xs:any
```

751 **4.7 Sequence Closed**

752 This fault is sent by an RM Destination to indicate that the specified sequence has been closed. This fault
753 MUST be generated when an RM Destination is asked to receive a message for a sequence that is
754 closed.

755 Properties:

756 [Code] Sender

757 [Subcode] wsrm:SequenceClosed

758 [Reason] The sequence is closed and can not receive new messages.

759 [Detail]

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

761 5 Security Considerations

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

768 Because Sequences are expected to exchange a number of messages, it is recommended that a security
769 context be established using the mechanisms described in WS-Trust and WS-SecureConversation
770 [SecureConversation]. If a Sequence is bound to a specific destination, then the security context needs to
771 be established or shared with the destination servicing the Sequence. While the context can be
772 established at any time, it is critical that the messages establishing the Sequence be secured even if they
773 precede security context establishment. However, it is recommended that the security context be
774 established first. Security contexts are independent of reliable messaging Sequences. Consequently,
775 security contexts can come and go independent of the lifetime of the Sequence. In fact, it is
776 recommended that the lifetime of a security context be less than the lifetime of the Sequence unless the
777 Sequence is very short-lived.

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

- 782 • Closing and re-establishing a security context
- 783 • Exchanging new secrets between the parties
- 784 • Using a derived key sequence and switch "generations"
- 785 • Attaching a nonce to each message and using it in a derived key function with the shared secret

786 The security context may be re-established using the mechanisms described in WS-Trust and WS-
787 SecureConversation. Similarly, secrets can be exchanged using the mechanisms described in WS-Trust.
788 Note, however, that the current shared secret should not be used to encrypt the new shared secret.
789 Derived keys, the preferred solution from this list, can be specified using the mechanisms described in
790 WS-SecureConversation.

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

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

- 801 • **Message alteration** – Alteration is prevented by including signatures of the message information
802 using WS-Security.
- 803 • **Message disclosure** – Confidentiality is preserved by encrypting sensitive data using WS-Security.

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

818 **6 References**

819 **6.1 Normative**

820 **[KEYWORDS]**

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

823 **[SOAP 1.1]**

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

825 **[SOAP 1.2]**

826 W3C Recommendation, "[SOAP Version 1.2 Part 1: Messaging Framework](#)" June 2003.

827 **[URI]**

828 T. Berners-Lee, R. Fielding, L. Masinter, "[Uniform Resource Identifiers \(URI\): Generic Syntax](#)," RFC 3986,
829 MIT/LCS, U.C. Irvine, Xerox Corporation, January 2005.

830 **[XML]**

831 W3C Recommendation, "[Extensible Markup Language \(XML\) 1.0 \(Second Edition\)](#)", October 2000.

832 **[XML-ns]**

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

834 **[XML-Schema Part1]**

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

836 **[XML-Schema Part2]**

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

838 **[WSDL 1.1]**

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.

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 **[WS-Security]**

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

850 **[RTTM]**

- 851 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May
852 1992.
- 853 **[SecurityPolicy]**
- 854 G. Della-Libra, et. al. "[Web Services Security Policy Language \(WS-SecurityPolicy\)](#)", July 2005
- 855 **[SecureConversation]**
- 856 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," May 2004.

857 A. Schema

858 The normative schema that is defined for WS-ReliableMessaging using [XML-Schema Part1] and [XML-
859 Schema Part2] is located at:

860 <http://docs.oasis-open.org/ws-rx/wsrn/200510/wsrn-1.1-schema-200510.xsd>

861 The following copy is provided for reference.

```
862 <?xml version="1.0" encoding="UTF-8"?>
863 <!--
864 OASIS takes no position regarding the validity or scope of any intellectual
865 property or other rights that might be claimed to pertain to the
866 implementation or use of the technology described in this document or the
867 extent to which any license under such rights might or might not be available;
868 neither does it represent that it has made any effort to identify any such
869 rights. Information on OASIS's procedures with respect to rights in OASIS
870 specifications can be found at the OASIS website. Copies of claims of rights
871 made available for publication and any assurances of licenses to be made
872 available, or the result of an attempt made to obtain a general license or
873 permission for the use of such proprietary rights by implementors or users of
874 this specification, can be obtained from the OASIS Executive Director.
875 OASIS invites any interested party to bring to its attention any copyrights,
876 patents or patent applications, or other proprietary rights which may cover
877 technology that may be required to implement this specification. Please
878 address the information to the OASIS Executive Director.
879 Copyright © OASIS Open 2002-2006. All Rights Reserved.
880 This document and translations of it may be copied and furnished to others,
881 and derivative works that comment on or otherwise explain it or assist in its
882 implementation may be prepared, copied, published and distributed, in whole or
883 in part, without restriction of any kind, provided that the above copyright
884 notice and this paragraph are included on all such copies and derivative
885 works. However, this document itself does not be modified in any way, such as
886 by removing the copyright notice or references to OASIS, except as needed for
887 the purpose of developing OASIS specifications, in which case the procedures
888 for copyrights defined in the OASIS Intellectual Property Rights document must
889 be followed, or as required to translate it into languages other than English.
890 The limited permissions granted above are perpetual and will not be revoked by
891 OASIS or its successors or assigns.
892 This document and the information contained herein is provided on an "AS
893 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
894 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
895 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
896 FITNESS FOR A PARTICULAR PURPOSE.
897 -->
898 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
899 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
900 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200510"
901 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200510"
902 elementFormDefault="qualified" attributeFormDefault="unqualified">
903   <xs:import
904     namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
905     schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
906   <!-- Protocol Elements -->
907   <xs:complexType name="SequenceType">
908     <xs:sequence
909       <xs:element ref="wsrm:Identifier"/>
910       <xs:element name="MessageNumber"
911         type="xs:unsignedLong"/>
912     <xs:any namespace="##other" processContents="lax"/>
```

```

913 minOccurs="0" maxOccurs="unbounded"/>
914     </xs:sequence>
915     <xs:anyAttribute namespace="##other" processContents="lax"/>
916 </xs:complexType>
917 <xs:element name="Sequence" type="wsrm:SequenceType"/>
918 <xs:element name="SequenceAcknowledgement">
919     <xs:complexType>
920         <xs:sequence>
921             <xs:element ref="wsrm:Identifier"/>
922             <xs:choice>
923                 <xs:sequence>
924                     <xs:choice>
925                         <xs:element
926 name="AcknowledgementRange" maxOccurs="unbounded">
927         <xs:complexType>
928         <xs:sequence/>
929         <xs:attribute name="Upper" type="xs:unsignedLong" use="required"/>
930         <xs:attribute name="Lower" type="xs:unsignedLong" use="required"/>
931         <xs:anyAttribute namespace="##other" processContents="lax"/>
932     </xs:complexType>
933     </xs:element>
934 </xs:element>
935 name="None" minOccurs="0">
936     <xs:complexType>
937     <xs:sequence/>
938     </xs:complexType>
939     </xs:element>
940 </xs:choice>
941 <xs:element name="Final"
942 minOccurs="0">
943     <xs:complexType>
944     <xs:sequence/>
945     </xs:complexType>
946     </xs:element>
947 </xs:sequence>
948 <xs:element name="Nack"
949 type="xs:unsignedLong" maxOccurs="unbounded"/>
950     </xs:choice>
951     <xs:any namespace="##other"
952 processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
953     </xs:sequence>
954     <xs:anyAttribute namespace="##other"
955 processContents="lax"/>
956     </xs:complexType>
957 </xs:element>
958 <xs:complexType name="AckRequestedType">
959     <xs:sequence>
960         <xs:element ref="wsrm:Identifier"/>
961         <xs:any namespace="##other" processContents="lax"
962 minOccurs="0" maxOccurs="unbounded"/>
963     </xs:sequence>
964     <xs:anyAttribute namespace="##other" processContents="lax"/>

```

```

965     </xs:complexType>
966     <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
967     <xs:element name="Identifier">
968         <xs:complexType>
969             <xs:annotation>
970                 <xs:documentation>
971                     This type is for elements whose [children] is an anyURI and can have
972                     arbitrary attributes.
973                 </xs:documentation>
974             </xs:annotation>
975             <xs:simpleContent>
976                 <xs:extension base="xs:anyURI">
977                     <xs:anyAttribute namespace="##other"
978 processContents="lax"/>
979                 </xs:extension>
980             </xs:simpleContent>
981         </xs:complexType>
982     </xs:element>
983     <!-- Fault Container and Codes -->
984     <xs:simpleType name="FaultCodes">
985         <xs:restriction base="xs:QName">
986             <xs:enumeration value="wsrm:UnknownSequence"/>
987             <xs:enumeration value="wsrm:SequenceTerminated"/>
988             <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
989             <xs:enumeration value="wsrm:MessageNumberRollover"/>
990             <xs:enumeration value="wsrm:CreateSequenceRefused"/>
991         </xs:restriction>
992     </xs:simpleType>
993     <xs:complexType name="SequenceFaultType">
994         <xs:sequence>
995             <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
996             <xs:any namespace="##any" processContents="lax"
997 minOccurs="0" maxOccurs="unbounded"/>
998         </xs:sequence>
999         <xs:anyAttribute namespace="##any" processContents="lax"/>
1000     </xs:complexType>
1001     <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1002     <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1003     <xs:element name="CreateSequenceResponse"
1004 type="wsrm:CreateSequenceResponseType"/>
1005     <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1006     <xs:element name="CloseSequenceResponse"
1007 type="wsrm:CloseSequenceResponseType"/>
1008     <xs:element name="TerminateSequence"
1009 type="wsrm:TerminateSequenceType"/>
1010     <xs:complexType name="CreateSequenceType">
1011         <xs:sequence>
1012             <xs:element ref="wsrm:AcksTo"/>
1013             <xs:element ref="wsrm:Expires" minOccurs="0"/>
1014             <xs:element name="Offer" type="wsrm:OfferType"
1015 minOccurs="0"/>
1016             <xs:any namespace="##other" processContents="lax"
1017 minOccurs="0" maxOccurs="unbounded">
1018                 <xs:annotation>
1019                     <xs:documentation>
1020                         It is the authors intent that this extensibility be used to
1021                         transfer a Security Token Reference as defined in WS-Security.
1022                     </xs:documentation>
1023                 </xs:annotation>
1024             </xs:any>
1025         </xs:sequence>
1026         <xs:anyAttribute namespace="##other" processContents="lax"/>

```

```

1027     </xs:complexType>
1028     <xs:complexType name="CreateSequenceResponseType">
1029         <xs:sequence>
1030             <xs:element ref="wsrm:Identifier"/>
1031             <xs:element ref="wsrm:Expires" minOccurs="0"/>
1032             <xs:element name="Accept" type="wsrm:AcceptType"
minOccurs="0"/>
1033
1034             <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
1035         </xs:sequence>
1036         <xs:anyAttribute namespace="##other" processContents="lax"/>
1037     </xs:complexType>
1038     <xs:complexType name="CloseSequenceType">
1039         <xs:sequence>
1040             <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
1041         </xs:sequence>
1042         <xs:attribute name="Identifier" type="xs:anyURI"
use="required"/>
1043         <xs:anyAttribute namespace="##other" processContents="lax"/>
1044     </xs:complexType>
1045     <xs:complexType name="CloseSequenceResponseType">
1046         <xs:sequence>
1047             <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
1048         </xs:sequence>
1049         <xs:anyAttribute namespace="##other" processContents="lax"/>
1050     </xs:complexType>
1051     <xs:complexType name="TerminateSequenceType">
1052         <xs:sequence>
1053             <xs:element ref="wsrm:Identifier"/>
1054             <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
1055         </xs:sequence>
1056         <xs:anyAttribute namespace="##other" processContents="lax"/>
1057     </xs:complexType>
1058     <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
1059     <xs:complexType name="OfferType">
1060         <xs:sequence>
1061             <xs:element ref="wsrm:Identifier"/>
1062             <xs:element ref="wsrm:Expires" minOccurs="0"/>
1063             <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
1064         </xs:sequence>
1065         <xs:anyAttribute namespace="##other" processContents="lax"/>
1066     </xs:complexType>
1067     <xs:complexType name="AcceptType">
1068         <xs:sequence>
1069             <xs:element ref="wsrm:AcksTo"/>
1070             <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
1071         </xs:sequence>
1072         <xs:anyAttribute namespace="##other" processContents="lax"/>
1073     </xs:complexType>
1074     <xs:element name="Expires">
1075         <xs:complexType>
1076             <xs:simpleContent>
1077                 <xs:extension base="xs:duration">
1078                     <xs:anyAttribute namespace="##other"
processContents="lax"/>
1079                 </xs:extension>
1080             </xs:simpleContent>
1081         </xs:complexType>
1082     </xs:element>

```

1089
1090
1091

```
                </xs:complexType>  
            </xs:element>  
</xs:schema>
```


1092 B. Message Examples

1093 B.1 Create Sequence

1094 Create Sequence

```
1095 <?xml version="1.0" encoding="UTF-8"?>
1096 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1097 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200510"
1098 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1099   <S:Header>
1100     <wsa:MessageID>
1101       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
1102     </wsa:MessageID>
1103     <wsa:To>http://example.com/serviceB/123</wsa:To>
1104     <wsa:Action>http://docs.oasis-open.org/ws-
1105 rx/wsmr/200510/CreateSequence</wsa:Action>
1106     <wsa:ReplyTo>
1107       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1108     </wsa:ReplyTo>
1109   </S:Header>
1110   <S:Body>
1111     <wsmr:CreateSequence>
1112       <wsmr:AcksTo>
1113         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1114       </wsmr:AcksTo>
1115     </wsmr:CreateSequence>
1116   </S:Body>
1117 </S:Envelope>
```

1118 Create Sequence Response

```
1119 <?xml version="1.0" encoding="UTF-8"?>
1120 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1121 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200510"
1122 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1123   <S:Header>
1124     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1125     <wsa:RelatesTo>
1126       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
1127     </wsa:RelatesTo>
1128     <wsa:Action>
1129       http://docs.oasis-open.org/ws-rx/wsmr/200510/CreateSequenceResponse
1130     </wsa:Action>
1131   </S:Header>
1132   <S:Body>
1133     <wsmr:CreateSequenceResponse>
1134       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1135     </wsmr:CreateSequenceResponse>
1136   </S:Body>
1137 </S:Envelope>
```

1138 B.2 Initial Transmission

1139 The following example WS-ReliableMessaging headers illustrate the message exchange in the above
1140 figure. The three messages have the following headers; the third message is identified as the last
1141 message in the sequence:

1142 **Message 1**

```
1143 <?xml version="1.0" encoding="UTF-8"?>
1144 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1145 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200510"
1146 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1147   <S:Header>
1148     <wsa:MessageID>
1149       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
1150     </wsa:MessageID>
1151     <wsa:To>http://example.com/serviceB/123</wsa:To>
1152     <wsa:From>
1153       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1154     </wsa:From>
1155     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1156     <wsmr:Sequence>
1157       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1158       <wsmr:MessageNumber>1</wsmr:MessageNumber>
1159     </wsmr:Sequence>
1160   </S:Header>
1161   <S:Body>
1162     <!-- Some Application Data -->
1163   </S:Body>
1164 </S:Envelope>
```

1165 **Message 2**

```
1166 <?xml version="1.0" encoding="UTF-8"?>
1167 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1168 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200510"
1169 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1170   <S:Header>
1171     <wsa:MessageID>
1172       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1173     </wsa:MessageID>
1174     <wsa:To>http://example.com/serviceB/123</wsa:To>
1175     <wsa:From>
1176       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1177     </wsa:From>
1178     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1179     <wsmr:Sequence>
1180       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1181       <wsmr:MessageNumber>2</wsmr:MessageNumber>
1182     </wsmr:Sequence>
1183   </S:Header>
1184   <S:Body>
1185     <!-- Some Application Data -->
1186   </S:Body>
1187 </S:Envelope>
```

1188 **Message 3**

```
1189 <?xml version="1.0" encoding="UTF-8"?>
1190 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1191 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200510"
1192 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1193   <S:Header>
1194     <wsa:MessageID>
1195       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1196     </wsa:MessageID>
1197     <wsa:To>http://example.com/serviceB/123</wsa:To>
1198     <wsa:From>
1199       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
```

```

1200 </wsa:From>
1201 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1202 <wsrm:Sequence>
1203 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1204 <wsrm:MessageNumber>3</wsrm:MessageNumber>
1205 </wsrm:Sequence>
1206 <wsrm:AckRequested>
1207 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1208 </wsrm:AckRequested>
1209 </S:Header>
1210 <S:Body>
1211 <!-- Some Application Data -->
1212 </S:Body>
1213 </S:Envelope>

```

1214 B.3 First Acknowledgement

1215 Message number 2 has not been received by the RM Destination due to some transmission error so it
1216 responds with an acknowledgement for messages 1 and 3:

```

1217 <?xml version="1.0" encoding="UTF-8"?>
1218 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1219 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200510"
1220 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1221 <S:Header>
1222 <wsa:MessageID>
1223 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1224 </wsa:MessageID>
1225 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1226 <wsa:From>
1227 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1228 </wsa:From>
1229 <wsa:Action>
1230 http://docs.oasis-open.org/ws-rx/wsrn/200510/SequenceAcknowledgement
1231 </wsa:Action>
1232 <wsrm:SequenceAcknowledgement>
1233 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1234 <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
1235 <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
1236 </wsrm:SequenceAcknowledgement>
1237 </S:Header>
1238 <S:Body/>
1239 </S:Envelope>

```

1240 B.4 Retransmission

1241 The RM Sourcediscovers that message number 2 was not received so it resends the message and
1242 requests an acknowledgement:

```

1243 <?xml version="1.0" encoding="UTF-8"?>
1244 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1245 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200510"
1246 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1247 <S:Header>
1248 <wsa:MessageID>
1249 http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1250 </wsa:MessageID>
1251 <wsa:To>http://example.com/serviceB/123</wsa:To>
1252 <wsa:From>
1253 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1254 </wsa:From>

```

```

1255 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1256 <wsrm:Sequence>
1257   <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1258   <wsrm:MessageNumber>2</wsrm:MessageNumber>
1259 </wsrm:Sequence>
1260 <wsrm:AckRequested>
1261   <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1262 </wsrm:AckRequested>
1263 </S:Header>
1264 <S:Body>
1265   <!-- Some Application Data -->
1266 </S:Body>
1267 </S:Envelope>

```

1268 B.5 Termination

1269 The RM Destination now responds with an acknowledgement for the complete sequence which can then
1270 be terminated:

```

1271 <?xml version="1.0" encoding="UTF-8"?>
1272 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1273 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200510"
1274 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1275   <S:Header>
1276     <wsa:MessageID>
1277       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1278     </wsa:MessageID>
1279     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1280     <wsa:From>
1281       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1282     </wsa:From>
1283     <wsa:Action>
1284       http://docs.oasis-open.org/ws-rx/wsrn/200510/SequenceAcknowledgement
1285     </wsa:Action>
1286     <wsrm:SequenceAcknowledgement>
1287       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1288       <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
1289     </wsrm:SequenceAcknowledgement>
1290   </S:Header>
1291   <S:Body/>
1292 </S:Envelope>

```

1293 Terminate Sequence

```

1294 <?xml version="1.0" encoding="UTF-8"?>
1295 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1296 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200510"
1297 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1298   <S:Header>
1299     <wsa:MessageID>
1300       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1301     </wsa:MessageID>
1302     <wsa:To>http://example.com/serviceB/123</wsa:To>
1303     <wsa:Action>
1304       http://docs.oasis-open.org/ws-rx/wsrn/200510/TerminateSequence
1305     </wsa:Action>
1306     <wsa:From>
1307       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1308     </wsa:From>
1309   </S:Header>
1310   <S:Body>
1311     <wsrm:TerminateSequence>

```

```
1312     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1313     </wsrm:TerminateSequence>
1314     </S:Body>
1315 </S:Envelope>
```

1316 C. WSDL

1317 The non-normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

1318 <http://docs.oasis-open.org/ws-rx/wsrn/200510/wsd/wsrn-1.1-wsd-200510.wsd>

1319 The following non-normative copy is provided for reference.

```
1320 <?xml version="1.0" encoding="utf-8"?>
1321 <!--
1322 OASIS takes no position regarding the validity or scope of any intellectual
1323 property or other rights that might be claimed to pertain to the
1324 implementation or use of the technology described in this document or the
1325 extent to which any license under such rights might or might not be available;
1326 neither does it represent that it has made any effort to identify any such
1327 rights. Information on OASIS's procedures with respect to rights in OASIS
1328 specifications can be found at the OASIS website. Copies of claims of rights
1329 made available for publication and any assurances of licenses to be made
1330 available, or the result of an attempt made to obtain a general license or
1331 permission for the use of such proprietary rights by implementors or users of
1332 this specification, can be obtained from the OASIS Executive Director.
1333 OASIS invites any interested party to bring to its attention any copyrights,
1334 patents or patent applications, or other proprietary rights which may cover
1335 technology that may be required to implement this specification. Please
1336 address the information to the OASIS Executive Director.
1337 Copyright © OASIS Open 2002-2006. All Rights Reserved.
1338 This document and translations of it may be copied and furnished to others,
1339 and derivative works that comment on or otherwise explain it or assist in its
1340 implementation may be prepared, copied, published and distributed, in whole or
1341 in part, without restriction of any kind, provided that the above copyright
1342 notice and this paragraph are included on all such copies and derivative
1343 works. However, this document itself does not be modified in any way, such as
1344 by removing the copyright notice or references to OASIS, except as needed for
1345 the purpose of developing OASIS specifications, in which case the procedures
1346 for copyrights defined in the OASIS Intellectual Property Rights document must
1347 be followed, or as required to translate it into languages other than English.
1348 The limited permissions granted above are perpetual and will not be revoked by
1349 OASIS or its successors or assigns.
1350 This document and the information contained herein is provided on an "AS
1351 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
1352 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
1353 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
1354 FITNESS FOR A PARTICULAR PURPOSE.
1355 -->
1356 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
1357 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1358 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1359 xmlns:rm="http://docs.oasis-open.org/ws-rx/wsrn/200510"
1360 xmlns:tns="http://docs.oasis-open.org/ws-rx/wsrn/200510/wsd"
1361 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200510/wsd">
1362   <wsdl:types>
1363     <xs:schema>
1364       <xs:import namespace="http://docs.oasis-open.org/ws-
1365 rx/wsrn/200510" schemaLocation="http://docs.oasis-open.org/ws-
1366 rx/wsrn/200510/wsrn-1.1-schema-200510.xsd"/>
1367     </xs:schema>
1368   </wsdl:types>
1369   <wsdl:message name="CreateSequence">
1370     <wsdl:part name="create" element="rm:CreateSequence"/>
1371   </wsdl:message>
```

```

1372     <wsdl:message name="CreateSequenceResponse">
1373         <wsdl:part name="createResponse"
1374 element="rm:CreateSequenceResponse"/>
1375     </wsdl:message>
1376     <wsdl:message name="CloseSequence">
1377         <wsdl:part name="close" element="rm:CloseSequence"/>
1378     </wsdl:message>
1379     <wsdl:message name="CloseSequenceResponse">
1380         <wsdl:part name="closeResponse"
1381 element="rm:CloseSequenceResponse"/>
1382     </wsdl:message>
1383     <wsdl:message name="TerminateSequence">
1384         <wsdl:part name="terminate" element="rm:TerminateSequence"/>
1385     </wsdl:message>
1386     <wsdl:portType name="SequenceAbstractPortType">
1387         <wsdl:operation name="CreateSequence">
1388             <wsdl:input message="tns:CreateSequence"
1389 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200510/CreateSequence"/>
1390             <wsdl:output message="tns:CreateSequenceResponse"
1391 wsa:Action="http://docs.oasis-open.org/ws-
1392 rx/wsrn/200510/CreateSequenceResponse"/>
1393         </wsdl:operation>
1394         <wsdl:operation name="CloseSequence">
1395             <wsdl:input message="tns:CloseSequence"
1396 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200510/CloseSequence"/>
1397             <wsdl:output message="tns:CloseSequenceResponse"
1398 wsa:Action="http://docs.oasis-open.org/ws-
1399 rx/wsrn/200510/CloseSequenceResponse"/>
1400         </wsdl:operation>
1401         <wsdl:operation name="TerminateSequence">
1402             <wsdl:input message="tns:TerminateSequence"
1403 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200510/TerminateSequence"/>
1404             </wsdl:operation>
1405     </wsdl:portType>
1406 </wsdl:definitions>

```

1407 **D. Acknowledgments**

1408 This document is based on initial contribution to OASIS WS-RX Technical Committee by the following
1409 authors:

1410 Ruslan Bilorusets, BEA, Don Box, Microsoft, Luis Felipe Cabrera, Microsoft, Doug Davis, IBM,
1411 Donald Ferguson, IBM, Christopher Ferris, IBM (Editor), Tom Freund, IBM, Mary Ann Hondo,
1412 IBM, John Ibbotson, IBM, Lei Jin, BEA, Chris Kaler, Microsoft, David Langworthy, Microsoft
1413 (Editor), Amelia Lewis, TIBCO Software, Rodney Limprecht, Microsoft, Steve Lucco, Microsoft,
1414 Don Mullen, TIBCO Software, Anthony Nadalin, IBM, Mark Nottingham, BEA, David Orchard,
1415 BEA, Jamie Roots, IBM, Shivajee Samdarshi, TIBCO Software, John Shewchuk, Microsoft, Tony
1416 Storey, IBM.

1417 The following individuals have provided invaluable input into the initial contribution:

1418 Keith Ballinger, Microsoft, Stefan Batres, Microsoft, Allen Brown, Microsoft, Michael Conner, IBM,
1419 George Copeland, Microsoft, Francisco Curbera, IBM, Paul Fremantle, IBM, Steve Graham, IBM,
1420 Pat Helland, Microsoft, Rick Hill, Microsoft, Scott Hinkelman, IBM, Tim Holloway, IBM, Efim Hudis,
1421 Microsoft, Gopal Kakivaya, Microsoft, Johannes Klein, Microsoft, Frank Leymann, IBM, Martin
1422 Nally, IBM, Peter Niblett, IBM, Jeffrey Schlimmer, Microsoft, James Snell, IBM, Keith Stobie,
1423 Microsoft, Satish Thatte, Microsoft, Stephen Todd, IBM, Sanjiva Weerawarana, IBM, Roger
1424 Wolter, Microsoft.

1425 The following individuals were members of the committee during the development of this specification:

1426 *TBD*

E. Revision History

Rev	Date	By Whom	What
wd-01	2005-07-07	Christopher Ferris	Initial version created based on submission by the authors.
ws-02	2005-07-21	Doug Davis	I011 (PT0S) added
wd-02	2005-08-16	Anish Karmarkar	Trivial editorial changes
ws-03	2005-09-15	Doug Davis	I019 and i028 (CloseSeq) added
wd-05	2005-09-26	Gilbert Pilz	i005 (Source resend of nacks messages when ack already received) added.
wd-05	2005-09-27	Doug Davis	i027 (InOrder delivery assurance spanning multiple sequences) added
wd-05	2005-09-27	Doug Davis	i020 (Semantics of "At most once" Delivery Assurance) added
wd-05	2005-09-27	Doug Davis	i034 (Fault while processing a piggy-backed RM header) added
wd-05	2005-09-27	Doug Davis	i033 (Processing model of NACKs) added
wd-05	2005-09-27	Doug Davis	i031 (AckRequested schema inconsistency) added
wd-05	2005-09-27	Doug Davis	i025 (SeqAck/None) added
wd-05	2005-09-27	Doug Davis	i029 (Remove dependency on WS-Security) added
wd-05	2005-09-27	Doug Davis	i039 (What does 'have a mU attribute' mean) added
wd-05	2005-09-27	Doug Davis	i040 (Change 'optiona'/'required' to 'OPTIONAL'/'REQUIRED') added
wd-05	2005-09-30	Anish Karmarkar	i017 (Change NS to http://docs.oasis-open.org/wsrn/200510/)
wd-05	2005-09-30	Anish Karmarkar	i045 (Include SecureConversation as a reference and move it to non-normative citation)
wd-05	2005-09-30	Anish Karmarkar	i046 (change the type of wsrn:FaultCode element)
wd-06	2005-11-02	Gilbert Pilz	Start wd-06 by changing title page from cd-01.
wd-06	2005-11-03	Gilbert Pilz	i047 (Reorder spec sections)
wd-07	2005-11-17	Gilbert Pilz	Start wd-07
wd-07	2005-11-28	Doug Davis	i071 – except for period in Appendix headings
wd-07	2005-11-28	Doug Davis	i10
wd-07	2005-11-28	Doug Davis	i030
wd-07	2005-11-28	Doug Davis	i037
wd-07	2005-11-28	Doug Davis	i038
wd-07	2005-11-28	Doug Davis	i041
wd-07	2005-11-28	Doug Davis	i043
wd-07	2005-11-28	Doug Davis	i044

Rev	Date	By Whom	What
wd-07	2005-11-28	Doug Davis	i048
wd-07	2005-11-28	Doug Davis	i051
wd-07	2005-11-28	Doug Davis	i053
wd-07	2005-11-28	Doug Davis	i059
wd-07	2005-11-28	Doug Davis	i062
wd-07	2005-11-28	Doug Davis	i063
wd-07	2005-11-28	Doug Davis	i065
wd-07	2005-11-28	Doug Davis	i067
wd-07	2005-11-28	Doug Davis	i068
wd-07	2005-11-28	Doug Davis	i069
wd-07	2005-11-28	Doug Davis	Fix bulleted list (#2) in section 2.3
wd-07	2005-11-29	Gilbert Pilz	i074 (Use of [tcShortName] in artifact locations namespaces, etc)
wd-07	2005-11-29	Gilbert Pilz	i071 – Fixed styles and formatting for TOC. Fixed styles of the appendix headings.
wd-07	2005-11-30	Doug Davis	Removed dup definition of "Receive"
wd-07	2005-11-30	Gilbert Pilz	Fixed lost formatting from heading for Namespace section. Fixed style of text body elements to match OASIS example documents. Fixed tables to match OASIS example documents.
wd-07	2005-12-01	Gilbert Pilz	Updated fix for i074 to eliminate trailing '/'. Added corresponding text around action IRI composition.
wd-07	2005-12-01	Gilbert Pilz	Use non-fixed fields for date values on both title page and body footers.
wd-07	2005-12-01	Doug Davis	Alphabetize the glossary
wd-07	2005-12-02	Doug Davis	i064
wd-07	2005-12-02	Doug Davis	i066
wd-08	2005-12-15	Doug Davis	Add back in RM Source to glossary
wd-08	2005-12-15	Steve Winkler	Doug added Steve's editorial nits
wd-08	2005-12-21	Doug Davis	i050
wd-08	2005-12-21	Doug Davis	i081
wd-08	2005-12-21	Doug Davis	i080 – but i050 negates the need for any changes
wd-08	2005-12-21	Doug Davis	i079
wd-08	2005-12-21	Doug Davis	i076 – didn't add text about "replies" since the RMD to RMS sequence could be used for any message not just replies
wd-08	2005-12-21	Umit Yalcinalp	Action Su03: removed wsse from Table 1
wd-08	2005-12-21	Umit Yalcinalp	i057 per Sunnyvale F2F 2005, Cleaned up some formatting errors in contributors
wd-08	2005-12-27	Doug Davis	i060
wd-08	2005-12-27	Gilbert Pilz	Moved schema and WSDL files to their own artifacts. Converted source document to

Rev	Date	By Whom	What
			OpenDocument Text format. Changed line numbers to be a single style.
wd-08	2005-12-28	Anish Karmarkar	Included a section link to c:\temp\wsrm-1.1-schema-200510.xsd and to c:\temp\wsrm-1.1-wsdl-200510.wsdl
wd-08	2006-01-04	Gilbert Pilz	Fixed formatting for included sections.
wd-08	2006-01-05	Gilbert Pilz	Created links for unused references. Fixed exemplars for CloseSequence and CloseSequenceResponse.
wd-09	2006-01-11	Doug Davis	Minor tweaks to text/typos.
cd-02	2006-01-13	Gilbert Pilz	Titles, boilerplate, etc. for cd-02

1428 F. Notices

1429 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
1430 might be claimed to pertain to the implementation or use of the technology described in this document or
1431 the extent to which any license under such rights might or might not be available; neither does it represent
1432 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to
1433 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made
1434 available for publication and any assurances of licenses to be made available, or the result of an attempt
1435 made to obtain a general license or permission for the use of such proprietary rights by implementors or
1436 users of this specification, can be obtained from the OASIS Executive Director.

1437 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or
1438 other proprietary rights which may cover technology that may be required to implement this specification.
1439 Please address the information to the OASIS Executive Director.

1440 Copyright (C) OASIS Open (2006). All Rights Reserved.

1441 This document and translations of it may be copied and furnished to others, and derivative works that
1442 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
1443 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
1444 this paragraph are included on all such copies and derivative works. However, this document itself may
1445 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as
1446 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights
1447 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it
1448 into languages other than English.

1449 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
1450 or assigns.

1451 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1452 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
1453 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR
1454 ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.