



1 Web Services ReliableMessaging 2 (WS-Reliable Messaging)

3 Working Draft 10, February 22, 2006

4 Document identifier:

5 wsrn-1.1-spec-wd-10

6 Location:

7 Editors:

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

13 Contributors:

14 TBD

15 Abstract:

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

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

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

30 Status:

31 This document is a work in progress and will be updated to reflect issues as they are resolved by the
32 Web Services Reliable Exchange (WS-RX) Technical Committee.

33 **Table of Contents**

34	1 Introduction.....	4
35	1.1 Goals and Requirements.....	4
36	1.1.1 Requirements.....	4
37	1.2 Notational Conventions.....	4
38	1.3 Namespace.....	4
39	1.4 Compliance.....	5
40	2 Reliable Messaging Model.....	6
41	2.1 Glossary.....	6
42	2.2 Protocol Preconditions.....	7
43	2.3 Protocol Invariants.....	7
44	2.4 Example Message Exchange.....	7
45	3 RM Protocol Elements.....	10
46	3.1 Sequence Creation.....	10
47	3.2 Closing A Sequence.....	13
48	3.3 Sequence Termination.....	15
49	3.4 Sequences.....	16
50	3.5 Request Acknowledgement.....	17
51	3.6 Sequence Acknowledgement.....	18
52	4 Faults.....	21
53	4.1 SequenceFault Element.....	22
54	4.2 Sequence Terminated.....	23
55	4.3 Unknown Sequence.....	23
56	4.4 Invalid Acknowledgement.....	23
57	4.5 Message Number Rollover.....	24
58	4.6 Create Sequence Refused.....	24
59	4.7 Sequence Closed.....	24
60	4.8 WSRM Required.....	24
61	5 Security Considerations.....	26
62	6 References.....	28
63	6.1 Normative.....	28
64	6.2 Non-Normative.....	28
65	A. Schema.....	30
66	B. Message Examples.....	36
67	B.1 Create Sequence.....	36
68	B.2 Initial Transmission.....	36
69	B.3 First Acknowledgement.....	38
70	B.4 Retransmission.....	38

71	B.5 Termination.....	39
72	C. WSDL.....	41
73	D. State Tables.....	44
74	E. Acknowledgments.....	48
75	F. Revision History.....	49
76	G. Notices.....	52

77 1 Introduction

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

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

86 1.1 Goals and Requirements

87 1.1.1 Requirements

88 1.2 Notational Conventions

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

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

- 93 • The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- 94 • Characters are appended to elements and attributes to indicate cardinality:
 - 95 ○ "?" (0 or 1)
 - 96 ○ "*" (0 or more)
 - 97 ○ "+" (1 or more)
- 98 • The character "|" is used to indicate a choice between alternatives.
- 99 • The characters "[" and "]" are used to indicate that contained items are to be treated as a group
100 with respect to cardinality or choice.
- 101 • An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content
102 specified in this document. Additional children elements and/or attributes MAY be added at the
103 indicated extension points but they MUST NOT contradict the semantics of the parent and/or
104 owner, respectively. If an extension is not recognized it SHOULD be ignored.
- 105 • XML namespace prefixes (See Section [Namespace](#)) are used to indicate the namespace of the
106 element being defined.

107 1.3 Namespace

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

109 <http://docs.oasis-open.org/ws-rx/wsrn/200602>

110 Dereferencing the above URI will produce the Resource Directory Description Language [[RDDL 2.0](#)]
111 document that describes this namespace.

112 Table 1 lists the XML namespaces that are used in this specification. The choice of any namespace prefix
113 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/
wsm	http://docs.oasis-open.org/ws-rx/wsm/200602
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing
xs	http://www.w3.org/2001/XMLSchema

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

117 <http://docs.oasis-open.org/ws-rx/wsm/200602/wsm-1.1.xsd>

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

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

122 `http://docs.oasis-open.org/ws-rx/wsm/200602/SequenceAcknowledgement`

123 **1.4 Compliance**

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

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

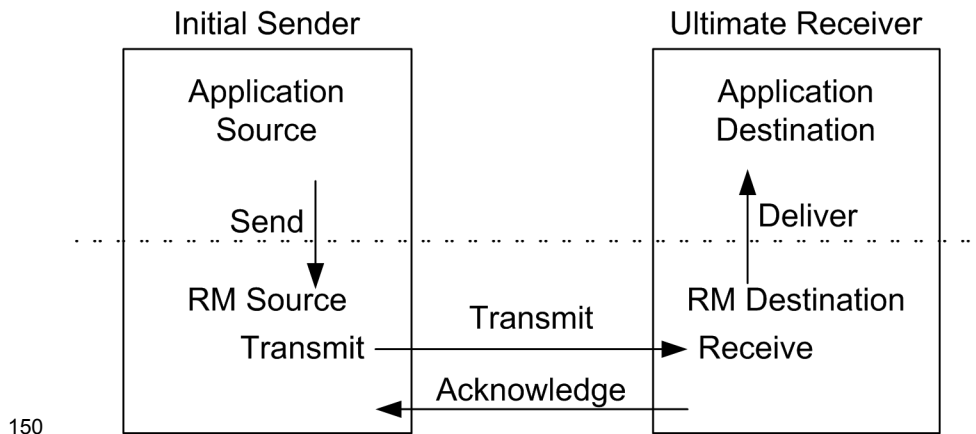
130 2 Reliable Messaging Model

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

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

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

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



150
151 Figure 1: Reliable Messaging Model

152 2.1 Glossary

153 The following definitions are used throughout this specification:

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

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

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

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

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

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

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

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

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

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

170 **2.2 Protocol Preconditions**

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

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

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

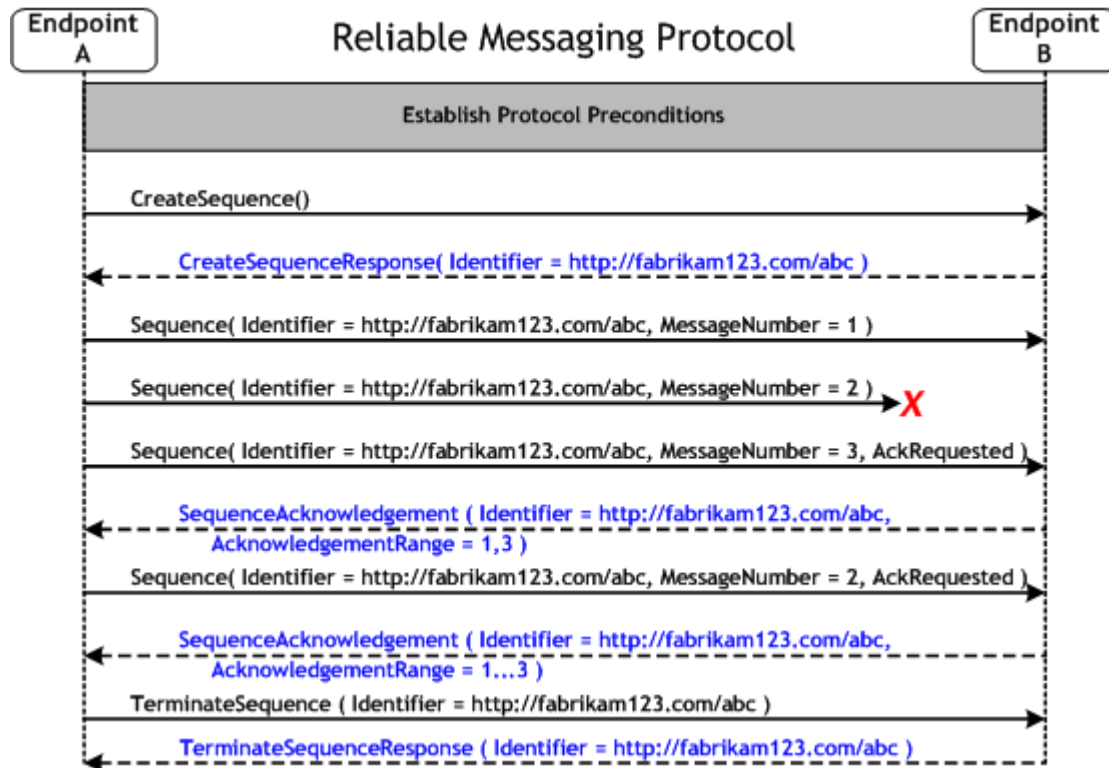
179 **2.3 Protocol Invariants**

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

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

187 **2.4 Example Message Exchange**

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



189 Figure 2: The WS-ReliableMessaging Protocol

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

214 The RM Source will expect to receive acknowledgements from the RM Destination during the course of a
215 message exchange at occasions described in Section 3 below. Should an acknowledgement not be
216 received in a timely fashion, the RM Source MUST re-transmit the request since either the request or the
217 associated acknowledgement may have been lost. Since the nature and dynamic characteristics of the
218 underlying transport and potential intermediaries are unknown in the general case, the timing of re-
219 transmissions cannot be specified. Additionally, over-aggressive re-transmissions have been
220 demonstrated to cause transport or intermediary flooding which are counterproductive to the intention of
221 providing a reliable exchange of messages. Consequently, implementers are encouraged to utilize
222 adaptive mechanisms that dynamically adjust re-transmission time and the back-off intervals that are
223 appropriate to the nature of the transports and intermediaries envisioned. For the case of TCP/IP
224 transports, a mechanism similar to that described as RTTM in RFC 1323 [RTTM] should be considered.

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

227 3 RM Protocol Elements

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

232 3.1 Sequence Creation

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

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

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

252 `/wsrm:CreateSequence`

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

257 `/wsrm:CreateSequence/wsrm:AcksTo`

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

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

264 `/wsrm:CreateSequence/wsrm:Expires`

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

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

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

272 /wsmr:CreateSequence/wsmr:Offer

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

275 /wsmr:CreateSequence/wsmr:Offer/wsmr:Identifier

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

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

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

281 /wsmr:CreateSequence/wsmr:Offer/wsmr:Expires

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

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

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

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

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

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

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

294 /wsmr:CreateSequence/{any}

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

297 /wsmr:CreateSequence/@{any}

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

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

304 The following exemplar defines the `<wsmr:CreateSequenceResponse>` syntax:

```
305 <wsmr:CreateSequenceResponse ...>  
306   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
307   <wsmr:Expires> xs:duration </wsmr:Expires> ?  
308   <wsmr:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?  
309   <wsmr:Accept ...>
```

```

310     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
311     ...
312     </wsrm:Accept> ?
313     ...
314 </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:AcknowledgementInterval

334 This element, if present, specifies the duration after which the RM Destination will transmit an
335 acknowledgement. If omitted, there is no implied value.

336 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@Milliseconds

337 The acknowledgement interval, specified in milliseconds.

338 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@{any}

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

341 /wsrm:CreateSequenceResponse/wsrm:Accept

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

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

347 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo

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

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

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

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

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

357 /wsrm:CreateSequenceResponse/{any}

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

360 /wsrm:CreateSequenceResponse/@{any}

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

363 **3.2 Closing A Sequence**

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

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

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

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

385 The following exemplar defines the CloseSequence syntax:

```
386 <wsrm:CloseSequence ...>  
387   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
388   ...  
389 </wsrm:CloseSequence>
```

390 /wsrm:CloseSequence

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

394 /wsrm:CloseSequence/wsrm:Identifier

395 This REQUIRED element MUST contain an absolute URI conformant with RFC3986 of the Sequence that
396 is being closed.

397 /wsrm:CloseSequence/wsrm:Identifier/@{any}

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

400 /wsrm:CloseSequence/{any}

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

403 /wsrm:CloseSequence@{any}

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

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

409 The following exemplar defines the <wsrm:CloseSequenceResponse> syntax:

```
410 <wsrm:CloseSequenceResponse ...>  
411   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
412   ...  
413 </wsrm:CloseSequenceResponse>
```

414 /wsrm:CloseSequenceResponse

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

418 /wsrm:CloseSequenceResponse/wsrm:Identifier

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

421 /wsrm:CloseSequenceResponse/wsrm:Identifier/@{any}

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

424 /wsrm:CloseSequenceResponse/{any}

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

427 /wsrm:CloseSequenceResponse@{any}

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

430 3.3 Sequence Termination

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

438 The following exemplar defines the TerminateSequence syntax:

```
439 <wsrm:TerminateSequence ...>  
440   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
441   ...  
442 </wsrm:TerminateSequence>
```

443 /wsrm:TerminateSequence

444 This element is sent by an RM Source to indicate it has completed its use of the Sequence. It indicates
445 that the RM Destination can safely reclaim any resources related to the identified Sequence. This element
446 MUST NOT be sent as a header block. The RM Source MAY retransmit this element. Once this element
447 is sent, other than this element, the RM Source MUST NOT send any additional message to the RM
448 Destination referencing this sequence.

449 /wsrm:TerminateSequence/wsrm:Identifier

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

452 /wsrm:TerminateSequence/wsrm:Identifier/@{any}

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

455 /wsrm:TerminateSequence/{any}

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

458 /wsrm:TerminateSequence/@{any}

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

461 A `<wsrm:TerminateSequenceResponse>` is sent in the body of a response message by an RM
462 Destination in response to receipt of a `<wsrm:TerminateSequence>` request message. It indicates that
463 the RM Destination has terminated the sequence.

464 The following exemplar defines the `<wsrm:TerminateSequenceResponse>` syntax:

```
465 <wsrm:TerminateSequenceResponse ...>  
466   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
467   ...  
468 </wsrm:TerminateSequenceResponse>
```

469 /wsrm:TerminateSequenceResponse

470 This element is sent in the body of a response message by an RM Destination in response to receipt of a
471 `<wsrm:TerminateSequence>` request message. It indicates that the RM Destination has terminated
472 the sequence. This element MUST NOT be sent as a header block.

473 /wsm:TerminateSequenceResponse/wsm:Identifier

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

476 /wsm:TerminateSequenceResponse/wsm:Identifier/@{any}

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

479 /wsm:TerminateSequenceResponse/{any}

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

482 /wsm:TerminateSequenceresponse/@{any}

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

485 On receipt of a <wsm:TerminateSequence> message an RM Destination MUST respond with a
486 corresponding <wsm:TerminateSequenceResponse> message or generate a fault.

487 3.4 Sequences

488 The RM protocol uses a <wsm:Sequence> header block to track and manage the reliable delivery of
489 messages. Messages for which a reliable delivery is required MUST contain a <wsm:Sequence>
490 header block. Each Sequence MUST have a unique <wsm:Identifier> element and each message
491 within a Sequence MUST have a <wsm:MessageNumber> element that increments by 1 from an initial
492 value of 1. These values are contained within a <wsm:Sequence> header block accompanying each
493 message being delivered in the context of a Sequence.

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

495 A following exemplar defines its syntax:

```
496 <wsm:Sequence ...>  
497   <wsm:Identifier ...> xs:anyURI </wsm:Identifier>  
498   <wsm:MessageNumber> wsm:MessageNumberType </wsm:MessageNumber>  
499   ...  
500 </wsm:Sequence>
```

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

502 /wsm:Sequence

503 This is the element containing Sequence information for WS-ReliableMessaging. The <wsm:Sequence>
504 element MUST be understood by the RM Destination. The <wsm:Sequence> element MUST have a
505 mustUnderstand attribute with a value 1/true from the namespace corresponding to the version of
506 SOAP to which the <wsm:Sequence> SOAP header block is bound.

507 /wsm:Sequence/wsm:Identifier

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

510 /wsm:Sequence/wsm:Identifier/@{any}

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

513 /wsmr:Sequence/wsmr:MessageNumber

514 This REQUIRED element MUST contain a `wsmr:MessageNumberType` representing the ordinal position
515 of the message within a Sequence. Sequence MessageNumbers start at 1 and monotonically increase
516 throughout the Sequence. If the message number exceeds the internal limitations of an RM Source or RM
517 Destination or reaches the maximum value of 9,223,372,036,854,775,807 the RM Source or Destination
518 MUST issue a MessageNumberRollover fault.

519 /wsmr:Sequence/{any}

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

522 /wsmr:Sequence/@{any}

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

525 The following example illustrates a Sequence header block.

```
526 <wsmr:Sequence>  
527   <wsmr:Identifier>http://example.com/abc</wsmr:Identifier>  
528   <wsmr:MessageNumber>10</wsmr:MessageNumber>  
529 </wsmr:Sequence>
```

530 3.5 Request Acknowledgement

531 The purpose of the `<wsmr:AckRequested>` header block is to signal to the RM Destination that the RM
532 Source is requesting that a `<wsmr:SequenceAcknowledgement>` be returned.

533 The RM Source may request an acknowledgement message from the RM Destination at any time by
534 including an `<wsmr:AckRequested>` header block in the message. An RM Destination that receives a
535 message that contains an `<wsmr:AckRequested>` header block MUST respond with a message
536 containing a `<wsmr:SequenceAcknowledgement>` header block. If a non-mustUnderstand fault occurs
537 when processing an RM Header that was piggy-backed on another message, a fault MUST be generated,
538 but the processing of the original message MUST NOT be affected.

539 The following exemplar defines its syntax:

```
540 <wsmr:AckRequested ...>  
541   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
  
542   ...  
543 </wsmr:AckRequested>
```

544 /wsmr:AckRequested

545 This element requests an acknowledgement for the identified sequence.

546 /wsmr:AckRequested/wsmr:Identifier

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

549 /wsmr:AckRequested/wsmr:Identifier/@{any}

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

552 /wsmr:AckRequested/{any}

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

555 /wsrm:AckRequested/@{any}

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

558 3.6 Sequence Acknowledgement

559 The RM Destination informs the RM Source of successful message receipt using a
560 <wsrm:SequenceAcknowledgement> header block. The <wsrm:SequenceAcknowledgement>
561 header block MAY be transmitted independently or included on return messages. The RM Destination
562 MAY send a <wsrm:SequenceAcknowledgement> header block at any point during which the
563 sequence is valid. The timing of acknowledgements can be advertised using policy and
564 acknowledgements can be explicitly requested using the <wsrm:AckRequested> directive (see Section
565 [Request Acknowledgement](#)). If a non-mustUnderstand fault occurs when processing an RM Header that
566 was piggy-backed on another message, a fault MUST be generated, but the processing of the original
567 message MUST NOT be affected.

568 The following exemplar defines its syntax:

```
569 <wsrm:SequenceAcknowledgement ...>  
570   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
571   [ [ <wsrm:AcknowledgementRange ...  
572     Upper="wsrm:MessageNumberType"  
573     Lower="wsrm:MessageNumberType"/> +  
  
574     | <wsrm:None/> ]  
575     <wsrm:Final/> ?  
576     | <wsrm:Nack> wsrm:MessageNumberType </wsrm:Nack> + ]  
577   ...  
578   ...  
579 </wsrm:SequenceAcknowledgement>
```

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

581 /wsrm:SequenceAcknowledgement

582 This element contains the Sequence acknowledgement information.

583 /wsrm:SequenceAcknowledgement/wsrm:Identifier

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

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

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

590 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange

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

595 /wsmr:SequenceAcknowledgement/wsmr:AcknowledgementRange/@Upper
596 This REQUIRED attribute contains a wsmr:MessageNumberType representing the
597 <wsmr:MessageNumber> of the highest contiguous message in a Sequence range received by the RM
598 Destination.

599 /wsmr:SequenceAcknowledgement/wsmr:AcknowledgementRange/@Lower
600 This REQUIRED attribute contains a wsmr:MessageNumberType representing the
601 <wsmr:MessageNumber> of the lowest contiguous message in a Sequence range received by the RM
602 Destination.

603 /wsmr:SequenceAcknowledgement/wsmr:AcknowledgementRange/@{any}
604 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
605 element.

606 /wsmr:SequenceAcknowledgement/wsmr:Final
607 This OPTIONAL element, if present, indicates that the RM Destination is not receiving new messages for
608 the specified Sequence. The RM Source can be assured that the ranges of messages acknowledged by
609 this SequenceAcknowledgement header block will not change in the future. This element MUST be
610 present when the Sequence is no longer receiving new message for the specified sequence. Note: this
611 element MUST NOT be used when sending a Nack, it can only be used when sending
612 AcknowledgementRanges or <wsmr:None>.

613 /wsmr:SequenceAcknowledgement/wsmr:Nack
614 This OPTIONAL element, if present, MUST contain a wsmr:MessageNumberType representing the
615 <wsmr:MessageNumber> of an unreceived message in a Sequence. This element permits the gap
616 analysis of the <wsmr:AcknowledgementRange> elements to be performed at the RM Destination
617 rather than at the RM Source which may yield performance benefits in certain environments. The
618 <wsmr:Nack> element MUST NOT be present if a sibling <wsmr:AcknowledgementRange> or
619 <wsmr:None> element is also present as a child of <wsmr:SequenceAcknowledgement>. Upon the
620 receipt of a Nack, an RM Source SHOULD retransmit the message identified by the Nack. The RM
621 Destination MUST NOT issue a <wsmr:SequenceAcknowledgement> containing a <wsmr:Nack> for
622 a message that it has previously acknowledged within a <wsmr:AcknowledgementRange>. The RM
623 Source SHOULD ignore a <wsmr:SequenceAcknowledgement> containing a <wsmr:Nack> for a
624 message that has previously been acknowledged within a <wsmr:AcknowledgementRange>.

625 /wsmr:SequenceAcknowledgement/wsmr:None
626 This OPTIONAL element, if present, MUST be used when the RM Destination has not received any
627 messages for the specified sequence. The <wsmr:None> element MUST NOT be present if a sibling
628 <wsmr:AcknowledgementRange> or <wsmr:Nack> element is also present as a child of the
629 <wsmr:SequenceAcknowledgement>.

630 /wsmr:SequenceAcknowledgement/{any}
631 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
632 to be passed.

633 /wsmr:SequenceAcknowledgement/@{any}
634 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
635 element.

636 The following examples illustrate <wsmr:SequenceAcknowledgement> elements:

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

```
638 <wsrm:SequenceAcknowledgement>
639     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
640     <wsrm:AcknowledgementRange Upper="10" Lower="1"/>
641 </wsrm:SequenceAcknowledgement>
```

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

```
644 <wsrm:SequenceAcknowledgement>
645     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
646     <wsrm:AcknowledgementRange Upper="2" Lower="1"/>
647     <wsrm:AcknowledgementRange Upper="6" Lower="4"/>
648     <wsrm:AcknowledgementRange Upper="10" Lower="8"/>
649 </wsrm:SequenceAcknowledgement>
```

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

```
651 <wsrm:SequenceAcknowledgement>
652     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
653     <wsrm:Nack>3</wsrm:Nack>
654 </wsrm:SequenceAcknowledgement>
```

655 4 Faults

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

659 Faults for this operation are treated as defined in WS-Addressing. CreateSequenceRefused is a possible
660 fault reply for this operation. UnknownSequence is a fault generated by endpoints when messages
661 carrying RM header blocks targeted at unrecognized or terminated sequences are detected, these faults
662 are also treated as defined in WS-Addressing. All other faults in this section relate to the processing of RM
663 header blocks targeted at known sequences and are collectively referred to as sequence faults. Sequence
664 faults SHOULD be sent to the same [destination] as `<wsrm:SequenceAcknowledgement>` messages.
665 These faults are correlated using the Sequence identifier carried in the detail.

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

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

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

672 The definitions of faults use the following properties:

673 [Code] The fault code.

674 [Subcode] The fault subcode.

675 [Reason] The English language reason element.

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

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

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

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

```
680 <S:Envelope>  
681   <S:Header>  
682     <wsa:Action>  
683       http://schemas.xmlsoap.org/ws/2004/08/addressing/fault  
684     </wsa:Action>  
685     <!-- Headers elided for clarity. -->  
686   </S:Header>  
687   <S:Body>  
688     <S:Fault>  
689       <S:Code>  
690         <S:Value [Code] </S:Value>  
691         <S:Subcode>  
692           <S:Value [Subcode] </S:Value>  
693         </S:Subcode>  
694       </S:Code>  
695       <S:Reason>  
696         <S:Text xml:lang="en"> [Reason] </S:Text>
```

```
697     </S:Reason>
698     <S:Detail>
699         [Detail]
700         ...
701     </S:Detail>
702 </S:Fault>
703 </S:Body>
704 </S:Envelope>
```

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

```
707 <S11:Envelope>
708   <S11:Header>
709     <wsrm:SequenceFault>
710       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
711       ...
712     </wsrm:SequenceFault>
713     <!-- Headers elided for clarity. -->
714   </S11:Header>
715   <S11:Body>
716     <S11:Fault>
717       <faultcode> [Code] </faultcode>
718       <faultstring> [Reason] </faultstring>
719     </S11:Fault>
720   </S11:Body>
721 </S11:Envelope>
```

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

```
724 <S11:Envelope>
725   <S11:Body>
726     <S11:Fault>
727       <faultcode> [Subcode] </faultcode>
728       <faultstring xml:lang="en"> [Reason] </faultstring>
729     </S11:Fault>
730   </S11:Body>
731 </S11:Envelope>
```

732 4.1 SequenceFault Element

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

737 The following exemplar defines its syntax:

```
738 <wsrm:SequenceFault ...>
739   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
740   ...
741 </wsrm:SequenceFault>
```

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

743 /wsrm:SequenceFault

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

745 /wsrm:SequenceFault/wsrm:FaultCode

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

747 /wsrm:SequenceFault/{any}

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

750 /wsrm:SequenceFault/@{any}

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

753 4.2 Sequence Terminated

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

758 Properties:

759 [Code] Sender or Receiver

760 [Subcode] wsrm:SequenceTerminated

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

762 [Detail]

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

764 4.3 Unknown Sequence

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

767 Properties:

768 [Code] Sender

769 [Subcode] wsrm:UnknownSequence

770 [Reason] The value of wsrm:Identifier is not a known Sequence identifier.

771 [Detail]

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

773 4.4 Invalid Acknowledgement

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

777 [Code] Sender

778 [Subcode] wsrm:InvalidAcknowledgement

779 [Reason] The `SequenceAcknowledgement` violates the cumulative acknowledgement invariant.

780 [Detail]

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

782 **4.5 Message Number Rollover**

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

784 Properties:

785 [Code] Sender

786 [Subcode] wsrm:MessageNumberRollover

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

788 [Detail]

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

790 **4.6 Create Sequence Refused**

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

792 Properties:

793 [Code] Sender

794 [Subcode] wsrm:CreateSequenceRefused

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

796 [Detail]

797 `xs:any`

798 **4.7 Sequence Closed**

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

802 Properties:

803 [Code] Sender

804 [Subcode] wsrm:SequenceClosed

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

806 [Detail]

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

808 **4.8 WSRM Required**

809 If an RM Destination requires the use of WS-RM, this fault is generated when it receives an incoming
810 message that did not use this protocol.

811 Properties:

812 [Code] Sender

813 [Subcode] wsrn:WSRMRequired

814 [Reason] The RM Destination requires the use of WSRM.

815 [Detail]

816 *xs:any*

817 5 Security Considerations

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

824 Because Sequences are expected to exchange a number of messages, it is recommended that a security
825 context be established using the mechanisms described in WS-Trust[Trust] and WS-SecureConversation
826 [SecureConversation]. If a Sequence is bound to a specific destination, then the security context needs to
827 be established or shared with the destination servicing the Sequence. While the context can be
828 established at any time, it is critical that the messages establishing the Sequence be secured even if they
829 precede security context establishment. However, it is recommended that the security context be
830 established first. Security contexts are independent of reliable messaging Sequences. Consequently,
831 security contexts can come and go independent of the lifetime of the Sequence. In fact, it is
832 recommended that the lifetime of a security context be less than the lifetime of the Sequence unless the
833 Sequence is very short-lived.

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

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

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

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

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

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

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

874 **6 References**

875 **6.1 Normative**

876 **[KEYWORDS]**

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

879 **[SOAP 1.1]**

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

881 **[SOAP 1.2]**

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

883 **[URI]**

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

886 **[XML]**

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

888 **[XML-ns]**

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

890 **[XML-Schema Part1]**

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

892 **[XML-Schema Part2]**

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

894 **[WSDL 1.1]**

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

896 **[WS-Addressing]**

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

898 **6.2 Non-Normative**

899 **[RDDL 2.0]**

900 Johnathan Borden, Tim Bray, eds. "[Resource Directory Description Language \(RDDL\) 2.0](#)," January 2004

901 **[WS-Policy]**

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

903 **[WS-PolicyAttachment]**

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

905 **[WS-Security]**

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

908 **[RTTM]**

909 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May
910 1992.

911 **[SecurityPolicy]**

912 G. Della-Libra, et. al. "[Web Services Security Policy Language \(WS-SecurityPolicy\)](#)", July 2005

913 **[SecureConversation]**

914 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," February
915 2005.

916 **[Trust]**

917 S. Anderson, et al, "[Web Services Trust Language \(WS-Trust\)](#)," February 2005.

918 A. Schema

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

921 <http://docs.oasis-open.org/ws-rx/wsrn/200602/wsrn-1.1-schema-200602.xsd>

922 The following copy is provided for reference.

```
923 <?xml version="1.0" encoding="UTF-8"?>
924 <!--
925 OASIS takes no position regarding the validity or scope of any intellectual
926 property or other rights that might be claimed to pertain to the
927 implementation or use of the technology described in this document or the
928 extent to which any license under such rights might or might not be
929 available; neither does it represent that it has made any effort to identify
930 any such rights. Information on OASIS's procedures with respect to rights in
931 OASIS specifications can be found at the OASIS website. Copies of claims of
932 rights made available for publication and any assurances of licenses to be
933 made available, or the result of an attempt made to obtain a general license
934 or permission for the use of such proprietary rights by implementors or users
935 of this specification, can be obtained from the OASIS Executive Director.
936 OASIS invites any interested party to bring to its attention any copyrights,
937 patents or patent applications, or other proprietary rights which may cover
938 technology that may be required to implement this specification. Please
939 address the information to the OASIS Executive Director.
940 Copyright © OASIS Open 2002-2006. All Rights Reserved.
941 This document and translations of it may be copied and furnished to others,
942 and derivative works that comment on or otherwise explain it or assist in its
943 implementation may be prepared, copied, published and distributed, in whole
944 or in part, without restriction of any kind, provided that the above
945 copyright notice and this paragraph are included on all such copies and
946 derivative works. However, this document itself does not be modified in any
947 way, such as by removing the copyright notice or references to OASIS, except
948 as needed for the purpose of developing OASIS specifications, in which case
949 the procedures for copyrights defined in the OASIS Intellectual Property
950 Rights document must be followed, or as required to translate it into
951 languages other than English.
952 The limited permissions granted above are perpetual and will not be revoked
953 by OASIS or its successors or assigns.
954 This document and the information contained herein is provided on an "AS
955 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
956 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
957 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
958 FITNESS FOR A PARTICULAR PURPOSE.
959 -->
960 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
961 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
962 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200602"
```

```

963 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200602"
964 elementFormDefault="qualified" attributeFormDefault="unqualified">
965   <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
966   schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
967   <!-- Protocol Elements -->
968   <xs:complexType name="SequenceType">
969     <xs:sequence>
970       <xs:element ref="wsrm:Identifier"/>
971       <xs:element name="MessageNumber" type="wsrm:MessageNumberType"/>
972       <xs:any namespace="##other" processContents="lax" minOccurs="0"
973   maxOccurs="unbounded"/>
974     </xs:sequence>
975     <xs:anyAttribute namespace="##other" processContents="lax"/>
976   </xs:complexType>
977   <xs:element name="Sequence" type="wsrm:SequenceType"/>
978   <xs:element name="SequenceAcknowledgement">
979     <xs:complexType>
980       <xs:sequence>
981         <xs:element ref="wsrm:Identifier"/>
982         <xs:choice>
983           <xs:sequence>
984             <xs:choice>
985               <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
986                 <xs:complexType>
987                   <xs:sequence/>
988                   <xs:attribute name="Upper" type="xs:unsignedLong"
989   use="required"/>
990                   <xs:attribute name="Lower" type="xs:unsignedLong"
991   use="required"/>
992                   <xs:anyAttribute namespace="##other" processContents="lax"/>
993                 </xs:complexType>
994               </xs:element>
995               <xs:element name="None" minOccurs="0">
996                 <xs:complexType>
997                   <xs:sequence/>
998                 </xs:complexType>
999               </xs:element>
1000             </xs:choice>
1001             <xs:element name="Final" minOccurs="0">
1002               <xs:complexType>
1003                 <xs:sequence/>
1004               </xs:complexType>
1005             </xs:element>
1006           </xs:sequence>
1007           <xs:element name="Nack" type="xs:unsignedLong"
1008   maxOccurs="unbounded"/>
1009         </xs:choice>
1010       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1011   maxOccurs="unbounded"/>
1012     </xs:sequence>

```

```

1013     <xs:anyAttribute namespace="##other" processContents="lax"/>
1014   </xs:complexType>
1015 </xs:element>
1016 <xs:complexType name="AckRequestedType">
1017   <xs:sequence>
1018     <xs:element ref="wsrm:Identifier"/>
1019     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1020 maxOccurs="unbounded"/>
1021   </xs:sequence>
1022   <xs:anyAttribute namespace="##other" processContents="lax"/>
1023 </xs:complexType>
1024 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1025 <xs:element name="Identifier">
1026   <xs:complexType>
1027     <xs:annotation>
1028       <xs:documentation>
1029         This type is for elements whose [children] is an anyURI and can
1030 have arbitrary attributes.
1031       </xs:documentation>
1032     </xs:annotation>
1033     <xs:simpleContent>
1034       <xs:extension base="xs:anyURI">
1035         <xs:anyAttribute namespace="##other" processContents="lax"/>
1036       </xs:extension>
1037     </xs:simpleContent>
1038   </xs:complexType>
1039 </xs:element>
1040 <xs:simpleType name="MessageNumberType">
1041   <xs:restriction base="xs:unsignedLong">
1042     <xs:minInclusive value="1"/>
1043     <xs:maxInclusive value="9223372036854775807"/>
1044   </xs:restriction>
1045 </xs:simpleType>

1046 <!-- Fault Container and Codes -->
1047 <xs:simpleType name="FaultCodes">
1048   <xs:restriction base="xs:QName">
1049     <xs:enumeration value="wsrm:SequenceTerminated"/>
1050     <xs:enumeration value="wsrm:UnknownSequence"/>
1051     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1052     <xs:enumeration value="wsrm:MessageNumberRollover"/>
1053     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1054     <xs:enumeration value="wsrm:SequenceClosed"/>
1055     <xs:enumeration value="wsrm:WSRMRequired"/>
1056   </xs:restriction>
1057 </xs:simpleType>
1058 <xs:complexType name="SequenceFaultType">
1059   <xs:sequence>
1060     <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1061     <xs:any namespace="##any" processContents="lax" minOccurs="0"

```

```

1062 maxOccurs="unbounded"/>
1063     </xs:sequence>
1064     <xs:anyAttribute namespace="##any" processContents="lax"/>
1065 </xs:complexType>
1066 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1067 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1068 <xs:element name="CreateSequenceResponse"
1069 type="wsrm:CreateSequenceResponseType"/>
1070 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1071 <xs:element name="CloseSequenceResponse"
1072 type="wsrm:CloseSequenceResponseType"/>
1073 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1074 <xs:element name="TerminateSequenceResponse"
1075 type="wsrm:TerminateSequenceResponseType"/>

1076 <xs:complexType name="CreateSequenceType">
1077     <xs:sequence>
1078         <xs:element ref="wsrm:AcksTo"/>
1079         <xs:element ref="wsrm:Expires" minOccurs="0"/>
1080         <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1081         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1082 maxOccurs="unbounded">
1083             <xs:annotation>
1084                 <xs:documentation>
1085                     It is the authors intent that this extensibility be used to
1086 transfer a Security Token Reference as defined in WS-Security.
1087                 </xs:documentation>
1088             </xs:annotation>
1089         </xs:any>
1090     </xs:sequence>
1091     <xs:anyAttribute namespace="##other" processContents="lax"/>
1092 </xs:complexType>
1093 <xs:complexType name="CreateSequenceResponseType">
1094     <xs:sequence>
1095         <xs:element ref="wsrm:Identifier"/>
1096         <xs:element ref="wsrm:Expires" minOccurs="0"/>
1097         <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>
1098         <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1099         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1100 maxOccurs="unbounded"/>
1101     </xs:sequence>
1102     <xs:anyAttribute namespace="##other" processContents="lax"/>
1103 </xs:complexType>
1104 <xs:complexType name="CloseSequenceType">
1105     <xs:sequence>
1106         <xs:element ref="wsrm:Identifier"/>
1107         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1108 maxOccurs="unbounded"/>
1109     </xs:sequence>
1110     <xs:anyAttribute namespace="##other" processContents="lax"/>

```

```

1111 </xs:complexType>
1112 <xs:complexType name="CloseSequenceResponseType">
1113   <xs:sequence>
1114     <xs:element ref="wsrm:Identifier"/>
1115     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1116 maxOccurs="unbounded"/>
1117   </xs:sequence>
1118   <xs:anyAttribute namespace="##other" processContents="lax"/>
1119 </xs:complexType>
1120 <xs:complexType name="TerminateSequenceType">
1121   <xs:sequence>
1122     <xs:element ref="wsrm:Identifier"/>
1123     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1124 maxOccurs="unbounded"/>
1125   </xs:sequence>
1126   <xs:anyAttribute namespace="##other" processContents="lax"/>
1127 </xs:complexType>
1128 <xs:complexType name="TerminateSequenceResponseType">
1129   <xs:sequence>
1130     <xs:element ref="wsrm:Identifier"/>
1131     <xs:any namespace="##other" processContents="lax"
1132       minOccurs="0" maxOccurs="unbounded"/>
1133   </xs:sequence>
1134   <xs:anyAttribute namespace="##other" processContents="lax"/>
1135 </xs:complexType>

1136 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
1137 <xs:complexType name="OfferType">
1138   <xs:sequence>
1139     <xs:element ref="wsrm:Identifier"/>
1140     <xs:element ref="wsrm:Expires" minOccurs="0"/>
1141     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1142 maxOccurs="unbounded"/>
1143
1144   </xs:sequence>
1145   <xs:anyAttribute namespace="##other" processContents="lax"/>
1146 </xs:complexType>
1147 <xs:complexType name="AcceptType">
1148   <xs:sequence>
1149     <xs:element ref="wsrm:AcksTo"/>
1150     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1151 maxOccurs="unbounded"/>
1152   </xs:sequence>
1153   <xs:anyAttribute namespace="##other" processContents="lax"/>
1154 </xs:complexType>
1155 <xs:element name="Expires">
1156   <xs:complexType>
1157     <xs:simpleContent>
1158       <xs:extension base="xs:duration">

```

```
1159         <xs:anyAttribute namespace="##other" processContents="lax"/>
1160     </xs:extension>
1161 </xs:simpleContent>
1162 </xs:complexType>
1163 </xs:element>
1164 <xs:element name="AcknowledgementInterval">
1165     <xs:complexType>
1166         <xs:sequence/>
1167         <xs:attribute name="Milliseconds" type="xs:unsignedLong"
1168 use="required"/>
1169     <xs:anyAttribute namespace="##other" processContents="lax"/>
1170 </xs:complexType>
1171 </xs:element>
1172 </xs:schema>
```

1173 B. Message Examples

1173 B.1 Create Sequence

1173 Create Sequence

```
1173 <?xml version="1.0" encoding="UTF-8"?>
1173 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1173 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200602"
1173 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1173   <S:Header>
1173     <wsa:MessageID>
1173       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
1173     </wsa:MessageID>
1173     <wsa:To>http://example.com/serviceB/123</wsa:To>
1173     <wsa:Action>http://docs.oasis-open.org/ws-
1174 rx/wsmr/200602/CreateSequence</wsa:Action>
1173     <wsa:ReplyTo>
1173       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1173     </wsa:ReplyTo>
1173   </S:Header>
1173   <S:Body>
1173     <wsmr:CreateSequence>
1173       <wsmr:AcksTo>
1173         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1173       </wsmr:AcksTo>
1173     </wsmr:CreateSequence>
1173   </S:Body>
1173 </S:Envelope>
```

1173 Create Sequence Response

```
1173 <?xml version="1.0" encoding="UTF-8"?>
1173 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1174 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200602"
1175 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1173   <S:Header>
1173     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1173     <wsa:RelatesTo>
1173       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
1173     </wsa:RelatesTo>
1173     <wsa:Action>
1173       http://docs.oasis-open.org/ws-rx/wsmr/200602/CreateSequenceResponse
1173     </wsa:Action>
1173   </S:Header>
1173   <S:Body>
1173     <wsmr:CreateSequenceResponse>
1173       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1173     </wsmr:CreateSequenceResponse>
1173   </S:Body>
1173 </S:Envelope>
```

1173 B.2 Initial Transmission

1173 The following example WS-ReliableMessaging headers illustrate the message exchange in the above
1174 figure. The three messages have the following headers; the third message is identified as the last
1175 message in the sequence:

1173 **Message 1**

```
1173 <?xml version="1.0" encoding="UTF-8"?>
1173 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1173 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200602"
1173 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1173   <S:Header>
1173     <wsa:MessageID>
1173       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
1173     </wsa:MessageID>
1173     <wsa:To>http://example.com/serviceB/123</wsa:To>
1173     <wsa:From>
1173       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1173     </wsa:From>
1173     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1173     <wsmr:Sequence>
1173       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1173       <wsmr:MessageNumber>1</wsmr:MessageNumber>
1173     </wsmr:Sequence>
1173   </S:Header>
1173   <S:Body>
1173     <!-- Some Application Data -->
1173   </S:Body>
1173 </S:Envelope>
```

1173 **Message 2**

```
1173 <?xml version="1.0" encoding="UTF-8"?>
1173 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1173 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200602"
1173 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1173   <S:Header>
1173     <wsa:MessageID>
1173       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1173     </wsa:MessageID>
1173     <wsa:To>http://example.com/serviceB/123</wsa:To>
1173     <wsa:From>
1173       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1173     </wsa:From>
1173     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1173     <wsmr:Sequence>
1173       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1173       <wsmr:MessageNumber>2</wsmr:MessageNumber>
1173     </wsmr:Sequence>
1173   </S:Header>
1173   <S:Body>
1173     <!-- Some Application Data -->
1173   </S:Body>
1173 </S:Envelope>
```

1173 **Message 3**

```
1173 <?xml version="1.0" encoding="UTF-8"?>
1173 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1173 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200602"
1173 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1173   <S:Header>
1173     <wsa:MessageID>
1173       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1173     </wsa:MessageID>
1173     <wsa:To>http://example.com/serviceB/123</wsa:To>
1173     <wsa:From>
1173       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
```

```

1173 </wsa:From>
1174 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1174 <wsrm:Sequence>
1174 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1174 <wsrm:MessageNumber>3</wsrm:MessageNumber>
1174 </wsrm:Sequence>
1174 <wsrm:AckRequested>
1174 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1174 </wsrm:AckRequested>
1174 </S:Header>
1174 <S:Body>
1174 <!-- Some Application Data -->
1174 </S:Body>
1174 </S:Envelope>

```

1174 B.3 First Acknowledgement

1174 Message number 2 has not been received by the RM Destination due to some transmission error so it
1175 responds with an acknowledgement for messages 1 and 3:

```

1174 <?xml version="1.0" encoding="UTF-8"?>
1174 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1174 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200602"
1174 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1174 <S:Header>
1174 <wsa:MessageID>
1174 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1174 </wsa:MessageID>
1174 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1174 <wsa:From>
1174 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1174 </wsa:From>
1174 <wsa:Action>
1174 http://docs.oasis-open.org/ws-rx/wsrn/200602/SequenceAcknowledgement
1174 </wsa:Action>
1174 <wsrm:SequenceAcknowledgement>
1174 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1174 <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
1174 <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
1174 </wsrm:SequenceAcknowledgement>
1174 </S:Header>
1174 <S:Body/>
1174 </S:Envelope>

```

1174 B.4 Retransmission

1174 The RM Sourcediscovers that message number 2 was not received so it resends the message and
1175 requests an acknowledgement:

```

1174 <?xml version="1.0" encoding="UTF-8"?>
1174 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1174 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200602"
1174 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1174 <S:Header>
1174 <wsa:MessageID>
1174 http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1174 </wsa:MessageID>
1174 <wsa:To>http://example.com/serviceB/123</wsa:To>
1174 <wsa:From>
1174 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1174 </wsa:From>

```

```

1174 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1174 <wsrm:Sequence>
1174 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1174 <wsrm:MessageNumber>2</wsrm:MessageNumber>
1174 </wsrm:Sequence>
1174 <wsrm:AckRequested>
1174 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1174 </wsrm:AckRequested>
1174 </S:Header>
1174 <S:Body>
1174 <!-- Some Application Data -->
1174 </S:Body>
1174 </S:Envelope>

```

1174 B.5 Termination

1174 The RM Destination now responds with an acknowledgement for the complete sequence which can then
1175 be terminated:

```

1174 <?xml version="1.0" encoding="UTF-8"?>
1174 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1174 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200602"
1174 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1174 <S:Header>
1174 <wsa:MessageID>
1174 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1174 </wsa:MessageID>
1174 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1174 <wsa:From>
1174 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1174 </wsa:From>
1174 <wsa:Action>
1174 http://docs.oasis-open.org/ws-rx/wsr/200602/SequenceAcknowledgement
1174 </wsa:Action>
1174 <wsrm:SequenceAcknowledgement>
1174 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1174 <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
1174 </wsrm:SequenceAcknowledgement>
1174 </S:Header>
1174 <S:Body/>
1174 </S:Envelope>

```

1174 Terminate Sequence

```

1174 <?xml version="1.0" encoding="UTF-8"?>
1174 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1174 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200602"
1174 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1174 <S:Header>
1174 <wsa:MessageID>
1174 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1174 </wsa:MessageID>
1174 <wsa:To>http://example.com/serviceB/123</wsa:To>
1174 <wsa:Action>
1174 http://docs.oasis-open.org/ws-rx/wsr/200602/TerminateSequence
1174 </wsa:Action>
1174 <wsa:From>
1174 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1174 </wsa:From>
1174 </S:Header>
1174 <S:Body>
1174 <wsrm:TerminateSequence>

```

```
1174     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1174     </wsrm:TerminateSequence>
1174   </S:Body>
1174 </S:Envelope>
```

1174 Terminate Sequence Response

```
1174 <?xml version="1.0" encoding="UTF-8"?>
1174 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1174   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrml/200602"
1174   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1174   <S:Header>
1174     <wsa:MessageID>
1174       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
1174     </wsa:MessageID>
1174     <wsa:To>http://example.com/serviceA/789</wsa:To>
1174     <wsa:Action>
1174       http://docs.oasis-open.org/ws-rx/wsrml/200602/TerminateSequenceResponse
1174     </wsa:Action>
1174     <wsa:RelatesTo>
1174       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1174     </wsa:RelatesTo>
1174     <wsa:From>
1174       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1174     </wsa:From>
1174   </S:Header>
1174   <S:Body>
1174     <wsrm:TerminateSequenceResponse>
1174       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1174     </wsrm:TerminateSequenceResponse>
1174   </S:Body>
1174 </S:Envelope>
```

1174 C. WSDL

1174 The non-normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

1174 <http://docs.oasis-open.org/ws-rx/wsrn/200602/wsd/wsrn-1.1-wsd-200602.wsd>

1174 The following non-normative copy is provided for reference.

```
1174 <?xml version="1.0" encoding="utf-8"?>
1175 <!--
1176 OASIS takes no position regarding the validity or scope of any intellectual
1177 property or other rights that might be claimed to pertain to the
1178 implementation or use of the technology described in this document or the
1179 extent to which any license under such rights might or might not be
1180 available; neither does it represent that it has made any effort to identify
1181 any such rights. Information on OASIS's procedures with respect to rights in
1182 OASIS specifications can be found at the OASIS website. Copies of claims of
1183 rights made available for publication and any assurances of licenses to be
1184 made available, or the result of an attempt made to obtain a general license
1185 or permission for the use of such proprietary rights by implementors or users
1186 of this specification, can be obtained from the OASIS Executive Director.
1187 OASIS invites any interested party to bring to its attention any copyrights,
1188 patents or patent applications, or other proprietary rights which may cover
1189 technology that may be required to implement this specification. Please
1190 address the information to the OASIS Executive Director.
1191 Copyright © OASIS Open 2002-2006. All Rights Reserved.
1192 This document and translations of it may be copied and furnished to others,
1193 and derivative works that comment on or otherwise explain it or assist in its
1194 implementation may be prepared, copied, published and distributed, in whole
1195 or in part, without restriction of any kind, provided that the above
1196 copyright notice and this paragraph are included on all such copies and
1197 derivative works. However, this document itself does not be modified in any
1198 way, such as by removing the copyright notice or references to OASIS, except
1199 as needed for the purpose of developing OASIS specifications, in which case
1200 the procedures for copyrights defined in the OASIS Intellectual Property
1201 Rights document must be followed, or as required to translate it into
1202 languages other than English.
1203 The limited permissions granted above are perpetual and will not be revoked
1204 by OASIS or its successors or assigns.
1205 This document and the information contained herein is provided on an "AS
1206 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
1207 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
1208 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
1209 FITNESS FOR A PARTICULAR PURPOSE.
1210 -->
1211 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
1212 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1213 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1214 xmlns:rm="http://docs.oasis-open.org/ws-rx/wsrn/200602"
```

```

1215 xmlns:tns="http://docs.oasis-open.org/ws-rx/wsrn/200602/wsdl"
1216 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200602/wsdl">
1217   <wsdl:types>
1218     <xs:schema>
1219       <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsrn/200602"
1220 schemaLocation="http://docs.oasis-open.org/ws-rx/wsrn/200602/wsrn-1.1-schema-
1221 200602.xsd"/>
1222     </xs:schema>
1223   </wsdl:types>
1224   <wsdl:message name="CreateSequence">
1225     <wsdl:part name="create" element="rm:CreateSequence"/>
1226   </wsdl:message>
1227   <wsdl:message name="CreateSequenceResponse">
1228     <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
1229   </wsdl:message>
1230   <wsdl:message name="CloseSequence">
1231     <wsdl:part name="close" element="rm:CloseSequence"/>
1232   </wsdl:message>
1233   <wsdl:message name="CloseSequenceResponse">
1234     <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
1235   </wsdl:message>
1236   <wsdl:message name="TerminateSequence">
1237     <wsdl:part name="terminate" element="rm:TerminateSequence"/>
1238   </wsdl:message>
1239   <wsdl:message name="TerminateSequenceResponse">
1240     <wsdl:part name="terminateResponse"
1241 element="rm:TerminateSequenceResponse"/>
1242   </wsdl:message>
1243   <wsdl:portType name="SequenceAbstractPortType">
1244     <wsdl:operation name="CreateSequence">
1245       <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
1246 open.org/ws-rx/wsrn/200602/CreateSequence"/>
1247       <wsdl:output message="tns:CreateSequenceResponse"
1248 wsa:Action="http://docs.oasis-open.org/ws-
1249 rx/wsrn/200602/CreateSequenceResponse"/>
1250     </wsdl:operation>
1251     <wsdl:operation name="CloseSequence">
1252       <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
1253 open.org/ws-rx/wsrn/200602/CloseSequence"/>
1254       <wsdl:output message="tns:CloseSequenceResponse"
1255 wsa:Action="http://docs.oasis-open.org/ws-
1256 rx/wsrn/200602/CloseSequenceResponse"/>
1257     </wsdl:operation>
1258     <wsdl:operation name="TerminateSequence">
1259       <wsdl:input message="tns:TerminateSequence"
1260 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200602/TerminateSequence"/>
1261       <wsdl:output message="tns:TerminateSequenceResponse"
1262 wsa:Action="http://docs.oasis-open.org/ws-
1263 rx/wsrn/200602/TerminateSequenceResponse"/>
1264     </wsdl:operation>

```

```
1265     </wsdl:portType>
1266 </wsdl:definitions>
```

1267 D. State Tables

1267 This appendix specifies the non-normative state transition tables for RM Source and RM Destination.

1267 Each cell in the tables in this appendix uses the following convention:

Legend
<i>action to take next state</i>

1267 Table 2 RM Source State Transition Table

Events	States							
	None	Connecting	Connected	Rollover	Closing	Closed	Terminating	Terminated
Create Sequence	<i>Transmit Create Sequence</i> Connecting	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Create Sequence Response	N/A	Connected	N/A	N/A	N/A	N/A	N/A	N/A
Create Sequence Refused Fault	N/A	Terminated	N/A	N/A	N/A	N/A	N/A	N/A
New Message	N/A	N/A	<i>Transmit message</i> Connected	<i>Inhibited</i>	<i>Inhibited?</i> Closing	N/A	N/A	N/A
Retransmit of unack message	N/A	N/A	<i>Transmit message</i> Connected	<i>Transmit message</i> Rollover	<i>Trasmit message?</i> Closing	<i>Transmit message</i> Closed	N/A	N/A
SeqAck (non-final)	N/A	N/A	Connected	Rollover	Closing	Closed	<i>Ignore?</i>	<i>Transmit Unknown Sequence Fault</i> Terminated
Nack	N/A	N/A	<i>Transmit message</i> Connected	<i>Transmit message</i> Rollover	<i>Transmit message?</i> Closing	<i>Transmit message?</i> Closed	<i>Ignore?</i>	<i>Transmit Unknown Sequence fault</i> Terminated
Reached max msg number	N/A	N/A	Rollover	Rollover	N/A	N/A	N/A	N/A

Events	States							
	None	Connecting	Connected	Rollover	Closing	Closed	Terminating	Terminated
Create Sequence	<i>Transmit Create Sequence</i> Connecting	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Message Number Rollover Fault	N/A	N/A	Rollover	Rollover	N/A	Closed?	<i>Ignore?</i>	<i>Transmit Unknown Sequence Fault</i> Terminated
Close sequence	N/A	N/A	<i>Transmit Close Sequence</i> Closing	<i>Transmit Close Sequence</i> Closing	<i>Transmit Close Sequence</i> Closing	<i>Transmit Close Sequence</i> Closed	N/A?	N/A
Close sequence Response	N/A	N/A	N/A	N/A	Closed	Closed	<i>Ignore?</i>	<i>Transmit Unknown Sequence Fault</i> Terminated
SeqAck (final)	N/A	N/A	Closed?	Closed?	Closed?	Closed?	<i>Ignore?</i>	<i>Transmit Unknown Sequence fault</i> Terminated
Sequence Closed Fault	N/A	N/A	?	?	?	?	<i>Ignore?</i>	<i>Transmit Unknown Sequence Fault</i> Terminated
Unknown Sequence Fault	N/A	N/A	Terminated?	Terminated?	Terminated?	Terminated?	Terminated?	<i>Ignore</i> Terminated
Sequence Terminated Fault	N/A	Terminated?	Terminated?	Terminated?	Terminated?	Terminated?	Terminated?	<i>Ignored</i> Terminated
Terminate sequence	N/A	N/A	<i>Transmit Terminate Sequence</i> Terminating	<i>Transmit Terminate Sequence</i> Terminating	<i>Transmit Terminate Sequence</i> Terminating	<i>Transmit Terminate Sequence</i> Terminating	<i>Transmit Terminate Sequence</i> Terminating	N/A
Terminate Sequence Response	N/A	N/A	N/A	N/A	N/A	N/A	Terminated	Terminated
Elapse Expires duration	N/A	N/A	Terminated	Terminated	Terminated	Terminated	Terminated?	N/A

1267 In Table 2 above, the rows consists of events that occur at the RM Source throughout the lifetime of an
 1268 RM Sequence and the columns consists of various RM Source states. Each cell in the table above lists
 1269 the action that the RM Source takes on occurrence of a particular event and the next state that it
 1270 transitions.

1267 Table 3 RM Destination State Transition Table

Events	States						
	None	Connecting	Connected	Rollover	Rollover Closed	Closed	Terminated
Creation request not satisfied	N/A	<i>Send Create Sequence Refused Fault</i> Terminated	N/A	N/A	N/A	N/A	
Unrecoverable error on creation	N/A	<i>Send Sequence Terminated Fault?</i> Terminated	N/A	N/A	N/A	N/A	
New message	N/A	N/A	<i>Send SequenceAck</i> Connection	<i>Send Message Number Rollover Fault</i> Rollover	<i>Send Message Number Rollover or Sequence Closed Fault?(with SeqAck+Final)</i> Rollover Closed	<i>Send Sequence Closed Fault (with SeqAck+Final)</i> Closed	<i>Send Unknown Seq Fault?</i> Terminated
Retransmitted message	N/A	N/A	<i>Send SequenceAck</i> Connected	<i>Send SequenceAck</i> Rollover	<i>Send SeqAck+Final</i> Rollover Closed	<i>Send SeqAck+Final</i> Closed	<i>Send Unknown Seq Fault</i> Terminated
Ack requested	N/A	N/A	<i>Send SequenceAck</i> Connected	<i>Send SequenceAck</i> Rollover	<i>Send SeqAck+Final</i> Rollover Closed	<i>Send SeqAck+Final</i> Closed	<i>Send Unknown Seq Fault</i> Terminated
Reach max message number	N/A	N/A	Rollover	Rollover	Rollover Closed	N/A	N/A
Message Number Rollover Fault	N/A	N/A	Rollover	Rollover	Rollover Closed	Closed?	<i>Send Unknown Sequence Fault</i> Terminated
Close sequence	N/A	N/A	<i>Send CloseSequen</i>	<i>Send CloseSequen</i>	<i>Send Close Sequence</i>	<i>Send Close Sequence</i>	<i>Send Unknown</i>

Events	States						
	None	Connecting	Connected	Rollover	Rollover Closed	Closed	Terminated
Creation request not satisfied	N/A	Send Create Sequence Refused Fault Terminated	N/A	N/A	N/A	N/A	
			ceResponse with SequenceAck (Final) Close	ceResponse with SequenceAck Final Rollver Closed	Response with SeqAck+Final Rollover Closed	Response with SeqAck+Final Closed	Sequence Fault Terminated
Close sequence itself	N/A	N/A	Closed	Rollover Closed	Rollover Closed	Closed	N/A
Terminate sequence	N/A	N/A	Terminated	Terminated	Terminated	Terminated	Terminated
Unknown Sequence Fault	N/A	N/A	Terminated?	Terminated?	Terminated?	Terminated?	Ignore Terminated
Sequence Terminated Fault	N/A	N/A	Terminated?	Terminated?	Terminated?	Terminated?	Ignore Terminated
Terminate sequence	N/A	N/A	Terminated	Terminated	Terminated	Terminated	N/A
Elapse Expires duration	N/A	N/A	Terminated	Terminated	Terminated	Terminated	N/A

1267 In Table 3 above, the rows consists of events that occur at the RM Destination throughout the lifetime of
1268 an RM Sequence and the columns consists of various RM Destination states. Each cell in the table above
1269 lists the action that the RM Destination takes on occurrence of a particular event and the next state that it
1270 transitions.

1267 E. Acknowledgments

1267 This document is based on initial contribution to OASIS WS-RX Technical Committee by the following
1268 authors:

1267 Ruslan Bilorusets, BEA, Don Box, Microsoft, Luis Felipe Cabrera, Microsoft, Doug Davis, IBM,
1268 Donald Ferguson, IBM, Christopher Ferris, IBM (Editor), Tom Freund, IBM, Mary Ann Hondo,
1269 IBM, John Ibbotson, IBM, Lei Jin, BEA, Chris Kaler, Microsoft, David Langworthy, Microsoft
1270 (Editor), Amelia Lewis, TIBCO Software, Rodney Limprecht, Microsoft, Steve Lucco, Microsoft,
1271 Don Mullen, TIBCO Software, Anthony Nadalin, IBM, Mark Nottingham, BEA, David Orchard,
1272 BEA, Jamie Roots, IBM, Shivajee Samdarshi, TIBCO Software, John Shewchuk, Microsoft, Tony
1273 Storey, IBM.

1267 The following individuals have provided invaluable input into the initial contribution:

1267 Keith Ballinger, Microsoft, Stefan Batres, Microsoft, Allen Brown, Microsoft, Michael Conner, IBM,
1268 George Copeland, Microsoft, Francisco Curbera, IBM, Paul Fremantle, IBM, Steve Graham, IBM,
1269 Pat Helland, Microsoft, Rick Hill, Microsoft, Scott Hinkelman, IBM, Tim Holloway, IBM, Efim Hudis,
1270 Microsoft, Gopal Kakivaya, Microsoft, Johannes Klein, Microsoft, Frank Leymann, IBM, Martin
1271 Nally, IBM, Peter Niblett, IBM, Jeffrey Schlimmer, Microsoft, James Snell, IBM, Keith Stobie,
1272 Microsoft, Satish Thatte, Microsoft, Stephen Todd, IBM, Sanjiva Weerawarana, IBM, Roger
1273 Wolter, Microsoft.

1267 The following individuals were members of the committee during the development of this specification:

1267 *TBD*

F. 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.
wd-10	2006-01-23	Doug Davis	Accept all changes from wd-09 Make some minor editorial tweaks from Marc's comments.
wd-10	2006-02-14	Doug Davis	Issue 082 resolution
wd-10	2006-02-14	Doug Davis	Issue 083 resolution
wd-10	2006-02-14	Doug Davis	Issue 085 resolution
wd-10	2006-02-14	Doug Davis	Issues 086, 087 resolutions Defined MessageNumberType
wd-10	2006-02-15	Doug Davis	Issue 078 resolution
wd-10	2006-02-15	Doug Davis	Issue 094 resolution
wd-10	2006-02-15	Doug Davis	Issue 095 resolution
wd-10	2006-02-15	Gilbert Pilz	Issue 088 – added namespace URI link to namespace URI; added text explaining that this URI could be dereferenced to produce the RDDDL doc; added non-normative reference to RDDDL 2.0
wd-10	2006-02-17	Anish Karmarkar	Namespace changed to 200602 for both WSDL and XSD docs.
wd-10	2006-02-17	Anish Karmarkar	Issue i087 as it applies to WSRM spec.
wd-10	2006-02-17	Anish Karmarkar	Added titles and minor text for state table (issue i058).

1267 G. Notices

1267 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
1268 might be claimed to pertain to the implementation or use of the technology described in this document or
1269 the extent to which any license under such rights might or might not be available; neither does it represent
1270 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to
1271 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made
1272 available for publication and any assurances of licenses to be made available, or the result of an attempt
1273 made to obtain a general license or permission for the use of such proprietary rights by implementors or
1274 users of this specification, can be obtained from the OASIS Executive Director.

1267 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or
1268 other proprietary rights which may cover technology that may be required to implement this specification.
1269 Please address the information to the OASIS Executive Director.

1267 Copyright (C) OASIS Open (2006). All Rights Reserved.

1267 This document and translations of it may be copied and furnished to others, and derivative works that
1268 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
1269 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
1270 this paragraph are included on all such copies and derivative works. However, this document itself may
1271 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as
1272 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights
1273 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it
1274 into languages other than English.

1267 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
1268 or assigns.

1267 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1268 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
1269 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR
1270 ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.