



# 1 Web Services ReliableMessaging 2 (WS-Reliable Messaging)

3 Committee Draft 03, March 14, 2006

4 **Document identifier:**

5 wsrn-1.1-spec-cd-03

6 **Location:**

7 <http://docs.oasis-open.org/ws-rx/wsrn/200602/wsrn-1.1-spec-cd-03.pdf>

8 **Editors:**

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

14 **Contributors:**

15 TBD

16 **Abstract:**

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

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

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

31 **Status:**

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

## 34 Table of Contents

35	1 Introduction.....	4
36	1.1 Goals and Requirements.....	4
37	1.1.1 Requirements.....	4
38	1.2 Notational Conventions.....	4
39	1.3 Namespace.....	4
40	1.4 Compliance.....	5
41	2 Reliable Messaging Model.....	6
42	2.1 Glossary.....	6
43	2.2 Protocol Preconditions.....	7
44	2.3 Protocol Invariants.....	7
45	2.4 Example Message Exchange.....	7
46	3 RM Protocol Elements.....	10
47	3.1 Sequence Creation.....	10
48	3.2 Closing A Sequence.....	13
49	3.3 Sequence Termination.....	14
50	3.4 Sequences.....	16
51	3.5 Request Acknowledgement.....	17
52	3.6 Sequence Acknowledgement.....	18
53	4 Faults.....	21
54	4.1 SequenceFault Element.....	22
55	4.2 Sequence Terminated.....	23
56	4.3 Unknown Sequence.....	23
57	4.4 Invalid Acknowledgement.....	23
58	4.5 Message Number Rollover.....	24
59	4.6 Create Sequence Refused.....	24
60	4.7 Sequence Closed.....	24
61	4.8 WSRM Required.....	24
62	5 Security Considerations.....	26
63	6 References.....	28
64	6.1 Normative.....	28
65	6.2 Non-Normative.....	28
66	A. Schema.....	30
67	B. Message Examples.....	36
68	B.1 Create Sequence.....	36
69	B.2 Initial Transmission.....	36
70	B.3 First Acknowledgement.....	38
71	B.4 Retransmission.....	38

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

# 78 1 Introduction

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

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

## 87 1.1 Goals and Requirements

### 88 1.1.1 Requirements

### 89 1.2 Notational Conventions

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

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

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

## 108 1.3 Namespace

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

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

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

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

115 The following namespaces are used in this document:

116 *Table 1*

Prefix	Namespace
S	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>
S11	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>
wsm	<a href="http://docs.oasis-open.org/ws-rx/wsm/200602">http://docs.oasis-open.org/ws-rx/wsm/200602</a>
wsa	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

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

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

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

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

123 <http://docs.oasis-open.org/ws-rx/wsm/200602/SequenceAcknowledgement>

## 124 **1.4 Compliance**

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

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

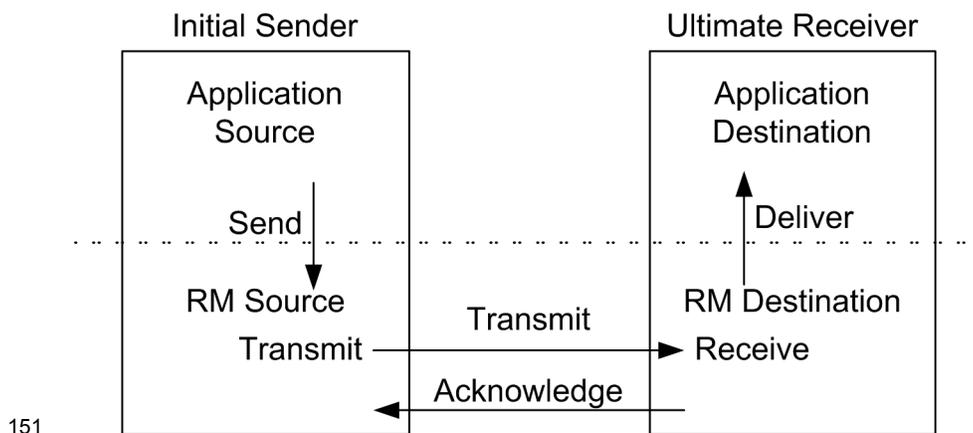
## 131 2 Reliable Messaging Model

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

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

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

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



151  
152 Figure 1: Reliable Messaging Model

### 153 2.1 Glossary

154 The following definitions are used throughout this specification:

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

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

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

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

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

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

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

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

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

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

## 171 2.2 Protocol Preconditions

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

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

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

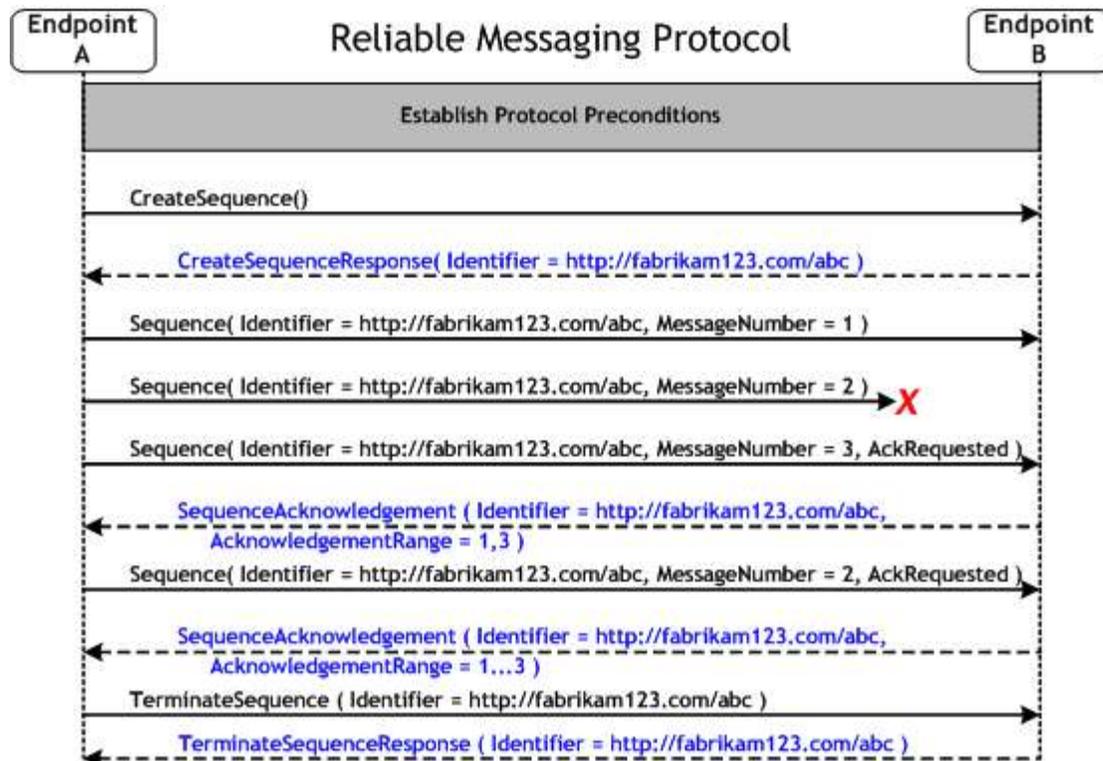
## 180 2.3 Protocol Invariants

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

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

## 188 2.4 Example Message Exchange

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



190 Figure 2: The WS-ReliableMessaging Protocol

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

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

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

## 228 3 RM Protocol Elements

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

### 233 3.1 Sequence Creation

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

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

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

253 `/wsrm:CreateSequence`

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

258 `/wsrm:CreateSequence/wsrm:AcksTo`

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

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

265 `/wsrm:CreateSequence/wsrm:Expires`

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

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

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

273 `/wsmr:CreateSequence/wsmr:Offer`

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

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

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

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

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

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

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

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

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

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

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

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

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

295 `/wsmr:CreateSequence/{any}`

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

298 `/wsmr:CreateSequence/@{any}`

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

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

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

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

```
312     ...
313     </wsrm:Accept> ?
314     ...
315 </wsrm:CreateSequenceResponse>
```

316 /wsrm:CreateSequenceResponse

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

320 /wsrm:CreateSequenceResponse/wsrm:Identifier

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

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

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

326 /wsrm:CreateSequenceResponse/wsrm:Expires

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

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

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

334 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval

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

337 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@Milliseconds

338 The acknowledgement interval, specified in milliseconds.

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

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

342 /wsrm:CreateSequenceResponse/wsrm:Accept

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

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

348 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo

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

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

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

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

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

358 /wsrm:CreateSequenceResponse/{any}

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

361 /wsrm:CreateSequenceResponse/@{any}

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

## 364 3.2 Closing A Sequence

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

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

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

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

386 The following exemplar defines the CloseSequence syntax:

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

391 /wsrm:CloseSequence

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

395 /wsmr:CloseSequence/wsmr:Identifier

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

398 /wsmr:CloseSequence/wsmr:Identifier/@{any}

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

401 /wsmr:CloseSequence/{any}

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

404 /wsmr:CloseSequence@{any}

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

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

410 The following exemplar defines the <wsmr:CloseSequenceResponse> syntax:

```
411 <wsmr:CloseSequenceResponse ...>  
412   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
413   ...  
414 </wsmr:CloseSequenceResponse>
```

415 /wsmr:CloseSequenceResponse

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

419 /wsmr:CloseSequenceResponse/wsmr:Identifier

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

422 /wsmr:CloseSequenceResponse/wsmr:Identifier/@{any}

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

425 /wsmr:CloseSequenceResponse/{any}

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

428 /wsmr:CloseSequenceResponse@{any}

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

### 431 3.3 Sequence Termination

432 When the RM Source has completed its use of the Sequence, it sends a <wsmr:TerminateSequence>  
433 element, in the body of a message to the RM Destination to indicate that the Sequence is complete, and

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

439 The following exemplar defines the TerminateSequence syntax:

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

444 `/wsrm:TerminateSequence`

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

450 `/wsrm:TerminateSequence/wsrm:Identifier`

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

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

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

456 `/wsrm:TerminateSequence/{any}`

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

459 `/wsrm:TerminateSequence/@{any}`

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

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

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

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

470 `/wsrm:TerminateSequenceResponse`

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

474 `/wsrm:TerminateSequenceResponse/wsrm:Identifier`

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

477 /wsmr:TerminateSequenceResponse/wsmr:Identifier/{any}

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

480 /wsmr:TerminateSequenceResponse/{any}

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

483 /wsmr:TerminateSequenceresponse/{any}

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

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

### 488 3.4 Sequences

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

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

496 A following exemplar defines its syntax:

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

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

503 /wsmr:Sequence

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

508 /wsmr:Sequence/wsmr:Identifier

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

511 /wsmr:Sequence/wsmr:Identifier/{any}

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

514 /wsmr:Sequence/wsmr:MessageNumber

515 This REQUIRED element MUST contain a wsmr:MessageNumberType representing the ordinal position  
516 of the message within a Sequence. Sequence MessageNumbers start at 1 and monotonically increase  
517 throughout the Sequence. If the message number exceeds the internal limitations of an RM Source or RM

518 Destination or reaches the maximum value of 9,223,372,036,854,775,807 the RM Source or Destination  
519 MUST issue a MessageNumberRollover fault.

520 /wsm:Sequence/{any}

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

523 /wsm:Sequence/@{any}

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

526 The following example illustrates a Sequence header block.

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

### 531 **3.5 Request Acknowledgement**

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

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

540 The following exemplar defines its syntax:

```
541 <wsm:AckRequested ...>  
542   <wsm:Identifier ...> xs:anyURI </wsm:Identifier>  
  
543   ...  
544 </wsm:AckRequested>
```

545 /wsm:AckRequested

546 This element requests an acknowledgement for the identified sequence.

547 /wsm:AckRequested/wsm:Identifier

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

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

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

553 /wsm:AckRequested/{any}

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

556 /wsm:AckRequested/@{any}

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

### 559 3.6 Sequence Acknowledgement

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

569 The following exemplar defines its syntax:

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

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

582 `/wsrm:SequenceAcknowledgement`

583 This element contains the Sequence acknowledgement information.

584 `/wsrm:SequenceAcknowledgement/wsrm:Identifier`

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

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

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

591 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

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

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

597 This REQUIRED attribute contains a `wsrm:MessageNumberType` representing the  
598 `<wsrm:MessageNumber>` of the highest contiguous message in a Sequence range received by the RM  
599 Destination.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 656 4 Faults

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

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

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

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

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

673 The definitions of faults use the following properties:

674 [Code] The fault code.

675 [Subcode] The fault subcode.

676 [Reason] The English language reason element.

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

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

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

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

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

```

699     <S:Detail>
700         [Detail]
701         ...
702     </S:Detail>
703 </S:Fault>
704 </S:Body>
705 </S:Envelope>

```

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

```

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

```

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

```

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

```

## 733 4.1 SequenceFault Element

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

738 The following exemplar defines its syntax:

```

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

```

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

744 /wsrm:SequenceFault

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

746 /wsrm:SequenceFault/wsrm:FaultCode

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

748 /wsrm:SequenceFault/{any}

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

751 /wsmr:SequenceFault/{any}

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

## 754 4.2 Sequence Terminated

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

759 Properties:

760 [Code] Sender or Receiver

761 [Subcode] wsmr:SequenceTerminated

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

763 [Detail]

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

## 765 4.3 Unknown Sequence

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

768 Properties:

769 [Code] Sender

770 [Subcode] wsmr:UnknownSequence

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

772 [Detail]

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

## 774 4.4 Invalid Acknowledgement

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

778 [Code] Sender

779 [Subcode] wsmr:InvalidAcknowledgement

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

781 [Detail]

782 `<wsmr:SequenceAcknowledgement ...> ... </wsmr:SequenceAcknowledgement>`

## 783 **4.5 Message Number Rollover**

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

785 Properties:

786 [Code] Sender

787 [Subcode] wsrn:MessageNumberRollover

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

789 [Detail]

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

## 791 **4.6 Create Sequence Refused**

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

793 Properties:

794 [Code] Sender

795 [Subcode] wsrn:CreateSequenceRefused

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

797 [Detail]

```
798 xs:any
```

## 799 **4.7 Sequence Closed**

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

803 Properties:

804 [Code] Sender

805 [Subcode] wsrn:SequenceClosed

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

807 [Detail]

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

## 809 **4.8 WSRM Required**

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

812 Properties:

813 [Code] Sender

814 [Subcode] wsrn:WSRMRequired

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

816 [Detail]



## 818 5 Security Considerations

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

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

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

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

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

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

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

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

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

## 875 **6 References**

### 876 **6.1 Normative**

#### 877 **[KEYWORDS]**

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

#### 880 **[SOAP 1.1]**

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

#### 882 **[SOAP 1.2]**

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

#### 884 **[URI]**

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

#### 887 **[XML]**

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

#### 889 **[XML-ns]**

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

#### 891 **[XML-Schema Part1]**

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

#### 893 **[XML-Schema Part2]**

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

#### 895 **[WSDL 1.1]**

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

#### 897 **[WS-Addressing]**

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

### 899 **6.2 Non-Normative**

#### 900 **[RDDL 2.0]**

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

#### 902 **[WS-Policy]**

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

#### 904 **[WS-PolicyAttachment]**

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

#### 906 **[WS-Security]**

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

909 **[RTTM]**

910 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May  
911 1992.

912 **[SecurityPolicy]**

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

914 **[SecureConversation]**

915 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," February  
916 2005.

917 **[Trust]**

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

## 919 A. Schema

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

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

923 The following copy is provided for reference.

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

```

964 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200602"
965 elementFormDefault="qualified" attributeFormDefault="unqualified">
966   <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
967   schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
968   <!-- Protocol Elements -->
969   <xs:complexType name="SequenceType">
970     <xs:sequence>
971       <xs:element ref="wsrm:Identifier"/>
972       <xs:element name="MessageNumber" type="wsrm:MessageNumberType"/>
973       <xs:any namespace="##other" processContents="lax" minOccurs="0"
974   maxOccurs="unbounded"/>
975     </xs:sequence>
976     <xs:anyAttribute namespace="##other" processContents="lax"/>
977   </xs:complexType>
978   <xs:element name="Sequence" type="wsrm:SequenceType"/>
979   <xs:element name="SequenceAcknowledgement">
980     <xs:complexType>
981       <xs:sequence>
982         <xs:element ref="wsrm:Identifier"/>
983         <xs:choice>
984           <xs:sequence>
985             <xs:choice>
986               <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
987                 <xs:complexType>
988                   <xs:sequence/>
989                   <xs:attribute name="Upper" type="xs:unsignedLong"
990   use="required"/>
991                   <xs:attribute name="Lower" type="xs:unsignedLong"
992   use="required"/>
993                   <xs:anyAttribute namespace="##other" processContents="lax"/>
994                 </xs:complexType>
995               </xs:element>
996               <xs:element name="None" minOccurs="0">
997                 <xs:complexType>
998                   <xs:sequence/>
999                 </xs:complexType>
1000               </xs:element>
1001             </xs:choice>
1002           <xs:element name="Final" minOccurs="0">
1003             <xs:complexType>
1004               <xs:sequence/>
1005             </xs:complexType>
1006           </xs:element>
1007         </xs:choice>
1008         <xs:element name="Nack" type="xs:unsignedLong"
1009   maxOccurs="unbounded"/>
1010       </xs:sequence>
1011       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1012   maxOccurs="unbounded"/>
1013     </xs:complexType>

```

```

1014     <xs:anyAttribute namespace="##other" processContents="lax"/>
1015   </xs:complexType>
1016 </xs:element>
1017 <xs:complexType name="AckRequestedType">
1018   <xs:sequence>
1019     <xs:element ref="wsrm:Identifier"/>
1020     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1021 maxOccurs="unbounded"/>
1022   </xs:sequence>
1023   <xs:anyAttribute namespace="##other" processContents="lax"/>
1024 </xs:complexType>
1025 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1026 <xs:element name="Identifier">
1027   <xs:complexType>
1028     <xs:annotation>
1029       <xs:documentation>
1030         This type is for elements whose [children] is an anyURI and can
1031 have arbitrary attributes.
1032       </xs:documentation>
1033     </xs:annotation>
1034     <xs:simpleContent>
1035       <xs:extension base="xs:anyURI">
1036         <xs:anyAttribute namespace="##other" processContents="lax"/>
1037       </xs:extension>
1038     </xs:simpleContent>
1039   </xs:complexType>
1040 </xs:element>
1041 <xs:simpleType name="MessageNumberType">
1042   <xs:restriction base="xs:unsignedLong">
1043     <xs:minInclusive value="1"/>
1044     <xs:maxInclusive value="9223372036854775807"/>
1045   </xs:restriction>
1046 </xs:simpleType>

1047 <!-- Fault Container and Codes -->
1048 <xs:simpleType name="FaultCodes">
1049   <xs:restriction base="xs:QName">
1050     <xs:enumeration value="wsrm:SequenceTerminated"/>
1051     <xs:enumeration value="wsrm:UnknownSequence"/>
1052     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1053     <xs:enumeration value="wsrm:MessageNumberRollover"/>
1054     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1055     <xs:enumeration value="wsrm:SequenceClosed"/>
1056     <xs:enumeration value="wsrm:WSRMRequired"/>
1057   </xs:restriction>
1058 </xs:simpleType>
1059 <xs:complexType name="SequenceFaultType">
1060   <xs:sequence>
1061     <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1062     <xs:any namespace="##any" processContents="lax" minOccurs="0"

```

```

1063 maxOccurs="unbounded"/>
1064     </xs:sequence>
1065     <xs:anyAttribute namespace="##any" processContents="lax"/>
1066 </xs:complexType>
1067 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1068 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1069 <xs:element name="CreateSequenceResponse"
1070 type="wsrm:CreateSequenceResponseType"/>
1071 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1072 <xs:element name="CloseSequenceResponse"
1073 type="wsrm:CloseSequenceResponseType"/>
1074 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1075 <xs:element name="TerminateSequenceResponse"
1076 type="wsrm:TerminateSequenceResponseType"/>

1077 <xs:complexType name="CreateSequenceType">
1078     <xs:sequence>
1079         <xs:element ref="wsrm:AcksTo"/>
1080         <xs:element ref="wsrm:Expires" minOccurs="0"/>
1081         <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1082         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1083 maxOccurs="unbounded">
1084             <xs:annotation>
1085                 <xs:documentation>
1086                     It is the authors intent that this extensibility be used to
1087 transfer a Security Token Reference as defined in WS-Security.
1088                 </xs:documentation>
1089             </xs:annotation>
1090         </xs:any>
1091     </xs:sequence>
1092     <xs:anyAttribute namespace="##other" processContents="lax"/>
1093 </xs:complexType>
1094 <xs:complexType name="CreateSequenceResponseType">
1095     <xs:sequence>
1096         <xs:element ref="wsrm:Identifier"/>
1097         <xs:element ref="wsrm:Expires" minOccurs="0"/>
1098         <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>
1099         <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1100         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1101 maxOccurs="unbounded"/>
1102     </xs:sequence>
1103     <xs:anyAttribute namespace="##other" processContents="lax"/>
1104 </xs:complexType>
1105 <xs:complexType name="CloseSequenceType">
1106     <xs:sequence>
1107         <xs:element ref="wsrm:Identifier"/>
1108         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1109 maxOccurs="unbounded"/>
1110     </xs:sequence>
1111     <xs:anyAttribute namespace="##other" processContents="lax"/>

```

```

1112 </xs:complexType>
1113 <xs:complexType name="CloseSequenceResponseType">
1114   <xs:sequence>
1115     <xs:element ref="wsrm:Identifier"/>
1116     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1117 maxOccurs="unbounded"/>
1118   </xs:sequence>
1119   <xs:anyAttribute namespace="##other" processContents="lax"/>
1120 </xs:complexType>
1121 <xs:complexType name="TerminateSequenceType">
1122   <xs:sequence>
1123     <xs:element ref="wsrm:Identifier"/>
1124     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1125 maxOccurs="unbounded"/>
1126   </xs:sequence>
1127   <xs:anyAttribute namespace="##other" processContents="lax"/>
1128 </xs:complexType>
1129 <xs:complexType name="TerminateSequenceResponseType">
1130   <xs:sequence>
1131     <xs:element ref="wsrm:Identifier"/>
1132     <xs:any namespace="##other" processContents="lax"
1133       minOccurs="0" maxOccurs="unbounded"/>
1134   </xs:sequence>
1135   <xs:anyAttribute namespace="##other" processContents="lax"/>
1136 </xs:complexType>

1137 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
1138 <xs:complexType name="OfferType">
1139   <xs:sequence>
1140     <xs:element ref="wsrm:Identifier"/>
1141     <xs:element ref="wsrm:Expires" minOccurs="0"/>
1142     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1143 maxOccurs="unbounded"/>
1144
1145   </xs:sequence>
1146   <xs:anyAttribute namespace="##other" processContents="lax"/>
1147 </xs:complexType>
1148 <xs:complexType name="AcceptType">
1149   <xs:sequence>
1150     <xs:element ref="wsrm:AcksTo"/>
1151     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1152 maxOccurs="unbounded"/>
1153   </xs:sequence>
1154   <xs:anyAttribute namespace="##other" processContents="lax"/>
1155 </xs:complexType>
1156 <xs:element name="Expires">
1157   <xs:complexType>
1158     <xs:simpleContent>
1159       <xs:extension base="xs:duration">

```

```
1160         <xs:anyAttribute namespace="##other" processContents="lax"/>
1161     </xs:extension>
1162 </xs:simpleContent>
1163 </xs:complexType>
1164 </xs:element>
1165 <xs:element name="AcknowledgementInterval">
1166     <xs:complexType>
1167         <xs:sequence/>
1168         <xs:attribute name="Milliseconds" type="xs:unsignedLong"
1169 use="required"/>
1170     <xs:anyAttribute namespace="##other" processContents="lax"/>
1171 </xs:complexType>
1172 </xs:element>
1173 </xs:schema>
```

## 1174 B. Message Examples

### 1175 B.1 Create Sequence

#### 1176 Create Sequence

```
1177 <?xml version="1.0" encoding="UTF-8"?>
1178 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1179 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200602"
1180 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1181   <S:Header>
1182     <wsa:MessageID>
1183       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
1184     </wsa:MessageID>
1185     <wsa:To>http://example.com/serviceB/123</wsa:To>
1186     <wsa:Action>http://docs.oasis-open.org/ws-
1187 rx/wsmr/200602/CreateSequence</wsa:Action>
1188     <wsa:ReplyTo>
1189       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1190     </wsa:ReplyTo>
1191   </S:Header>
1192   <S:Body>
1193     <wsmr:CreateSequence>
1194       <wsmr:AcksTo>
1195         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1196       </wsmr:AcksTo>
1197     </wsmr:CreateSequence>
1198   </S:Body>
1199 </S:Envelope>
```

#### 1200 Create Sequence Response

```
1201 <?xml version="1.0" encoding="UTF-8"?>
1202 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1203 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200602"
1204 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1205   <S:Header>
1206     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1207     <wsa:RelatesTo>
1208       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
1209     </wsa:RelatesTo>
1210     <wsa:Action>
1211       http://docs.oasis-open.org/ws-rx/wsmr/200602/CreateSequenceResponse
1212     </wsa:Action>
1213   </S:Header>
1214   <S:Body>
1215     <wsmr:CreateSequenceResponse>
1216       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1217     </wsmr:CreateSequenceResponse>
1218   </S:Body>
1219 </S:Envelope>
```

### 1220 B.2 Initial Transmission

1221 The following example WS-ReliableMessaging headers illustrate the message exchange in the above  
1222 figure. The three messages have the following headers; the third message is identified as the last  
1223 message in the sequence:

#### 1224 Message 1

```

1225 <?xml version="1.0" encoding="UTF-8"?>
1226 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1227 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200602"
1228 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1229   <S:Header>
1230     <wsa:MessageID>
1231       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
1232     </wsa:MessageID>
1233     <wsa:To>http://example.com/serviceB/123</wsa:To>
1234     <wsa:From>
1235       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1236     </wsa:From>
1237     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1238     <wsmr:Sequence>
1239       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1240       <wsmr:MessageNumber>1</wsmr:MessageNumber>
1241     </wsmr:Sequence>
1242   </S:Header>
1243   <S:Body>
1244     <!-- Some Application Data -->
1245   </S:Body>
1246 </S:Envelope>

```

## 1247 Message 2

```

1248 <?xml version="1.0" encoding="UTF-8"?>
1249 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1250 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200602"
1251 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1252   <S:Header>
1253     <wsa:MessageID>
1254       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1255     </wsa:MessageID>
1256     <wsa:To>http://example.com/serviceB/123</wsa:To>
1257     <wsa:From>
1258       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1259     </wsa:From>
1260     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1261     <wsmr:Sequence>
1262       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1263       <wsmr:MessageNumber>2</wsmr:MessageNumber>
1264     </wsmr:Sequence>
1265   </S:Header>
1266   <S:Body>
1267     <!-- Some Application Data -->
1268   </S:Body>
1269 </S:Envelope>

```

## 1270 Message 3

```

1271 <?xml version="1.0" encoding="UTF-8"?>
1272 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1273 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200602"
1274 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1275   <S:Header>
1276     <wsa:MessageID>
1277       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1278     </wsa:MessageID>
1279     <wsa:To>http://example.com/serviceB/123</wsa:To>
1280     <wsa:From>
1281       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1282     </wsa:From>
1283     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1284     <wsmr:Sequence>
1285       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>

```

```

1286     <wsrm:MessageNumber>3</wsrm:MessageNumber>
1287     </wsrm:Sequence>
1288     <wsrm:AckRequested>
1289         <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1290     </wsrm:AckRequested>
1291 </S:Header>
1292 <S:Body>
1293     <!-- Some Application Data -->
1294 </S:Body>
1295 </S:Envelope>

```

### 1296 **B.3 First Acknowledgement**

1297 Message number 2 has not been received by the RM Destination due to some transmission error so it  
1298 responds with an acknowledgement for messages 1 and 3:

```

1299 <?xml version="1.0" encoding="UTF-8"?>
1300 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1301 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200602"
1302 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1303     <S:Header>
1304         <wsa:MessageID>
1305             http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1306         </wsa:MessageID>
1307         <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1308         <wsa:From>
1309             <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1310         </wsa:From>
1311         <wsa:Action>
1312             http://docs.oasis-open.org/ws-rx/wsrm/200602/SequenceAcknowledgement
1313         </wsa:Action>
1314         <wsrm:SequenceAcknowledgement>
1315             <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1316             <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
1317             <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
1318         </wsrm:SequenceAcknowledgement>
1319     </S:Header>
1320     <S:Body/>
1321 </S:Envelope>

```

### 1322 **B.4 Retransmission**

1323 The RM Sourcediscovers that message number 2 was not received so it resends the message and  
1324 requests an acknowledgement:

```

1325 <?xml version="1.0" encoding="UTF-8"?>
1326 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1327 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200602"
1328 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1329     <S:Header>
1330         <wsa:MessageID>
1331             http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1332         </wsa:MessageID>
1333         <wsa:To>http://example.com/serviceB/123</wsa:To>
1334         <wsa:From>
1335             <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1336         </wsa:From>
1337         <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1338         <wsrm:Sequence>
1339             <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1340             <wsrm:MessageNumber>2</wsrm:MessageNumber>
1341         </wsrm:Sequence>

```

```

1342     <wsrm:AckRequested>
1343     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1344     </wsrm:AckRequested>
1345   </S:Header>
1346   <S:Body>
1347     <!-- Some Application Data -->
1348   </S:Body>
1349 </S:Envelope>

```

## 1350 B.5 Termination

1351 The RM Destination now responds with an acknowledgement for the complete sequence which can then  
 1352 be terminated:

```

1353 <?xml version="1.0" encoding="UTF-8"?>
1354 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1355 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200602"
1356 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1357   <S:Header>
1358     <wsa:MessageID>
1359       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1360     </wsa:MessageID>
1361     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1362     <wsa:From>
1363       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1364     </wsa:From>
1365     <wsa:Action>
1366       http://docs.oasis-open.org/ws-rx/wsr/200602/SequenceAcknowledgement
1367     </wsa:Action>
1368     <wsrm:SequenceAcknowledgement>
1369       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1370       <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
1371     </wsrm:SequenceAcknowledgement>
1372   </S:Header>
1373   <S:Body/>
1374 </S:Envelope>

```

## 1375 Terminate Sequence

```

1376 <?xml version="1.0" encoding="UTF-8"?>
1377 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1378 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200602"
1379 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1380   <S:Header>
1381     <wsa:MessageID>
1382       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1383     </wsa:MessageID>
1384     <wsa:To>http://example.com/serviceB/123</wsa:To>
1385     <wsa:Action>
1386       http://docs.oasis-open.org/ws-rx/wsr/200602/TerminateSequence
1387     </wsa:Action>
1388     <wsa:From>
1389       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1390     </wsa:From>
1391   </S:Header>
1392   <S:Body>
1393     <wsrm:TerminateSequence>
1394       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1395     </wsrm:TerminateSequence>
1396   </S:Body>
1397 </S:Envelope>

```

## 1398 Terminate Sequence Response

```

1399 <?xml version="1.0" encoding="UTF-8"?>

```

```
1400 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1401 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200602"
1402 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1403   <S:Header>
1404     <wsa:MessageID>
1405       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
1406     </wsa:MessageID>
1407     <wsa:To>http://example.com/serviceA/789</wsa:To>
1408     <wsa:Action>
1409       http://docs.oasis-open.org/ws-rx/wsm/200602/TerminateSequenceResponse
1410     </wsa:Action>
1411     <wsa:RelatesTo>
1412       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1413     </wsa:RelatesTo>
1414     <wsa:From>
1415       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1416     </wsa:From>
1417   </S:Header>
1418   <S:Body>
1419     <wsm:TerminateSequenceResponse>
1420       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1421     </wsm:TerminateSequenceResponse>
1422   </S:Body>
1423 </S:Envelope>
```

## 1424 C. WSDL

1425 The non-normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

1426 <http://docs.oasis-open.org/ws-rx/wsrn/200602/wsd/wsrn-1.1-wsd-200602.wsd>

1427 The following non-normative copy is provided for reference.

```
1428 <?xml version="1.0" encoding="utf-8"?>
1429 <!--
1430 OASIS takes no position regarding the validity or scope of any intellectual
1431 property or other rights that might be claimed to pertain to the
1432 implementation or use of the technology described in this document or the
1433 extent to which any license under such rights might or might not be
1434 available; neither does it represent that it has made any effort to identify
1435 any such rights. Information on OASIS's procedures with respect to rights in
1436 OASIS specifications can be found at the OASIS website. Copies of claims of
1437 rights made available for publication and any assurances of licenses to be
1438 made available, or the result of an attempt made to obtain a general license
1439 or permission for the use of such proprietary rights by implementors or users
1440 of this specification, can be obtained from the OASIS Executive Director.
1441 OASIS invites any interested party to bring to its attention any copyrights,
1442 patents or patent applications, or other proprietary rights which may cover
1443 technology that may be required to implement this specification. Please
1444 address the information to the OASIS Executive Director.
1445 Copyright © OASIS Open 2002-2006. All Rights Reserved.
1446 This document and translations of it may be copied and furnished to others,
1447 and derivative works that comment on or otherwise explain it or assist in its
1448 implementation may be prepared, copied, published and distributed, in whole
1449 or in part, without restriction of any kind, provided that the above
1450 copyright notice and this paragraph are included on all such copies and
1451 derivative works. However, this document itself does not be modified in any
1452 way, such as by removing the copyright notice or references to OASIS, except
1453 as needed for the purpose of developing OASIS specifications, in which case
1454 the procedures for copyrights defined in the OASIS Intellectual Property
1455 Rights document must be followed, or as required to translate it into
1456 languages other than English.
1457 The limited permissions granted above are perpetual and will not be revoked
1458 by OASIS or its successors or assigns.
1459 This document and the information contained herein is provided on an "AS
1460 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
1461 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
1462 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
1463 FITNESS FOR A PARTICULAR PURPOSE.
1464 -->
1465 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
1466 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1467 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1468 xmlns:rm="http://docs.oasis-open.org/ws-rx/wsrn/200602"
```

```

1469 xmlns:tns="http://docs.oasis-open.org/ws-rx/wsr/200602/wsdl"
1470 targetNamespace="http://docs.oasis-open.org/ws-rx/wsr/200602/wsdl">
1471   <wsdl:types>
1472     <xs:schema>
1473       <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsr/200602"
1474       schemaLocation="http://docs.oasis-open.org/ws-rx/wsr/200602/wsr-1.1-schema-
1475       200602.xsd"/>
1476     </xs:schema>
1477   </wsdl:types>
1478   <wsdl:message name="CreateSequence">
1479     <wsdl:part name="create" element="rm:CreateSequence"/>
1480   </wsdl:message>
1481   <wsdl:message name="CreateSequenceResponse">
1482     <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
1483   </wsdl:message>
1484   <wsdl:message name="CloseSequence">
1485     <wsdl:part name="close" element="rm:CloseSequence"/>
1486   </wsdl:message>
1487   <wsdl:message name="CloseSequenceResponse">
1488     <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
1489   </wsdl:message>
1490   <wsdl:message name="TerminateSequence">
1491     <wsdl:part name="terminate" element="rm:TerminateSequence"/>
1492   </wsdl:message>
1493   <wsdl:message name="TerminateSequenceResponse">
1494     <wsdl:part name="terminateResponse"
1495     element="rm:TerminateSequenceResponse"/>
1496   </wsdl:message>
1497   <wsdl:portType name="SequenceAbstractPortType">
1498     <wsdl:operation name="CreateSequence">
1499       <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
1500 open.org/ws-rx/wsr/200602/CreateSequence"/>
1501       <wsdl:output message="tns:CreateSequenceResponse"
1502 wsa:Action="http://docs.oasis-open.org/ws-
1503 rx/wsr/200602/CreateSequenceResponse"/>
1504     </wsdl:operation>
1505     <wsdl:operation name="CloseSequence">
1506       <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
1507 open.org/ws-rx/wsr/200602/CloseSequence"/>
1508       <wsdl:output message="tns:CloseSequenceResponse"
1509 wsa:Action="http://docs.oasis-open.org/ws-
1510 rx/wsr/200602/CloseSequenceResponse"/>
1511     </wsdl:operation>
1512     <wsdl:operation name="TerminateSequence">
1513       <wsdl:input message="tns:TerminateSequence"
1514 wsa:Action="http://docs.oasis-open.org/ws-rx/wsr/200602/TerminateSequence"/>
1515       <wsdl:output message="tns:TerminateSequenceResponse"
1516 wsa:Action="http://docs.oasis-open.org/ws-
1517 rx/wsr/200602/TerminateSequenceResponse"/>
1518     </wsdl:operation>

```

```
1519 </wsdl:portType>
1520 </wsdl:definitions>
```

1521 **D. State Tables**

1522 This appendix specifies the non-normative state transition tables for RM Source and RM Destination.

1523 Each cell in the tables in this appendix uses the following convention:

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

1524 Table 2 RM Source State Transition Table

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

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

1525 In Table 2 above, the rows consists of events that occur at the RM Source throughout the lifetime of an  
1526 RM Sequence and the columns consists of various RM Source states. Each cell in the table above lists  
1527 the action that the RM Source takes on occurrence of a particular event and the next state that it  
1528 transitions.

1529 Table 3 RM Destination State Transition Table

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

Events	States						
	None	Connecting	Connected	Rollover	Rollover Closed	Closed	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?	<i>Ignore</i> Terminated
Sequence Terminated Fault	N/A	N/A	Terminated?	Terminated?	Terminated?	Terminated?	<i>Ignore</i> 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

1530 In Table 3 above, the rows consists of events that occur at the RM Destination throughout the lifetime of  
1531 an RM Sequence and the columns consists of various RM Destination states. Each cell in the table above  
1532 lists the action that the RM Destination takes on occurrence of a particular event and the next state that it  
1533 transitions.

## 1534 E. Acknowledgments

1535 This document is based on initial contribution to OASIS WS-RX Technical Committee by the following  
1536 authors:

1537 Ruslan Bilorusets, BEA, Don Box, Microsoft, Luis Felipe Cabrera, Microsoft, Doug Davis, IBM,  
1538 Donald Ferguson, IBM, Christopher Ferris, IBM (Editor), Tom Freund, IBM, Mary Ann Hondo,  
1539 IBM, John Ibbotson, IBM, Lei Jin, BEA, Chris Kaler, Microsoft, David Langworthy, Microsoft  
1540 (Editor), Amelia Lewis, TIBCO Software, Rodney Limprecht, Microsoft, Steve Lucco, Microsoft,  
1541 Don Mullen, TIBCO Software, Anthony Nadalin, IBM, Mark Nottingham, BEA, David Orchard,  
1542 BEA, Jamie Roots, IBM, Shivajee Samdarshi, TIBCO Software, John Shewchuk, Microsoft, Tony  
1543 Storey, IBM.

1544 The following individuals have provided invaluable input into the initial contribution:

1545 Keith Ballinger, Microsoft, Stefan Batres, Microsoft, Allen Brown, Microsoft, Michael Conner, IBM,  
1546 George Copeland, Microsoft, Francisco Curbera, IBM, Paul Fremantle, IBM, Steve Graham, IBM,  
1547 Pat Helland, Microsoft, Rick Hill, Microsoft, Scott Hinkelman, IBM, Tim Holloway, IBM, Efim Hudis,  
1548 Microsoft, Gopal Kakivaya, Microsoft, Johannes Klein, Microsoft, Frank Leymann, IBM, Martin  
1549 Nally, IBM, Peter Niblett, IBM, Jeffrey Schlimmer, Microsoft, James Snell, IBM, Keith Stobie,  
1550 Microsoft, Satish Thatte, Microsoft, Stephen Todd, IBM, Sanjiva Weerawarana, IBM, Roger  
1551 Wolter, Microsoft.

1552 The following individuals were members of the committee during the development of this specification:

1553 *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 <a href="http://docs.oasis-open.org/wsrn/200510/">http://docs.oasis-open.org/wsrn/200510/</a> )
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

Rev	Date	By Whom	What
wd-08	2005-12-27	Gilbert Pilz	Moved schema and WSDL files to their own artifacts. Converted source document to 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).

## 1555 G. Notices

1556 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that  
1557 might be claimed to pertain to the implementation or use of the technology described in this document or  
1558 the extent to which any license under such rights might or might not be available; neither does it represent  
1559 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to  
1560 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made  
1561 available for publication and any assurances of licenses to be made available, or the result of an attempt  
1562 made to obtain a general license or permission for the use of such proprietary rights by implementors or  
1563 users of this specification, can be obtained from the OASIS Executive Director.

1564 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or  
1565 other proprietary rights which may cover technology that may be required to implement this specification.  
1566 Please address the information to the OASIS Executive Director.

1567 Copyright (C) OASIS Open (2006). All Rights Reserved.

1568 This document and translations of it may be copied and furnished to others, and derivative works that  
1569 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
1570 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and  
1571 this paragraph are included on all such copies and derivative works. However, this document itself may  
1572 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as  
1573 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights  
1574 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it  
1575 into languages other than English.

1576 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors  
1577 or assigns.

1578 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
1579 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY  
1580 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR  
1581 ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.