



1 Web Services ReliableMessaging 2 (WS-Reliable Messaging)

3 Working Draft 1²⁴, April 24, 2006

4 Document identifier:

5 wsrn-1.1-spec-wd-1²⁴

6 Location:

7 Editors:

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

13 Contributors:

14 TBD

15 Abstract:

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

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

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

30 Status:

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

Table of Contents

1	Introduction.....	4
1.1	Goals and Requirements.....	4
1.1.1	Requirements.....	4
1.2	Notational Conventions.....	4
1.3	Namespace.....	4
1.4	Compliance.....	5
2	Reliable Messaging Model.....	6
2.1	Glossary.....	6
2.2	Protocol Preconditions.....	7
2.3	Protocol Invariants.....	7
2.4	Example Message Exchange.....	7
3	RM Protocol Elements.....	10
3.1	Sequence Creation.....	10
3.2	Closing A Sequence.....	13
3.3	Sequence Termination.....	15
3.4	Sequences.....	16
3.5	Request Acknowledgement.....	17
3.6	Sequence Acknowledgement.....	18
4	Faults.....	21
4.1	SequenceFault Element.....	22
4.2	Sequence Terminated.....	23
4.3	Unknown Sequence.....	23
4.4	Invalid Acknowledgement.....	24
4.5	Message Number Rollover.....	24
4.6	Create Sequence Refused.....	24
4.7	Sequence Closed.....	25
4.8	WSRM Required.....	25
5	Security Considerations.....	26
6	References.....	28
6.1	Normative.....	28
6.2	Non-Normative.....	28
A.	Schema.....	30
B.	Message Examples.....	35
B.1	Create Sequence.....	35
B.2	Initial Transmission.....	35
B.3	First Acknowledgement.....	37
B.4	Retransmission.....	37

71	B.5 Termination.....	38
72	C. WSDL.....	40
73	D. State Tables.....	42
74	E. Acknowledgments.....	46
75	F. Revision History.....	47
76	G. Notices.....	51

1 Introduction

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

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

1.1 Goals and Requirements

1.1.1 Requirements

1.2 Notational Conventions

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

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

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

1.3 Namespace

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

<http://docs.oasis-open.org/ws-rx/wsrn/2006042>

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	(Either SOAP 1.1 or 1.2)
S11	http://schemas.xmlsoap.org/soap/envelope/
S12	http://www.w3.org/2003/05/soap-envelope
wsm	http://docs.oasis-open.org/ws-rx/wsm/200604
wsa	http://www.w3.org/2005/08/addressing
xs	http://www.w3.org/2001/XMLSchema

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

118 <http://docs.oasis-open.org/ws-rx/wsm/200604/wsm-1.1-schema-2006042/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/2006042/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.

2 Reliable Messaging Model

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

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

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

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

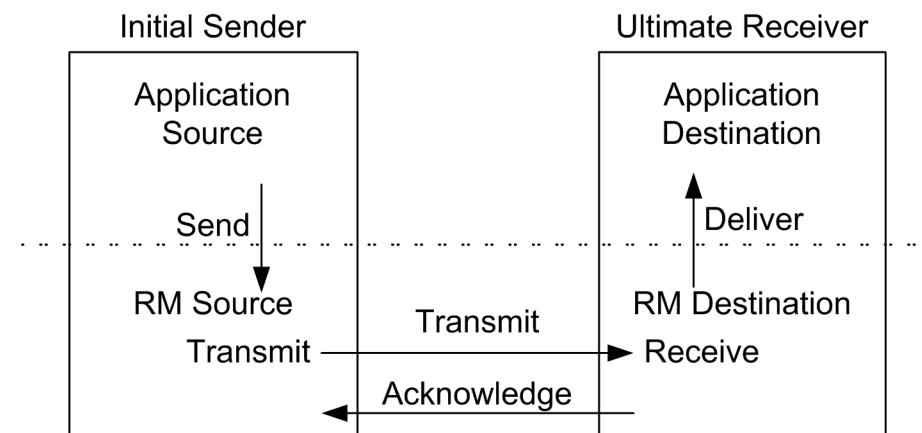


Figure 1: Reliable Messaging Model

2.1 Glossary

The following definitions are used throughout this specification:

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

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

Application Source: The endpoint that Sends a message.

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

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

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

158 **RM Destination:** For any one reliably sent message the endpoint that receives the message.

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

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

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

163 2.2 Protocol Preconditions

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

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

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

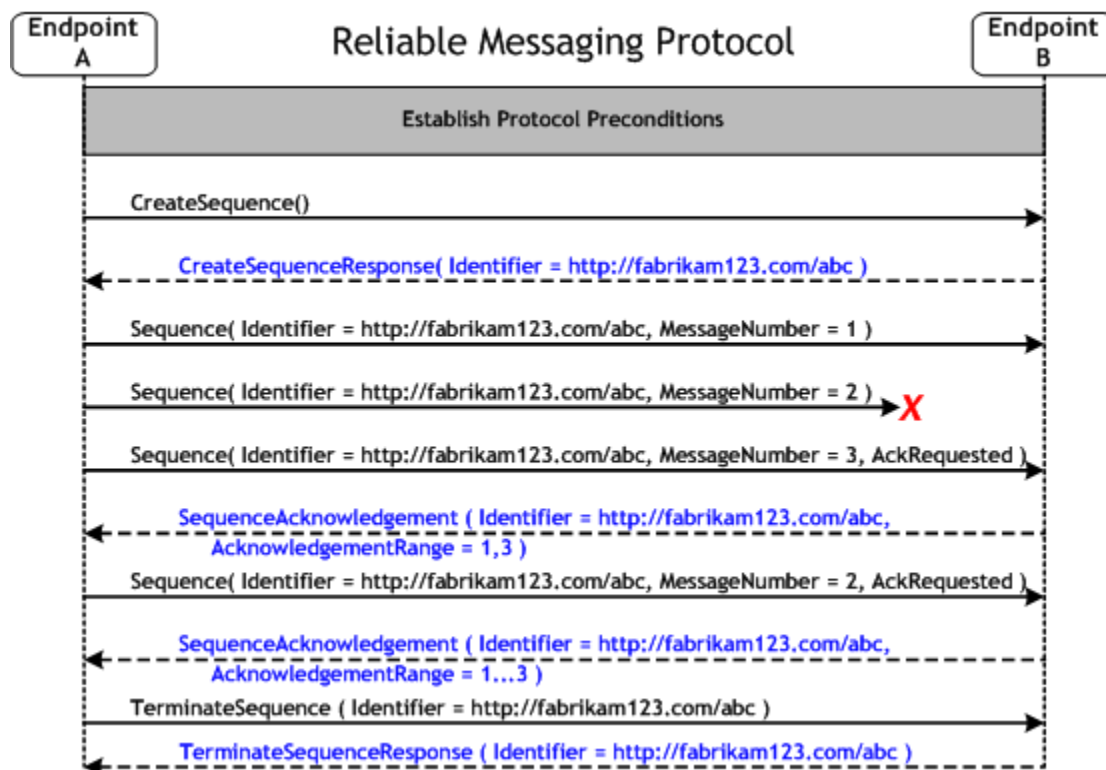
172 2.3 Protocol Invariants

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

- 174 • The RM Source **MUST** assign each message within a Sequence a message number (defined
175 below) beginning at 1 and increasing by exactly 1 for each subsequent message. These numbers
176 **MUST** be assigned in the same order in which messages are sent by the Application Source.
- 177 • Every acknowledgement issued by the RM Destination **MUST** include within an acknowledgement
178 range or ranges the Sequence number of every message successfully received by the RM
179 Destination and **MUST** exclude Ssequence number of every message successfully received by the
180 RM Destination and **MUST** exclude ssequence numbers of any messages not yet received.

181 2.4 Example Message Exchange

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



183 Figure 2: The WS-ReliableMessaging Protocol

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

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

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

3 RM Protocol Elements

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

3.1 Sequence Creation

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

The SOAP version used for the `CreateSequence` message SHOULD be used for all subsequent messages in or for that Sequence, sent by either the RMS or the RMD.

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

```
<wsrm:CreateSequence ...>
  <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
  <wsrm:Expires ...> xs:duration </wsrm:Expires> ?
  <wsrm:Offer ...>
    <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
    <wsrm:Endpoint> wsa:EndpointReferenceType </wsrm:Endpoint>
    <wsrm:Expires ...> xs:duration </wsrm:Expires> ?
    ...
  </wsrm:Offer> ?
  ...
</wsrm:CreateSequence>
```

`/wsrm:CreateSequence`

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

`/wsrm:CreateSequence/wsrm:AcksTo`

This REQUIRED element, of type `wsa:EndpointReferenceType` as specified by WS-Addressing [WS-Addressing] specifies the endpoint reference to which all `<wsrm:SequenceAcknowledgement>` messages and faults related to the created Sequence are to be sent, unless otherwise noted in this specification (for example, see Section 3.2).

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

`/wsrm:CreateSequence/wsrm:Expires`

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

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

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

245 `/wsrm:CreateSequence/wsrm:Offer`

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

248 `/wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier`

249 This REQUIRED element MUST contain an absolute URI conformant with RFC3986 [\[URI\]](#) that uniquely
250 identifies the offered Sequence.

251 `/wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier/@{any}`

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

254 [`/wsrm:CreateSequence/wsrm:Offer/wsrm:Endpoint`](#)

255 [This REQUIRED element, of type `wsa:EndpointReferenceType` as specified by WS-Addressing](#)
256 [`\[WSAddressing\]` specifies the endpoint reference to which WS-RM protocol messages related to the](#)
257 [offered Sequence are to be sent.](#)

258 `/wsrm:CreateSequence/wsrm:Offer/wsrm:Expires`

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

262 `/wsrm:CreateSequence/wsrm:Offer/wsrm:Expires/@{any}`

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

265 `/wsrm:CreateSequence/wsrm:Offer/{any}`

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

268 `/wsrm:CreateSequence/wsrm:Offer/@{any}`

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

271 `/wsrm:CreateSequence/{any}`

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

274 `/wsrm:CreateSequence/@{any}`

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

277 A `<wsrm:CreateSequenceResponse>` is sent in the body of a response message by an RM
278 Destination in response to receipt of a `<wsrm:CreateSequence>` request message. It carries the
279 `<wsrm:Identifier>` of the created Sequence and indicates that the RM Source may begin sending
280 messages in the context of the identified Sequence.

281 The following exemplar defines the `<wsrm:CreateSequenceResponse>` syntax:

```
282 <wsrm:CreateSequenceResponse ...>
283   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
284   <wsrm:Expires> xs:duration </wsrm:Expires> ?
285   <wsrm:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?
286   <wsrm:Accept ...>
287     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
288     ...
289   </wsrm:Accept> ?
290   ...
291 </wsrm:CreateSequenceResponse>
```

292 `/wsrm:CreateSequenceResponse`

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

296 `/wsrm:CreateSequenceResponse/wsrm:Identifier`

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

299 `/wsrm:CreateSequenceResponse/wsrm:Identifier/@{any}`

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

302 `/wsrm:CreateSequenceResponse/wsrm:Expires`

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

307 `/wsrm:CreateSequenceResponse/wsrm:Expires/@{any}`

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

310 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval`

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

313 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@Milliseconds`

314 The acknowledgement interval, specified in milliseconds.

315 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@{any}`

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

318 `/wsrm:CreateSequenceResponse/wsrm:Accept`

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

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

324 `/wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo`

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

328 `/wsrm:CreateSequenceResponse/wsrm:Accept/{any}`

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

331 `/wsrm:CreateSequenceResponse/wsrm:Accept/@{any}`

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

334 `/wsrm:CreateSequenceResponse/{any}`

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

337 `/wsrm:CreateSequenceResponse/@{any}`

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

340 **3.2 Closing A Sequence**

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

346 If the RM Source wishes to close the Sequence, then it sends a `<wsrm:CloseSequence>` element, in
347 the body of a message, to the RM Destination. This message indicates that the RM Destination MUST
348 NOT receive any new messages for the specified Sequence, other than those already received at the time
349 the `<wsrm:CloseSequence>` element is interpreted by the RMD. Upon receipt of this message, or
350 subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST
351 include a final SequenceAcknowledgement (that MUST include the `<wsrm:Final>` element) header block
352 on any messages associated with the Sequence destined to the RM Source, including the
353 CloseSequenceResponse message or then it sends a `<wsrm:CloseSequence>` element, in the body of
354 a message, to the RM Destination. This message indicates that the RM Destination MUST NOT receive
355 any new messages for the specified sequence, other than those already received at the time the
356 `<wsrm:CloseSequence>` element is interpreted by the RMD. Upon receipt of this message, or
357 subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST
358 include a final SequenceAcknowledgement (that MUST include the `<wsrm:Final>` element) header block
359 on each message destined to the RM Source, including the CloseSequenceResponse message and on
360 any Sequence Fault transmitted to the RMS.

361 While the RM Destination MUST NOT receive any new messages for the specified Sequence it MUST still
362 process RM protocol messages. For example, it MUST respond to AckRequested, TerminateSequence
363 as well as CloseSequence messages. Note, subsequent CloseSequence messages have no effect on the
364 state of the Ssequence it MUST still process RM protocol messages. For example, it MUST respond to
365 AckRequested, TerminateSequence as well as CloseSequence messages. Note, subsequent
366 CloseSequence messages have no effect on the state of the sequence.

367 In the case where the RM Destination wishes to discontinue use of a Sequence it is RECOMMENDED
368 that it close the Sequence~~sequence it may close the sequence itself~~. Please see <wsrm:Final> and the
369 SequenceClosed fault. Whenever possible the SequenceClosed Fault SHOULD be used in place of the
370 SequenceTerminated Fault, whenever possible, to allow the RM Source to still receive
371 Acknowledgements.

372 The following exemplar defines the CloseSequence syntax:

```
373 <wsrm:CloseSequence ...>  
374   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
375   ...  
376 </wsrm:CloseSequence>
```

377 /wsrm:CloseSequence

378 This element is sent by an RM Source to indicate that the RM Destination MUST NOT receive any new
379 messages for this Sequence. A SequenceClosed fault MUST be generated by the RM Destination when it
380 receives a message for a Ssequence. A SequenceClosed fault MUST be generated by the RM-
381 Destination when it receives a message for a sequence that is closed.

382 /wsrm:CloseSequence/wsrm:Identifier

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

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

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

388 /wsrm:CloseSequence/{any}

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

391 /wsrm:CloseSequence@{any}

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

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

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

```
398 <wsrm:CloseSequenceResponse ...>  
399   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
400   ...  
401 </wsrm:CloseSequenceResponse>
```

402 /wsrm:CloseSequenceResponse

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

406 `/wsrm:CloseSequenceResponse/wsrm:Identifier`

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

409 `/wsrm:CloseSequenceResponse/wsrm:Identifier/@{any}`

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

412 `/wsrm:CloseSequenceResponse/{any}`

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

415 `/wsrm:CloseSequenceResponse@{any}`

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

418 **3.3 Sequence Termination**

419 When the RM Source has completed its use of the Sequence it sends a `<wsrm:TerminateSequence>`
420 element, in the body of a message, to the RM Destination to indicate that the Sequence is complete and
421 that it will not be sending any further messages related to the Sequence. The RM Destination can safely
422 reclaim any resources associated with the Sequence upon receipt of the `<wsrm:TerminateSequence>`
423 message. Under normal usage the RM ~~Source will complete its use of the S~~source will complete its use of
424 ~~the s~~sequence when all of the messages in the Sequence have been acknowledged. However, the RM
425 Source is free to Terminate or Close a Sequence at any time regardless of the acknowledgement state of
426 the messages.

427 The following exemplar defines the TerminateSequence syntax:

```
428 <wsrm:TerminateSequence ...>  
429   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
430   ...  
431 </wsrm:TerminateSequence>
```

432 `/wsrm:TerminateSequence`

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

438 `/wsrm:TerminateSequence/wsrm:Identifier`

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

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

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

444 /wsrm:TerminateSequence/{any}

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

447 /wsrm:TerminateSequence/@{any}

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

450 A <wsrm:TerminateSequenceResponse> is sent in the body of a response message by an RM
451 Destination in response to receipt of a <wsrm:TerminateSequence> request message. It indicates that
452 the RM Destination has terminated the Sequence.

453 The following exemplar defines the <wsrm:TerminateSequenceResponse> syntax:

```
454 <wsrm:TerminateSequenceResponse ...>  
455   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
456   ...  
457 </wsrm:TerminateSequenceResponse>
```

458 /wsrm:TerminateSequenceResponse

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

462 /wsrm:TerminateSequenceResponse/wsrm:Identifier

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

465 /wsrm:TerminateSequenceResponse/wsrm:Identifier/@{any}

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

468 /wsrm:TerminateSequenceResponse/{any}

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

471 /wsrm:TerminateSequenceResponse/@{any}

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

474 On receipt of a <wsrm:TerminateSequence> message an RM Destination MUST respond with a
475 corresponding <wsrm:TerminateSequenceResponse> message or generate a fault.

476 3.4 Sequences

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

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

484 A following exemplar defines its syntax:

```
485 <wsrm:Sequence ...>
486   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
487   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>
488   ...
489 </wsrm:Sequence>
```

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

491 /wsrm:Sequence

492 This [protocol element associates the message in which it is contained with a previously established RM Sequence. It contains the Sequence's unique identifier and the containing message's ordinal position within that Sequence](#) is the element containing Sequence information for WS-ReliableMessaging. The
493
494
495 <wsrm:Sequence> element MUST be understood by the RM Destination. The <wsrm:Sequence>
496 element MUST have a `mustUnderstand` attribute with a value 1/true from the namespace corresponding
497 to the version of SOAP to which the <wsrm:Sequence> SOAP header block is bound.

498 /wsrm:Sequence/wsrm:Identifier

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

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

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

504 /wsrm:Sequence/wsrm:MessageNumber

505 This REQUIRED element MUST contain a `wsrm:MessageNumberType` representing the ordinal position
506 of the message within a Sequence. Sequence MessageNumbers start at 1 and monotonically increase
507 throughout the Sequence. If the message number exceeds the internal limitations of an RM Source or RM
508 Destination or reaches the maximum value of 9,223,372,036,854,775,807 the RM Source or Destination
509 MUST [generate](#) a MessageNumberRollover fault.

510 /wsrm:Sequence/{any}

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

513 /wsrm:Sequence/@{any}

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

516 The following example illustrates a Sequence header block.

```
517 <wsrm:Sequence>
518   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
519   <wsrm:MessageNumber>10</wsrm:MessageNumber>
520 </wsrm:Sequence>
```

521 3.5 Request Acknowledgement

522 The purpose of the <wsrm:AckRequested> header block is to signal to the RM Destination that the RM
523 Source is requesting that a <wsrm:SequenceAcknowledgement> be sent.

524 The RM Source MAY request an acknowledgement message from the RM Destination at any time by
 525 including an <wsrm:AckRequested> header block in any message targeted to the RM Destination. An
 526 RM Destination that receives a message that contains an <wsrm:AckRequested> header block MUST
 527 send a message containing a <wsrm:SequenceAcknowledgement> header block to the wsrm:AcksTo
 528 endpoint reference (see Section 3.1). If a non-mustUnderstand fault occurs when processing an RM
 529 Header that was piggy-backed on another message, a fault MUST be generated, but the processing of
 530 the original message MUST NOT be affected. It is RECOMMENDED that the RMD return a
 531 <wsrm:AcknowledgementRange> or <wsrm:None> element instead of a <wsrm:Nack> element (see
 532 below). ~~may request an acknowledgement message from the RM Destination at any time by including an~~
 533 ~~<wsrm:AckRequested> header block in any message targeted to the RM Destination. An RM-~~
 534 ~~Destination that receives a message that contains an <wsrm:AckRequested> header block MUST send~~
 535 ~~a message containing a <wsrm:SequenceAcknowledgement> header block to the wsrm:AcksTo-~~
 536 ~~endpoint reference (see Section 3.1). If a non-mustUnderstand fault occurs when processing an RM-~~
 537 ~~Header that was piggy-backed on another message, a fault MUST be generated, but the processing of~~
 538 ~~the original message MUST NOT be affected.~~

539 The following exemplar defines its syntax:

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

544 /wsrm:AckRequested

545 This element requests an acknowledgement for the identified Ssequence.

546 /wsrm:AckRequested/wsrm:Identifier

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

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

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

552 /wsrm:AckRequested/{any}

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

555 /wsrm:AckRequested/@{any}

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

558 3.6 Sequence Acknowledgement

559 The RM Destination informs the RM Source of successful message receipt using a
 560 <wsrm:SequenceAcknowledgement> header block. The <wsrm:SequenceAcknowledgement>
 561 header block MAY be transmitted independently or included on any message targeted to the AcksTo EPR.
 562 The RM Destination MAY send a <wsrm:SequenceAcknowledgement> header block at any point
 563 during which the S- ~~The RM Destination MAY send a <wsrm:SequenceAcknowledgement> header-~~
 564 ~~block at any point during which the~~ sequence is valid. Acknowledgements can be explicitly requested
 565 using the <wsrm:AckRequested> directive (see Section [Request Acknowledgement](#)). If a non-

566 mustUnderstand fault occurs when processing an RM Header that was piggy-backed on another
567 message, a fault MUST be generated, but the processing of the original message MUST NOT be
568 affected.

569 A RMD MAY include a wsrn:SequenceAcknowledgement header block on any SOAP envelope targetted
570 to the endpoint referenced by the wsrn:AcksTo EPR. This concept is often referred to as "piggy-backing"
571 Sequence acknowledgements.

572 A wsrn:AcksTo EPR MAY specify the WS-Addressing anonymous IRI as its address. When the
573 wsrn:AcksTo EPR specifies the WS-Addressing anonymous IRI as its address, the RM Destination
574 MUST transmit any wsrn:SequenceAcknowledgement headers for the created Sequence in a SOAP
575 envelope to be transmitted on the protocol binding-specific channel. Such a channel is provided by the
576 context of a received message containing a SOAP envelope that contains a wsrn:Sequence header block
577 and/or a wsrn:AckRequested header block for that same Sequence identifier.

578 The following exemplar defines its syntax:

```
579 <wsrn:SequenceAcknowledgement ...>
580   <wsrn:Identifier ...> xs:anyURI </wsrn:Identifier>
581   [ [ [ <wsrn:AcknowledgementRange ...
582         Upper="wsrn:MessageNumberType"
583         Lower="wsrn:MessageNumberType"/> +
584
585         | <wsrn:None/> ]
586         <wsrn:Final/> ? ]
587   | <wsrn:Nack> wsrn:MessageNumberType </wsrn:Nack> + ]
588   ...
589 </wsrn:SequenceAcknowledgement>
```

590 The following describes the content model of the <wsrn:SequenceAcknowledgement> header block.

591 /wsrn:SequenceAcknowledgement

592 This element contains the Sequence acknowledgement information.

593 /wsrn:SequenceAcknowledgement/wsrn:Identifier

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

597 /wsrn:SequenceAcknowledgement/wsrn:Identifier/@{any}

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

600 /wsrn:SequenceAcknowledgement/wsrn:AcknowledgementRange

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

605 /wsrn:SequenceAcknowledgement/wsrn:AcknowledgementRange/@Upper

606 This REQUIRED attribute contains a wsrn:MessageNumberType representing the
607 <wsrn:MessageNumber> of the highest contiguous message in a Sequence range received by the RM
608 Destination.

609 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Lower
 610 This REQUIRED attribute contains a wsrm:MessageNumberType representing the
 611 <wsrm:MessageNumber> of the lowest contiguous message in a Sequence range received by the RM
 612 Destination.

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

616 /wsrm:SequenceAcknowledgement/wsrm:Final
 617 This OPTIONAL element, if present, indicates that the RM Destination is not receiving new messages for
 618 the specified Sequence. The RM Source can be assured that the ranges of messages acknowledged by
 619 this SequenceAcknowledgement header block will not change in the future. This element MUST be
 620 present when the Sequence is closed. Note: this element MUST NOT be used when sending a Nack, it
 621 can only be used when sending AcknowledgementRanges or <wsrm:None>.

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

634 /wsrm:SequenceAcknowledgement/wsrm:None
 635 This OPTIONAL element, if present, MUST be used when the RM Destination has not received any
 636 messages for the specified Sequence. The <wsrm:None> element MUST NOT be present if a sibling
 637 <wsrm:AcknowledgementRange> or <wsrm:Nack> element is also present as a child of the
 638 <wsrm:SequenceAcknowledgement>.

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

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

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

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

```

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

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

```
653 <wsrm:SequenceAcknowledgement>  
654     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
655     <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
656     <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
657     <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
658 </wsrm:SequenceAcknowledgement>
```

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

```
660 <wsrm:SequenceAcknowledgement>  
661     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
662     <wsrm:Nack>3</wsrm:Nack>  
663 </wsrm:SequenceAcknowledgement>
```

4 Faults

The faults defined in this section fall into one of two categories: those faults that are the result of messages or operations within a specific Sequence and those faults that are not. By their nature the `CreateSequenceRefused`, `UnknownSequence`, and `WSRMRequired` faults cannot be correlated with a Sequence. All other faults defined in this section relate to the processing of WS-RM protocol messages or messages containing WS-RM header blocks targeted at a specific Sequence and are collectively referred to as "Sequence faults" definitions defined in this section reference certain abstract properties, such as [fault endpoint], that are defined in section 3 of the WS-Addressing [WS-Addressing] specification. Endpoints compliant with this specification MUST include required Message Addressing Properties on all fault messages.

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

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

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

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

The definitions of faults use the following properties:

[Code] The fault code.

[Subcode] The fault subcode.

[Reason] The English language reason element.

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

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

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

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

```
<S:Envelope>
  <S:Header>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/08/addressing/fault
    </wsa:Action>
  <!-- Headers elided for clarity. -->
```

```

704 </S:Header>
705 <S:Body>
706   <S:Fault>
707     <S:Code>
708       <S:Value> [Code] </S:Value>
709       <S:Subcode>
710         <S:Value> [Subcode] </S:Value>
711       </S:Subcode>
712     </S:Code>
713     <S:Reason>
714       <S:Text xml:lang="en"> [Reason] </S:Text>
715     </S:Reason>
716     <S:Detail>
717       [Detail]
718       ...
719     </S:Detail>
720   </S:Fault>
721 </S:Body>
722 </S:Envelope>

```

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

```

725 <S11:Envelope>
726   <S11:Header>
727     <wsrm:SequenceFault>
728       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
729       <wsrm:Detail> [Detail] </wsrm:Detail>
730       ...
731     </wsrm:SequenceFault>
732     <!-- Headers elided for clarity. -->
733   </S11:Header>
734   <S11:Body>
735     <S11:Fault>
736       <faultcode> [Code] </faultcode>
737       <faultstring> [Reason] </faultstring>
738     </S11:Fault>
739   </S11:Body>
740 </S11:Envelope>

```

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

```

743 <S11:Envelope>
744   <S11:Body>
745     <S11:Fault>
746       <faultcode> [Subcode] </faultcode>
747       <faultstring> [Reason] </faultstring>
748     </S11:Fault>
749   </S11:Body>
750 </S11:Envelope>

```

751 4.1 SequenceFault Element

752 The purpose of the <wsrm:SequenceFault> element is to carry the specific details of a fault generated
 753 during the reliable messaging specific processing of a message belonging to a Sequence. The
 754 <wsrm:SequenceFault> container MUST only be used in conjunction with the SOAP [1.1 fault](#)
 755 [mechanism. It MUST NOT be used in conjunction with the SOAP 1.1 fault mechanism. It MUST NOT be](#)
 756 [used in conjunction with the SOAP 1.2 binding.](#)

757 The following exemplar defines its syntax:

```

758 <wsrm:SequenceFault ...>
759   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
760   <wsrm:Detail> ... </wsrm:Detail> ?
761   ...
762 </wsrm:SequenceFault>

```

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

764 `/wsrm:SequenceFault`

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

766 `/wsrm:SequenceFault/wsrm:FaultCode`

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

768 `/wsrm:SequenceFault/wsrm:Detail`

769 This optional element is intended for carrying application specific error information related to the fault being described.

771 `/wsrm:SequenceFault/wsrm:Detail/{any}`

772 The application specific error information related to the fault being described.

773 `/wsrm:SequenceFault/wsrm:Detail/@{any}`

774 The application specific error information related to the fault being described.

775 `/wsrm:SequenceFault/{any}`

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

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

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

781 4.2 Sequence Terminated

782 This fault is generated by either the RM Source or the RM Destination to indicate that it has either
 783 encountered an unrecoverable condition, or has detected a violation of the protocol and as a
 784 consequence, has chosen to terminate the Ssent by either the RM Source or the RM Destination to
 785 indicate that it has either encountered an unrecoverable condition, or has detected a violation of the
 786 protocol and as a consequence, has chosen to terminate the sequence. The endpoint that generates this
 787 fault should make every reasonable effort to notify the corresponding endpoint of this decision.

788 Receipt of SequenceTerminated by either the RMD or the RMS shall terminate the Ssequence if it is not
 789 otherwise terminated.

790 Properties:

791 [Code] Sender or Receiver

792 [Subcode] `wsrm:SequenceTerminated`

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

794 [Detail]

```

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

```


4.3 Unknown Sequence

This fault is ~~generated by either the RM Source or the RM Destination in response to a message containing an unknown or terminated Sequence identifier. Receipt of UnknownSequence by either the RMD or the RMS shall terminate the Ssent by either the RM Source or the RM Destination in response to a message containing an unknown or terminated sequence identifier. Receipt of UnknownSequence by either the RMD or the RMS shall terminate the~~ sequence if it is not otherwise terminated.

Properties:

[Code] Sender

[Subcode] wsrn:UnknownSequence

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

[Detail]

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

4.4 Invalid Acknowledgement

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

[Code] Sender

[Subcode] wsrn:InvalidAcknowledgement

[Reason] The SequenceAcknowledgement violates the cumulative acknowledgement invariant.

[Detail]

```
<wsrn:SequenceAcknowledgement ...> ... </wsrn:SequenceAcknowledgement>
```

4.5 Message Number Rollover

This fault is ~~generated to indicate that message numbers for a Ssent to indicate that message numbers for a~~ sequence have been exhausted.

Properties:

[Code] Sender

[Subcode] wsrn:MessageNumberRollover

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

[Detail]

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

4.6 Create Sequence Refused

This fault is ~~generated in response to a create Ssent in response to a create~~ sequence request that cannot be satisfied.

Properties:

[Code] Sender

831 [Subcode] wsrn:CreateSequenceRefused

832 [Reason] The create ~~S~~sequence request has been refused by the RM Destination.

833 [Detail]

834 `xs:any`

835 4.7 Sequence Closed

836 This fault is ~~generated by an RM Destination to indicate that the specified Sequence has been closed.~~

837 ~~This fault MUST be generated when an RM Destination is asked to receive a message for a Ssent by an~~

838 ~~RM Destination to indicate that the specified sequence has been closed. This fault MUST be generated~~

839 ~~when an RM Destination is asked to receive a message for a~~ sequence that is closed.

840 Properties:

841 [Code] Sender

842 [Subcode] wsrn:SequenceClosed

843 [Reason] The ~~S~~sequence is closed and can not receive new messages.

844 [Detail]

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

846 4.8 WSRM Required

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

849 Properties:

850 [Code] Sender

851 [Subcode] wsrn:WSRMRequired

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

853 [Detail]

854 `xs:any`

5 Security Considerations

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

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

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

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

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

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

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

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

- 897 • **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by comparing
898 secured policies – see WS-Policy and WS-SecurityPolicy).
- 899 • **Authentication** – Authentication is established using the mechanisms described in WS-Security
900 and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- 901 • **Accountability** – Accountability is a function of the type of and string of the key and algorithms
902 being used. In many cases, a strong symmetric key provides sufficient accountability. However, in
903 some environments, strong PKI signatures are required.
- 904 • **Availability** – All reliable messaging services are subject to a variety of availability attacks. Replay
905 detection is a common attack and it is recommended that this be addressed by the mechanisms
906 described in WS-Security. (Note that because of legitimate message replays, detection should
907 include a differentiator besides message id such as a timestamp). Other attacks, such as network-
908 level denial of service attacks are harder to avoid and are outside the scope of this specification.
909 That said, care should be taken to ensure that minimal state is saved prior to any authenticating
910 Ssequences.

6 References

6.1 Normative

[KEYWORDS]

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

[SOAP 1.1]

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

[SOAP 1.2]

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

[URI]

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

[XML]

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

[XML-ns]

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

[XML-Schema Part1]

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

[XML-Schema Part2]

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

[WSDL 1.1]

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

[WS-Addressing]

[W3C Proposed Recommendation, "Web Services Addressing 1.0 - Core", March 2006](#)~~D. Box, et al, "Web Services Addressing (WS-Addressing)," August 2004.~~

[W3C Proposed Recommendation, "Web Services Addressing 1.0 – SOAP Binding", March 2006.](#)

6.2 Non-Normative

[RDDL 2.0]

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

[WS-Policy]

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

[WS-PolicyAttachment]

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

944 **[WS-Security]**

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

947 [Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "OASIS Web Services Security:](#)
948 [SOAP Message Security 1.1 \(WS-Security 2004\)](#)", OASIS Standard 200602, February 2006.

949 **[RTTM]**

950 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May
951 1992.

952 **[SecurityPolicy]**

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

954 **[SecureConversation]**

955 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," February
956 2005.

957 **[Trust]**

958 S. Anderson, et al, "Web Services Trust Language (WS-Trust)," February 2005.

A. Schema

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

<http://docs.oasis-open.org/ws-rx/wsrn/200604/wsrn-1.1-schema-2006042/wsrn-1.1-schema-200602.xsd>

The following copy is provided for reference.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
OASIS takes no position regarding the validity or scope of any intellectual
property or other rights that might be claimed to pertain to the
implementation or use of the technology described in this document or the
extent to which any license under such rights might or might not be available;
neither does it represent that it has made any effort to identify any such
rights. Information on OASIS's procedures with respect to rights in OASIS
specifications can be found at the OASIS website. Copies of claims of rights
made available for publication and any assurances of licenses to be made
available, or the result of an attempt made to obtain a general license or
permission for the use of such proprietary rights by implementors or users of
this specification, can be obtained from the OASIS Executive Director.
OASIS invites any interested party to bring to its attention any copyrights,
patents or patent applications, or other proprietary rights which may cover
technology that may be required to implement this specification. Please
address the information to the OASIS Executive Director.
Copyright © OASIS Open 2002-2006. All Rights Reserved.
This document and translations of it may be copied and furnished to others,
and derivative works that comment on or otherwise explain it or assist in its
implementation may be prepared, copied, published and distributed, in whole or
in part, without restriction of any kind, provided that the above copyright
notice and this paragraph are included on all such copies and derivative
works. However, this document itself does not be modified in any way, such as
by removing the copyright notice or references to OASIS, except as needed for
the purpose of developing OASIS specifications, in which case the procedures
for copyrights defined in the OASIS Intellectual Property Rights document must
be followed, or as required to translate it into languages other than English.
The limited permissions granted above are perpetual and will not be revoked by
OASIS or its successors or assigns.
This document and the information contained herein is provided on an "AS
IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
FITNESS FOR A PARTICULAR PURPOSE.
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200604"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
  <!-- Protocol Elements -->
  <xs:complexType name="SequenceType">
    <xs:sequence>
      <xs:element ref="wsrm:Identifier"/>
      <xs:element name="MessageNumber" type="wsrm:MessageNumberType"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

1015     </xs:sequence>
1016     <xs:anyAttribute namespace="##other" processContents="lax"/>
1017 </xs:complexType>
1018 <xs:element name="Sequence" type="wsrm:SequenceType"/>
1019 <xs:element name="SequenceAcknowledgement">
1020   <xs:complexType>
1021     <xs:sequence>
1022       <xs:element ref="wsrm:Identifier"/>
1023       <xs:choice>
1024         <xs:sequence>
1025           <xs:choice>
1026             <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
1027               <xs:complexType>
1028                 <xs:sequence/>
1029                 <xs:attribute name="Upper" type="xs:unsignedLong"
1030 use="required"/>
1031                 <xs:attribute name="Lower" type="xs:unsignedLong"
1032 use="required"/>
1033               <xs:anyAttribute namespace="##other" processContents="lax"/>
1034             </xs:complexType>
1035             </xs:element>
1036             <xs:element name="None" minOccurs="0">
1037               <xs:complexType>
1038                 <xs:sequence/>
1039               </xs:complexType>
1040             </xs:element>
1041             </xs:choice>
1042             <xs:element name="Final" minOccurs="0">
1043               <xs:complexType>
1044                 <xs:sequence/>
1045               </xs:complexType>
1046             </xs:element>
1047             </xs:sequence>
1048             <xs:element name="Nack" type="xs:unsignedLong"
1049 maxOccurs="unbounded"/>
1050           </xs:choice>
1051           <xs:any namespace="##other" processContents="lax" minOccurs="0"
1052 maxOccurs="unbounded"/>
1053         </xs:sequence>
1054         <xs:anyAttribute namespace="##other" processContents="lax"/>
1055       </xs:complexType>
1056     </xs:element>
1057     <xs:complexType name="AckRequestedType">
1058       <xs:sequence>
1059         <xs:element ref="wsrm:Identifier"/>
1060         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1061 maxOccurs="unbounded"/>
1062       </xs:sequence>
1063       <xs:anyAttribute namespace="##other" processContents="lax"/>
1064     </xs:complexType>
1065     <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1066     <xs:element name="Identifier">
1067       <xs:complexType>
1068         <xs:annotation>
1069           <xs:documentation>
1070             This type is for elements whose [children] is an anyURI and can have
1071             arbitrary attributes.
1072           </xs:documentation>
1073         </xs:annotation>
1074         <xs:simpleContent>
1075           <xs:extension base="xs:anyURI">
1076             <xs:anyAttribute namespace="##other" processContents="lax"/>
1077           </xs:extension>

```



```

1078     </xs:simpleContent>
1079   </xs:complexType>
1080 </xs:element>
1081   <xs:simpleType name="MessageNumberType">
1082     <xs:restriction base="xs:unsignedLong">
1083       <xs:minInclusive value="1"/>
1084       <xs:maxInclusive value="9223372036854775807"/>
1085     </xs:restriction>
1086   </xs:simpleType>
1087   <!-- Fault Container and Codes -->
1088   <xs:simpleType name="FaultCodes">
1089     <xs:restriction base="xs:QName">
1090       <xs:enumeration value="wsrm:SequenceTerminated"/>
1091       <xs:enumeration value="wsrm:UnknownSequence"/>
1092       <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1093       <xs:enumeration value="wsrm:MessageNumberRollover"/>
1094       <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1095       <xs:enumeration value="wsrm:SequenceClosed"/>
1096       <xs:enumeration value="wsrm:WSRMRequired"/>
1097     </xs:restriction>
1098   </xs:simpleType>
1099   <xs:complexType name="SequenceFaultType">
1100     <xs:sequence>
1101       <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1102       <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
1103       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1104 maxOccurs="unbounded"/>
1105     </xs:sequence>
1106     <xs:anyAttribute namespace="##other" processContents="lax"/>
1107   </xs:complexType>
1108   <xs:complexType name="DetailType">
1109     <xs:sequence>
1110       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1111 maxOccurs="unbounded"/>
1112     </xs:sequence>
1113     <xs:anyAttribute namespace="##other" processContents="lax"/>
1114   </xs:complexType>
1115   <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1116   <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1117   <xs:element name="CreateSequenceResponse"
1118 type="wsrm:CreateSequenceResponseType"/>
1119   <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1120   <xs:element name="CloseSequenceResponse"
1121 type="wsrm:CloseSequenceResponseType"/>
1122   <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1123   <xs:element name="TerminateSequenceResponse"
1124 type="wsrm:TerminateSequenceResponseType"/>
1125   <xs:complexType name="CreateSequenceType">
1126     <xs:sequence>
1127       <xs:element ref="wsrm:AcksTo"/>
1128       <xs:element ref="wsrm:Expires" minOccurs="0"/>
1129       <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1130       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1131 maxOccurs="unbounded"/>
1132     <xs:annotation>
1133       <xs:documentation>
1134         It is the authors intent that this extensibility be used to
1135 transfer a Security Token Reference as defined in WS-Security.
1136       </xs:documentation>
1137     </xs:annotation>
1138   </xs:complexType>
1139   </xs:sequence>
1140   <xs:anyAttribute namespace="##other" processContents="lax"/>

```

```

1141 </xs:complexType>
1142 <xs:complexType name="CreateSequenceResponseType">
1143   <xs:sequence>
1144     <xs:element ref="wsrm:Identifier"/>
1145     <xs:element ref="wsrm:Expires" minOccurs="0"/>
1146     <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>
1147     <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1148     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1149 maxOccurs="unbounded"/>
1150   </xs:sequence>
1151   <xs:anyAttribute namespace="##other" processContents="lax"/>
1152 </xs:complexType>
1153 <xs:complexType name="CloseSequenceType">
1154   <xs:sequence>
1155     <xs:element ref="wsrm:Identifier"/>
1156     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1157 maxOccurs="unbounded"/>
1158   </xs:sequence>
1159   <xs:anyAttribute namespace="##other" processContents="lax"/>
1160 </xs:complexType>
1161 <xs:complexType name="CloseSequenceResponseType">
1162   <xs:sequence>
1163     <xs:element ref="wsrm:Identifier"/>
1164     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1165 maxOccurs="unbounded"/>
1166   </xs:sequence>
1167   <xs:anyAttribute namespace="##other" processContents="lax"/>
1168 </xs:complexType>
1169 <xs:complexType name="TerminateSequenceType">
1170   <xs:sequence>
1171     <xs:element ref="wsrm:Identifier"/>
1172     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1173 maxOccurs="unbounded"/>
1174   </xs:sequence>
1175   <xs:anyAttribute namespace="##other" processContents="lax"/>
1176 </xs:complexType>
1177 <xs:complexType name="TerminateSequenceResponseType">
1178   <xs:sequence>
1179     <xs:element ref="wsrm:Identifier"/>
1180     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1181 maxOccurs="unbounded"/>
1182   </xs:sequence>
1183   <xs:anyAttribute namespace="##other" processContents="lax"/>
1184 </xs:complexType>
1185 <xs:element

```

```

1186     name="AcksTo" type="wsa:EndpointReferenceType"/>
1187     <xs:complexType name="OfferType">
1188         <xs:sequence>
1189             <xs:element ref="wsrm:Identifier"/>
1190             <xs:element ref="wsrm:Expires" minOccurs="0"/>
1191             <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
1192             <xs:any namespace="##other" processContents="lax" minOccurs="0"
1193 maxOccurs="unbounded"/>
1194         </xs:sequence>
1195         <xs:anyAttribute namespace="##other" processContents="lax"/>
1196     </xs:complexType>
1197     <xs:complexType name="AcceptType">
1198         <xs:sequence>
1199             <xs:element ref="wsrm:AcksTo"/>
1200             <xs:any namespace="##other" processContents="lax" minOccurs="0"
1201 maxOccurs="unbounded"/>
1202         </xs:sequence>
1203         <xs:anyAttribute namespace="##other" processContents="lax"/>
1204     </xs:complexType>
1205     <xs:element name="Expires">
1206         <xs:complexType>
1207             <xs:simpleContent>
1208                 <xs:extension base="xs:duration">
1209                     <xs:anyAttribute namespace="##other" processContents="lax"/>
1210                 </xs:extension>
1211             </xs:simpleContent>
1212         </xs:complexType>
1213     </xs:element>
1214     <xs:element name="AcknowledgementInterval">
1215         <xs:complexType>
1216             <xs:sequence/>
1217             <xs:attribute name="Milliseconds" type="xs:unsignedLong"
1218 use="required"/>
1219             <xs:anyAttribute namespace="##other" processContents="lax"/>
1220         </xs:complexType>
1221     </xs:element>
1222 </xs:schema>

```

```

1223 <?xml version="1.0" encoding="UTF-8"?>
1224 <!--
1225 OASIS takes no position regarding the validity or scope of any intellectual
1226 property or other rights that might be claimed to pertain to the
1227 implementation or use of the technology described in this document or the
1228 extent to which any license under such rights might or might not be available;
1229 neither does it represent that it has made any effort to identify any such
1230 rights. Information on OASIS's procedures with respect to rights in OASIS
1231 specifications can be found at the OASIS website. Copies of claims of rights
1232 made available for publication and any assurances of licenses to be made
1233 available, or the result of an attempt made to obtain a general license or
1234 permission for the use of such proprietary rights by implementors or users of
1235 this specification, can be obtained from the OASIS Executive Director.
1236 OASIS invites any interested party to bring to its attention any copyrights,
1237 patents or patent applications, or other proprietary rights which may cover
1238 technology that may be required to implement this specification. Please
1239 address the information to the OASIS Executive Director.
1240 Copyright &copy; OASIS Open 2002-2006. All Rights Reserved.
1241 This document and translations of it may be copied and furnished to others,
1242 and derivative works that comment on or otherwise explain it or assist in its
1243 implementation may be prepared, copied, published and distributed, in whole or
1244 in part, without restriction of any kind, provided that the above copyright
1245 notice and this paragraph are included on all such copies and derivative
1246 works. However, this document itself does not be modified in any way, such as
1247 by removing the copyright notice or references to OASIS, except as needed for
1248 the purpose of developing OASIS specifications, in which case the procedures
1249 for copyrights defined in the OASIS Intellectual Property Rights document must
1250 be followed, or as required to translate it into languages other than English.
1251 The limited permissions granted above are perpetual and will not be revoked by
1252 OASIS or its successors or assigns.
1253 This document and the information contained herein is provided on an "AS
1254 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
1255 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
1256 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
1257 FITNESS FOR A PARTICULAR PURPOSE.
1258 -->
1259 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1260 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1261 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200602"
1262 targetNamespace="http://docs.oasis-open.org/ws-rx/wsm/200602"
1263 elementFormDefault="qualified" attributeFormDefault="unqualified">
1264 <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1265 schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
1266 <!-- Protocol Elements -->
1267 <xs:complexType name="SequenceType">
1268 <xs:sequence>
1269 <xs:element ref="wsm:Identifier"/>
1270 <xs:element name="MessageNumber" type="wsm:MessageNumberType"/>
1271 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1272 maxOccurs="unbounded"/>
1273 </xs:sequence>
1274 <xs:anyAttribute namespace="##other" processContents="lax"/>
1275 </xs:complexType>
1276 <xs:element name="Sequence" type="wsm:SequenceType"/>
1277 <xs:element name="SequenceAcknowledgement">
1278 <xs:complexType>
1279 <xs:sequence>
1280 <xs:element ref="wsm:Identifier"/>
1281 <xs:choice>
1282 <xs:sequence>
1283 <xs:choice>
1284 <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
1285 <xs:complexType>

```

```

1286         <xs:sequence/>
1287         <xs:attribute name="Upper" type="xs:unsignedLong"
1288 use="required"/>
1289         <xs:attribute name="Lower" type="xs:unsignedLong"
1290 use="required"/>
1291         <xs:anyAttribute namespace="##other" processContents="lax"/>
1292     </xs:complexType>
1293 </xs:element>
1294     <xs:element name="None" minOccurs="0">
1295         <xs:complexType>
1296             <xs:sequence/>
1297         </xs:complexType>
1298     </xs:element>
1299 </xs:choice>
1300     <xs:element name="Final" minOccurs="0">
1301         <xs:complexType>
1302             <xs:sequence/>
1303         </xs:complexType>
1304     </xs:element>
1305 </xs:sequence>
1306     <xs:element name="Nack" type="xs:unsignedLong"
1307 minOccurs="unbounded"/>
1308 </xs:choice>
1309     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1310 minOccurs="unbounded"/>
1311 </xs:sequence>
1312 <xs:anyAttribute namespace="##other" processContents="lax"/>
1313 </xs:complexType>
1314 </xs:element>
1315 <xs:complexType name="AckRequestedType">
1316     <xs:sequence>
1317         <xs:element ref="wsrm:Identifier"/>
1318         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1319 minOccurs="unbounded"/>
1320     </xs:sequence>
1321     <xs:anyAttribute namespace="##other" processContents="lax"/>
1322 </xs:complexType>
1323 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1324 <xs:element name="Identifier">
1325     <xs:complexType>
1326         <xs:annotation>
1327             <xs:documentation>
1328                 This type is for elements whose [children] is an anyURI and can have
1329 arbitrary attributes.
1330             </xs:documentation>
1331         </xs:annotation>
1332         <xs:simpleContent>
1333             <xs:extension base="xs:anyURI">
1334                 <xs:anyAttribute namespace="##other" processContents="lax"/>
1335             </xs:extension>
1336         </xs:simpleContent>
1337     </xs:complexType>
1338 </xs:element>
1339     <xs:simpleType name="MessageNumberType">
1340         <xs:restriction base="xs:unsignedLong">
1341             <xs:minInclusive value="1"/>
1342             <xs:maxInclusive value="9223372036854775807"/>
1343         </xs:restriction>
1344     </xs:simpleType>
1345 <!-- Fault Container and Codes -->
1346 <xs:simpleType name="FaultCodes">
1347     <xs:restriction base="xs:QName">

```

```

1348 <xs:enumeration value="wsrm:SequenceTerminated"/>
1349 <xs:enumeration value="wsrm:UnknownSequence"/>
1350 <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1351 <xs:enumeration value="wsrm:MessageNumberRollover"/>
1352 <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1353 <xs:enumeration value="wsrm:SequenceClosed"/>
1354 <xs:enumeration value="wsrm:WSRMRequired"/>
1355 </xs:restriction>
1356 </xs:simpleType>
1357 <xs:complexType name="SequenceFaultType">
1358 <xs:sequence>
1359 <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1360 <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
1361 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1362 maxOccurs="unbounded"/>
1363 </xs:sequence>
1364 <xs:anyAttribute namespace="##other" processContents="lax"/>
1365 </xs:complexType>
1366 <xs:complexType name="DetailType">
1367 <xs:sequence>
1368 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1369 maxOccurs="unbounded"/>
1370 </xs:sequence>
1371 <xs:anyAttribute namespace="##other" processContents="lax"/>
1372 </xs:complexType>
1373 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1374 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1375 <xs:element name="CreateSequenceResponse"
1376 type="wsrm:CreateSequenceResponseType"/>
1377 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1378 <xs:element name="CloseSequenceResponse"
1379 type="wsrm:CloseSequenceResponseType"/>
1380 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1381 <xs:element name="TerminateSequenceResponse"
1382 type="wsrm:TerminateSequenceResponseType"/>
1383 <xs:complexType name="CreateSequenceType">
1384 <xs:sequence>
1385 <xs:element ref="wsrm:AcksTo"/>
1386 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1387 <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1388 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1389 maxOccurs="unbounded">
1390 <xs:annotation>
1391 <xs:documentation>
1392 It is the authors intent that this extensibility be used to
1393 transfer a Security Token Reference as defined in WS-Security.
1394 </xs:documentation>
1395 </xs:annotation>
1396 </xs:any>
1397 </xs:sequence>
1398 <xs:anyAttribute namespace="##other" processContents="lax"/>
1399 </xs:complexType>
1400 <xs:complexType name="CreateSequenceResponseType">
1401 <xs:sequence>
1402 <xs:element ref="wsrm:Identifier"/>
1403 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1404 <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>
1405 <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1406 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1407 maxOccurs="unbounded"/>
1408 </xs:sequence>
1409 <xs:anyAttribute namespace="##other" processContents="lax"/>

```

```

1410 </xs:complexType>
1411 <xs:complexType name="CloseSequenceType">
1412 <xs:sequence>
1413 <xs:element ref="wsrm:Identifier"/>
1414 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1415 maxOccurs="unbounded"/>
1416 </xs:sequence>
1417 <xs:anyAttribute namespace="##other" processContents="lax"/>
1418 </xs:complexType>
1419 <xs:complexType name="CloseSequenceResponseType">
1420 <xs:sequence>
1421 <xs:element ref="wsrm:Identifier"/>
1422 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1423 maxOccurs="unbounded"/>
1424 </xs:sequence>
1425 <xs:anyAttribute namespace="##other" processContents="lax"/>
1426 </xs:complexType>
1427 <xs:complexType name="TerminateSequenceType">
1428 <xs:sequence>
1429 <xs:element ref="wsrm:Identifier"/>
1430 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1431 maxOccurs="unbounded"/>
1432 </xs:sequence>
1433 <xs:anyAttribute namespace="##other" processContents="lax"/>
1434 </xs:complexType>
1435 <xs:complexType name="TerminateSequenceResponseType">
1436 <xs:sequence>
1437 <xs:element ref="wsrm:Identifier"/>
1438 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1439 maxOccurs="unbounded"/>
1440 </xs:sequence>
1441 <xs:anyAttribute namespace="##other" processContents="lax"/>
1442 </xs:complexType>

```

```

1443 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
1444 <xs:complexType name="OfferType">
1445 <xs:sequence>
1446 <xs:element ref="wsrm:Identifier"/>
1447 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1448 <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
1449 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1450 maxOccurs="unbounded"/>
1451 </xs:sequence>
1452 <xs:anyAttribute namespace="##other" processContents="lax"/>
1453 </xs:complexType>
1454 <xs:complexType name="AcceptType">
1455 <xs:sequence>
1456 <xs:element ref="wsrm:AcksTo"/>
1457 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1458 maxOccurs="unbounded"/>
1459 </xs:sequence>
1460 <xs:anyAttribute namespace="##other" processContents="lax"/>
1461 </xs:complexType>
1462 <xs:element name="Expires">
1463 <xs:complexType>
1464 <xs:simpleContent>
1465 <xs:extension base="xs:duration">
1466 <xs:anyAttribute namespace="##other" processContents="lax"/>
1467 </xs:extension>
1468 </xs:simpleContent>
1469 </xs:complexType>
1470 </xs:element>
1471 <xs:element name="AcknowledgementInterval">
1472 <xs:complexType>
1473 <xs:sequence/>
1474 <xs:attribute name="Milliseconds" type="xs:unsignedLong"
1475 use="required"/>
1476 <xs:anyAttribute namespace="##other" processContents="lax"/>
1477 </xs:complexType>
1478 </xs:element>
1479 </xs:schema>

```


B. Message Examples

B.1 Create Sequence

Create Sequence

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/2006042"
  xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
  <S:Header>
    <wsa:MessageID>
      http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
    </wsa:MessageID>
    <wsa:To>http://example.com/serviceB/123</wsa:To>
    <wsa:Action>http://docs.oasis-open.org/ws-
rx/wsrn/2006042/CreateSequence</wsa:Action>
    <wsa:ReplyTo>
      <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
    </wsa:ReplyTo>
  </S:Header>
  <S:Body>
    <wsrm:CreateSequence>
      <wsrm:AcksTo>
        <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
      </wsrm:AcksTo>
    </wsrm:CreateSequence>
  </S:Body>
</S:Envelope>
```

Create Sequence Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/2006042"
  xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
  <S:Header>
    <wsa:To>http://Business456.com/serviceA/789</wsa:To>
    <wsa:RelatesTo>
      http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
    </wsa:RelatesTo>
    <wsa:Action>
      http://docs.oasis-open.org/ws-rx/wsrn/2006042/CreateSequenceResponse
    </wsa:Action>
  </S:Header>
  <S:Body>
    <wsrm:CreateSequenceResponse>
      <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
    </wsrm:CreateSequenceResponse>
  </S:Body>
</S:Envelope>
```

B.2 Initial Transmission

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

1531 Message 1

```
1532 <?xml version="1.0" encoding="UTF-8"?>
1533 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1534 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/2006042"
1535 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
1536   <S:Header>
1537     <wsa:MessageID>
1538       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
1539     </wsa:MessageID>
1540     <wsa:To>http://example.com/serviceB/123</wsa:To>
1541     <wsa:From>
1542       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1543     </wsa:From>
1544     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1545     <wsm:Sequence>
1546       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1547       <wsm:MessageNumber>1</wsm:MessageNumber>
1548     </wsm:Sequence>
1549   </S:Header>
1550   <S:Body>
1551     <!-- Some Application Data -->
1552   </S:Body>
1553 </S:Envelope>
```

1554 Message 2

```
1555 <?xml version="1.0" encoding="UTF-8"?>
1556 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1557 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/2006042"
1558 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
1559   <S:Header>
1560     <wsa:MessageID>
1561       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1562     </wsa:MessageID>
1563     <wsa:To>http://example.com/serviceB/123</wsa:To>
1564     <wsa:From>
1565       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1566     </wsa:From>
1567     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1568     <wsm:Sequence>
1569       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1570       <wsm:MessageNumber>2</wsm:MessageNumber>
1571     </wsm:Sequence>
1572   </S:Header>
1573   <S:Body>
1574     <!-- Some Application Data -->
1575   </S:Body>
1576 </S:Envelope>
```

1577 Message 3

```
1578 <?xml version="1.0" encoding="UTF-8"?>
1579 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1580 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/2006042"
1581 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
1582   <S:Header>
1583     <wsa:MessageID>
1584       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1585     </wsa:MessageID>
1586     <wsa:To>http://example.com/serviceB/123</wsa:To>
1587     <wsa:From>
1588       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
```

```

1589     </wsa:From>
1590     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1591     <wsrm:Sequence>
1592         <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1593         <wsrm:MessageNumber>3</wsrm:MessageNumber>
1594     </wsrm:Sequence>
1595     <wsrm:AckRequested>
1596         <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1597     </wsrm:AckRequested>
1598 </S:Header>
1599 <S:Body>
1600     <!-- Some Application Data -->
1601 </S:Body>
1602 </S:Envelope>

```

1603 B.3 First Acknowledgement

1604 Message number 2 has not been received by the RM Destination due to some transmission error so it
1605 responds with an acknowledgement for messages 1 and 3:

```

1606 <?xml version="1.0" encoding="UTF-8"?>
1607 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1608 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/2006042"
1609 xmlns:wsa="http://www.w3.org/2005schemas+xmlsoap.org/ws/2004/08/addressing">
1610   <S:Header>
1611     <wsa:MessageID>
1612       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1613     </wsa:MessageID>
1614     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1615     <wsa:From>
1616       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1617     </wsa:From>
1618     <wsa:Action>
1619       http://docs.oasis-open.org/ws-rx/wsr/2006042/SequenceAcknowledgement
1620     </wsa:Action>
1621     <wsrm:SequenceAcknowledgement>
1622       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1623       <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
1624       <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
1625     </wsrm:SequenceAcknowledgement>
1626   </S:Header>
1627   <S:Body/>
1628 </S:Envelope>

```

1629 B.4 Retransmission

1630 The RM Sourcediscovers that message number 2 was not received so it resends the message and
1631 requests an acknowledgement:

```

1632 <?xml version="1.0" encoding="UTF-8"?>
1633 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1634 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/2006042"
1635 xmlns:wsa="http://www.w3.org/2005schemas+xmlsoap.org/ws/2004/08/addressing">
1636   <S:Header>
1637     <wsa:MessageID>
1638       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1639     </wsa:MessageID>
1640     <wsa:To>http://example.com/serviceB/123</wsa:To>
1641     <wsa:From>
1642       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1643     </wsa:From>

```

```

1644 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1645 <wsrm:Sequence>
1646   <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1647   <wsrm:MessageNumber>2</wsrm:MessageNumber>
1648 </wsrm:Sequence>
1649 <wsrm:AckRequested>
1650   <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1651 </wsrm:AckRequested>
1652 </S:Header>
1653 <S:Body>
1654   <!-- Some Application Data -->
1655 </S:Body>
1656 </S:Envelope>

```

B.5 Termination

The RM Destination now responds with an acknowledgement for the complete [Sequence](#) which can then be terminated:

```

1660 <?xml version="1.0" encoding="UTF-8"?>
1661 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1662   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/2006042"
1663   xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
1664   <S:Header>
1665     <wsa:MessageID>
1666       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1667     </wsa:MessageID>
1668     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1669     <wsa:From>
1670       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1671     </wsa:From>
1672     <wsa:Action>
1673       http://docs.oasis-open.org/ws-rx/wsrn/2006042/SequenceAcknowledgement
1674     </wsa:Action>
1675     <wsrm:SequenceAcknowledgement>
1676       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1677       <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
1678     </wsrm:SequenceAcknowledgement>
1679   </S:Header>
1680   <S:Body/>
1681 </S:Envelope>

```

Terminate Sequence

```

1683 <?xml version="1.0" encoding="UTF-8"?>
1684 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1685   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/2006042"
1686   xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
1687   <S:Header>
1688     <wsa:MessageID>
1689       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1690     </wsa:MessageID>
1691     <wsa:To>http://example.com/serviceB/123</wsa:To>
1692     <wsa:Action>
1693       http://docs.oasis-open.org/ws-rx/wsrn/2006042/TerminateSequence
1694     </wsa:Action>
1695     <wsa:From>
1696       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1697     </wsa:From>
1698   </S:Header>
1699   <S:Body>
1700     <wsrm:TerminateSequence>

```

```

1701     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1702   </wsrm:TerminateSequence>
1703 </S:Body>
1704 </S:Envelope>

```

1705 Terminate Sequence Response

```

1706 <?xml version="1.0" encoding="UTF-8"?>
1707 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1708   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/2006042"
1709   xmlns:wsa="http://www.w3.org/2005schemas-xmisoap.org/ws/2004/08/addressing">
1710   <S:Header>
1711     <wsa:MessageID>
1712       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
1713     </wsa:MessageID>
1714     <wsa:To>http://example.com/serviceA/789</wsa:To>
1715     <wsa:Action>
1716       http://docs.oasis-open.org/ws-rx/wsrn/2006042/TerminateSequenceResponse
1717     </wsa:Action>
1718     <wsa:RelatesTo>
1719       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1720     </wsa:RelatesTo>
1721     <wsa:From>
1722       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1723     </wsa:From>
1724   </S:Header>
1725   <S:Body>
1726     <wsrm:TerminateSequenceResponse>
1727       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1728     </wsrm:TerminateSequenceResponse>
1729   </S:Body>
1730 </S:Envelope>

```

C. WSDL

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

<http://docs.oasis-open.org/ws-rx/wsrn/200604/wsd/wsrn-1.1-wsd-200602.wsd>

The following non-normative copy is provided for reference.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
OASIS takes no position regarding the validity or scope of any intellectual
property or other rights that might be claimed to pertain to the
implementation or use of the technology described in this document or the
extent to which any license under such rights might or might not be available;
neither does it represent that it has made any effort to identify any such
rights. Information on OASIS's procedures with respect to rights in OASIS
specifications can be found at the OASIS website. Copies of claims of rights
made available for publication and any assurances of licenses to be made
available, or the result of an attempt made to obtain a general license or
permission for the use of such proprietary rights by implementors or users of
this specification, can be obtained from the OASIS Executive Director.
OASIS invites any interested party to bring to its attention any copyrights,
patents or patent applications, or other proprietary rights which may cover
technology that may be required to implement this specification. Please
address the information to the OASIS Executive Director.
Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
This document and translations of it may be copied and furnished to others,
and derivative works that comment on or otherwise explain it or assist in its
implementation may be prepared, copied, published and distributed, in whole or
in part, without restriction of any kind, provided that the above copyright
notice and this paragraph are included on all such copies and derivative
works. However, this document itself does not be modified in any way, such as
by removing the copyright notice or references to OASIS, except as needed for
the purpose of developing OASIS specifications, in which case the procedures
for copyrights defined in the OASIS Intellectual Property Rights document must
be followed, or as required to translate it into languages other than English.
The limited permissions granted above are perpetual and will not be revoked by
OASIS or its successors or assigns.
This document and the information contained herein is provided on an "AS
IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
FITNESS FOR A PARTICULAR PURPOSE.
-->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
open.org/ws-rx/wsrn/200604" xmlns:tns="http://docs.oasis-open.org/ws-
rx/wsrn/200604/wsd" targetNamespace="http://docs.oasis-open.org/ws-
rx/wsrn/200604/wsd">

  <wsdl:types>
    <xs:schema>
      <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsrn/200604"
schemaLocation="http://docs.oasis-open.org/ws-rx/wsrn/200604/wsrn-1.1-schema-
200604.xsd"/>
    </xs:schema>
  </wsdl:types>
```

```

1785 <wsdl:message name="CreateSequence">
1786 <wsdl:part name="create" element="rm:CreateSequence"/>
1787 </wsdl:message>
1788 <wsdl:message name="CreateSequenceResponse">
1789 <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
1790 </wsdl:message>
1791 <wsdl:message name="CloseSequence">
1792 <wsdl:part name="close" element="rm:CloseSequence"/>
1793 </wsdl:message>
1794 <wsdl:message name="CloseSequenceResponse">
1795 <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
1796 </wsdl:message>
1797 <wsdl:message name="TerminateSequence">
1798 <wsdl:part name="terminate" element="rm:TerminateSequence"/>
1799 </wsdl:message>
1800 <wsdl:message name="TerminateSequenceResponse">
1801 <wsdl:part name="terminateResponse"
1802 element="rm:TerminateSequenceResponse"/>
1803 </wsdl:message>
1804
1805 <wsdl:portType name="SequenceAbstractPortType">
1806 <wsdl:operation name="CreateSequence">
1807 <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
1808 open.org/ws-rx/wsrn/200604/CreateSequence"/>
1809 <wsdl:output message="tns:CreateSequenceResponse"
1810 wsa:Action="http://docs.oasis-open.org/ws-
1811 rx/wsrn/200604/CreateSequenceResponse"/>
1812 </wsdl:operation>
1813 <wsdl:operation name="CloseSequence">
1814 <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
1815 open.org/ws-rx/wsrn/200604/CloseSequence"/>
1816 <wsdl:output message="tns:CloseSequenceResponse"
1817 wsa:Action="http://docs.oasis-open.org/ws-
1818 rx/wsrn/200604/CloseSequenceResponse"/>
1819 </wsdl:operation>
1820 <wsdl:operation name="TerminateSequence">
1821 <wsdl:input message="tns:TerminateSequence"
1822 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequence"/>
1823 <wsdl:output message="tns:TerminateSequenceResponse"
1824 wsa:Action="http://docs.oasis-open.org/ws-
1825 rx/wsrn/200604/TerminateSequenceResponse"/>
1826 </wsdl:operation>
1827 </wsdl:portType>
</wsdl:definitions>

```

```

1828 <?xml version="1.0" encoding="utf-8"?>
1829 <!--
1830 OASIS takes no position regarding the validity or scope of any intellectual
1831 property or other rights that might be claimed to pertain to the
1832 implementation or use of the technology described in this document or the
1833 extent to which any license under such rights might or might not be available;
1834 neither does it represent that it has made any effort to identify any such
1835 rights. Information on OASIS's procedures with respect to rights in OASIS
1836 specifications can be found at the OASIS website. Copies of claims of rights
1837 made available for publication and any assurances of licenses to be made
1838 available, or the result of an attempt made to obtain a general license or
1839 permission for the use of such proprietary rights by implementors or users of
1840 this specification, can be obtained from the OASIS Executive Director.
1841 OASIS invites any interested party to bring to its attention any copyrights,
1842 patents or patent applications, or other proprietary rights which may cover
1843 technology that may be required to implement this specification. Please

```


address the information to the OASIS Executive Director.
Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
This document and translations of it may be copied and furnished to others,
and derivative works that comment on or otherwise explain it or assist in its
implementation may be prepared, copied, published and distributed, in whole or
in part, without restriction of any kind, provided that the above copyright
notice and this paragraph are included on all such copies and derivative
works. However, this document itself does not be modified in any way, such as
by removing the copyright notice or references to OASIS, except as needed for
the purpose of developing OASIS specifications, in which case the procedures
for copyrights defined in the OASIS Intellectual Property Rights document must
be followed, or as required to translate it into languages other than English.
The limited permissions granted above are perpetual and will not be revoked by
OASIS or its successors or assigns.
This document and the information contained herein is provided on an "AS-
IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
FITNESS FOR A PARTICULAR PURPOSE.
—>
<wsdl:definitions xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:rm="http://docs.oasis-open.org/ws-rx/wsr/200602"
xmlns:tns="http://docs.oasis-open.org/ws-rx/wsr/200602/wsdl"
targetNamespace="http://docs.oasis-open.org/ws-rx/wsr/200602/wsdl">
<wsdl:types>
<xs:schema>
<xs:import namespace="http://docs.oasis-open.org/ws-rx/wsr/200602"
schemaLocation="http://docs.oasis-open.org/ws-rx/wsr/200602/wsr-1.1-schema-
200602.xsd"/>
</xs:schema>
</wsdl:types>
<wsdl:message name="CreateSequence">
<wsdl:part name="create" element="rm:CreateSequence"/>
</wsdl:message>
<wsdl:message name="CreateSequenceResponse">
<wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
</wsdl:message>
<wsdl:message name="CloseSequence">
<wsdl:part name="close" element="rm:CloseSequence"/>
</wsdl:message>
<wsdl:message name="CloseSequenceResponse">
<wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
</wsdl:message>
<wsdl:message name="TerminateSequence">
<wsdl:part name="terminate" element="rm:TerminateSequence"/>
</wsdl:message>
<wsdl:message name="TerminateSequenceResponse">
<wsdl:part name="terminateResponse"
element="rm:TerminateSequenceResponse"/>
</wsdl:message>
<wsdl:portType name="SequenceAbstractPortType">
<wsdl:operation name="CreateSequence">
<wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
open.org/ws-rx/wsr/200602/CreateSequence"/>
<wsdl:output message="tns:CreateSequenceResponse"
wsa:Action="http://docs.oasis-open.org/ws-
rx/wsr/200602/CreateSequenceResponse"/>
</wsdl:operation>
<wsdl:operation name="CloseSequence">
<wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-


```
1906 open.org/ws-rx/wsrn/200602/CloseSequence"/>
1907 <wsdl:output-message="tns:CloseSequenceResponse"
1908 wsa:Action="http://docs.oasis-open.org/ws-
1909 rx/wsrn/200602/CloseSequenceResponse"/>
1910 </wsdl:operation>
1911 <wsdl:operation-name="TerminateSequence">
1912 <wsdl:input-message="tns:TerminateSequence"
1913 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200602/TerminateSequence"/>
1914 <wsdl:output-message="tns:TerminateSequenceResponse"
1915 wsa:Action="http://docs.oasis-open.org/ws-
1916 rx/wsrn/200602/TerminateSequenceResponse"/>
1917 </wsdl:operation>
1918 </wsdl:portType>
1919 </wsdl:definitions>
```

D. State Tables

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

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

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

Table 2 RM Source State Transition Table

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

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

1924 In Table 2 above, the rows consists of events that occur at the RM Source throughout the lifetime of an
 1925 RM Sequence and the columns consists of various RM Source states. Each cell in the table above lists
 1926 the action that the RM Source takes on occurrence of a particular event and the next state that it
 1927 transitions.

1928 Table 3 RM Destination State Transition Table

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

Events	States						
	None	Connecting	Connected	Rollover	Rollover Closed	Closed	Terminated
			<i>ceResponse with SequenceAck (Final)</i> Close	<i>ceResponse with SequenceAck Final</i> Rollver Closed	<i>Response with SeqAck+Final</i> Rollover Closed	<i>Response with SeqAck+Final</i> Closed	<i>Sequence Fault</i> 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
EIapse Expires duration	N/A	N/A	Terminated	Terminated	Terminated	Terminated	N/A

1929 In Table 3 above, the rows consists of events that occur at the RM Destination throughout the lifetime of
1930 an RM Sequence and the columns consists of various RM Destination states. Each cell in the table above
1931 lists the action that the RM Destination takes on occurrence of a particular event and the next state that it
1932 transitions.

E. Acknowledgments

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

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

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

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

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

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

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

Rev	Date	By Whom	What
			OpenDocument Text format. Changed line numbers to be a single style.
wd-08	2005-12-28	Anish Karmarkar	Included a section link to c:\temp\wsrm-1.1-schema-200510.xsd and to c:\temp\wsrm-1.1-wsdl-200510.wsdl
wd-08	2006-01-04	Gilbert Pilz	Fixed formatting for included sections.
wd-08	2006-01-05	Gilbert Pilz	Created links for unused references. Fixed exemplars for CloseSequence and CloseSequenceResponse.
wd-09	2006-01-11	Doug Davis	Minor tweaks to text/typos.
wd-10	2006-01-23	Doug Davis	Accept all changes from wd-09 Make some minor editorial tweaks from Marc's comments.
wd-10	2006-02-14	Doug Davis	Issue 082 resolution
wd-10	2006-02-14	Doug Davis	Issue 083 resolution
wd-10	2006-02-14	Doug Davis	Issue 085 resolution
wd-10	2006-02-14	Doug Davis	Issues 086, 087 resolutions Defined MessageNumberType
wd-10	2006-02-15	Doug Davis	Issue 078 resolution
wd-10	2006-02-15	Doug Davis	Issue 094 resolution
wd-10	2006-02-15	Doug Davis	Issue 095 resolution
wd-10	2006-02-15	Gilbert Pilz	Issue 088 – added namespace URI link to namespace URI; added text explaining that this URI could be dereferenced to produce the RDDDL doc; added non-normative reference to RDDDL 2.0
wd-10	2006-02-17	Anish Karmarkar	Namespace changed to 200602 for both WSDL and XSD docs.
wd-10	2006-02-17	Anish Karmarkar	Issue i087 as it applies to WSRM spec.
wd-10	2006-02-17	Anish Karmarkar	Added titles and minor text for state table (issue i058).
wd-11	2006-02-22	Doug Davis	Accept all changes for new WD Minor typos fixed
wd-11	2006-02-23	Doug Davis	s/'close'/close/g – per Marc Goodner Added first ref to [URI] – per Marc G again
wd-11	2006-02-27	Doug Davis	Issue i061 applied
wd-11	2006-02-28	Doug Davis	Fixed typo around the use of "above" and "below"
wd-11	2006-03-01	Doug Davis	Minor typos found by Marc Goodner
wd-11	2006-03-02	Doug Davis	Minor typos found by Matt Lovett
wd-11	2006-03-08	Doug Davis	Issue 091 applied
wd-11	2006-03-08	Doug Davis	Issue 092 applied
wd-11	2006-03-08	Doug Davis	Issue 100 applied

Rev	Date	By Whom	What
wd-12	2006-03-20	Doug Davis	Added space in "SOAP1.x" – PaulCotton
wd-12	2006-04-11	Doug Davis	Issue 007 applied
wd-12	2006-04-11	Doug Davis	Issue 090 applied
wd-12	2006-04-11	Doug Davis	Issue 098 applied
wd-12	2006-04-11	Doug Davis	Issue 099 applied
wd-12	2006-04-11	Doug Davis	Issue 101 applied
wd-12	2006-04-11	Doug Davis	Issue 103 applied
wd-12	2006-04-11	Doug Davis	Issue 104 applied
wd-12	2006-04-11	Doug Davis	Issue 105 applied
wd-12	2006-04-11	Doug Davis	Issue 107 applied
wd-12	2006-04-11	Doug Davis	Issue 109 applied
wd-12	2006-04-11	Doug Davis	Issue 110 applied
wd-12	2006-04-12	Doug Davis	Used "generated" instead of "issue" or "send" when talking about faults.
wd-12	2006-04-24	Gilbert Pilz	Update references to WS-Addressing to the Proposed Recommendations; update WS-RM namespace to "200604".
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 'optional'/'required' to 'OPTIONAL'/'REQUIRED') added
wd-05	2005-09-30	Anish Karmarkar	i017 (Change NS to http://docs.oasis-open.org/wsrn/200510/-)
wd-05	2005-09-30	Anish Karmarkar	i045 (Include SecureConversation as a reference and move it to non-normative

Rev	Date	By Whom	What
			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
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

Rev	Date	By Whom	What
wd-08	2005-12-15	Steve Winkler	Doug added Steve's editorial nits
wd-08	2005-12-21	Doug Davis	i050
wd-08	2005-12-21	Doug Davis	i081
wd-08	2005-12-21	Doug Davis	i080 — but i050 negates the need for any changes
wd-08	2005-12-21	Doug Davis	i079
wd-08	2005-12-21	Doug Davis	i076 — didn't add text about "replies" since the RMD to RMS sequence could be used for any message not just replies
wd-08	2005-12-21	Umit Yalcinalp	Action Su03: removed wsse from Table 1
wd-08	2005-12-21	Umit Yalcinalp	i057 per Sunnyvale F2F 2005, Cleaned up some formatting errors in contributors
wd-08	2005-12-27	Doug Davis	i060
wd-08	2005-12-27	Gilbert Pilz	Moved schema and WSDL files to their own artifacts. Converted source document to 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/types.
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 RDDL doc; added non-normative reference to RDDL 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

Rev	Date	By Whom	What
			(issue i058):
wd-11	2006-02-22	Doug Davis	Accept all changes for new WD Minor typos fixed
wd-11	2006-02-23	Doug Davis	s/'close'/close/g — per Marc Goodner Added first ref to [URI] — per Marc G again
wd-11	2006-02-27	Doug Davis	Issue i061 applied
wd-11	2006-02-28	Doug Davis	Fixed typo around the use of "above" and "below"
wd-11	2006-03-01	Doug Davis	Minor typos found by Marc Goodner
wd-11	2006-03-02	Doug Davis	Minor typos found by Matt Lovett
wd-11	2006-03-08	Doug Davis	Issue 091 applied
wd-11	2006-03-08	Doug Davis	Issue 092 applied
wd-11	2006-03-08	Doug Davis	Issue 100 applied

G. Notices

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

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

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

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

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

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