



1 Web Services Reliable Messaging 2 (WS-ReliableMessaging)

3 Working Draft 13, May 30, 2006

4 Document identifier:

5 wsrn-1.1-spec-wd-13

6 Location:

7 Editors:

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

9 Anish Karmarkar, Oracle <Anish.Karmarkar@oracle.com>

10 Gilbert Pilz, BEA <gpilz@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

33	1 Introduction.....	4
34	1.1 Goals and Requirements.....	4
35	1.1.1 Requirements.....	4
36	1.2 Notational Conventions.....	4
37	1.3 Namespace.....	4
38	1.4 Compliance.....	5
39	2 Reliable Messaging Model.....	6
40	2.1 Glossary.....	6
41	2.2 Protocol Preconditions.....	7
42	2.3 Protocol Invariants.....	7
43	2.4 Example Message Exchange.....	7
44	3 RM Protocol Elements.....	10
45	3.1 Sequence Creation.....	10
46	3.2 Closing A Sequence.....	13
47	3.3 Sequence Termination.....	15
48	3.4 Sequences.....	16
49	3.5 Request Acknowledgement.....	17
50	3.6 Sequence Acknowledgement.....	18
51	4 Faults.....	21
52	4.1 SequenceFault Element.....	22
53	4.2 Sequence Terminated.....	23
54	4.3 Unknown Sequence.....	23
55	4.4 Invalid Acknowledgement.....	24
56	4.5 Message Number Rollover.....	24
57	4.6 Create Sequence Refused.....	24
58	4.7 Sequence Closed.....	25
59	4.8 WSRM Required.....	25
60	5 Security Considerations.....	26
61	6 References.....	28
62	6.1 Normative.....	28
63	6.2 Non-Normative.....	28
64	A. Schema.....	30
65	B. Message Examples.....	35
66	B.1 Create Sequence.....	35
67	B.2 Initial Transmission.....	35
68	B.3 First Acknowledgement.....	37
69	B.4 Retransmission.....	37

70	B.5 Termination.....	38
71	C. WSDL.....	40
72	D. State Tables.....	42
73	E. Acknowledgments.....	46
74	F. Revision History.....	47
75	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/200604>

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

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

114 The following namespaces are used in this document:

115 *Table 1*

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

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

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

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

119 **1.4 Compliance**

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

124 Normative text within this specification takes precedence over normative outlines, which in turn take
125 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, 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.

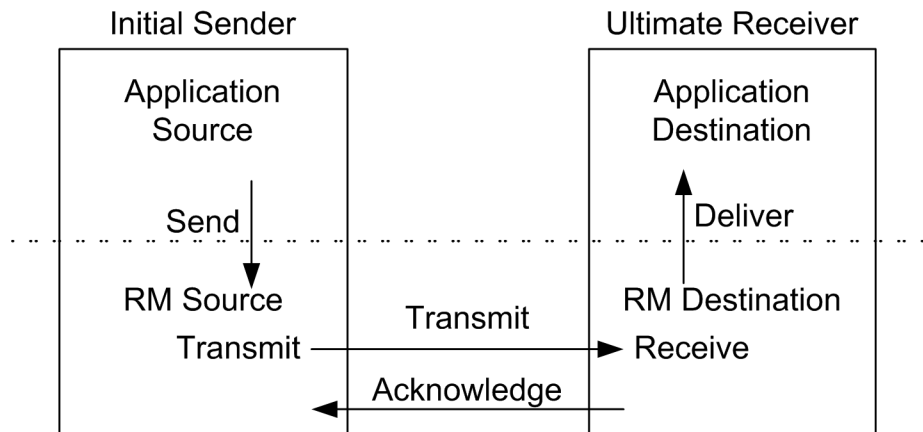


Figure 1: Reliable Messaging Model

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.

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

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

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

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

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

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

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

126 2.2 Protocol Preconditions

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

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

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

128 2.3 Protocol Invariants

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

- 130 • The RM Source **MUST** assign each message within a Sequence a message number (defined
131 below) beginning at 1 and increasing by exactly 1 for each subsequent message. These numbers
132 **MUST** be assigned in the same order in which messages are sent by the Application Source.
- 133 • Within every acknowledgement it issues, the RM Destination **MUST** include one or more
134 acknowledgement ranges that contain the message number of every message successfully
135 received by the RM Destination. The RM Destination **MUST** exclude the message numbers of any
136 messages it has not received.

137 2.4 Example Message Exchange

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

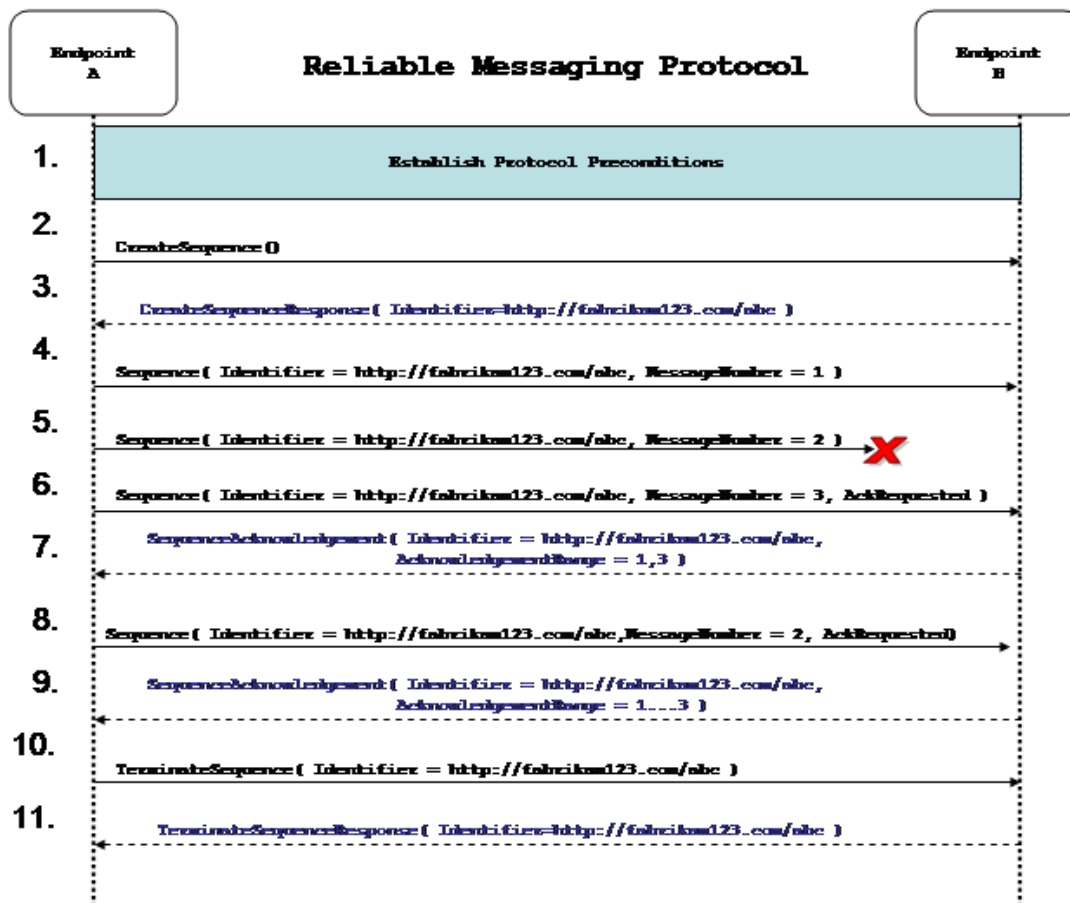


Figure 2: The WS-ReliableMessaging Protocol

- 139 1. The protocol preconditions are established. These include policy exchange, endpoint resolution,
140 and establishing trust.
- 141 2. The RM Source requests creation of a new Sequence.
- 142 3. The RM Destination creates a new Sequence and returns its globally unique identifier.
- 143 4. The RM Source begins transmitting messages in the Sequence beginning with MessageNumber 1.
144 In the figure above, the RM Source sends 3 messages in the Sequence.
- 145 5. The 2nd message in the Sequence is lost in transit.
- 146 6. The 3rd message is the last in this Sequence and the RM Source includes a
147 `<wsrm:AckRequested>` header to ensure that it gets a timely
148 `<wsrm:SequenceAcknowledgement>` for the Sequence.
- 149 7. The RM Destination acknowledges receipt of message numbers 1 and 3 as a result of receiving the
150 RM Source's `<wsrm:AckRequested>` header.
- 151 8. The RM Source retransmits the unacknowledged message with MessageNumber 2. This is a new
152 message from the perspective of the underlying transport, but it has the same Sequence Identifier
153 and MessageNumber so the RM Destination can recognize it as a duplicate of the earlier message,
154 in case the original and retransmitted messages are both received. The RM Source includes an
155 `<wsrm:AckRequested>` header in the retransmitted message so the RM Destination will expedite
156 an acknowledgement.

157 9. The RM Destination receives the second transmission of the message with MessageNumber 2 and
158 acknowledges receipt of message numbers 1, 2, and 3.

159 10. The RM Source receives this acknowledgement and sends a TerminateSequence message to the
160 RM Destination indicating that the Sequence is completed and reclaims any resources associated
161 with the Sequence.

162 11. The RM Destination receives the TerminateSequence message indicating that the RM Source will
163 not be sending any more messages. The RM Destination sends a TerminateSequenceResponse
164 message to the RM Source and reclaims any resources associated with the Sequence.

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

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

3 RM Protocol Elements

The following protocol elements define extensibility points at various places. Implementations MAY add child elements and/or attributes 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.

If action IRIs are used by either the RM Source or the RM Destination, and one is not already defined as per the rules of the WS-Addressing specification [WS-Addressing], then the RM Source or the RM Destination MUST use an action IRI that consists of the WS-RM namespace URI concatenated with a '/', followed by the message element name. For example:

```
http://docs.oasis-open.org/ws-rx/wsrn/200604/SequenceAcknowledgement
```

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 RM Source MAY include an offer to create an inbound Sequence within the `<wsrm:CreateSequence>` message. This offer is either accepted or rejected by the RM Destination in the `<wsrm:CreateSequenceResponse>` message.

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. The RM Source MUST NOT send this element as a header block. The RM Destination MUST respond either with a `<wsrm:CreateSequenceResponse>` response message or a `CreateSequenceRefused` fault.

`/wsrm:CreateSequence/wsrn:AcksTo`

The RM Source MUST include this element in any `CreateSequence` message it sends. This element is of type `wsa:EndpointReferenceType` (as specified by WS-Addressing [WS-Addressing]). It specifies the endpoint reference to which messages containing `<wsrm:SequenceAcknowledgement>` header blocks and faults related to the created Sequence are to be sent, unless otherwise noted in this specification (for example, see Section 3.2).

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

223 /wsrm:CreateSequence/wsrm:Expires

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

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

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

231 /wsrm:CreateSequence/wsrm:Offer

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

234 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier

235 The RM Source MUST set the value of this element to an absolute URI (conformant with RFC3986 [\[URI\]](#))
236 that will uniquely identify the offered Sequence.

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

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

240 /wsrm:CreateSequence/wsrm:Offer/wsrm:Endpoint

241 An RM Source MUST include this element, of type `wsa:EndpointReferenceType` (as specified by
242 WS-Addressing [\[WSAddressing\]](#)) This element specifies the endpoint reference to which WS-RM protocol
243 messages related to the offered Sequence are to be sent.

244 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires

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

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

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

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

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

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

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

257 /wsrm:CreateSequence/{any}

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

260 /wsrm:CreateSequence/@{any}

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

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

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

```
268 <wsrm:CreateSequenceResponse ...>
269   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
270   <wsrm:Expires> xs:duration </wsrm:Expires> ?
271   <wsrm:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?
272   <wsrm:IncompleteSequenceBehavior> wsrm:IncompleteSequenceBehaviorType
273 </wsrm:IncompleteSequenceBehavior> ?
274   <wsrm:Accept ...>
275     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
276     ...
277   </wsrm:Accept> ?
278   ...
279 </wsrm:CreateSequenceResponse>
```

280 /wsrm:CreateSequenceResponse

281 This element is sent in the body of the response message in response to a <wsrm:CreateSequence>
282 request message. It indicates that the RM Destination has created a new Sequence at the request of the
283 RM Source. The RM Destination MUST NOT send this element as a header block.

284 /wsrm:CreateSequenceResponse/wsrm:Identifier

285 The RM Destination MUST include this element within any CreateSequenceResponse message it sends.
286 The RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986
287 [URI]) of the Sequence that has been created by the RM Destination.

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

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

291 /wsrm:CreateSequenceResponse/wsrm:Expires

292 This element, if present, of type xs:duration accepts or refines the RM Source's requested duration for
293 the Sequence. A value of 'PT0S' indicates that the Sequence will never expire. Absence of the element
294 indicates an implied value of 'PT0S'. The RM Destination MUST set the value of this element to be equal
295 to or less than the value requested by the RM Source in the corresponding <wsrm:CreateSequence>
296 message.

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

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

300 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval

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

303 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@Milliseconds`
304 The acknowledgement interval, specified in milliseconds.

305 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@{any}`
306 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
307 element.

308 `/wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior`
309 This optional element, if present, specifies the behavior that the RM Destination will exhibit upon the
310 closure of an incomplete sequence.

311 A value of "DiscardEntireSequence" indicates that the entire sequence will be discarded by the RM
312 Destination if the sequence is closed when there are one or more gaps in the
313 `SequenceAcknowledgement/Final`.

314 A value of "DiscardFollowingFirstGap" indicates that messages in the sequence beyond the first gap will
315 be discarded by the RM Destination when there are one or more gaps in the
316 `SequenceAcknowledgement/Final`.

317 The default value of "NoDiscard" indicates that no acknowledged messages in the sequence will be
318 discarded by the RM Destination.

319 `/wsrm:CreateSequenceResponse/wsrm:Accept`
320 This element, if present, enables an RM Destination to accept the offer of a corresponding Sequence for
321 the reliable exchange of messages transmitted from RM Destination to RM Source.

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

325 `/wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo`
326 The RM Destination MUST include this element, of type `wsa:EndpointReferenceType` (as specified
327 by WS-Addressing [WS-Addressing]). The RM Source SHOULD send messages with
328 `<wsrm:SequenceAcknowledgement>` header blocks related to the accepted Sequence to the
329 referenced endpoint.

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

333 `/wsrm:CreateSequenceResponse/wsrm:Accept/@{any}`
334 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
335 to be passed.

336 `/wsrm:CreateSequenceResponse/{any}`
337 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
338 to be passed.

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

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

342 **3.2 Closing A Sequence**

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

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

357 While the RM Destination MUST NOT receive any new messages for the specified Sequence it MUST still
358 process RM protocol messages. For example, it MUST respond to `AckRequested`, `TerminateSequence`
359 as well as `CloseSequence` messages. Note, subsequent `CloseSequence` messages have no effect on the
360 state of the Sequence.

361 In the case where the RM Destination wishes to discontinue use of a Sequence it is RECOMMENDED
362 that it close the Sequence. Please see `<wsrm:Final>` and the `SequenceClosed` fault. Whenever
363 possible the `SequenceClosed` Fault SHOULD be used in place of the `SequenceTerminated` Fault,
364 whenever possible, to allow the RM Source to still receive Acknowledgements.

365 The following exemplar defines the `CloseSequence` syntax:

```
366 <wsrm:CloseSequence ...>  
367   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
368   ...  
369 </wsrm:CloseSequence>
```

370 `/wsrm:CloseSequence`

371 This element is sent by an RM Source to indicate that the RM Destination MUST NOT receive any new
372 messages for this Sequence. A `SequenceClosed` fault MUST be generated by the RM Destination when it
373 receives a message for a Sequence that is closed.

374 `/wsrm:CloseSequence/wsrm:Identifier`

375 The RM Source MUST include this element in any `CloseSequence` messages it sends. The RM Source
376 MUST set the value of this element to the absolute URI (conformant with RFC3986 [URI]) of the
377 Sequence that is being closed.

378 `/wsrm:CloseSequence/wsrm:Identifier/@{any}`

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

381 `/wsrm:CloseSequence/{any}`

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

384 /wsrm:CloseSequence@{any}

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

387 A <wsrm:CloseSequenceResponse> is sent in the body of a response message by an RM Destination
388 in response to receipt of a <wsrm:CloseSequence> request message. It indicates that the RM
389 Destination has closed the Sequence.

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

```
391 <wsrm:CloseSequenceResponse ...>  
392   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
393   ...  
394 </wsrm:CloseSequenceResponse>
```

395 /wsrm:CloseSequenceResponse

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

399 /wsrm:CloseSequenceResponse/wsrm:Identifier

400 The RM Destination MUST include this element in any CloseSequenceResponse message it sends. The
401 RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986 [URI])
402 of the Sequence that is being closed.

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

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

406 /wsrm:CloseSequenceResponse/{any}

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

409 /wsrm:CloseSequenceResponse@{any}

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

412 3.3 Sequence Termination

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

420 The following exemplar defines the TerminateSequence syntax:

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


423
424

```
...  
</wsrm:TerminateSequence>
```

425 /wsrm:TerminateSequence

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

431 /wsrm:TerminateSequence/wsrm:Identifier

432 The RM Source MUST include this element in any TerminateSequence message it sends. The RM
433 Source MUST set the value of this element to the absolute URI (conformant with RFC3986 [URI]) of the
434 Sequence that is being terminated.

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

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

438 /wsrm:TerminateSequence/{any}

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

441 /wsrm:TerminateSequence/@{any}

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

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

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

```
448 <wsrm:TerminateSequenceResponse ...>  
449   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
450   ...  
451 </wsrm:TerminateSequenceResponse>
```

452 /wsrm:TerminateSequenceResponse

453 This element is sent in the body of a response message by an RM Destination in response to receipt of a
454 <wsrm:TerminateSequence> request message. It indicates that the RM Destination has terminated
455 the sequence. The RM Destination MUST NOT send this element as a header block.

456 /wsrm:TerminateSequenceResponse/wsrm:Identifier

457 The RMD Destination MUST include this element in any TerminateSequenceResponse message it sends.
458 The RMD Destination MUST set the value of this element to the absolute URI (conformant with RFC3986
459 [URI]) of the Sequence that is being terminated.

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

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

463 /wsrm:TerminateSequenceResponse/{any}

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

466 /wsrm:TerminateSequenceResponse/{any}

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

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

471 3.4 Sequences

472 The RM protocol uses a <wsrm:Sequence> header block to track and manage the reliable delivery of
473 messages. The RM Source MUST include a <wsrm:Sequence> header block in all messages for
474 which reliable delivery is required. The RM Source MUST identify Sequences with unique
475 <wsrm:Identifier> elements and the RM Source MUST assign each message within a Sequence a
476 <wsrm:MessageNumber> element that increments by 1 from an initial value of 1. These values are
477 contained within a <wsrm:Sequence> header block accompanying each message being delivered in the
478 context of a Sequence.

479 The RM Source MUST NOT include more than one <wsrm:Sequence> header block in any message.

480 A following exemplar defines its syntax:

```
481 <wsrm:Sequence ...>  
482   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
483   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>  
484   ...  
485 </wsrm:Sequence>
```

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

487 /wsrm:Sequence

488 This protocol element associates the message in which it is contained with a previously established RM
489 Sequence. It contains the Sequence's unique identifier and the containing message's ordinal position
490 within that Sequence. The RM Destination MUST understand the <wsrm:Sequence> header block. The
491 RM Source MUST assign a `mustUnderstand` attribute with a value 1/true (from the namespace
492 corresponding to the version of SOAP to which the <wsrm:Sequence> SOAP header block is bound) to
493 the <wsrm:Sequence> header block element.

494 /wsrm:Sequence/wsrm:Identifier

495 An RM Source that includes a <wsrm:Sequence> header block in a SOAP envelope MUST include
496 this element in that header block. The RM Source MUST set the value of this element to the absolute URI
497 (conformant with RFC3986 [\[URI\]](#)) that uniquely identifies the Sequence.

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

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

501 /wsrm:Sequence/wsrm:MessageNumber

502 The RM Source MUST include this element within any Sequence headers it creates. This element is of
503 type `wsrm:MessageNumberType`. It represents the ordinal position of the message within a Sequence.
504 Sequence message numbers start at 1 and monotonically increase throughout the Sequence. If the

505 message number exceeds the internal limitations of an RM Source or RM Destination or reaches the
506 maximum value of 9,223,372,036,854,775,807 the RM Source or Destination MUST generate a
507 MessageNumberRollover fault.

508 /wsrm:Sequence/{any}

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

511 /wsrm:Sequence/@{any}

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

514 The following example illustrates a Sequence header block.

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

519 3.5 Request Acknowledgement

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

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

531 The following exemplar defines its syntax:

```
532 <wsrm:AckRequested ...>  
533   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
534   ...  
535 </wsrm:AckRequested>
```

536 /wsrm:AckRequested

537 This element requests an acknowledgement for the identified Sequence.

538 /wsrm:AckRequested/wsrm:Identifier

539 An RM Source that includes a <wsrm:AckRequested> header block in a SOAP envelope MUST include
540 this element in that header block. The RM Source MUST set the value of this element to the absolute URI,
541 (conformant with RFC3986 [URI]), that uniquely identifies the Sequence to which the request applies.

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

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

545 /wsrm:AckRequested/{any}

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

548 /wsrm:AckRequested/@{any}

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

551 3.6 Sequence Acknowledgement

552 The RM Destination informs the RM Source of successful message receipt using a
553 <wsrm:SequenceAcknowledgement> header block. The RM Destination MAY transmit the
554 <wsrm:SequenceAcknowledgement> header block independently or it MAY include the
555 <wsrm:SequenceAcknowledgement> header block on any message targeted to the AcksTo EPR. The
556 RM Destination MAY send a <wsrm:SequenceAcknowledgement> header block at any point during
557 which the Sequence is valid. Acknowledgements can be explicitly requested using the
558 <wsrm:AckRequested> directive (see Section [Request Acknowledgement](#)). If a non-mustUnderstand
559 fault occurs when processing an RM Header that was piggy-backed on another message, a fault MUST
560 be generated, but the processing of the original message MUST NOT be affected.

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

564 During creation of a Sequence the RM Source MAY specify the WS-Addressing anonymous IRI as the
565 address of the <wsrm:AcksTo> EPR for that Sequence. When the RM Source specifies the WS-
566 Addressing anonymous IRI as the address of the <wsrm:AcksTo> EPR, the RM Destination MUST
567 transmit any <wsrm:SequenceAcknowledgement> headers for the created Sequence in a SOAP
568 envelope to be transmitted on the protocol binding-specific channel. Such a channel is provided by the
569 context of a received message containing a SOAP envelope that contains a <wsrm:Sequence> header
570 block and/or a <wsrm:AckRequested> header block for that same Sequence identifier.

571 The following exemplar defines its syntax:

```
572 <wsrm:SequenceAcknowledgement ...>  
573   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
574   [ [ [ <wsrm:AcknowledgementRange ...  
575         Upper="wsrm:MessageNumberType"  
576         Lower="wsrm:MessageNumberType"/> +  
  
577         | <wsrm:None/> ]  
578         <wsrm:Final/> ? ]  
579         | <wsrm:Nack> wsrm:MessageNumberType </wsrm:Nack> + ]  
580  
581   ...  
582 </wsrm:SequenceAcknowledgement>
```

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

584 /wsrm:SequenceAcknowledgement

585 This element contains the Sequence acknowledgement information.

586 /wsrm:SequenceAcknowledgement/wsrm:Identifier

587 An RM Destination that includes a <wsrm:SequenceAcknowledgement> header block in a SOAP
588 envelope MUST include this element in that header block. The RM Destination MUST set the value of this

589 element to the absolute URI (conformant with RFC3986 [URI]) that uniquely identifies the Sequence. The
590 RM Destination MUST NOT include multiple <wsrm:SequenceAcknowledgement> header blocks that
591 share the same value for <wsrm:Identifier> within the same SOAP envelope.

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

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

595 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange

596 The RM Destination MAY include one or more instances of this element within a
597 <wsrm:SequenceAcknowledgement> header block. It contains a range of Sequence
598 MessageNumbers successfully received by the RM Destination. The ranges SHOULD NOT overlap. The
599 RM Destination MUST NOT include this element if a sibling <wsrm:Nack> or <wsrm:None> element is
600 also present as a child of <wsrm:SequenceAcknowledgement>.

601 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Upper

602 The RM Destination MUST set the value of this attribute equal to the message number of the highest
603 contiguous message in a Sequence range received by the RM Destination.

604 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Lower

605 The RM Destination MUST set the value of this attribute equal to the message number of the lowest
606 contiguous message in a Sequence range received by the RM Destination.

607 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@{any}

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

610 /wsrm:SequenceAcknowledgement/wsrm:Final

611 The RM Destination MAY include this element within a <wsrm:SequenceAcknowledgement> header
612 block. This element indicates that the RM Destination is not receiving new messages for the specified
613 Sequence. The RM Source can be assured that the ranges of messages acknowledged by this
614 SequenceAcknowledgement header block will not change in the future. The RM Destination MUST
615 include this element when the Sequence is closed. Note: the RM Destination MUST NOT include this
616 element when sending a Nack; it can only be used when sending <wsrm:AcknowledgementRange>s or
617 <wsrm:None>.

618 /wsrm:SequenceAcknowledgement/wsrm:Nack

619 The RM Destination MAY include this element within a <wsrm:SequenceAcknowledgement> header
620 block. If used, the RM Destination MUST set the value of this element to a wsrm:MessageNumberType
621 representing the <wsrm:MessageNumber> of an unreceived message in a Sequence. The RM
622 Destination MUST NOT include a <wsrm:Nack> element if a sibling
623 <wsrm:AcknowledgementRange> or <wsrm:None> element is also present as a child of
624 <wsrm:SequenceAcknowledgement>. Upon the receipt of a Nack, an RM Source SHOULD retransmit
625 the message identified by the Nack. The RM Destination MUST NOT issue a
626 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that it has previously
627 acknowledged within a <wsrm:AcknowledgementRange>. The RM Source SHOULD ignore a
628 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that has previously
629 been acknowledged within a <wsrm:AcknowledgementRange>.

630 /wsrm:SequenceAcknowledgement/wsrm:None

631 The RM Destination MUST include this element within a `<wsrm:SequenceAcknowledgement>` header
632 block if the RM Destination has not received any messages for the specified Sequence. The RM
633 Destination MUST NOT include this element if a sibling `<wsrm:AcknowledgementRange>` or
634 `<wsrm:Nack>` element is also present as a child of the `<wsrm:SequenceAcknowledgement>`.

635 `/wsrm:SequenceAcknowledgement/{any}`

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

638 `/wsrm:SequenceAcknowledgement/@{any}`

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

641 The following examples illustrate `<wsrm:SequenceAcknowledgement>` elements:

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

```
643 <wsrm:SequenceAcknowledgement>  
644   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
645   <wsrm:AcknowledgementRange Upper="10" Lower="1"/>  
646 </wsrm:SequenceAcknowledgement>
```

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

```
649 <wsrm:SequenceAcknowledgement>  
650   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
651   <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
652   <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
653   <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
654 </wsrm:SequenceAcknowledgement>
```

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

```
656 <wsrm:SequenceAcknowledgement>  
657   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
658   <wsrm:Nack>3</wsrm:Nack>  
659 </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".

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 Sequence faults. Entities that generate Sequence faults SHOULD send those faults to the same [destination] as `<wsrm:SequenceAcknowledgement>` messages. These faults are correlated using the Sequence identifier carried in the detail.

Entities that generate 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(s). If absent, no detail element is defined for the fault. If more than one detail element is defined for a fault, implementations MUST include the elements in the order that they are specified.

Entities that generate WS-ReliableMessaging faults MUST set the [Code] property to 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. -->
  </S:Header>
  <S:Body>
    <S:Fault>
```

```

700     <S:Code>
701       <S:Value> [Code] </S:Value>
702       <S:Subcode>
703         <S:Value> [Subcode] </S:Value>
704       </S:Subcode>
705     </S:Code>
706     <S:Reason>
707       <S:Text xml:lang="en"> [Reason] </S:Text>
708     </S:Reason>
709     <S:Detail>
710       [Detail]
711     ...
712   </S:Detail>
713 </S:Fault>
714 </S:Body>
715 </S:Envelope>

```

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

```

718 <S11:Envelope>
719   <S11:Header>
720     <wsrm:SequenceFault>
721       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
722       <wsrm:Detail> [Detail] </wsrm:Detail>
723     ...
724   </wsrm:SequenceFault>
725   <!-- Headers elided for clarity. -->
726 </S11:Header>
727 <S11:Body>
728   <S11:Fault>
729     <faultcode> [Code] </faultcode>
730     <faultstring> [Reason] </faultstring>
731   </S11:Fault>
732 </S11:Body>
733 </S11:Envelope>

```

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

```

736 <S11:Envelope>
737   <S11:Body>
738     <S11:Fault>
739       <faultcode> [Subcode] </faultcode>
740       <faultstring> [Reason] </faultstring>
741     </S11:Fault>
742   </S11:Body>
743 </S11:Envelope>

```

744 4.1 SequenceFault Element

745 The purpose of the <wsrm:SequenceFault> element is to carry the specific details of a fault generated
746 during the reliable messaging specific processing of a message belonging to a Sequence. WS-
747 ReliableMessaging nodes MUST use the <wsrm:SequenceFault> container only in conjunction with
748 the SOAP 1.1 fault mechanism. WS-ReliableMessaging nodes MUST NOT use the
749 <wsrm:SequenceFault> container in conjunction with the SOAP 1.2 binding.

750 The following exemplar defines its syntax:

```

751 <wsrm:SequenceFault ...>

```



```

752     <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
753     <wsrm:Detail> ... </wsrm:Detail> ?
754     ...
755 </wsrm:SequenceFault>

```

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

757 `/wsrm:SequenceFault`

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

759 `/wsrm:SequenceFault/wsrm:FaultCode`

760 WS-ReliableMessaging nodes that generate a `<wsrm:SequenceFault>` MUST set the value of this element to a qualified name from the set of fault [Subcodes] defined below.

762 `/wsrm:SequenceFault/wsrm:Detail`

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

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

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

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

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

769 `/wsrm:SequenceFault/{any}`

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

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

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

775 4.2 Sequence Terminated

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

780 Receipt of `SequenceTerminated` by either the RMD or the RMS shall terminate the Sequence if it is not otherwise terminated.

782 Properties:

783 [Code] Sender or Receiver

784 [Subcode] `wsrm:SequenceTerminated`

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

786 [Detail]

```

787 <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 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 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 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>  
<wsrn:MaxMessageNumber> wsrn:MessageNumberType </wsrn:MaxMessageNumber>
```

4.6 Create Sequence Refused

This fault is generated in response to a create Sequence request that cannot be satisfied.

Properties:

[Code] Sender

[Subcode] wsrn:CreateSequenceRefused

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

822 [Detail]

823 `xs:any`

824 **4.7 Sequence Closed**

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

826 This fault MUST be generated when an RM Destination is asked to receive a message for a Sequence
827 that is closed.

828 Properties:

829 [Code] Sender

830 [Subcode] wsrn:SequenceClosed

831 [Reason] The Sequence is closed and can not receive new messages.

832 [Detail]

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

834 **4.8 WSRM Required**

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

837 Properties:

838 [Code] Sender

839 [Subcode] wsrn:WSRMRequired

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

841 [Detail]

842 `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.

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

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.

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.

[WS-Security]

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

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

936 **[RTTM]**

937 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May
938 1992.

939 **[SecurityPolicy]**

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

941 **[SecureConversation]**

942 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," February
943 2005.

944 **[Trust]**

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

946 **A. Schema**

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

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

950 The following copy is provided for reference.

```

951 <?xml version="1.0" encoding="UTF-8"?>
952 <!--
953 OASIS takes no position regarding the validity or scope of any intellectual
954 property or other rights that might be claimed to pertain to the
955 implementation or use of the technology described in this document or the
956 extent to which any license under such rights might or might not be available;
957 neither does it represent that it has made any effort to identify any such
958 rights. Information on OASIS's procedures with respect to rights in OASIS
959 specifications can be found at the OASIS website. Copies of claims of rights
960 made available for publication and any assurances of licenses to be made
961 available, or the result of an attempt made to obtain a general license or
962 permission for the use of such proprietary rights by implementors or users of
963 this specification, can be obtained from the OASIS Executive Director.
964 OASIS invites any interested party to bring to its attention any copyrights,
965 patents or patent applications, or other proprietary rights which may cover
966 technology that may be required to implement this specification. Please
967 address the information to the OASIS Executive Director.
968 Copyright © OASIS Open 2002-2006. All Rights Reserved.
969 This document and translations of it may be copied and furnished to others,
970 and derivative works that comment on or otherwise explain it or assist in its
971 implementation may be prepared, copied, published and distributed, in whole or
972 in part, without restriction of any kind, provided that the above copyright
973 notice and this paragraph are included on all such copies and derivative
974 works. However, this document itself does not be modified in any way, such as
975 by removing the copyright notice or references to OASIS, except as needed for
976 the purpose of developing OASIS specifications, in which case the procedures
977 for copyrights defined in the OASIS Intellectual Property Rights document must
978 be followed, or as required to translate it into languages other than English.
979 The limited permissions granted above are perpetual and will not be revoked by
980 OASIS or its successors or assigns.
981 This document and the information contained herein is provided on an "AS
982 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
983 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
984 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
985 FITNESS FOR A PARTICULAR PURPOSE.
986 -->
987 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
988 xmlns:wsa="http://www.w3.org/2005/08/addressing"
989 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200604"
990 targetNamespace="http://docs.oasis-open.org/ws-rx/wsm/200604"
991 elementFormDefault="qualified" attributeFormDefault="unqualified">
992   <xs:import namespace="http://www.w3.org/2005/08/addressing"
993   schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
994   <!-- Protocol Elements -->
995   <xs:complexType name="SequenceType">
996     <xs:sequence>
997       <xs:element ref="wsm:Identifier"/>
998       <xs:element name="MessageNumber" type="wsm:MessageNumberType"/>
999       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1000 maxOccurs="unbounded"/>
1001     </xs:sequence>
1002     <xs:anyAttribute namespace="##other" processContents="lax"/>
1003   </xs:complexType>
1004   <xs:element name="Sequence" type="wsm:SequenceType"/>
1005   <xs:element name="SequenceAcknowledgement">
1006     <xs:complexType>
1007       <xs:sequence>
1008         <xs:element ref="wsm:Identifier"/>
1009         <xs:choice>
1010           <xs:sequence>
1011             <xs:choice>
1012               <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
1013                 <xs:complexType>

```



```

1014         <xs:sequence/>
1015         <xs:attribute name="Upper" type="xs:unsignedLong"
1016 use="required"/>
1017         <xs:attribute name="Lower" type="xs:unsignedLong"
1018 use="required"/>
1019         <xs:anyAttribute namespace="##other" processContents="lax"/>
1020     </xs:complexType>
1021 </xs:element>
1022     <xs:element name="None" minOccurs="0">
1023         <xs:complexType>
1024             <xs:sequence/>
1025         </xs:complexType>
1026     </xs:element>
1027 </xs:choice>
1028     <xs:element name="Final" minOccurs="0">
1029         <xs:complexType>
1030             <xs:sequence/>
1031         </xs:complexType>
1032     </xs:element>
1033 </xs:sequence>
1034     <xs:element name="Nack" type="xs:unsignedLong"
1035 maxOccurs="unbounded"/>
1036 </xs:choice>
1037     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1038 maxOccurs="unbounded"/>
1039 </xs:sequence>
1040     <xs:anyAttribute namespace="##other" processContents="lax"/>
1041 </xs:complexType>
1042 </xs:element>
1043 <xs:complexType name="AckRequestedType">
1044     <xs:sequence>
1045         <xs:element ref="wsrm:Identifier"/>
1046         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1047 maxOccurs="unbounded"/>
1048     </xs:sequence>
1049     <xs:anyAttribute namespace="##other" processContents="lax"/>
1050 </xs:complexType>
1051 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1052 <xs:element name="Identifier">
1053     <xs:complexType>
1054         <xs:annotation>
1055             <xs:documentation>
1056                 This type is for elements whose [children] is an anyURI and can have
1057 arbitrary attributes.
1058             </xs:documentation>
1059         </xs:annotation>
1060         <xs:simpleContent>
1061             <xs:extension base="xs:anyURI">
1062                 <xs:anyAttribute namespace="##other" processContents="lax"/>
1063             </xs:extension>
1064         </xs:simpleContent>
1065     </xs:complexType>
1066 </xs:element>
1067 <xs:simpleType name="MessageNumberType">
1068     <xs:restriction base="xs:unsignedLong">
1069         <xs:minInclusive value="1"/>
1070         <xs:maxInclusive value="9223372036854775807"/>
1071     </xs:restriction>
1072 </xs:simpleType>
1073 <!-- Fault Container and Codes -->
1074 <xs:simpleType name="FaultCodes">
1075     <xs:restriction base="xs:QName">
1076         <xs:enumeration value="wsrm:SequenceTerminated"/>

```

```

1077     <xs:enumeration value="wsrm:UnknownSequence"/>
1078     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1079     <xs:enumeration value="wsrm:MessageNumberRollover"/>
1080     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1081     <xs:enumeration value="wsrm:SequenceClosed"/>
1082     <xs:enumeration value="wsrm:WSRMRequired"/>
1083   </xs:restriction>
1084 </xs:simpleType>
1085 <xs:complexType name="SequenceFaultType">
1086   <xs:sequence>
1087     <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1088     <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
1089     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1090 maxOccurs="unbounded"/>
1091   </xs:sequence>
1092   <xs:anyAttribute namespace="##other" processContents="lax"/>
1093 </xs:complexType>
1094 <xs:complexType name="DetailType">
1095   <xs:sequence>
1096     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1097 maxOccurs="unbounded"/>
1098   </xs:sequence>
1099   <xs:anyAttribute namespace="##other" processContents="lax"/>
1100 </xs:complexType>
1101 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1102 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1103 <xs:element name="CreateSequenceResponse"
1104 type="wsrm:CreateSequenceResponseType"/>
1105 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1106 <xs:element name="CloseSequenceResponse"
1107 type="wsrm:CloseSequenceResponseType"/>
1108 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1109 <xs:element name="TerminateSequenceResponse"
1110 type="wsrm:TerminateSequenceResponseType"/>
1111 <xs:complexType name="CreateSequenceType">
1112   <xs:sequence>
1113     <xs:element ref="wsrm:AcksTo"/>
1114     <xs:element ref="wsrm:Expires" minOccurs="0"/>
1115     <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1116     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1117 maxOccurs="unbounded">
1118       <xs:annotation>
1119         <xs:documentation>
1120           It is the authors intent that this extensibility be used to
1121 transfer a Security Token Reference as defined in WS-Security.
1122         </xs:documentation>
1123       </xs:annotation>
1124     </xs:any>
1125   </xs:sequence>
1126   <xs:anyAttribute namespace="##other" processContents="lax"/>
1127 </xs:complexType>
1128 <xs:complexType name="CreateSequenceResponseType">
1129   <xs:sequence>
1130     <xs:element ref="wsrm:Identifier"/>
1131     <xs:element ref="wsrm:Expires" minOccurs="0"/>
1132     <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>

```

```

1133     <xs:element name="IncompleteSequenceBehaviour"
1134 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
1135     <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1136     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1137 maxOccurs="unbounded"/>
1138   </xs:sequence>
1139   <xs:anyAttribute namespace="##other" processContents="lax"/>
1140 </xs:complexType>
1141 <xs:complexType name="CloseSequenceType">
1142   <xs:sequence>
1143     <xs:element ref="wsrm:Identifier"/>
1144     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1145 maxOccurs="unbounded"/>
1146   </xs:sequence>
1147   <xs:anyAttribute namespace="##other" processContents="lax"/>
1148 </xs:complexType>
1149 <xs:complexType name="CloseSequenceResponseType">
1150   <xs:sequence>
1151     <xs:element ref="wsrm:Identifier"/>
1152     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1153 maxOccurs="unbounded"/>
1154   </xs:sequence>
1155   <xs:anyAttribute namespace="##other" processContents="lax"/>
1156 </xs:complexType>
1157 <xs:complexType name="TerminateSequenceType">
1158   <xs:sequence>
1159     <xs:element ref="wsrm:Identifier"/>
1160     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1161 maxOccurs="unbounded"/>
1162   </xs:sequence>
1163   <xs:anyAttribute namespace="##other" processContents="lax"/>
1164 </xs:complexType>
1165 <xs:complexType name="TerminateSequenceResponseType">
1166   <xs:sequence>
1167     <xs:element ref="wsrm:Identifier"/>
1168     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1169 maxOccurs="unbounded"/>
1170   </xs:sequence>
1171   <xs:anyAttribute namespace="##other" processContents="lax"/>
1172 </xs:complexType>
1173 <xs:element name="AcksTo"

```

```

1174     type="wsa:EndpointReferenceType"/>
1175     <xs:complexType name="OfferType">
1176       <xs:sequence>
1177         <xs:element ref="wsrm:Identifier"/>
1178         <xs:element ref="wsrm:Expires" minOccurs="0"/>
1179         <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
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:complexType name="AcceptType">
1186       <xs:sequence>
1187         <xs:element ref="wsrm:AcksTo"/>
1188         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1189 maxOccurs="unbounded"/>
1190       </xs:sequence>
1191       <xs:anyAttribute namespace="##other" processContents="lax"/>
1192     </xs:complexType>
1193     <xs:element name="Expires">
1194       <xs:complexType>
1195         <xs:simpleContent>
1196           <xs:extension base="xs:duration">
1197             <xs:anyAttribute namespace="##other" processContents="lax"/>
1198           </xs:extension>
1199         </xs:simpleContent>
1200       </xs:complexType>
1201     </xs:element>
1202     <xs:element name="AcknowledgementInterval">
1203       <xs:complexType>
1204         <xs:sequence/>
1205         <xs:attribute name="Milliseconds" type="xs:unsignedLong"
1206 use="required"/>
1207         <xs:anyAttribute namespace="##other" processContents="lax"/>
1208       </xs:complexType>
1209     </xs:element>
1210     <xs:simpleType name="IncompleteSequenceBehaviorType">
1211       <xs:restriction base="xs:string">
1212         <xs:enumeration value="DiscardEntireSequence"/>
1213         <xs:enumeration value="DiscardFollowingFirstGap"/>
1214         <xs:enumeration value="NoDiscard"/>
1215       </xs:restriction>
1216     </xs:simpleType>
1217 </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:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
  xmlns:wsa="http://www.w3.org/2005/08/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/wsmr/200604/CreateSequence</wsa:Action>
    <wsa:ReplyTo>
      <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
    </wsa:ReplyTo>
  </S:Header>
  <S:Body>
    <wsmr:CreateSequence>
      <wsmr:AcksTo>
        <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
      </wsmr:AcksTo>
    </wsmr: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:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
  xmlns:wsa="http://www.w3.org/2005/08/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/wsmr/200604/CreateSequenceResponse
    </wsa:Action>
  </S:Header>
  <S:Body>
    <wsmr:CreateSequenceResponse>
      <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
    </wsmr: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:

1268 Message 1

```
1269 <?xml version="1.0" encoding="UTF-8"?>
1270 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1271 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
1272 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1273   <S:Header>
1274     <wsa:MessageID>
1275       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
1276     </wsa:MessageID>
1277     <wsa:To>http://example.com/serviceB/123</wsa:To>
1278     <wsa:From>
1279       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1280     </wsa:From>
1281     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1282     <wsmr:Sequence>
1283       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1284       <wsmr:MessageNumber>1</wsmr:MessageNumber>
1285     </wsmr:Sequence>
1286   </S:Header>
1287   <S:Body>
1288     <!-- Some Application Data -->
1289   </S:Body>
1290 </S:Envelope>
```

1291 Message 2

```
1292 <?xml version="1.0" encoding="UTF-8"?>
1293 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1294 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
1295 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1296   <S:Header>
1297     <wsa:MessageID>
1298       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1299     </wsa:MessageID>
1300     <wsa:To>http://example.com/serviceB/123</wsa:To>
1301     <wsa:From>
1302       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1303     </wsa:From>
1304     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1305     <wsmr:Sequence>
1306       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1307       <wsmr:MessageNumber>2</wsmr:MessageNumber>
1308     </wsmr:Sequence>
1309   </S:Header>
1310   <S:Body>
1311     <!-- Some Application Data -->
1312   </S:Body>
1313 </S:Envelope>
```

1314 Message 3

```
1315 <?xml version="1.0" encoding="UTF-8"?>
1316 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1317 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
1318 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1319   <S:Header>
1320     <wsa:MessageID>
1321       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1322     </wsa:MessageID>
1323     <wsa:To>http://example.com/serviceB/123</wsa:To>
1324     <wsa:From>
1325       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
```

```

1326 </wsa:From>
1327 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1328 <wsrm:Sequence>
1329 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1330 <wsrm:MessageNumber>3</wsrm:MessageNumber>
1331 </wsrm:Sequence>
1332 <wsrm:AckRequested>
1333 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1334 </wsrm:AckRequested>
1335 </S:Header>
1336 <S:Body>
1337 <!-- Some Application Data -->
1338 </S:Body>
1339 </S:Envelope>

```

1340 B.3 First Acknowledgement

1341 Message number 2 has not been received by the RM Destination due to some transmission error so it
1342 responds with an acknowledgement for messages 1 and 3:

```

1343 <?xml version="1.0" encoding="UTF-8"?>
1344 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1345 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
1346 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1347 <S:Header>
1348 <wsa:MessageID>
1349 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1350 </wsa:MessageID>
1351 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1352 <wsa:From>
1353 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1354 </wsa:From>
1355 <wsa:Action>
1356 http://docs.oasis-open.org/ws-rx/wsr/200604/SequenceAcknowledgement
1357 </wsa:Action>
1358 <wsrm:SequenceAcknowledgement>
1359 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1360 <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
1361 <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
1362 </wsrm:SequenceAcknowledgement>
1363 </S:Header>
1364 <S:Body/>
1365 </S:Envelope>

```

1366 B.4 Retransmission

1367 The RM Sourcediscovers that message number 2 was not received so it resends the message and
1368 requests an acknowledgement:

```

1369 <?xml version="1.0" encoding="UTF-8"?>
1370 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1371 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
1372 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1373 <S:Header>
1374 <wsa:MessageID>
1375 http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1376 </wsa:MessageID>
1377 <wsa:To>http://example.com/serviceB/123</wsa:To>
1378 <wsa:From>
1379 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1380 </wsa:From>

```

```

1381 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1382 <wsrm:Sequence>
1383 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1384 <wsrm:MessageNumber>2</wsrm:MessageNumber>
1385 </wsrm:Sequence>
1386 <wsrm:AckRequested>
1387 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1388 </wsrm:AckRequested>
1389 </S:Header>
1390 <S:Body>
1391 <!-- Some Application Data -->
1392 </S:Body>
1393 </S:Envelope>

```

1394 B.5 Termination

1395 The RM Destination now responds with an acknowledgement for the complete Sequence which can then
1396 be terminated:

```

1397 <?xml version="1.0" encoding="UTF-8"?>
1398 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1399 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
1400 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1401 <S:Header>
1402 <wsa:MessageID>
1403 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1404 </wsa:MessageID>
1405 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1406 <wsa:From>
1407 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1408 </wsa:From>
1409 <wsa:Action>
1410 http://docs.oasis-open.org/ws-rx/wsr/200604/SequenceAcknowledgement
1411 </wsa:Action>
1412 <wsrm:SequenceAcknowledgement>
1413 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1414 <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
1415 </wsrm:SequenceAcknowledgement>
1416 </S:Header>
1417 <S:Body/>
1418 </S:Envelope>

```

1419 Terminate Sequence

```

1420 <?xml version="1.0" encoding="UTF-8"?>
1421 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1422 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
1423 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1424 <S:Header>
1425 <wsa:MessageID>
1426 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1427 </wsa:MessageID>
1428 <wsa:To>http://example.com/serviceB/123</wsa:To>
1429 <wsa:Action>
1430 http://docs.oasis-open.org/ws-rx/wsr/200604/TerminateSequence
1431 </wsa:Action>
1432 <wsa:From>
1433 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1434 </wsa:From>
1435 </S:Header>
1436 <S:Body>
1437 <wsrm:TerminateSequence>

```



```

1438     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1439   </wsrm:TerminateSequence>
1440 </S:Body>
1441 </S:Envelope>

```

1442 Terminate Sequence Response

```

1443 <?xml version="1.0" encoding="UTF-8"?>
1444 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1445   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
1446   xmlns:wsa="http://www.w3.org/2005/08/addressing">
1447   <S:Header>
1448     <wsa:MessageID>
1449       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
1450     </wsa:MessageID>
1451     <wsa:To>http://example.com/serviceA/789</wsa:To>
1452     <wsa:Action>
1453       http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequenceResponse
1454     </wsa:Action>
1455     <wsa:RelatesTo>
1456       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1457     </wsa:RelatesTo>
1458     <wsa:From>
1459       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1460     </wsa:From>
1461   </S:Header>
1462   <S:Body>
1463     <wsrm:TerminateSequenceResponse>
1464       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1465     </wsrm:TerminateSequenceResponse>
1466   </S:Body>
1467 </S:Envelope>

```

C. WSDL

1468

1469 The non-normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

1470 <http://docs.oasis-open.org/ws-rx/wsrn/200604/wsd/wsrn-1.1-wsd-200604.wsd>

1471 The following non-normative copy is provided for reference.

```

1472 <?xml version="1.0" encoding="utf-8"?>
1473 <!--
1474 OASIS takes no position regarding the validity or scope of any intellectual
1475 property or other rights that might be claimed to pertain to the
1476 implementation or use of the technology described in this document or the
1477 extent to which any license under such rights might or might not be available;
1478 neither does it represent that it has made any effort to identify any such
1479 rights. Information on OASIS's procedures with respect to rights in OASIS
1480 specifications can be found at the OASIS website. Copies of claims of rights
1481 made available for publication and any assurances of licenses to be made
1482 available, or the result of an attempt made to obtain a general license or
1483 permission for the use of such proprietary rights by implementors or users of
1484 this specification, can be obtained from the OASIS Executive Director.
1485 OASIS invites any interested party to bring to its attention any copyrights,
1486 patents or patent applications, or other proprietary rights which may cover
1487 technology that may be required to implement this specification. Please
1488 address the information to the OASIS Executive Director.
1489 Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
1490 This document and translations of it may be copied and furnished to others,
1491 and derivative works that comment on or otherwise explain it or assist in its
1492 implementation may be prepared, copied, published and distributed, in whole or
1493 in part, without restriction of any kind, provided that the above copyright
1494 notice and this paragraph are included on all such copies and derivative
1495 works. However, this document itself does not be modified in any way, such as
1496 by removing the copyright notice or references to OASIS, except as needed for
1497 the purpose of developing OASIS specifications, in which case the procedures
1498 for copyrights defined in the OASIS Intellectual Property Rights document must
1499 be followed, or as required to translate it into languages other than English.
1500 The limited permissions granted above are perpetual and will not be revoked by
1501 OASIS or its successors or assigns.
1502 This document and the information contained herein is provided on an "AS
1503 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
1504 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
1505 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
1506 FITNESS FOR A PARTICULAR PURPOSE.
1507 -->
1508 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
1509 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1510 xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
1511 open.org/ws-rx/wsr/200604" xmlns:tns="http://docs.oasis-open.org/ws-
1512 rx/wsr/200604/wsdl" targetNamespace="http://docs.oasis-open.org/ws-
1513 rx/wsr/200604/wsdl">
1514     <wsdl:types>
1515         <xs:schema
1516             <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsr/200604"
1517             schemaLocation="http://docs.oasis-open.org/ws-rx/wsr/200604/wsr-1.1-schema-
1518             200604.xsd"/>
1519         </xs:schema>
1520     </wsdl:types>
1521
1522     <wsdl:message name="CreateSequence">
1523         <wsdl:part name="create" element="rm:CreateSequence"/>
1524     </wsdl:message>
1525     <wsdl:message name="CreateSequenceResponse">
1526         <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
1527     </wsdl:message>
1528     <wsdl:message name="CloseSequence">
1529         <wsdl:part name="close" element="rm:CloseSequence"/>
1530     </wsdl:message>
1531     <wsdl:message name="CloseSequenceResponse">
1532         <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
1533     </wsdl:message>

```

```

1533     <wsdl:message name="TerminateSequence">
1534         <wsdl:part name="terminate" element="rm:TerminateSequence"/>
1535     </wsdl:message>
1536     <wsdl:message name="TerminateSequenceResponse">
1537         <wsdl:part name="terminateResponse"
1538 element="rm:TerminateSequenceResponse"/>
1539     </wsdl:message>

1540     <wsdl:portType name="SequenceAbstractPortType">
1541         <wsdl:operation name="CreateSequence">
1542             <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
1543 open.org/ws-rx/wsrn/200604/CreateSequence"/>
1544             <wsdl:output message="tns:CreateSequenceResponse"
1545 wsa:Action="http://docs.oasis-open.org/ws-
1546 rx/wsrn/200604/CreateSequenceResponse"/>
1547         </wsdl:operation>
1548         <wsdl:operation name="CloseSequence">
1549             <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
1550 open.org/ws-rx/wsrn/200604/CloseSequence"/>
1551             <wsdl:output message="tns:CloseSequenceResponse"
1552 wsa:Action="http://docs.oasis-open.org/ws-
1553 rx/wsrn/200604/CloseSequenceResponse"/>
1554         </wsdl:operation>
1555         <wsdl:operation name="TerminateSequence">
1556             <wsdl:input message="tns:TerminateSequence"
1557 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequence"/>
1558             <wsdl:output message="tns:TerminateSequenceResponse"
1559 wsa:Action="http://docs.oasis-open.org/ws-
1560 rx/wsrn/200604/TerminateSequenceResponse"/>
1561         </wsdl:operation>
1562     </wsdl:portType>

1563 </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 Connecting</i>	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Create Sequence Response	N/A	No action Connected	N/A	N/A	N/A	N/A	N/A	N/A
Create Sequence Refused Fault	N/A	No action Terminated	N/A	N/A	N/A	N/A	N/A	N/A
New Message	N/A	N/A	<i>Transmit message Connected</i>	<i>no action Rollover</i>	<i>No action Closing</i>	N/A	N/A	N/A
Retransmit of unack message	N/A	N/A	<i>Transmit message Connected</i>	<i>Transmit message Rollover</i>	<i>Trasmit message? Closing</i>	<i>No action Closed</i>	N/A	N/A
SeqAck (non-final)	N/A	N/A	<i>Process Ack ranges Connected</i>	<i>Process Ack ranges Rollover</i>	<i>Process Ack ranges Closing</i>	<i>Process Ack ranages Closed</i>	<i>Process Ack ranages Terminating</i>	<i>Transmit Unknown Sequence Fault Terminated</i>
Nack	N/A	N/A	<i>Transmit message(s) Connected</i>	<i>Transmit message(s) Rollover</i>	<i>Transmit message(s) Closing</i>	<i>No action Closed</i>	<i>No action Terminating</i>	<i>Transmit Unknown Sequence fault Terminated</i>
Reached max msg number	N/A	N/A	<i>No action Rollover</i>	<i>No action Rollover</i>	N/A	N/A	N/A	N/A

Events	States							
	None	Connecting	Connected	Rollover	Closing	Closed	Terminating	Terminated
Message Number Rollover Fault	N/A	N/A	No action Rollover	No action Rollover	No action Closing	No action Closed	No action Terminating	Transmit Unknown Sequence Fault Terminated
Close Sequence	N/A	N/A	Transmit Close Sequence Closing	Transmit Close Sequence Closing	Transmit Close Sequence Closing	No action Closed	No action Terminating	N/A
Close Sequence Response	N/A	N/A	N/A	N/A	No action Closed	No action Closed	No action Terminating	Transmit Unknown Sequence Fault Terminated
SeqAck (final)	N/A	N/A	Process Ack/Nack ranges Closed	Process Ack/Nack ranges Closed	Process Ack/Nack ranges Closed	Process Ack/Nack ranges Closed	Process Ack/Nack ranges Terminating	Transmit Unknown Sequence fault Terminated
Sequence Closed Fault	N/A	N/A	No action Closed	No action Closed	No action Closed	No action Closed	No action Terminating	Transmit Unknown Sequence Fault Terminated
Unknown Sequence Fault	N/A	N/A	No action Terminated	No action Terminated	No action Terminated	No action Terminated	No action Terminated	No action Terminated
Sequence Terminated Fault	N/A	N/A	No action Terminated	No action Terminated	No action Terminated	No action Terminated	No action Terminated	No Action Terminated
Terminate Sequence	N/A	N/A	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	N/A
Terminate Sequence Response	N/A	N/A	N/A	N/A	N/A	N/A	No action Terminated	No action Terminated
Elapse Expires duration	N/A	N/A	Send SequenceTerminated Fault Terminated	Send SequenceTerminated Fault Terminated	Send SequenceTerminated Fault Terminated	Send SequenceTerminated Fault Terminated	Send SequenceTerminated Fault Terminated	N/A

1565 In Table 2 above, the rows consists of events that occur at the RM Source throughout the lifetime of an
1566 RM Sequence and the columns consists of various RM Source states. Each cell in the table above lists

1567 the action that the RM Source takes on occurrence of a particular event and the next state that it
1568 transitions.

1569 Table 3 RM Destination State Transition Table

Events	States						
	None	Connecting	Connected			Closed	Terminated
Creation request not satisfied	N/A	<i>Send Create Sequence Refused Fault</i> Terminated	N/A			N/A	
Message (with message number within range)	N/A	N/A	<i>No action</i> Connected			<i>Send Sequence Closed Fault (with SeqAck+Final)</i> Closed	<i>Send Unknown Seq Fault</i> Terminated
Ack requested	N/A	N/A	<i>Send SequenceAck</i> Connected			<i>Send SeqAck+Final</i> Closed	<i>Send Unknown Seq Fault</i> Terminated
Message (with message number outside of range)	N/A	N/A	<i>Send Message Number Rollover Fault</i> Connected			N/A	N/A
Close Sequence	N/A	N/A	<i>Send CloseSequenceResponse with SequenceAck (Final)</i> Closed			<i>Send Close Sequence Response with SeqAck+Final</i> Closed	<i>Send Unknown Sequence Fault</i> Terminated
Close Sequence itself	N/A	N/A	Closed			<i>Send Sequence Closed Fault</i> Closed	N/A
Terminate Sequence	N/A	N/A	<i>Send Terminate Sequence Response</i> Terminated			<i>Send Terminate Sequence Response</i> Terminated	<i>Send Unknown Sequence Fault</i> Terminated

Events	States						
	None	Connecting	Connected			Closed	Terminated
Unknown Sequence Fault	N/A	N/A	<i>No action</i> Terminated			<i>No action</i> Terminated	<i>No action</i> Terminated
Sequence Terminated Fault	N/A	N/A	<i>No action</i> Terminated			<i>No action</i> Terminated	<i>No action</i> Terminated
EIapse Expires duration	N/A	N/A	<i>Send Sequence Terminated Fault</i> Terminated			<i>Send Sequence Terminated Fault</i> Terminated	N/A

1570 In Table 3 above, the rows consists of events that occur at the RM Destination throughout the lifetime of
1571 an RM Sequence and the columns consists of various RM Destination states. Each cell in the table above
1572 lists the action that the RM Destination takes on occurrence of a particular event and the next state that it
1573 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 (PT0S) added
wd-02	2005-08-16	Anish Karmarkar	Trivial editorial changes
ws-03	2005-09-15	Doug Davis	i019 and i028 (CloseSeq) added
wd-05	2005-09-26	Gilbert Pilz	i005 (Source resend of nacks messages when ack already received) added.
wd-05	2005-09-27	Doug Davis	i027 (InOrder delivery assurance spanning multiple sequences) added
wd-05	2005-09-27	Doug Davis	i020 (Semantics of "At most once" Delivery Assurance) added
wd-05	2005-09-27	Doug Davis	i034 (Fault while processing a piggy-backed RM header) added
wd-05	2005-09-27	Doug Davis	i033 (Processing model of NACKs) added
wd-05	2005-09-27	Doug Davis	i031 (AckRequested schema inconsistency) added
wd-05	2005-09-27	Doug Davis	i025 (SeqAck/None) added
wd-05	2005-09-27	Doug Davis	i029 (Remove dependency on WS-Security) added
wd-05	2005-09-27	Doug Davis	i039 (What does 'have a mU attribute' mean) added
wd-05	2005-09-27	Doug Davis	i040 (Change 'optiona'/'required' to 'OPTIONAL'/'REQUIRED') added
wd-05	2005-09-30	Anish Karmarkar	i017 (Change NS to http://docs.oasis-open.org/wsrn/200510/)
wd-05	2005-09-30	Anish Karmarkar	i045 (Include SecureConversation as a reference and move it to non-normative citation)
wd-05	2005-09-30	Anish Karmarkar	i046 (change the type of wsrn:FaultCode element)
wd-06	2005-11-02	Gilbert Pilz	Start wd-06 by changing title page from cd-01.
wd-06	2005-11-03	Gilbert Pilz	i047 (Reorder spec sections)
wd-07	2005-11-17	Gilbert Pilz	Start wd-07
wd-07	2005-11-28	Doug Davis	i071 – except for period in Appendix headings
wd-07	2005-11-28	Doug Davis	i10
wd-07	2005-11-28	Doug Davis	i030
wd-07	2005-11-28	Doug Davis	i037
wd-07	2005-11-28	Doug Davis	i038
wd-07	2005-11-28	Doug Davis	i041
wd-07	2005-11-28	Doug Davis	i043
wd-07	2005-11-28	Doug Davis	i044

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

Rev	Date	By Whom	What
			OpenDocument Text format. Changed line numbers to be a single style.
wd-08	2005-12-28	Anish Karmarkar	Included a section link to c:\temp\wsrm-1.1-schema-200510.xsd and to c:\temp\wsrm-1.1-wsdl-200510.wsdl
wd-08	2006-01-04	Gilbert Pilz	Fixed formatting for included sections.
wd-08	2006-01-05	Gilbert Pilz	Created links for unused references. Fixed exemplars for CloseSequence and CloseSequenceResponse.
wd-09	2006-01-11	Doug Davis	Minor tweaks to text/typos.
wd-10	2006-01-23	Doug Davis	Accept all changes from wd-09 Make some minor editorial tweaks from Marc's comments.
wd-10	2006-02-14	Doug Davis	Issue 082 resolution
wd-10	2006-02-14	Doug Davis	Issue 083 resolution
wd-10	2006-02-14	Doug Davis	Issue 085 resolution
wd-10	2006-02-14	Doug Davis	Issues 086, 087 resolutions Defined MessageNumberType
wd-10	2006-02-15	Doug Davis	Issue 078 resolution
wd-10	2006-02-15	Doug Davis	Issue 094 resolution
wd-10	2006-02-15	Doug Davis	Issue 095 resolution
wd-10	2006-02-15	Gilbert Pilz	Issue 088 – added namespace URI link to namespace URI; added text explaining that this URI could be dereferenced to produce the RDDDL doc; added non-normative reference to RDDDL 2.0
wd-10	2006-02-17	Anish Karmarkar	Namespace changed to 200602 for both WSDL and XSD docs.
wd-10	2006-02-17	Anish Karmarkar	Issue i087 as it applies to WSRM spec.
wd-10	2006-02-17	Anish Karmarkar	Added titles and minor text for state table (issue i058).
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".
wd-13	2006-05-08	Gilbert Pilz	i093 part 1; more work needed
wd-13	2006-05-10	Doug Davis	Issue 096 applied
wd-13	2006-05-26	Gilbert Pilz	i093 part 2; reflects decisions from 2006-05-25 meeting
wd-13	2006-05-28	Gilbert Pilz	Issue 106 applied
wd-13	2006-05-29	Gilbert Pilz	Issue 118 applied
wd-13	2006-05-29	Gilbert Pilz	Issue 120 applied
wd-13	2006-05-30	Gilbert Pilz	Issue 114 applied
wd-13	2006-05-30	Gilbert Pilz	Issue 116 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.