



1 Web Services ReliableMessaging 2 (WS-Reliable Messaging)

3 Working Draft 12, April 24, 2006

4 Document identifier:

5 wsrn-1.1-spec-wd-12

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

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

1 Introduction

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

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

1.1 Goals and Requirements

1.1.1 Requirements

1.2 Notational Conventions

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

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

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

1.3 Namespace

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

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

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

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

115 The following namespaces are used in this document:

116 *Table 1*

Prefix	Namespace
S	(Either SOAP 1.1 or 1.2)
S11	http://schemas.xmlsoap.org/soap/envelope/
S12	http://www.w3.org/2003/05/soap-envelope
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

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

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

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

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

123 <http://docs.oasis-open.org/ws-rx/wsrn/200604/SequenceAcknowledgement>

124 1.4 Compliance

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

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

2 Reliable Messaging Model

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

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

The protocol supports reliability features that enable ordered delivery, duplicate elimination, and guaranteed receipt for the RMD. It is expected that the AD and RMD will implement as many of these or as few of these characteristics as necessary to implement the AD. Regardless of which of the reliability features are employed, 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.

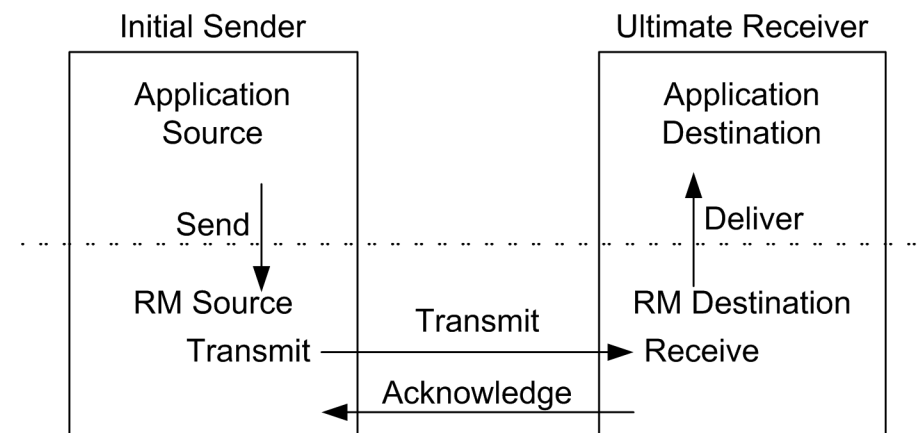


Figure 1: Reliable Messaging Model

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.

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

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

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

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

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

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

135 2.2 Protocol Preconditions

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

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

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

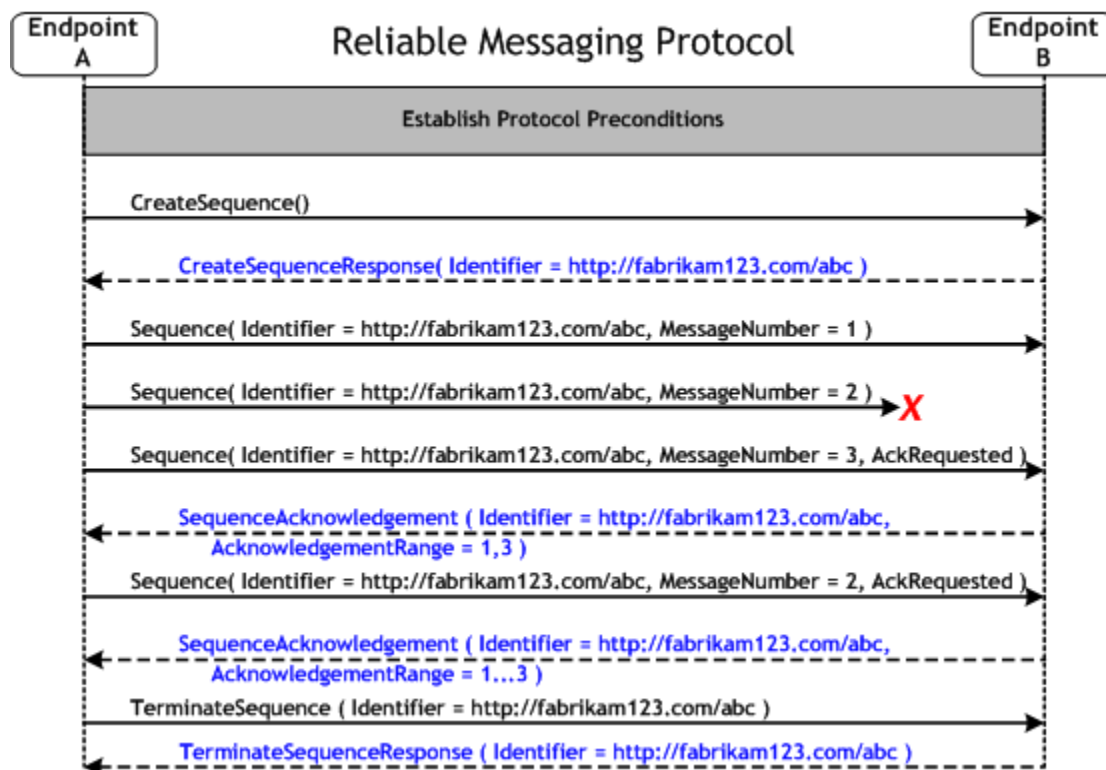
135 2.3 Protocol Invariants

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

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

138 2.4 Example Message Exchange

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



138 Figure 2: The WS-ReliableMessaging Protocol

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

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

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

3 RM Protocol Elements

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

3.1 Sequence Creation

The RM Source MUST request creation of an outbound Sequence by sending a `<wsrm:CreateSequence>` element in the body of a message to the RM Destination which in turn responds either with a message containing `<wsrm:CreateSequenceResponse>` or a `CreateSequenceRefused` fault. The `<wsrm:CreateSequence>` MAY carry an offer to create an inbound Sequence which is either accepted or rejected in the `<wsrm:CreateSequenceResponse>`.

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

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

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

`/wsrm:CreateSequence`

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

`/wsrm:CreateSequence/wsrm:AcksTo`

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

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

`/wsrm:CreateSequence/wsrm:Expires`

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

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

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

195 /wsrm:CreateSequence/wsrm:Offer

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

198 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier

199 This REQUIRED element MUST contain an absolute URI conformant with RFC3986 [URI] that uniquely
200 identifies the offered Sequence.

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

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

204 /wsrm:CreateSequence/wsrm:Offer/wsrm:Endpoint

205 This REQUIRED element, of type `wsa:EndpointReferenceType` as specified by WS-Addressing
206 [WSAddressing] specifies the endpoint reference to which WS-RM protocol messages related to the
207 offered Sequence are to be sent.

208 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires

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

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

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

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

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

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

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

221 /wsrm:CreateSequence/{any}

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

224 /wsrm:CreateSequence/@{any}

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

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

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

```
232 <wsrm:CreateSequenceResponse ...>
233   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
234   <wsrm:Expires> xs:duration </wsrm:Expires> ?
235   <wsrm:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?
236   <wsrm:Accept ...>
237     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
238     ...
239   </wsrm:Accept> ?
240   ...
241 </wsrm:CreateSequenceResponse>
```

242 `/wsrm:CreateSequenceResponse`

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

246 `/wsrm:CreateSequenceResponse/wsrm:Identifier`

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

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

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

252 `/wsrm:CreateSequenceResponse/wsrm:Expires`

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

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

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

260 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval`

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

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

264 The acknowledgement interval, specified in milliseconds.

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

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

268 `/wsrm:CreateSequenceResponse/wsrm:Accept`

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

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

274 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo
 275 This REQUIRED element, of type wsa:EndpointReferenceType as specified by WS-Addressing [WS-
 276 Addressing], specifies the endpoint reference to which <wsrm:SequenceAcknowledgement>
 277 messages related to the accepted Sequence are to be sent.

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

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

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

287 /wsrm:CreateSequenceResponse/@{any}
 288 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
 289 element.

290 3.2 Closing A Sequence

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

296 If the RM Source wishes to close the Sequence, then it sends a <wsrm:CloseSequence> element, in
 297 the body of a message, to the RM Destination. This message indicates that the RM Destination MUST
 298 NOT receive any new messages for the specified Sequence, other than those already received at the time
 299 the <wsrm:CloseSequence> element is interpreted by the RMD. Upon receipt of this message, or
 300 subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST
 301 include a final SequenceAcknowledgement (that MUST include the <wsrm:Final> element) header block
 302 on any messages associated with the Sequence destined to the RM Source, including the
 303 CloseSequenceResponse message or on any Sequence Fault transmitted to the RMS.

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

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

312 The following exemplar defines the CloseSequence syntax:

```
313 <wsrm:CloseSequence ...>
314   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
```

315
316

```
...  
</wsrm:CloseSequence>
```

317 /wsrm:CloseSequence

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

321 /wsrm:CloseSequence/wsrm:Identifier

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

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

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

327 /wsrm:CloseSequence/{any}

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

330 /wsrm:CloseSequence@{any}

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

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

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

337
338
339
340

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

341 /wsrm:CloseSequenceResponse

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

345 /wsrm:CloseSequenceResponse/wsrm:Identifier

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

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

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

351 /wsrm:CloseSequenceResponse/{any}

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

354 /wsrm:CloseSequenceResponse@{any}

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

357 3.3 Sequence Termination

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

365 The following exemplar defines the TerminateSequence syntax:

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

370 /wsrm:TerminateSequence

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

376 /wsrm:TerminateSequence/wsrm:Identifier

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

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

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

382 /wsrm:TerminateSequence/{any}

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

385 /wsrm:TerminateSequence/@{any}

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

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

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

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

395 `</wsrm:TerminateSequenceResponse>`

396 `/wsrm:TerminateSequenceResponse`

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

400 `/wsrm:TerminateSequenceResponse/wsrm:Identifier`

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

403 `/wsrm:TerminateSequenceResponse/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:TerminateSequenceResponse/{any}`

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

409 `/wsrm:TerminateSequenceResponse/@{any}`

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

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

414 3.4 Sequences

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

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

422 A following exemplar defines its syntax:

```
423 <wsrm:Sequence ...>
424   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
425   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>
426   ...
427 </wsrm:Sequence>
```

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

429 `/wsrm:Sequence`

430 This protocol element associates the message in which it is contained with a previously established RM
431 Sequence. It contains the Sequence's unique identifier and the containing message's ordinal position
432 within that Sequence. The `<wsrm:Sequence>` element MUST be understood by the RM Destination. The
433 `<wsrm:Sequence>` element MUST have a `mustUnderstand` attribute with a value 1/true from the
434 namespace corresponding to the version of SOAP to which the `<wsrm:Sequence>` SOAP header block
435 is bound.

436 /wsrm:Sequence/wsrm:Identifier

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

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

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

442 /wsrm:Sequence/wsrm:MessageNumber

443 This REQUIRED element MUST contain a `wsrm:MessageNumberType` representing the ordinal position
444 of the message within a Sequence. Sequence MessageNumbers start at 1 and monotonically increase
445 throughout the Sequence. If the message number exceeds the internal limitations of an RM Source or RM
446 Destination or reaches the maximum value of 9,223,372,036,854,775,807 the RM Source or Destination
447 MUST generate a MessageNumberRollover fault.

448 /wsrm:Sequence/{any}

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

451 /wsrm:Sequence/@{any}

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

454 The following example illustrates a Sequence header block.

```
455 <wsrm:Sequence>  
456   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
457   <wsrm:MessageNumber>10</wsrm:MessageNumber>  
458 </wsrm:Sequence>
```

459 3.5 Request Acknowledgement

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

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

471 The following exemplar defines its syntax:

```
472 <wsrm:AckRequested ...>  
473   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
474   ...  
475 </wsrm:AckRequested>
```

476 /wsrm:AckRequested

477 This element requests an acknowledgement for the identified Sequence.

478 /wsrm:AckRequested/wsrm:Identifier

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

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

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

484 /wsrm:AckRequested/{any}

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

487 /wsrm:AckRequested/@{any}

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

490 **3.6 Sequence Acknowledgement**

491 The RM Destination informs the RM Source of successful message receipt using a
492 <wsrm:SequenceAcknowledgement> header block. The <wsrm:SequenceAcknowledgement>
493 header block MAY be transmitted independently or included on any message targeted to the AcksTo EPR.
494 The RM Destination MAY send a <wsrm:SequenceAcknowledgement> header block at any point
495 during which the Sequence is valid. Acknowledgements can be explicitly requested using the
496 <wsrm:AckRequested> directive (see Section [Request Acknowledgement](#)). If a non-mustUnderstand
497 fault occurs when processing an RM Header that was piggy-backed on another message, a fault MUST
498 be generated, but the processing of the original message MUST NOT be affected.

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

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

508 The following exemplar defines its syntax:

```
509 <wsrm:SequenceAcknowledgement ...>
510   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
511   [ [ [ <wsrm:AcknowledgementRange ...
512         Upper="wsrm:MessageNumberType"
513         Lower="wsrm:MessageNumberType"/> +
514         | <wsrm:None/> ]
515         <wsrm:Final/> ? ]
516   | <wsrm:Nack> wsrm:MessageNumberType </wsrm:Nack> + ]
517   ...
518   ...
519 </wsrm:SequenceAcknowledgement>
```

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

521 `/wsrm:SequenceAcknowledgement`

522 This element contains the Sequence acknowledgement information.

523 `/wsrm:SequenceAcknowledgement/wsrm:Identifier`

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

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

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

530 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

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

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

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

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

540 This REQUIRED attribute contains a `wsrm:MessageNumberType` representing the
541 `<wsrm:MessageNumber>` of the lowest contiguous message in a Sequence range received by the RM
542 Destination.

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

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

546 `/wsrm:SequenceAcknowledgement/wsrm:Final`

547 This OPTIONAL element, if present, indicates that the RM Destination is not receiving new messages for
548 the specified Sequence. The RM Source can be assured that the ranges of messages acknowledged by
549 this `SequenceAcknowledgement` header block will not change in the future. This element MUST be
550 present when the Sequence is closed. Note: this element MUST NOT be used when sending a Nack, it
551 can only be used when sending `AcknowledgementRanges` or `<wsrm:None>`.

552 `/wsrm:SequenceAcknowledgement/wsrm:Nack`

553 This OPTIONAL element, if present, MUST contain a `wsrm:MessageNumberType` representing the
554 `<wsrm:MessageNumber>` of an unreceived message in a Sequence. The `<wsrm:Nack>` element
555 MUST NOT be present if a sibling `<wsrm:AcknowledgementRange>` or `<wsrm:None>` element is also
556 present as a child of `<wsrm:SequenceAcknowledgement>`. Upon the receipt of a Nack, an RM Source
557 SHOULD retransmit the message identified by the Nack. The RM Destination MUST NOT issue a
558 `<wsrm:SequenceAcknowledgement>` containing a `<wsrm:Nack>` for a message that it has previously
559 acknowledged within a `<wsrm:AcknowledgementRange>`. The RM Source SHOULD ignore a

560 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that has previously
561 been acknowledged within a <wsrm:AcknowledgementRange>.

562 /wsrm:SequenceAcknowledgement/wsrm:None

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

567 /wsrm:SequenceAcknowledgement/{any}

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

570 /wsrm:SequenceAcknowledgement/@{any}

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

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

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

```
575 <wsrm:SequenceAcknowledgement>  
576   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
577   <wsrm:AcknowledgementRange Upper="10" Lower="1"/>  
578 </wsrm:SequenceAcknowledgement>
```

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

```
581 <wsrm:SequenceAcknowledgement>  
582   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
583   <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
584   <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
585   <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
586 </wsrm:SequenceAcknowledgement>
```

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

```
588 <wsrm:SequenceAcknowledgement>  
589   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
590   <wsrm:Nack>3</wsrm:Nack>  
591 </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. Sequence faults SHOULD be sent to the same [destination] as <wsrm:SequenceAcknowledgement> messages. These faults are correlated using the Sequence identifier carried in the detail.

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

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

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

The definitions of faults use the following properties:

[Code] The fault code.

[Subcode] The fault subcode.

[Reason] The English language reason element.

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

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

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

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

```
<S:Envelope>
  <S:Header>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/08/addressing/fault
    </wsa:Action>
    <!-- Headers elided for clarity. -->
  </S:Header>
  <S:Body>
    <S:Fault>
      <S:Code>
        <S:Value> [Code] </S:Value>
      <S:Subcode>
```

```

632     <S:Value> [Subcode] </S:Value>
633     </S:Subcode>
634   </S:Code>
635   <S:Reason>
636     <S:Text xml:lang="en"> [Reason] </S:Text>
637   </S:Reason>
638   <S:Detail>
639     [Detail]
640     ...
641   </S:Detail>
642 </S:Fault>
643 </S:Body>
644 </S:Envelope>

```

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

```

647 <S11:Envelope>
648   <S11:Header>
649     <wsrm:SequenceFault>
650       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
651       <wsrm:Detail> [Detail] </wsrm:Detail>
652       ...
653     </wsrm:SequenceFault>
654     <!-- Headers elided for clarity. -->
655   </S11:Header>
656   <S11:Body>
657     <S11:Fault>
658       <faultcode> [Code] </faultcode>
659       <faultstring> [Reason] </faultstring>
660     </S11:Fault>
661   </S11:Body>
662 </S11:Envelope>

```

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

```

665 <S11:Envelope>
666   <S11:Body>
667     <S11:Fault>
668       <faultcode> [Subcode] </faultcode>
669       <faultstring> [Reason] </faultstring>
670     </S11:Fault>
671   </S11:Body>
672 </S11:Envelope>

```

673 4.1 SequenceFault Element

674 The purpose of the <wsrm:SequenceFault> element is to carry the specific details of a fault generated
675 during the reliable messaging specific processing of a message belonging to a Sequence. The
676 <wsrm:SequenceFault> container MUST only be used in conjunction with the SOAP 1.1 fault
677 mechanism. It MUST NOT be used in conjunction with the SOAP 1.2 binding.

678 The following exemplar defines its syntax:

```

679 <wsrm:SequenceFault ...>
680   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
681   <wsrm:Detail> ... </wsrm:Detail> ?
682   ...
683 </wsrm:SequenceFault>

```

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

685 `/wsrm:SequenceFault`

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

687 `/wsrm:SequenceFault/wsrm:FaultCode`

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

689 `/wsrm:SequenceFault/wsrm:Detail`

690 This optional element is intended for carrying application specific error information related to the fault

691 being described.

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

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

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

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

696 `/wsrm:SequenceFault/{any}`

697 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,

698 to be passed.

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

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

701 element.

702 4.2 Sequence Terminated

703 This fault is generated by either the RM Source or the RM Destination to indicate that it has either

704 encountered an unrecoverable condition, or has detected a violation of the protocol and as a

705 consequence, has chosen to terminate the Sequence. The endpoint that generates this fault should make

706 every reasonable effort to notify the corresponding endpoint of this decision.

707 Receipt of `SequenceTerminated` by either the RMD or the RMS shall terminate the Sequence if it is not

708 otherwise terminated.

709 Properties:

710 [Code] Sender or Receiver

711 [Subcode] `wsrm:SequenceTerminated`

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

713 [Detail]

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

715 4.3 Unknown Sequence

715 This fault is generated by either the RM Source or the RM Destination in response to a message

716 containing an unknown or terminated Sequence identifier. Receipt of `UnknownSequence` by either the

717 RMD or the RMS shall terminate the Sequence if it is not otherwise terminated.

718 Properties:

718 [Code] Sender

718 [Subcode] wsrn:UnknownSequence

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

718 [Detail]

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

718 4.4 Invalid Acknowledgement

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

721 [Code] Sender

721 [Subcode] wsrn:InvalidAcknowledgement

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

721 [Detail]

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

721 4.5 Message Number Rollover

721 This fault is generated to indicate that message numbers for a Sequence have been exhausted.

722 Properties:

722 [Code] Sender

722 [Subcode] wsrn:MessageNumberRollover

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

722 [Detail]

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

722 4.6 Create Sequence Refused

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

723 Properties:

723 [Code] Sender

723 [Subcode] wsrn:CreateSequenceRefused

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

724 [Detail]

724 `xs:any`

724 4.7 Sequence Closed

724 This fault is generated by an RM Destination to indicate that the specified Sequence has been closed.
725 This fault MUST be generated when an RM Destination is asked to receive a message for a Sequence
726 that is closed.

727 Properties:

727 [Code] Sender

727 [Subcode] wsrn:SequenceClosed

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

728 [Detail]

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

728 4.8 WSRM Required

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

728 Properties:

728 [Code] Sender

728 [Subcode] wsrn:WSRMRequired

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

728 [Detail]

728 `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.

- 735 • **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by comparing
736 secured policies – see WS-Policy and WS-SecurityPolicy).
- 737 • **Authentication** – Authentication is established using the mechanisms described in WS-Security
738 and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- 739 • **Accountability** – Accountability is a function of the type of and string of the key and algorithms
740 being used. In many cases, a strong symmetric key provides sufficient accountability. However, in
741 some environments, strong PKI signatures are required.
- 742 • **Availability** – All reliable messaging services are subject to a variety of availability attacks. Replay
743 detection is a common attack and it is recommended that this be addressed by the mechanisms
744 described in WS-Security. (Note that because of legitimate message replays, detection should
745 include a differentiator besides message id such as a timestamp). Other attacks, such as network-
746 level denial of service attacks are harder to avoid and are outside the scope of this specification.
747 That said, care should be taken to ensure that minimal state is saved prior to any authenticating
748 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]

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

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

763 **[RTTM]**

764 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May
765 1992.

766 **[SecurityPolicy]**

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

768 **[SecureConversation]**

769 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," February
770 2005.

771 **[Trust]**

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

773 A. Schema

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

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

777 The following copy is provided for reference.

```
778 <?xml version="1.0" encoding="UTF-8"?>
779 <!--
780 OASIS takes no position regarding the validity or scope of any intellectual
781 property or other rights that might be claimed to pertain to the
782 implementation or use of the technology described in this document or the
783 extent to which any license under such rights might or might not be available;
784 neither does it represent that it has made any effort to identify any such
785 rights. Information on OASIS's procedures with respect to rights in OASIS
786 specifications can be found at the OASIS website. Copies of claims of rights
787 made available for publication and any assurances of licenses to be made
788 available, or the result of an attempt made to obtain a general license or
789 permission for the use of such proprietary rights by implementors or users of
790 this specification, can be obtained from the OASIS Executive Director.
791 OASIS invites any interested party to bring to its attention any copyrights,
792 patents or patent applications, or other proprietary rights which may cover
793 technology that may be required to implement this specification. Please
794 address the information to the OASIS Executive Director.
795 Copyright © OASIS Open 2002-2006. All Rights Reserved.
796 This document and translations of it may be copied and furnished to others,
797 and derivative works that comment on or otherwise explain it or assist in its
798 implementation may be prepared, copied, published and distributed, in whole or
799 in part, without restriction of any kind, provided that the above copyright
800 notice and this paragraph are included on all such copies and derivative
801 works. However, this document itself does not be modified in any way, such as
802 by removing the copyright notice or references to OASIS, except as needed for
803 the purpose of developing OASIS specifications, in which case the procedures
804 for copyrights defined in the OASIS Intellectual Property Rights document must
805 be followed, or as required to translate it into languages other than English.
806 The limited permissions granted above are perpetual and will not be revoked by
807 OASIS or its successors or assigns.
808 This document and the information contained herein is provided on an "AS
809 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
810 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
811 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
812 FITNESS FOR A PARTICULAR PURPOSE.
813 -->
814 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
815 xmlns:wsa="http://www.w3.org/2005/08/addressing"
816 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
817 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200604"
818 elementFormDefault="qualified" attributeFormDefault="unqualified">
819   <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
820   schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
821   <!-- Protocol Elements -->
822   <xs:complexType name="SequenceType">
823     <xs:sequence>
824       <xs:element ref="wsrm:Identifier"/>
825       <xs:element name="MessageNumber" type="wsrm:MessageNumberType"/>
826       <xs:any namespace="##other" processContents="lax" minOccurs="0"
827 maxOccurs="unbounded"/>
828     </xs:sequence>
```

```

829     <xs:anyAttribute namespace="##other" processContents="lax"/>
830 </xs:complexType>
831 <xs:element name="Sequence" type="wsrm:SequenceType"/>
832 <xs:element name="SequenceAcknowledgement">
833     <xs:complexType>
834         <xs:sequence>
835             <xs:element ref="wsrm:Identifier"/>
836             <xs:choice>
837                 <xs:sequence>
838                     <xs:choice>
839                         <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
840                             <xs:complexType>
841                                 <xs:sequence/>
842                                 <xs:attribute name="Upper" type="xs:unsignedLong"
843 use="required"/>
844                                 <xs:attribute name="Lower" type="xs:unsignedLong"
845 use="required"/>
846                                 <xs:anyAttribute namespace="##other" processContents="lax"/>
847                             </xs:complexType>
848                         </xs:element>
849                         <xs:element name="None" minOccurs="0">
850                             <xs:complexType>
851                                 <xs:sequence/>
852                             </xs:complexType>
853                         </xs:element>
854                     </xs:choice>
855                     <xs:element name="Final" minOccurs="0">
856                         <xs:complexType>
857                             <xs:sequence/>
858                         </xs:complexType>
859                     </xs:element>
860                 </xs:sequence>
861                 <xs:element name="Nack" type="xs:unsignedLong"
862 maxOccurs="unbounded"/>
863             </xs:choice>
864             <xs:any namespace="##other" processContents="lax" minOccurs="0"
865 maxOccurs="unbounded"/>
866         </xs:sequence>
867         <xs:anyAttribute namespace="##other" processContents="lax"/>
868     </xs:complexType>
869 </xs:element>
870 <xs:complexType name="AckRequestedType">
871     <xs:sequence>
872         <xs:element ref="wsrm:Identifier"/>
873         <xs:any namespace="##other" processContents="lax" minOccurs="0"
874 maxOccurs="unbounded"/>
875     </xs:sequence>
876     <xs:anyAttribute namespace="##other" processContents="lax"/>
877 </xs:complexType>
878 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
879 <xs:element name="Identifier">
880     <xs:complexType>
881         <xs:annotation>
882             <xs:documentation>
883                 This type is for elements whose [children] is an anyURI and can have
884 arbitrary attributes.
885             </xs:documentation>
886         </xs:annotation>
887         <xs:simpleContent>
888             <xs:extension base="xs:anyURI">
889                 <xs:anyAttribute namespace="##other" processContents="lax"/>
890             </xs:extension>

```

```

891     </xs:simpleContent>
892   </xs:complexType>
893 </xs:element>
894 <xs:simpleType name="MessageNumberType">
895   <xs:restriction base="xs:unsignedLong">
896     <xs:minInclusive value="1"/>
897     <xs:maxInclusive value="9223372036854775807"/>
898   </xs:restriction>
899 </xs:simpleType>
900 <!-- Fault Container and Codes -->
901 <xs:simpleType name="FaultCodes">
902   <xs:restriction base="xs:QName">
903     <xs:enumeration value="wsrm:SequenceTerminated"/>
904     <xs:enumeration value="wsrm:UnknownSequence"/>
905     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
906     <xs:enumeration value="wsrm:MessageNumberRollover"/>
907     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
908     <xs:enumeration value="wsrm:SequenceClosed"/>
909     <xs:enumeration value="wsrm:WSRMRequired"/>
910   </xs:restriction>
911 </xs:simpleType>
912 <xs:complexType name="SequenceFaultType">
913   <xs:sequence>
914     <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
915     <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
916     <xs:any namespace="##other" processContents="lax" minOccurs="0"
917 maxOccurs="unbounded"/>
918   </xs:sequence>
919   <xs:anyAttribute namespace="##other" processContents="lax"/>
920 </xs:complexType>
921 <xs:complexType name="DetailType">
922   <xs:sequence>
923     <xs:any namespace="##other" processContents="lax" minOccurs="0"
924 maxOccurs="unbounded"/>
925   </xs:sequence>
926   <xs:anyAttribute namespace="##other" processContents="lax"/>
927 </xs:complexType>
928 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
929 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
930 <xs:element name="CreateSequenceResponse"
931 type="wsrm:CreateSequenceResponseType"/>
932 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
933 <xs:element name="CloseSequenceResponse"
934 type="wsrm:CloseSequenceResponseType"/>
935 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
936 <xs:element name="TerminateSequenceResponse"
937 type="wsrm:TerminateSequenceResponseType"/>
938 <xs:complexType name="CreateSequenceType">
939   <xs:sequence>
940     <xs:element ref="wsrm:AcksTo"/>
941     <xs:element ref="wsrm:Expires" minOccurs="0"/>
942     <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
943     <xs:any namespace="##other" processContents="lax" minOccurs="0"
944 maxOccurs="unbounded">
945       <xs:annotation>
946         <xs:documentation>
947           It is the authors intent that this extensibility be used to
948 transfer a Security Token Reference as defined in WS-Security.
949         </xs:documentation>
950       </xs:annotation>
951     </xs:any>
952   </xs:sequence>

```



```

953     <xs:anyAttribute namespace="##other" processContents="lax"/>
954 </xs:complexType>
955 <xs:complexType name="CreateSequenceResponseType">
956   <xs:sequence>
957     <xs:element ref="wsrm:Identifier"/>
958     <xs:element ref="wsrm:Expires" minOccurs="0"/>
959     <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>
960     <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
961     <xs:any namespace="##other" processContents="lax" minOccurs="0"
962 maxOccurs="unbounded"/>
963   </xs:sequence>
964   <xs:anyAttribute namespace="##other" processContents="lax"/>
965 </xs:complexType>
966 <xs:complexType name="CloseSequenceType">
967   <xs:sequence>
968     <xs:element ref="wsrm:Identifier"/>
969     <xs:any namespace="##other" processContents="lax" minOccurs="0"
970 maxOccurs="unbounded"/>
971   </xs:sequence>
972   <xs:anyAttribute namespace="##other" processContents="lax"/>
973 </xs:complexType>
974 <xs:complexType name="CloseSequenceResponseType">
975   <xs:sequence>
976     <xs:element ref="wsrm:Identifier"/>
977     <xs:any namespace="##other" processContents="lax" minOccurs="0"
978 maxOccurs="unbounded"/>
979   </xs:sequence>
980   <xs:anyAttribute namespace="##other" processContents="lax"/>
981 </xs:complexType>
982 <xs:complexType name="TerminateSequenceType">
983   <xs:sequence>
984     <xs:element ref="wsrm:Identifier"/>
985     <xs:any namespace="##other" processContents="lax" minOccurs="0"
986 maxOccurs="unbounded"/>
987   </xs:sequence>
988   <xs:anyAttribute namespace="##other" processContents="lax"/>
989 </xs:complexType>
990 <xs:complexType name="TerminateSequenceResponseType">
991   <xs:sequence>
992     <xs:element ref="wsrm:Identifier"/>
993     <xs:any namespace="##other" processContents="lax" minOccurs="0"
994 maxOccurs="unbounded"/>
995   </xs:sequence>
996   <xs:anyAttribute namespace="##other" processContents="lax"/>
997 </xs:complexType>
998 <xs:element

```

```

999     name="AcksTo" type="wsa:EndpointReferenceType"/>
1000     <xs:complexType name="OfferType">
1001         <xs:sequence>
1002             <xs:element ref="wsrm:Identifier"/>
1003             <xs:element ref="wsrm:Expires" minOccurs="0"/>
1004             <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
1005             <xs:any namespace="##other" processContents="lax" minOccurs="0"
1006 maxOccurs="unbounded"/>
1007         </xs:sequence>
1008         <xs:anyAttribute namespace="##other" processContents="lax"/>
1009     </xs:complexType>
1010     <xs:complexType name="AcceptType">
1011         <xs:sequence>
1012             <xs:element ref="wsrm:AcksTo"/>
1013             <xs:any namespace="##other" processContents="lax" minOccurs="0"
1014 maxOccurs="unbounded"/>
1015         </xs:sequence>
1016         <xs:anyAttribute namespace="##other" processContents="lax"/>
1017     </xs:complexType>
1018     <xs:element name="Expires">
1019         <xs:complexType>
1020             <xs:simpleContent>
1021                 <xs:extension base="xs:duration">
1022                     <xs:anyAttribute namespace="##other" processContents="lax"/>
1023                 </xs:extension>
1024             </xs:simpleContent>
1025         </xs:complexType>
1026     </xs:element>
1027     <xs:element name="AcknowledgementInterval">
1028         <xs:complexType>
1029             <xs:sequence/>
1030             <xs:attribute name="Milliseconds" type="xs:unsignedLong"
1031 use="required"/>
1032             <xs:anyAttribute namespace="##other" processContents="lax"/>
1033         </xs:complexType>
1034     </xs:element>
1035 </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:wsm="http://docs.oasis-open.org/ws-rx/wsm/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/wsm/200604/CreateSequence</wsa:Action>
    <wsa:ReplyTo>
      <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
    </wsa:ReplyTo>
  </S:Header>
  <S:Body>
    <wsm:CreateSequence>
      <wsm:AcksTo>
        <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
      </wsm:AcksTo>
    </wsm: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:wsm="http://docs.oasis-open.org/ws-rx/wsm/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/wsm/200604/CreateSequenceResponse
    </wsa:Action>
  </S:Header>
  <S:Body>
    <wsm:CreateSequenceResponse>
      <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
    </wsm: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:

1086 Message 1

```
1087 <?xml version="1.0" encoding="UTF-8"?>
1088 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1089 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
1090 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1091   <S:Header>
1092     <wsa:MessageID>
1093       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
1094     </wsa:MessageID>
1095     <wsa:To>http://example.com/serviceB/123</wsa:To>
1096     <wsa:From>
1097       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1098     </wsa:From>
1099     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1100     <wsmr:Sequence>
1101       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1102       <wsmr:MessageNumber>1</wsmr:MessageNumber>
1103     </wsmr:Sequence>
1104   </S:Header>
1105   <S:Body>
1106     <!-- Some Application Data -->
1107   </S:Body>
1108 </S:Envelope>
```

1109 Message 2

```
1110 <?xml version="1.0" encoding="UTF-8"?>
1111 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1112 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
1113 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1114   <S:Header>
1115     <wsa:MessageID>
1116       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1117     </wsa:MessageID>
1118     <wsa:To>http://example.com/serviceB/123</wsa:To>
1119     <wsa:From>
1120       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1121     </wsa:From>
1122     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1123     <wsmr:Sequence>
1124       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1125       <wsmr:MessageNumber>2</wsmr:MessageNumber>
1126     </wsmr:Sequence>
1127   </S:Header>
1128   <S:Body>
1129     <!-- Some Application Data -->
1130   </S:Body>
1131 </S:Envelope>
```

1132 Message 3

```
1133 <?xml version="1.0" encoding="UTF-8"?>
1134 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1135 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
1136 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1137   <S:Header>
1138     <wsa:MessageID>
1139       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1140     </wsa:MessageID>
1141     <wsa:To>http://example.com/serviceB/123</wsa:To>
1142     <wsa:From>
1143       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
```

```

1144     </wsa:From>
1145     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1146     <wsrm:Sequence>
1147       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1148       <wsrm:MessageNumber>3</wsrm:MessageNumber>
1149     </wsrm:Sequence>
1150     <wsrm:AckRequested>
1151       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1152     </wsrm:AckRequested>
1153   </S:Header>
1154   <S:Body>
1155     <!-- Some Application Data -->
1156   </S:Body>
1157 </S:Envelope>

```

1158 B.3 First Acknowledgement

1159 Message number 2 has not been received by the RM Destination due to some transmission error so it
 1160 responds with an acknowledgement for messages 1 and 3:

```

1161 <?xml version="1.0" encoding="UTF-8"?>
1162 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1163   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
1164   xmlns:wsa="http://www.w3.org/2005/08/addressing">
1165   <S:Header>
1166     <wsa:MessageID>
1167       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1168     </wsa:MessageID>
1169     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1170     <wsa:From>
1171       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1172     </wsa:From>
1173     <wsa:Action>
1174       http://docs.oasis-open.org/ws-rx/wsrn/200604/SequenceAcknowledgement
1175     </wsa:Action>
1176     <wsrm:SequenceAcknowledgement>
1177       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1178       <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
1179       <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
1180     </wsrm:SequenceAcknowledgement>
1181   </S:Header>
1182   <S:Body/>
1183 </S:Envelope>

```

1184 B.4 Retransmission

1185 The RM Sourcediscovers that message number 2 was not received so it resends the message and
 1186 requests an acknowledgement:

```

1187 <?xml version="1.0" encoding="UTF-8"?>
1188 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1189   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
1190   xmlns:wsa="http://www.w3.org/2005/08/addressing">
1191   <S:Header>
1192     <wsa:MessageID>
1193       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1194     </wsa:MessageID>
1195     <wsa:To>http://example.com/serviceB/123</wsa:To>
1196     <wsa:From>
1197       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1198     </wsa:From>

```

```

1199 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1200 <wsrm:Sequence>
1201   <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1202   <wsrm:MessageNumber>2</wsrm:MessageNumber>
1203 </wsrm:Sequence>
1204 <wsrm:AckRequested>
1205   <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1206 </wsrm:AckRequested>
1207 </S:Header>
1208 <S:Body>
1209   <!-- Some Application Data -->
1210 </S:Body>
1211 </S:Envelope>

```

1212 B.5 Termination

1213 The RM Destination now responds with an acknowledgement for the complete Sequence which can then
1214 be terminated:

```

1215 <?xml version="1.0" encoding="UTF-8"?>
1216 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1217 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
1218 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1219   <S:Header>
1220     <wsa:MessageID>
1221       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1222     </wsa:MessageID>
1223     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1224     <wsa:From>
1225       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1226     </wsa:From>
1227     <wsa:Action>
1228       http://docs.oasis-open.org/ws-rx/wsr/200604/SequenceAcknowledgement
1229     </wsa:Action>
1230     <wsrm:SequenceAcknowledgement>
1231       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1232       <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
1233     </wsrm:SequenceAcknowledgement>
1234   </S:Header>
1235   <S:Body/>
1236 </S:Envelope>

```

1237 Terminate Sequence

```

1238 <?xml version="1.0" encoding="UTF-8"?>
1239 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1240 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
1241 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1242   <S:Header>
1243     <wsa:MessageID>
1244       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1245     </wsa:MessageID>
1246     <wsa:To>http://example.com/serviceB/123</wsa:To>
1247     <wsa:Action>
1248       http://docs.oasis-open.org/ws-rx/wsr/200604/TerminateSequence
1249     </wsa:Action>
1250     <wsa:From>
1251       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1252     </wsa:From>
1253   </S:Header>
1254   <S:Body>
1255     <wsrm:TerminateSequence>

```

```

1256     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1257   </wsrm:TerminateSequence>
1258 </S:Body>
1259 </S:Envelope>

```

1260 Terminate Sequence Response

```

1261 <?xml version="1.0" encoding="UTF-8"?>
1262 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1263   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
1264   xmlns:wsa="http://www.w3.org/2005/08/addressing">
1265   <S:Header>
1266     <wsa:MessageID>
1267       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
1268     </wsa:MessageID>
1269     <wsa:To>http://example.com/serviceA/789</wsa:To>
1270     <wsa:Action>
1271       http://docs.oasis-open.org/ws-rx/wsr/200604/TerminateSequenceResponse
1272     </wsa:Action>
1273     <wsa:RelatesTo>
1274       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1275     </wsa:RelatesTo>
1276     <wsa:From>
1277       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1278     </wsa:From>
1279   </S:Header>
1280   <S:Body>
1281     <wsrm:TerminateSequenceResponse>
1282       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1283     </wsrm:TerminateSequenceResponse>
1284   </S:Body>
1285 </S:Envelope>

```

C. WSDL

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

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

The following non-normative copy is provided for reference.

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

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

  <wsdl:message name="CreateSequence">
```



```

1340     <wsdl:part name="create" element="rm:CreateSequence"/>
1341 </wsdl:message>
1342 <wsdl:message name="CreateSequenceResponse">
1343     <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
1344 </wsdl:message>
1345 <wsdl:message name="CloseSequence">
1346     <wsdl:part name="close" element="rm:CloseSequence"/>
1347 </wsdl:message>
1348 <wsdl:message name="CloseSequenceResponse">
1349     <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
1350 </wsdl:message>
1351 <wsdl:message name="TerminateSequence">
1352     <wsdl:part name="terminate" element="rm:TerminateSequence"/>
1353 </wsdl:message>
1354 <wsdl:message name="TerminateSequenceResponse">
1355     <wsdl:part name="terminateResponse"
1356 element="rm:TerminateSequenceResponse"/>
1357 </wsdl:message>

1358 <wsdl:portType name="SequenceAbstractPortType">
1359     <wsdl:operation name="CreateSequence">
1360         <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
1361 open.org/ws-rx/wsrn/200604/CreateSequence"/>
1362         <wsdl:output message="tns:CreateSequenceResponse"
1363 wsa:Action="http://docs.oasis-open.org/ws-
1364 rx/wsrn/200604/CreateSequenceResponse"/>
1365     </wsdl:operation>
1366     <wsdl:operation name="CloseSequence">
1367         <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
1368 open.org/ws-rx/wsrn/200604/CloseSequence"/>
1369         <wsdl:output message="tns:CloseSequenceResponse"
1370 wsa:Action="http://docs.oasis-open.org/ws-
1371 rx/wsrn/200604/CloseSequenceResponse"/>
1372     </wsdl:operation>
1373     <wsdl:operation name="TerminateSequence">
1374         <wsdl:input message="tns:TerminateSequence"
1375 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequence"/>
1376         <wsdl:output message="tns:TerminateSequenceResponse"
1377 wsa:Action="http://docs.oasis-open.org/ws-
1378 rx/wsrn/200604/TerminateSequenceResponse"/>
1379     </wsdl:operation>
1380 </wsdl:portType>

1381 </wsdl:definitions>

```

D. State Tables

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

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

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

Table 2 RM Source State Transition Table

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

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

1386 In Table 2 above, the rows consists of events that occur at the RM Source throughout the lifetime of an
 1387 RM Sequence and the columns consists of various RM Source states. Each cell in the table above lists
 1388 the action that the RM Source takes on occurrence of a particular event and the next state that it
 1389 transitions.

1390 Table 3 RM Destination State Transition Table

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

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

1391 In Table 3 above, the rows consists of events that occur at the RM Destination throughout the lifetime of
1392 an RM Sequence and the columns consists of various RM Destination states. Each cell in the table above
1393 lists the action that the RM Destination takes on occurrence of a particular event and the next state that it
1394 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".

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.