



1 Web Services ReliableMessaging 2 (WS-Reliable Messaging)

3 Committee Draft 03, April 12, 2006 (CBF)

4 Document identifier:

5 wsrn-1.1-spec-cd-03

6 Location:

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

8 Editors:

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

14 Contributors:

15 TBD

16 Abstract:

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

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

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

31 Status:

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

Table of Contents

1	Introduction.....	4
1.1	Goals and Requirements.....	4
1.1.1	Requirements.....	4
1.2	Notational Conventions.....	4
1.3	Namespace.....	4
1.4	Compliance.....	5
2	Reliable Messaging Model.....	6
2.1	Glossary.....	6
2.2	Protocol Preconditions.....	7
2.3	Protocol Invariants.....	7
2.4	Example Message Exchange.....	7
3	RM Protocol Elements.....	10
3.1	Sequence Creation.....	10
3.2	Closing A Sequence.....	13
3.3	Sequence Termination.....	14
3.4	Sequences.....	16
3.5	Request Acknowledgement.....	17
3.6	Sequence Acknowledgement.....	18
4	Faults.....	21
4.1	SequenceFault Element.....	22
4.2	Sequence Terminated.....	23
4.3	Unknown Sequence.....	23
4.4	Invalid Acknowledgement.....	23
4.5	Message Number Rollover.....	24
4.6	Create Sequence Refused.....	24
4.7	Sequence Closed.....	24
4.8	WSRM Required.....	24
5	Security Considerations.....	26
6	References.....	28
6.1	Normative.....	28
6.2	Non-Normative.....	28
A.	Schema.....	30
B.	Message Examples.....	36
B.1	Create Sequence.....	36
B.2	Initial Transmission.....	36
B.3	First Acknowledgement.....	38
B.4	Retransmission.....	38

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

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~~transfer of messages. It defines a messaging protocol to identify, track, and manage the reliable ~~delivery~~transfer 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-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/200602>

Dereferencing the above URI will produce the Resource Directory Description Language [RDDL 2.0] 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	http://www.w3.org/2003/05/soap-envelope
S11	http://schemas.xmlsoap.org/soap/envelope/
wsm	http://docs.oasis-open.org/ws-rx/wsm/200602
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing
xs	http://www.w3.org/2001/XMLSchema

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

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

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

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

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

124 1.4 Compliance

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

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

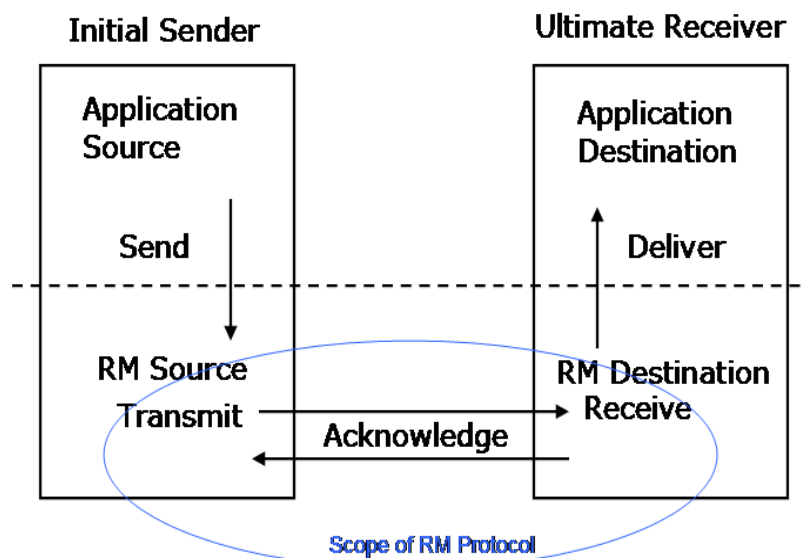
2 Reliable Messaging Model

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 regarding receipt of the messages transmitted. Note that this specification makes places 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 supportsenables the implementation of a broad range of reliability features whichthat include ordered delivery, duplicate elimination, and guaranteed receipt for the RMD. The protocol can also be implemented with a range of robustness characteristics ranging from in-memory persistence that is scoped to a single process lifetime, to replicated durable storage that is recoverable in all but the most extreme circumstances. It is expected that the AD and RMD endpoints will implement as many of these or as few of these reliability characteristics as necessary to implement the Ad for the correct operation of the application using the protocol. easeanyIn Regardless of which of these reliability features is enabled, 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 deliverytransfer. 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.



157 Figure 1: Reliable Messaging Model

158 2.1 Glossary

159 The following definitions are used throughout this specification:

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

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

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

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

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

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

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

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

173 **Send:** The act of submitting a message to the RM Source for reliable ~~deliverytransfer.~~ ~~The-reliability-~~
174 ~~guarantee-begins-at-this-point.~~

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

176 2.2 Protocol Preconditions

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

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

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

185 2.3 Protocol Invariants

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

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

2.4 Example Message Exchange

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

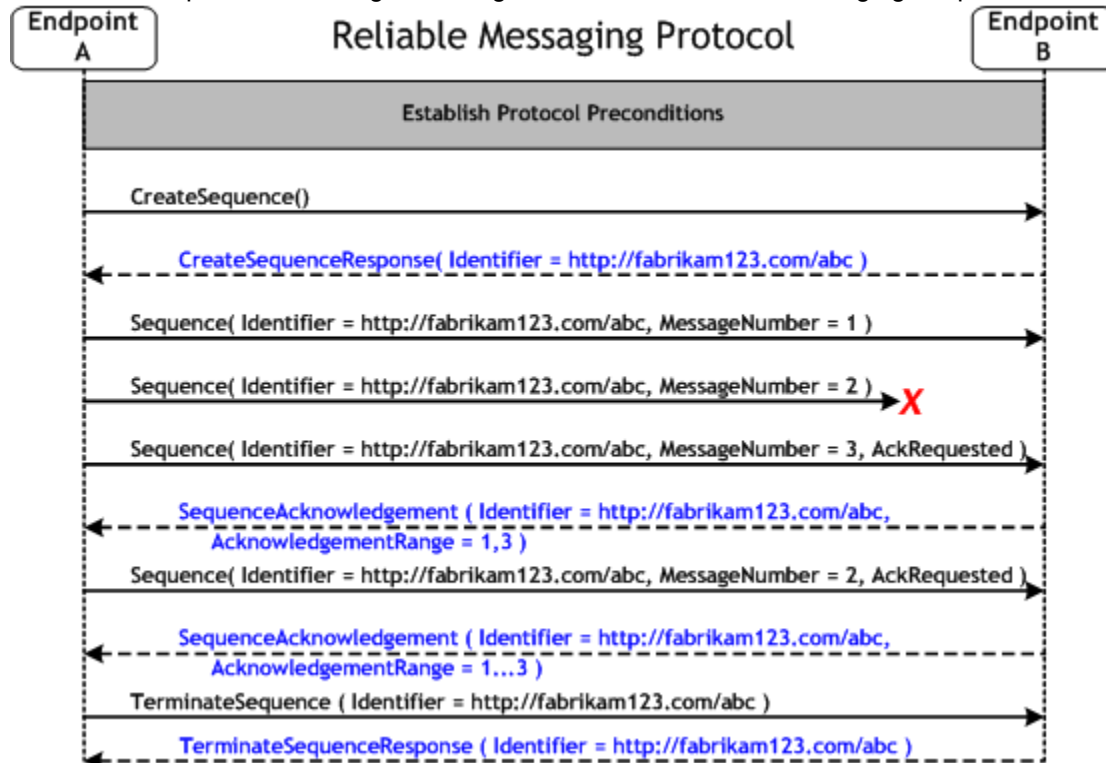


Figure 2: The WS-ReliableMessaging Protocol

1. The protocol preconditions are established. These include policy exchange, endpoint resolution, establishing trust.
2. The RM Source requests creation of a new Sequence.
3. The RM Destination creates a Sequence by returning a globally unique identifier.
4. The RM Source begins sending messages beginning with MessageNumber 1. In the figure above, the RM Source sends 3 messages.
5. Since the 3rd message is the last in this exchange, the RM Source includes a `<wsrm:AckRequested>` Header.
6. The 2nd message is lost in transit.
7. The RM Destination acknowledges receipt of message numbers 1 and 3 as a result of receiving the RM Source's `<wsrm:AckRequested>` Header.
8. The RM Source retransmits the 2nd message. This is a new message on the underlying transport, but it has the same sequence identifier and message number so the RM Destination can recognize it as equivalent to the earlier message, in case both are received.
9. The RM Source includes an `<wsrm:AckRequested>` element so the RM Destination will expedite an acknowledgement.
10. The RM Destination receives the second transmission of the message with MessageNumber 2 and acknowledges receipt of message numbers 1, 2, and 3.
11. The RM Source receives this acknowledgement and sends a `TerminateSequence` message to the RM Destination indicating that the sequence is completed and reclaims any resources associated with the Sequence.

217 12. The RM Destination receives the TerminateSequence message indicating that the RM Source will
218 not be sending any more messages. The RM Destination sends a TerminateSequenceResponse
219 message to the RM Source and and reclaims any resources associated with the Sequence.

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

231 Now that the basic model has been outlined, the details of the elements used in this protocol are now
232 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.. `<wsrm:CreateSequence>` MAY carry an offer to create an inbound sequence which is either accepted or rejected in the `<wsrm:CreateSequenceResponse>`. Note that offering a Sequence within the `<wsrm:CreateSequence>` element is simply a protocol optimization. There is no semantic difference between offering a Sequence, and choosing not to offer one and subsequently creating a new Sequence to carry messages from the RM Destination to the RM Source.

The 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: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 `<wsrm:SequenceAcknowledgement>` messages and faults related to the created Sequence are to be sent.

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'.

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

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

278 /wsrm:CreateSequence/wsrm:Offer

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

281 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier

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

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

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

287 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires

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

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

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

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

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

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

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

300 /wsrm:CreateSequence/{any}

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

303 /wsrm:CreateSequence/@{any}

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

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

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

```
311 <wsrm:CreateSequenceResponse ...>
312   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
313   <wsrm:Expires> xs:duration </wsrm:Expires> ?
314   <wsrm:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?
315   <wsrm:Accept ...>
316     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
```

```

317     ...
318     </wsrm:Accept> ?
319     ...
320 </wsrm:CreateSequenceResponse>

```

321 /wsrm:CreateSequenceResponse

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

325 /wsrm:CreateSequenceResponse/wsrm:Identifier

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

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

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

331 /wsrm:CreateSequenceResponse/wsrm:Expires

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

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

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

339 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval

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

342 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@Milliseconds

343 The acknowledgement interval, specified in milliseconds.

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

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

347 /wsrm:CreateSequenceResponse/wsrm:Accept

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

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

353 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo

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

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

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

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

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

369 3.2 Closing A Sequence

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

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

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

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

391 The following exemplar defines the CloseSequence syntax:

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

396 /wsrm:CloseSequence

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

400 /wsrm:CloseSequence/wsrm:Identifier

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

403 /wsrm:CloseSequence/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:CloseSequence/{any}

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

409 /wsrm:CloseSequence@{any}

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

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

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

```
416 <wsrm:CloseSequenceResponse ...>  
417   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
418   ...  
419 </wsrm:CloseSequenceResponse>
```

420 /wsrm:CloseSequenceResponse

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

424 /wsrm:CloseSequenceResponse/wsrm:Identifier

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

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

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

430 /wsrm:CloseSequenceResponse/{any}

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

433 /wsrm:CloseSequenceResponse@{any}

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

436 3.3 Sequence Termination

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

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

444 The following exemplar defines the TerminateSequence syntax:

```
445 <wsrm:TerminateSequence ...>  
446   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
447   ...  
448 </wsrm:TerminateSequence>
```

449 /wsrm:TerminateSequence

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

455 /wsrm:TerminateSequence/wsrm:Identifier

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

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

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

461 /wsrm:TerminateSequence/{any}

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

464 /wsrm:TerminateSequence/@{any}

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

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

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

```
471 <wsrm:TerminateSequenceResponse ...>  
472   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
473   ...  
474 </wsrm:TerminateSequenceResponse>
```

475 /wsrm:TerminateSequenceResponse

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

479 /wsrm:TerminateSequenceResponse/wsrm:Identifier

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

482 /wsrm:TerminateSequenceResponse/wsrm:Identifier/{any}

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

485 /wsrm:TerminateSequenceResponse/{any}

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

488 /wsrm:TerminateSequenceresponse/{any}

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

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

493 3.4 Sequences

494 The RM protocol uses a <wsrm:Sequence> header block to track and manage the reliable
495 **deliverytransfer** of messages. Messages for which a reliable **deliverytransfer** is required MUST contain a
496 <wsrm:Sequence> header block. Each Sequence MUST have a unique <wsrm:Identifier> element
497 and each message within a Sequence MUST have a <wsrm:MessageNumber> element that increments
498 by 1 from an initial value of 1. These values are contained within a <wsrm:Sequence> header block
499 accompanying each message being **deliveredtransferred** in the context of a Sequence.

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

501 A following exemplar defines its syntax:

```
502 <wsrm:Sequence ...>  
503   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
504   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>  
505   ...  
506 </wsrm:Sequence>
```

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

508 /wsrm:Sequence

509 This is the element containing Sequence information for WS-ReliableMessaging. The <wsrm:Sequence>
510 element MUST be understood by the RM Destination. The <wsrm:Sequence> element MUST have a
511 **mustUnderstand** attribute with a value 1/true from the namespace corresponding to the version of
512 SOAP to which the <wsrm:Sequence> SOAP header block is bound.

513 /wsrm:Sequence/wsrm:Identifier

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

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

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

519 /wsrm:Sequence/wsrm:MessageNumber

520 This REQUIRED element MUST contain a **wsrm:MessageNumberType** representing the ordinal position
521 of the message within a Sequence. Sequence MessageNumbers start at 1 and monotonically increase
522 throughout the Sequence. If the message number exceeds the internal limitations of an RM Source or RM

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

525 /wsrm:Sequence/{any}

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

528 /wsrm:Sequence/@{any}

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

531 The following example illustrates a Sequence header block.

```
532 <wsrm:Sequence>  
533   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
534   <wsrm:MessageNumber>10</wsrm:MessageNumber>  
535 </wsrm:Sequence>
```

536 3.5 Request Acknowledgement

537 The purpose of the <wsrm:AckRequested> header block is to signal to the RM Destination that the RM
538 Source is requesting that a <wsrm:SequenceAcknowledgement> be returned.

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

545 The following exemplar defines its syntax:

```
546 <wsrm:AckRequested ...>  
547   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
548   ...  
549 </wsrm:AckRequested>
```

550 /wsrm:AckRequested

551 This element requests an acknowledgement for the identified sequence.

552 /wsrm:AckRequested/wsrm:Identifier

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

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

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

558 /wsrm:AckRequested/{any}

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

561 /wsrm:AckRequested/@{any}

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

564 3.6 Sequence Acknowledgement

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

574 The following exemplar defines its syntax:

```
575 <wsrm:SequenceAcknowledgement ...>
576   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
577   [ [ <wsrm:AcknowledgementRange ...
578       Upper="wsrm:MessageNumberType"
579       Lower="wsrm:MessageNumberType"/> +
580
581       | <wsrm:None/> ]
582     <wsrm:Final/> ?
583     | <wsrm:Nack> wsrm:MessageNumberType </wsrm:Nack> + ]
584   ...
585 </wsrm:SequenceAcknowledgement>
```

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

587 `/wsrm:SequenceAcknowledgement`

588 This element contains the Sequence acknowledgement information.

589 `/wsrm:SequenceAcknowledgement/wsrm:Identifier`

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

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

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

596 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

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

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

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

605 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Lower
 606 This REQUIRED attribute contains a `wsrm:MessageNumberType` representing the
 607 `<wsrm:MessageNumber>` of the lowest contiguous message in a Sequence range received by the RM
 608 Destination.

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

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

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

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

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

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

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

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

```

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

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

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

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

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

4 Faults

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

Faults for this operation 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>
          <S:Value> [Subcode] </S:Value>
        </S:Subcode>
      </S:Code>
      <S:Reason>
        <S:Text xml:lang="en"> [Reason] </S:Text>
      </S:Reason>
    </S:Fault>
  </S:Body>
</S:Envelope>
```

```

704     <S:Detail>
705         [Detail]
706         ...
707     </S:Detail>
708 </S:Fault>
709 </S:Body>
710 </S:Envelope>

```

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

```

713 <S11:Envelope>
714   <S11:Header>
715     <wsrm:SequenceFault>
716       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
717       ...
718     </wsrm:SequenceFault>
719     <!-- Headers elided for clarity. -->
720   </S11:Header>
721   <S11:Body>
722     <S11:Fault>
723       <faultcode> [Code] </faultcode>
724       <faultstring> [Reason] </faultstring>
725     </S11:Fault>
726   </S11:Body>
727 </S11:Envelope>

```

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

```

730 <S11:Envelope>
731   <S11:Body>
732     <S11:Fault>
733       <faultcode> [Subcode] </faultcode>
734       <faultstring xml:lang="en"> [Reason] </faultstring>
735     </S11:Fault>
736   </S11:Body>
737 </S11:Envelope>

```

738 4.1 SequenceFault Element

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

743 The following exemplar defines its syntax:

```

744 <wsrm:SequenceFault ...>
745   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
746   ...
747 </wsrm:SequenceFault>

```

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

749 /wsrm:SequenceFault

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

751 /wsrm:SequenceFault/wsrm:FaultCode

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

753 /wsrm:SequenceFault/{any}

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

756 /wsrm:SequenceFault/{any}

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

759 4.2 Sequence Terminated

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

764 Properties:

765 [Code] Sender or Receiver

766 [Subcode] wsrm:SequenceTerminated

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

768 [Detail]

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

770 4.3 Unknown Sequence

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

773 Properties:

774 [Code] Sender

775 [Subcode] wsrm:UnknownSequence

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

777 [Detail]

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

779 4.4 Invalid Acknowledgement

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

783 [Code] Sender

784 [Subcode] wsrm:InvalidAcknowledgement

785 [Reason] The SequenceAcknowledgement violates the cumulative acknowledgement invariant.

786 [Detail]

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

4.5 Message Number Rollover

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

Properties:

[Code] Sender

[Subcode] wsrn:MessageNumberRollover

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

[Detail]

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

4.6 Create Sequence Refused

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

Properties:

[Code] Sender

[Subcode] wsrn:CreateSequenceRefused

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

[Detail]

```
xs:any
```

4.7 Sequence Closed

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

Properties:

[Code] Sender

[Subcode] wsrn:SequenceClosed

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

[Detail]

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

4.8 WSRM Required

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

Properties:

[Code] Sender

[Subcode] wsrn:WSRMRequired

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

[Detail]

5 Security Considerations

It is strongly recommended that the communication between services be secured using the mechanisms described in WS-Security [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.

- 866 • **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by comparing
867 secured policies – see WS-Policy and WS-SecurityPolicy).
- 868 • **Authentication** – Authentication is established using the mechanisms described in WS-Security
869 and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- 870 • **Accountability** – Accountability is a function of the type of and string of the key and algorithms
871 being used. In many cases, a strong symmetric key provides sufficient accountability. However, in
872 some environments, strong PKI signatures are required.
- 873 • **Availability** – All reliable messaging services are subject to a variety of availability attacks. Replay
874 detection is a common attack and it is recommended that this be addressed by the mechanisms
875 described in WS-Security. (Note that because of legitimate message replays, detection should
876 include a differentiator besides message id such as a timestamp). Other attacks, such as network-
877 level denial of service attacks are harder to avoid and are outside the scope of this specification.
878 That said, care should be taken to ensure that minimal state is saved prior to any authenticating
879 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]

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

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]

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

914 **[RTTM]**

915 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May
916 1992.

917 **[SecurityPolicy]**

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

919 **[SecureConversation]**

920 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," February
921 2005.

922 **[Trust]**

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

924 **A. Schema**

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

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

928 The following copy is provided for reference.

B. Message Examples

B.1 Create Sequence

Create Sequence

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

Create Sequence Response

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

B.2 Initial Transmission

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

Message 1

```

980 <?xml version="1.0" encoding="UTF-8"?>
981 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
982 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200602"
983 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
984   <S:Header>
985     <wsa:MessageID>
986       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
987     </wsa:MessageID>
988     <wsa:To>http://example.com/serviceB/123</wsa:To>
989     <wsa:From>
990       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
991     </wsa:From>
992     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
993     <wsm:Sequence>
994       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
995       <wsm:MessageNumber>1</wsm:MessageNumber>
996     </wsm:Sequence>
997   </S:Header>
998   <S:Body>
999     <!-- Some Application Data -->
1000   </S:Body>
1001 </S:Envelope>

```

1002 Message 2

```

1003 <?xml version="1.0" encoding="UTF-8"?>
1004 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1005 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200602"
1006 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1007   <S:Header>
1008     <wsa:MessageID>
1009       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1010     </wsa:MessageID>
1011     <wsa:To>http://example.com/serviceB/123</wsa:To>
1012     <wsa:From>
1013       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1014     </wsa:From>
1015     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1016     <wsm:Sequence>
1017       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1018       <wsm:MessageNumber>2</wsm:MessageNumber>
1019     </wsm:Sequence>
1020   </S:Header>
1021   <S:Body>
1022     <!-- Some Application Data -->
1023   </S:Body>
1024 </S:Envelope>

```

1025 Message 3

```

1026 <?xml version="1.0" encoding="UTF-8"?>
1027 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1028 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200602"
1029 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1030   <S:Header>
1031     <wsa:MessageID>
1032       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1033     </wsa:MessageID>
1034     <wsa:To>http://example.com/serviceB/123</wsa:To>
1035     <wsa:From>
1036       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1037     </wsa:From>
1038     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1039     <wsm:Sequence>
1040       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>

```



```

1041     <wsrm:MessageNumber>3</wsrm:MessageNumber>
1042   </wsrm:Sequence>
1043   <wsrm:AckRequested>
1044     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1045   </wsrm:AckRequested>
1046 </S:Header>
1047 <S:Body>
1048   <!-- Some Application Data -->
1049 </S:Body>
1050 </S:Envelope>

```

1051 B.3 First Acknowledgement

1052 Message number 2 has not been received by the RM Destination due to some transmission error so it
 1053 responds with an acknowledgement for messages 1 and 3:

```

1054 <?xml version="1.0" encoding="UTF-8"?>
1055 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1056   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200602"
1057   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1058   <S:Header>
1059     <wsa:MessageID>
1060       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1061     </wsa:MessageID>
1062     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1063     <wsa:From>
1064       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1065     </wsa:From>
1066     <wsa:Action>
1067       http://docs.oasis-open.org/ws-rx/wsrn/200602/SequenceAcknowledgement
1068     </wsa:Action>
1069     <wsrm:SequenceAcknowledgement>
1070       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1071       <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
1072       <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
1073     </wsrm:SequenceAcknowledgement>
1074   </S:Header>
1075   <S:Body/>
1076 </S:Envelope>

```

1077 B.4 Retransmission

1078 The RM Sourcediscovers that message number 2 was not received so it resends the message and
 1079 requests an acknowledgement:

```

1080 <?xml version="1.0" encoding="UTF-8"?>
1081 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1082   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200602"
1083   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1084   <S:Header>
1085     <wsa:MessageID>
1086       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1087     </wsa:MessageID>
1088     <wsa:To>http://example.com/serviceB/123</wsa:To>
1089     <wsa:From>
1090       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1091     </wsa:From>
1092     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1093     <wsrm:Sequence>
1094       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1095       <wsrm:MessageNumber>2</wsrm:MessageNumber>
1096     </wsrm:Sequence>

```

```

1097     <wsrm:AckRequested>
1098     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1099     </wsrm:AckRequested>
1100   </S:Header>
1101   <S:Body>
1102     <!-- Some Application Data -->
1103   </S:Body>
1104 </S:Envelope>

```

1105 B.5 Termination

1106 The RM Destination now responds with an acknowledgement for the complete sequence which can then
 1107 be terminated:

```

1108 <?xml version="1.0" encoding="UTF-8"?>
1109 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1110 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200602"
1111 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1112   <S:Header>
1113     <wsa:MessageID>
1114       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1115     </wsa:MessageID>
1116     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1117     <wsa:From>
1118       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1119     </wsa:From>
1120     <wsa:Action>
1121       http://docs.oasis-open.org/ws-rx/wsrn/200602/SequenceAcknowledgement
1122     </wsa:Action>
1123     <wsrm:SequenceAcknowledgement>
1124       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1125       <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
1126     </wsrm:SequenceAcknowledgement>
1127   </S:Header>
1128   <S:Body/>
1129 </S:Envelope>

```

1130 Terminate Sequence

```

1131 <?xml version="1.0" encoding="UTF-8"?>
1132 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1133 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200602"
1134 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1135   <S:Header>
1136     <wsa:MessageID>
1137       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1138     </wsa:MessageID>
1139     <wsa:To>http://example.com/serviceB/123</wsa:To>
1140     <wsa:Action>
1141       http://docs.oasis-open.org/ws-rx/wsrn/200602/TerminateSequence
1142     </wsa:Action>
1143     <wsa:From>
1144       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1145     </wsa:From>
1146   </S:Header>
1147   <S:Body>
1148     <wsrm:TerminateSequence>
1149       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1150     </wsrm:TerminateSequence>
1151   </S:Body>
1152 </S:Envelope>

```

1153 Terminate Sequence Response

```

1154 <?xml version="1.0" encoding="UTF-8"?>

```

```

1155 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1156 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200602"
1157 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1158   <S:Header>
1159     <wsa:MessageID>
1160       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
1161     </wsa:MessageID>
1162     <wsa:To>http://example.com/serviceA/789</wsa:To>
1163     <wsa:Action>
1164       http://docs.oasis-open.org/ws-rx/wsm/200602/TerminateSequenceResponse
1165     </wsa:Action>
1166     <wsa:RelatesTo>
1167       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1168     </wsa:RelatesTo>
1169     <wsa:From>
1170       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1171     </wsa:From>
1172   </S:Header>
1173   <S:Body>
1174     <wsm:TerminateSequenceResponse>
1175       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1176     </wsm:TerminateSequenceResponse>
1177   </S:Body>
1178 </S:Envelope>

```

C. WSDL

1179

1180 The non-normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

1181 <http://docs.oasis-open.org/ws-rx/wsrn/200602/wsd/wsrn-1.1-wsd-200602.wsd>

1182 The following non-normative copy is provided for reference.

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
Message Number Rollover Fault	N/A	N/A	Rollover	Rollover	N/A	Closed?	Ignore?	Transmit Unknown Sequence Fault Terminated
Close sequence	N/A	N/A	Transmit Close Sequence Closing	Transmit Close Sequence Closing	Transmit Close Sequence Closing	Transmit Close Sequence Closed	N/A?	N/A
Close sequence Response	N/A	N/A	N/A	N/A	Closed	Closed	Ignore?	Transmit Unknown Sequence Fault Terminated
SeqAck (final)	N/A	N/A	Closed?	Closed?	Closed?	Closed?	Ignore?	Transmit Unknown Sequence fault Terminated
Sequence Closed Fault	N/A	N/A	?	?	?	?	Ignore?	Transmit Unknown Sequence Fault Terminated
Unknown Sequence Fault	N/A	N/A	Terminated?	Terminated?	Terminated?	Terminated?	Terminated?	Ignore Terminated
Sequence Terminated Fault	N/A	Terminated?	Terminated?	Terminated?	Terminated?	Terminated?	Terminated?	Ignored Terminated
Terminate sequence	N/A	N/A	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	N/A
Terminate Sequence Response	N/A	N/A	N/A	N/A	N/A	N/A	Terminated	Terminated
Elapse Expires duration	N/A	N/A	Terminated	Terminated	Terminated	Terminated	Terminated?	N/A

1187 In Table 2 above, the rows consists of events that occur at the RM Source throughout the lifetime of an
1188 RM Sequence and the columns consists of various RM Source states. Each cell in the table above lists
1189 the action that the RM Source takes on occurrence of a particular event and the next state that it
1190 transitions.

1191 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 CloseSequenceResponse with SequenceAck (Final)</i> Close	<i>Send CloseSequenceResponse with SequenceAck Final</i> Rollover Closed	<i>Send Close Sequence Response with SeqAck+Final</i> Rollover Closed	<i>Send Close Sequence Response with SeqAck+Final</i> Closed	<i>Send Unknown Sequence Fault</i> Terminated
Close	N/A	N/A					N/A

Events	States						
	None	Connecting	Connected	Rollover	Rollover Closed	Closed	Terminated
sequence itself			Closed	Rollover Closed	Rollover Closed	Closed	
Terminate sequence	N/A	N/A	Terminated	Terminated	Terminated	Terminated	Terminated
Unknown Sequence Fault	N/A	N/A	Terminated?	Terminated?	Terminated?	Terminated?	<i>Ignore</i> Terminated
Sequence Terminated Fault	N/A	N/A	Terminated?	Terminated?	Terminated?	Terminated?	<i>Ignore</i> Terminated
Terminate sequence	N/A	N/A	Terminated	Terminated	Terminated	Terminated	N/A
Elapse Expires duration	N/A	N/A	Terminated	Terminated	Terminated	Terminated	N/A

1192 In Table 3 above, the rows consists of events that occur at the RM Destination throughout the lifetime of
 1193 an RM Sequence and the columns consists of various RM Destination states. Each cell in the table above
 1194 lists the action that the RM Destination takes on occurrence of a particular event and the next state that it
 1195 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).

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.