



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

3 Working Draft 1~~32~~, May 10, 2006

## 4 Document identifier:

5 wsrn-1.1-spec-wd-1~~32~~

## 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

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

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

# 1 Introduction

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

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

## 1.1 Goals and Requirements

### 1.1.1 Requirements

### 1.2 Notational Conventions

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

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

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

### 1.3 Namespace

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

<http://docs.oasis-open.org/ws-rx/wsrn/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	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>
S12	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>
wsm	<a href="http://docs.oasis-open.org/ws-rx/wsm/200604">http://docs.oasis-open.org/ws-rx/wsm/200604</a>
wsa	<a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

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

118 <http://docs.oasis-open.org/ws-rx/wsm/200604/wsm-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 by a system that uses the elements defined within this specification, and one is not  
121 already defined per the rules of the WS-Addressing specification [WS-Addressing], then said system  
122 MUST use an the-action IRI that MUST consist of the WS-RM namespace URI concatenated with a '/',  
123 followed by the message element name. For example:

124 <http://docs.oasis-open.org/ws-rx/wsm/200604/SequenceAcknowledgement>

## 125 1.4 Compliance

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

130 Normative text within this specification takes precedence over normative outlines, which in turn take  
131 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.

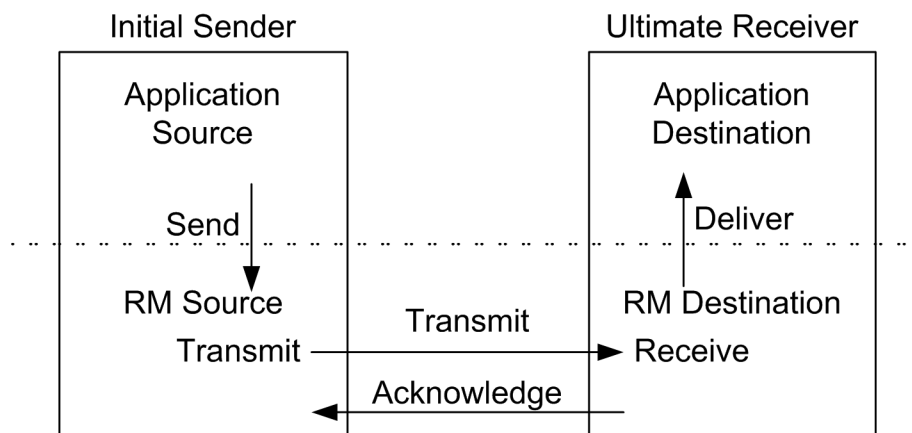


Figure 1: Reliable Messaging Model

### 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.

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

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

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

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

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

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

## 172 2.2 Protocol Preconditions

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

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

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

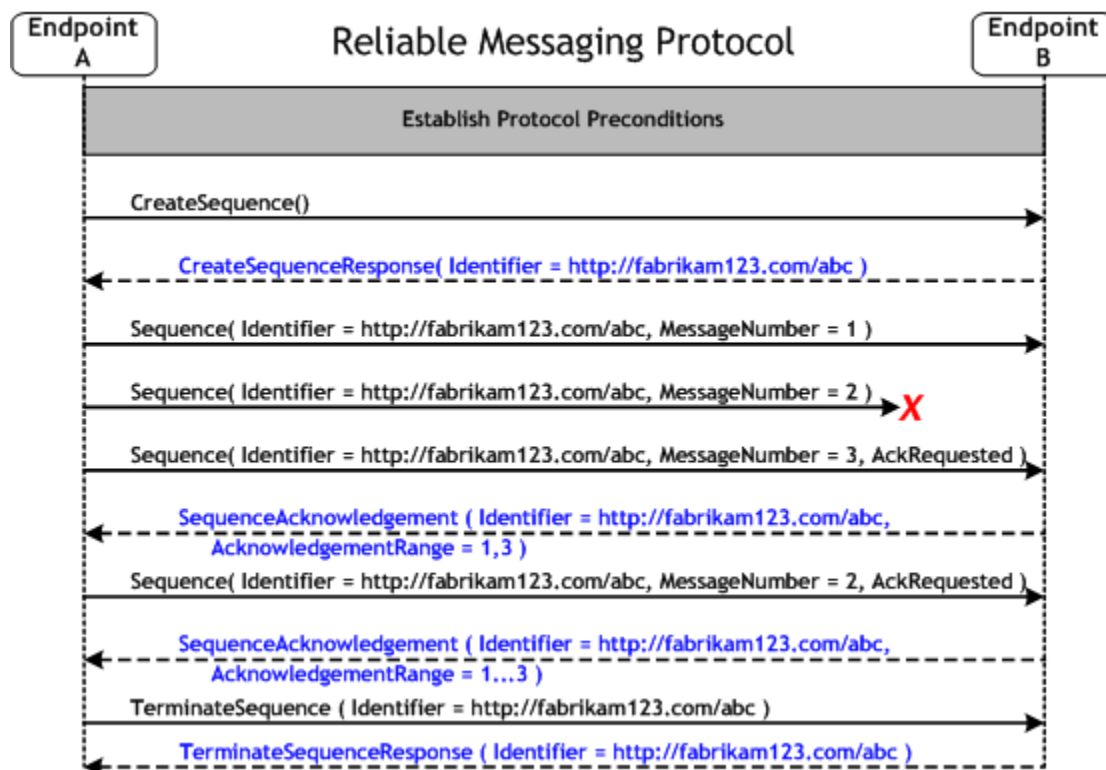
## 181 2.3 Protocol Invariants

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

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

## 191 2.4 Example Message Exchange

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



193 Figure 2: The WS-ReliableMessaging Protocol

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

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

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

## 3 RM Protocol Elements

The following protocol elements define extensibility points at various places. Implementations MAY add 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 RM Source MAY include an offer to create an inbound Sequence within the `<wsrm:CreateSequence>` message. This offer MAY carry an offer to create an inbound Sequence which is either accepted or rejected by the RM Destination in the `<wsrm:CreateSequenceResponse>` message.

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

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

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

`/wsrm:CreateSequence`

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

`/wsrm:CreateSequence/wsrm:AcksTo`

The RM Source MUST include this REQUIRED element in any `CreateSequence` message it sends. This element is of type `wsa:EndpointReferenceType` (as specified by WS-Addressing [WS-Addressing]). It specifies the endpoint reference to which messages containing all 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`

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

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

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

281 `/wsrm:CreateSequence/wsrm:Offer`

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

284 `/wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier`

285 An RM Source that includes a `<wsrm:Offer>` element within a `CreateSequence` message MUST include  
286 this element as a child of the `<wsrm:Offer>` element. This REQUIRED element The RM Source MUST  
287 set the value of this element to contain an absolute URI (conformant with RFC3986 [URI]) that will  
288 uniquely identify the offered Sequence.

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

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

292 `/wsrm:CreateSequence/wsrm:Offer/wsrm:Endpoint`

293 An RM Source that includes a `<wsrm:Offer>` element within a `CreateSequence` message MUST include  
294 this element as a child of the `<wsrm:Offer>` element. This REQUIRED element, This element, of type  
295 `wsa:EndpointReferenceType` (as specified by WS-Addressing [WSAddressing]), specifies the  
296 endpoint reference to which WS-RM protocol messages related to the offered Sequence are to be sent.

297 `/wsrm:CreateSequence/wsrm:Offer/wsrm:Expires`

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

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

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

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

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

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

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

310 `/wsrm:CreateSequence/{any}`

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

313 /wsrm:CreateSequence/@{any}

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

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

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

```
321 <wsrm:CreateSequenceResponse ...>
322   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
323   <wsrm:Expires> xs:duration </wsrm:Expires> ?
324   <wsrm:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?
325   <wsrm:Accept ...>
326     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
327     ...
328   </wsrm:Accept> ?
329   ...
330 </wsrm:CreateSequenceResponse>
```

331 /wsrm:CreateSequenceResponse

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

335 /wsrm:CreateSequenceResponse/wsrm:Identifier

336 ~~The RM Destination MUST include this~~ **REQUIRED** element **within any CreateSequenceResponse**  
337 **messages it sends. The RM Destination MUST set the value of this element to the**~~contain an~~ absolute URI  
338 ~~(conformant with RFC3986)~~ of the Sequence that has been created by the RM Destination.

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

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

342 /wsrm:CreateSequenceResponse/wsrm:Expires

343 This element, if present, of type xs:duration accepts or refines the RM Source's requested duration for  
344 the Sequence. A value of 'PT0S' indicates that the Sequence will never expire. Absence of the element  
345 indicates an implied value of 'PT0S'. ~~The RM Destination MUST set the value of this element to~~**This value-**  
346 **MUST** be equal to or less than the value requested by the RM Source in the corresponding  
347 <wsrm:CreateSequence> message.

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

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

351 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval

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

354 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@Milliseconds

355 **The acknowledgement interval, specified in milliseconds.**

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

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

359 /wsrm:CreateSequenceResponse/wsrm:Accept

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

362 Note: If a <wsrm:CreateSequenceResponse> is returned without a child <wsrm:Accept> in response  
 363 to a <wsrm:CreateSequence> that ~~did~~ contained a child <wsrm:Offer>, then the RM Source MAY  
 364 immediately reclaim any resources associated with the unused offered Sequence.-

365 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo

366 An RM Destination that includes a <wsrm:Accept> element within a CreateSequenceResponse  
 367 message MUST include this element as a child of the <wsrm:Accept> element. This ~~REQUIRED~~  
 368 element, of type wsam:EndpointReferenceType (as specified by WS-Addressing [WS-Addressing]),  
 369 specifies the endpoint reference to which <wsrm:SequenceAcknowledgement> messages related to  
 370 the accepted Sequence are to be sent.

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

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

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

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

377 /wsrm:CreateSequenceResponse/{any}

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

380 /wsrm:CreateSequenceResponse/@{any}

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

### 383 3.2 Closing A Sequence

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

389 If the RM Source wishes to close the Sequence, then it sends a <wsrm:CloseSequence> element, in  
 390 the body of a message, to the RM Destination. -This message indicates that the RM Destination MUST  
 391 NOT receive any new messages for the specified Sequence, other than those already received at the time  
 392 the <wsrm:CloseSequence> element is interpreted by the RMD. Upon receipt of this message, or  
 393 subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST  
 394 include a final <wsrm:SequenceAcknowledgement> (within which the RM Destination that MUST  
 395 include the <wsrm:Final> element) header block on any messages associated with the Sequence

396 destined to the RM Source, including the <wsrm:CloseSequenceResponse> message or on any  
397 Sequence Fault transmitted to the RMS.-

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

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

406 The following exemplar defines the CloseSequence syntax:

```
407 <wsrm:CloseSequence ...>  
408   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
409   ...  
410 </wsrm:CloseSequence>
```

411 /wsrm:CloseSequence

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

415 /wsrm:CloseSequence/wsrm:Identifier

416 The RM Source MUST include this element in any CloseSequence messages it sends. The RM Source  
417 MUST set the value of this element to the~~This REQUIRED element MUST contain an~~ absolute URI  
418 ~~(conformant with RFC3986)~~ of the Sequence that is being closed.-

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

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

422 /wsrm:CloseSequence/{any}

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

425 /wsrm:CloseSequence@{any}

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

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

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

```
432 <wsrm:CloseSequenceResponse ...>  
433   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
434   ...  
435 </wsrm:CloseSequenceResponse>
```

436 /wsrm:CloseSequenceResponse

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

440 `/wsrm:CloseSequenceResponse/wsrm:Identifier`

441 The RM Destination MUST include this element in any CloseSequenceResponse message it sends. The  
442 RM Destination MUST set the value of this element to the~~This REQUIRED element MUST contain an~~  
443 absolute URI (conformant with RFC3986) of the Sequence that is being ~~closed~~terminated.-

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

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

447 `/wsrm:CloseSequenceResponse/{any}`

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

450 `/wsrm:CloseSequenceResponse@{any}`

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

### 453 3.3 Sequence Termination

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

461 The following exemplar defines the TerminateSequence syntax:

```
462 <wsrm:TerminateSequence ...>  
463   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
464   ...  
465 </wsrm:TerminateSequence>
```

466 `/wsrm:TerminateSequence`

467 This element is sent by an RM Source to indicate it has completed its use of the Sequence. It indicates  
468 that the RM Destination can safely reclaim any resources related to the identified Sequence. The RM  
469 Source MUST NOT send T~~his element MUST NOT be sent~~ as a header block. The RM Source MAY  
470 retransmit this element. Once this element is sent, other than this element, the RM Source MUST NOT  
471 send any additional message to the RM Destination referencing this Sequence.

472 `/wsrm:TerminateSequence/wsrm:Identifier`

473 The RM Source MUST include this element in any TerminateSequence message it sends. The RM  
474 Source MUST set the value of this element to the~~This REQUIRED element MUST contain an~~ absolute  
475 URI (conformant with RFC3986) of the Sequence that is being terminated.-

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

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

479 /wsrm:TerminateSequence/{any}

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

482 /wsrm:TerminateSequence/@{any}

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

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

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

```
489 <wsrm:TerminateSequenceResponse ...>  
490   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
491   ...  
492 </wsrm:TerminateSequenceResponse>
```

493 /wsrm:TerminateSequenceResponse

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

498 /wsrm:TerminateSequenceResponse/wsrm:Identifier

499 The RM Destination MUST include this element in any TerminateSequenceResponse message it sends.  
500 The RM Destination MUST set the value of this element to the ~~This REQUIRED element MUST contain~~  
501 ~~an~~ absolute URI (conformant with RFC3986) of the Sequence that is being terminated.

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

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

505 /wsrm:TerminateSequenceResponse/{any}

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

508 /wsrm:TerminateSequenceResponse/@{any}

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

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

## 513 3.4 Sequences

514 The RM protocol uses a <wsrm:Sequence> header block to track and manage the reliable delivery of  
515 messages. The RM Source MUST include a <wsrm:Sequence> header block in all M ~~m~~ essages  
516 for which a ~~reliable delivery is required~~ MUST contain a <wsrm:Sequence> header block. The RM

517 ~~Source MUST identify Sequences with Each Sequence MUST have a~~ unique  
 518 ~~<wsrm:Identifier> elements and the RM Source MUST assign~~ each message within a Sequence  
 519 ~~MUST have a~~ `<wsrm:MessageNumber>` element that increments by 1 from an initial value of 1. These  
 520 values are contained within a `<wsrm:Sequence>` header block accompanying each message being  
 521 delivered in the context of a Sequence.

522 ~~The RM Source MUST NOT include There MUST be no~~ more than one `<wsrm:Sequence>` header block  
 523 in any message.

524 A following exemplar defines its syntax:

```

525 <wsrm:Sequence ...>
526   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
527   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>
528   ...
529 </wsrm:Sequence>
  
```

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

531 `/wsrm:Sequence`  
 532 This protocol element associates the message in which it is contained with a previously established RM  
 533 Sequence. It contains the Sequence's unique identifier and the containing message's ordinal position  
 534 within that Sequence. ~~The RM Destination MUST understand the~~ `<wsrm:Sequence>` header  
 535 ~~block element MUST be understood by the RM Destination. The RM Source MUST assign The-~~  
 536 ~~`<wsrm:Sequence>` element MUST have~~ a `mustUnderstand` attribute with a value 1/true (from the  
 537 namespace corresponding to the version of SOAP to which the `<wsrm:Sequence>` SOAP header block  
 538 is bound) to the `<wsrm:Sequence>` header block element.

539 `/wsrm:Sequence/wsrm:Identifier`  
 540 An RM Source that includes a `<wsrm:Sequence>` header block in a SOAP envelope MUST include this  
 541 element in that header block. The RM Source MUST set the value of this element to the~~This REQUIRED-~~  
 542 ~~element MUST contain an~~ absolute URI (conformant with RFC3986) that uniquely identifies the  
 543 Sequence.-

544 `/wsrm:Sequence/wsrm:Identifier/@{any}`  
 545 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
 546 element.

547 `/wsrm:Sequence/wsrm:MessageNumber`  
 548 The RM Source MUST include this element within any Sequence headers it creates. This ~~REQUIRED-~~  
 549 ~~element is of type~~MUST contain a `wsrm:MessageNumberType`. It representsing the ordinal position of  
 550 the message within a Sequence. Sequence ~~M~~message ~~N~~numbers start at 1 and monotonically increase  
 551 throughout the Sequence. If the message number exceeds the internal limitations of an RM Source or RM  
 552 Destination or reaches the maximum value of 9,223,372,036,854,775,807 the RM Source or Destination  
 553 MUST generate a `MessageNumberRollover` fault.

554 `/wsrm:Sequence/{any}`  
 555 This is an extensibility mechanism to allow different types of information, based on a schema, to be  
 556 passed.

557 `/wsrm:Sequence/@{any}`  
 558 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
 559 element.

560 The following example illustrates a Sequence header block.

```
561 <wsrm:Sequence>
562   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
563   <wsrm:MessageNumber>10</wsrm:MessageNumber>
564 </wsrm:Sequence>
```

### 565 3.5 Request Acknowledgement

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

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

577 The following exemplar defines its syntax:

```
578 <wsrm:AckRequested ...>
579   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
580   ...
581 </wsrm:AckRequested>
```

582 /wsrm:AckRequested

583 This element requests an acknowledgement for the identified Sequence.

584 /wsrm:AckRequested/wsrm:Identifier

585 An RM Source that includes a `<wsrm:AckRequested>` header block in a SOAP envelope MUST include  
586 this element in that header block. The RM Source MUST set the value of this element to the  
587 REQUIRED element MUST contain an absolute URI, (conformant with RFC3986), that uniquely identifies  
588 the Sequence to which the request applies.

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

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

592 /wsrm:AckRequested/{any}

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

595 /wsrm:AckRequested/@{any}

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

## 3.6 Sequence Acknowledgement

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

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

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

The following exemplar defines its syntax:

```
<wsrm:SequenceAcknowledgement ...>
  <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
  [ [ [ <wsrm:AcknowledgementRange ...
        Upper="wsrm:MessageNumberType"
        Lower="wsrm:MessageNumberType"/> +

        | <wsrm:None/> ]
        <wsrm:Final/> ? ]
    | <wsrm:Nack> wsrm:MessageNumberType </wsrm:Nack> + ]
  ...
</wsrm:SequenceAcknowledgement>
```

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

`/wsrm:SequenceAcknowledgement`

This element contains the Sequence acknowledgement information.

`/wsrm:SequenceAcknowledgement/wsrm:Identifier`

~~An RM Destination that includes a~~ `<wsrm:SequenceAcknowledgement>` header block in a SOAP envelope ~~MUST include this REQUIRED element in that header block~~. The RM Destination ~~MUST set the value of this element to the~~ ~~contain an~~ absolute URI (conformant with RFC3986) that uniquely identifies the Sequence. ~~The RM Destination A message MUST NOT include~~ ~~contain~~ multiple `<wsrm:SequenceAcknowledgement>` header blocks that share the same value for `<wsrm:Identifier>` within the same SOAP envelope.

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

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

644 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange

645 The RM Destination MAY include one or more instances of this element within a  
646 <wsrm:SequenceAcknowledgement> header block. ~~This OPTIONAL element, if present, can occur 1-~~  
647 ~~or more times.~~ It contains a range of Sequence MessageNumbers successfully received by the RM  
648 Destination. The ranges SHOULD NOT overlap. The RM Destination MUST NOT include t~~This element~~  
649 ~~MUST NOT be present~~ if a sibling <wsrm:Nack> or <wsrm:None> element is also present as a child of  
650 <wsrm:SequenceAcknowledgement>.

651 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Upper

652 An RM Destination that includes a <wsrm:SequenceAcknowledgement> header block containing one  
653 or more <wsrm:AcknowledgementRange> elements MUST include this ~~This REQUIRED~~ attribute as a  
654 child of the <wsrm:AcknowledgementRange>. The RM Destination MUST set the value of this attribute  
655 to contains a wsrm:MessageNumberType representing the <wsrm:MessageNumber> of the highest  
656 contiguous message in a Sequence range received by the RM Destination.

657 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Lower

658 An RM Destination that includes a <wsrm:SequenceAcknowledgement> header block containing one  
659 or more <wsrm:AcknowledgementRange> elements MUST include t~~This REQUIRED~~ attribute as a  
660 child of the <wsrm:AcknowledgementRange>. The RM Destination MUST set the value of this attribute  
661 to contains a wsrm:MessageNumberType representing the <wsrm:MessageNumber> of the lowest  
662 contiguous message in a Sequence range received by the RM Destination.

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

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

666 /wsrm:SequenceAcknowledgement/wsrm:Final

667 The RM Destination MAY include this element within a <wsrm:SequenceAcknowledgement> header  
668 block. ~~This OPTIONAL element, if present,~~ indicates that the RM Destination is not receiving new  
669 messages for the specified Sequence. The RM Source can be assured that the ranges of messages  
670 acknowledged by this SequenceAcknowledgement header block will not change in the future. The RM  
671 Destination~~This element~~ MUST include this element~~be present~~ when the Sequence is closed. Note: the  
672 RM Destination~~this element~~ MUST NOT include this element~~be used~~ when sending a Nack; it can only be  
673 used when sending <wsrm:AcknowledgementRange>~~AcknowledgementRange~~s or <wsrm:None>.

674 /wsrm:SequenceAcknowledgement/wsrm:Nack

675 The RM Destination MAY include this element within a <wsrm:SequenceAcknowledgement> header  
676 block. If used, the RM Destination MUST set the value of ~~t~~this ~~OPTIONAL element, if present, MUST~~  
677 ~~contain to~~ a wsrm:MessageNumberType representing the <wsrm:MessageNumber> of an unreceived  
678 message in a Sequence. The RM Destination MUST NOT include a~~The~~ <wsrm:Nack> element ~~MUST~~  
679 ~~NOT be present~~ if a sibling <wsrm:AcknowledgementRange> or <wsrm:None> element is also  
680 present as a child of <wsrm:SequenceAcknowledgement>. Upon the receipt of a Nack, an RM Source  
681 SHOULD retransmit the message identified by the Nack. The RM Destination MUST NOT issue a  
682 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that it has previously  
683 acknowledged within a <wsrm:AcknowledgementRange>. The RM Source SHOULD ignore a  
684 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that has previously  
685 been acknowledged within a <wsrm:AcknowledgementRange>.

686 /wsrm:SequenceAcknowledgement/wsrm:None

687 The RM Destination MUST include this element within a <wsrm:SequenceAcknowledgement> header  
688 block if~~This OPTIONAL element, if present, MUST be used when~~ the RM Destination has not received  
689 any messages for the specified Sequence. The RM Destination~~<wsrm:None> element~~ MUST NOT  
690 include this element~~be present~~ if a sibling <wsrm:AcknowledgementRange> or <wsrm:Nack> element  
691 is also present as a child of the <wsrm:SequenceAcknowledgement>.

692 /wsrm:SequenceAcknowledgement/{any}

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

695 /wsrm:SequenceAcknowledgement/@{any}

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

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

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

```
700 <wsrm:SequenceAcknowledgement>  
701   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
702   <wsrm:AcknowledgementRange Upper="10" Lower="1"/>  
703 </wsrm:SequenceAcknowledgement>
```

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

```
706 <wsrm:SequenceAcknowledgement>  
707   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
708   <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
709   <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
710   <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
711 </wsrm:SequenceAcknowledgement>
```

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

```
713 <wsrm:SequenceAcknowledgement>  
714   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
715   <wsrm:Nack>3</wsrm:Nack>  
716 </wsrm:SequenceAcknowledgement>
```

## 4 Faults

The faults defined in this section fall into one of two categories; those faults that are the result of messages or operations within a specific Sequence and those faults that are not. By their nature the CreateSequenceRefused, UnknownSequence, and WSRMRequired faults cannot be correlated with a Sequence. All other faults defined in this section relate to the processing of WS-RM protocol messages or messages containing WS-RM header blocks targeted at a specific Sequence and are collectively referred to as "Sequence faults".

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

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

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

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

The definitions of faults use the following properties:

[Code] The fault code.

[Subcode] The fault subcode.

[Reason] The English language reason element.

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

**Entities that generate WS-ReliableMessaging faults MUST set** the [Code] property **to 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>
```

```

730     <S:Value> [Subcode] </S:Value>
730     </S:Subcode>
730   </S:Code>
730   <S:Reason>
730     <S:Text xml:lang="en"> [Reason] </S:Text>
730   </S:Reason>
730   <S:Detail>
730     [Detail]
730     ...
730   </S:Detail>
730 </S:Fault>
730 </S:Body>
730 </S:Envelope>

```

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

```

730 <S11:Envelope>
730   <S11:Header>
730     <wsrm:SequenceFault>
730       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
730       <wsrm:Detail> [Detail] </wsrm:Detail>
730       ...
730     </wsrm:SequenceFault>
730     <!-- Headers elided for clarity. -->
730   </S11:Header>
730   <S11:Body>
730     <S11:Fault>
730       <faultcode> [Code] </faultcode>
730       <faultstring> [Reason] </faultstring>
730     </S11:Fault>
730   </S11:Body>
730 </S11:Envelope>

```

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

```

730 <S11:Envelope>
730   <S11:Body>
730     <S11:Fault>
730       <faultcode> [Subcode] </faultcode>
730       <faultstring> [Reason] </faultstring>
730     </S11:Fault>
730   </S11:Body>
730 </S11:Envelope>

```

## 730 4.1 SequenceFault Element

730 The purpose of the <wsrm:SequenceFault> element is to carry the specific details of a fault generated  
731 during the reliable messaging specific processing of a message belonging to a Sequence. WS-  
732 ReliableMessaging nodes MUST only use the <wsrm:SequenceFault> container MUST only be used  
733 in conjunction with the SOAP 1.1 fault mechanism. WS-ReliableMessaging nodes MUST NOT use the  
734 the <wsrm:SequenceFault> container be used in conjunction with the SOAP 1.2 binding.

735 The following exemplar defines its syntax:

```

735 <wsrm:SequenceFault ...>
735   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
735   <wsrm:Detail> ... </wsrm:Detail> ?
735   ...

```


735 `</wsrm:SequenceFault>`

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

735 `/wsrm:SequenceFault`

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

735 `/wsrm:SequenceFault/wsrm:FaultCode`

735 WS-ReliableMessaging nodes that generate a `<wsrm:SequenceFault>` MUST set the value of `F` this  
736 element, if present, MUST contain to a qualified name from the set of fault [Subcodes] defined below. 

737 `/wsrm:SequenceFault/wsrm:Detail`

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

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

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

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

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

737 `/wsrm:SequenceFault/{any}`

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

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

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

## 737 4.2 Sequence Terminated

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

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

737 Properties:

737 [Code] Sender or Receiver

737 [Subcode] `wsrm:SequenceTerminated`

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

737 [Detail]

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

### 4.3 Unknown Sequence

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

Properties:

[Code] Sender

[Subcode] wsrn:UnknownSequence

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

[Detail]

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

### 4.4 Invalid Acknowledgement

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

[Code] Sender

[Subcode] wsrn:InvalidAcknowledgement

[Reason] The SequenceAcknowledgement violates the cumulative acknowledgement invariant.

[Detail]

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

### 4.5 Message Number Rollover

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

Properties:

[Code] Sender

[Subcode] wsrn:MessageNumberRollover

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

[Detail]

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

### 4.6 Create Sequence Refused

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

Properties:

[Code] Sender

[Subcode] wsrn:CreateSequenceRefused

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

740 [Detail]

740 `xs:any`

## 740 **4.7 Sequence Closed**

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

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

740 Properties:

740 [Code] Sender

740 [Subcode] wsrn:SequenceClosed

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

740 [Detail]

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

## 740 **4.8 WSRM Required**

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

740 Properties:

740 [Code] Sender

740 [Subcode] wsrn:WSRMRequired

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

740 [Detail]

740 `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.

- 747 • **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by comparing  
748 secured policies – see WS-Policy and WS-SecurityPolicy).
- 749 • **Authentication** – Authentication is established using the mechanisms described in WS-Security  
750 and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- 751 • **Accountability** – Accountability is a function of the type of and string of the key and algorithms  
752 being used. In many cases, a strong symmetric key provides sufficient accountability. However, in  
753 some environments, strong PKI signatures are required.
- 754 • **Availability** – All reliable messaging services are subject to a variety of availability attacks. Replay  
755 detection is a common attack and it is recommended that this be addressed by the mechanisms  
756 described in WS-Security. (Note that because of legitimate message replays, detection should  
757 include a differentiator besides message id such as a timestamp). Other attacks, such as network-  
758 level denial of service attacks are harder to avoid and are outside the scope of this specification.  
759 That said, care should be taken to ensure that minimal state is saved prior to any authenticating  
760 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]

761 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "[OASIS Web Services Security:  
762 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.

761 **[RTTM]**

761 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May  
762 1992.

761 **[SecurityPolicy]**

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

761 **[SecureConversation]**

761 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," February  
762 2005.

761 **[Trust]**

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

## 761 **A. Schema**

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

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

761 The following copy is provided for reference.

```

761 <?xml version="1.0" encoding="UTF-8"?>
762 <!--
763 OASIS takes no position regarding the validity or scope of any intellectual
764 property or other rights that might be claimed to pertain to the
765 implementation or use of the technology described in this document or the
766 extent to which any license under such rights might or might not be available;
767 neither does it represent that it has made any effort to identify any such
768 rights. Information on OASIS's procedures with respect to rights in OASIS
769 specifications can be found at the OASIS website. Copies of claims of rights
770 made available for publication and any assurances of licenses to be made
771 available, or the result of an attempt made to obtain a general license or
772 permission for the use of such proprietary rights by implementors or users of
773 this specification, can be obtained from the OASIS Executive Director.
774 OASIS invites any interested party to bring to its attention any copyrights,
775 patents or patent applications, or other proprietary rights which may cover
776 technology that may be required to implement this specification. Please
777 address the information to the OASIS Executive Director.
778 Copyright © OASIS Open 2002-2006. All Rights Reserved.
779 This document and translations of it may be copied and furnished to others,
780 and derivative works that comment on or otherwise explain it or assist in its
781 implementation may be prepared, copied, published and distributed, in whole or
782 in part, without restriction of any kind, provided that the above copyright
783 notice and this paragraph are included on all such copies and derivative
784 works. However, this document itself does not be modified in any way, such as
785 by removing the copyright notice or references to OASIS, except as needed for
786 the purpose of developing OASIS specifications, in which case the procedures
787 for copyrights defined in the OASIS Intellectual Property Rights document must
788 be followed, or as required to translate it into languages other than English.
789 The limited permissions granted above are perpetual and will not be revoked by
790 OASIS or its successors or assigns.
791 This document and the information contained herein is provided on an "AS
792 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
793 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
794 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
795 FITNESS FOR A PARTICULAR PURPOSE.
796 -->
797 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
798 xmlns:wsa="http://www.w3.org/2005/08/addressing"
799 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200604"
800 targetNamespace="http://docs.oasis-open.org/ws-rx/wsm/200604"
801 elementFormDefault="qualified" attributeFormDefault="unqualified">
802   <xs:import namespace="http://www.w3.org/2005/08/addressing"
803   schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
804   <!-- Protocol Elements -->
805   <xs:complexType name="SequenceType">
806     <xs:sequence>
807       <xs:element ref="wsm:Identifier"/>
808       <xs:element name="MessageNumber" type="wsm:MessageNumberType"/>
809       <xs:any namespace="##other" processContents="lax" minOccurs="0"
810 maxOccurs="unbounded"/>
811     </xs:sequence>
812     <xs:anyAttribute namespace="##other" processContents="lax"/>
813   </xs:complexType>
814   <xs:element name="Sequence" type="wsm:SequenceType"/>
815   <xs:element name="SequenceAcknowledgement">
816     <xs:complexType>
817       <xs:sequence>
818         <xs:element ref="wsm:Identifier"/>
819         <xs:choice>
820           <xs:sequence>
821             <xs:choice>
822               <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
823                 <xs:complexType>

```

```

761         <xs:sequence/>
762         <xs:attribute name="Upper" type="xs:unsignedLong"
763 use="required"/>
764         <xs:attribute name="Lower" type="xs:unsignedLong"
765 use="required"/>
766         <xs:anyAttribute namespace="##other" processContents="lax"/>
767     </xs:complexType>
768 </xs:element>
769     <xs:element name="None" minOccurs="0">
770         <xs:complexType>
771             <xs:sequence/>
772         </xs:complexType>
773     </xs:element>
774 </xs:choice>
775     <xs:element name="Final" minOccurs="0">
776         <xs:complexType>
777             <xs:sequence/>
778         </xs:complexType>
779     </xs:element>
780 </xs:sequence>
781     <xs:element name="Nack" type="xs:unsignedLong"
782 maxOccurs="unbounded"/>
783 </xs:choice>
784     <xs:any namespace="##other" processContents="lax" minOccurs="0"
785 maxOccurs="unbounded"/>
786 </xs:sequence>
787     <xs:anyAttribute namespace="##other" processContents="lax"/>
788 </xs:complexType>
789 </xs:element>
790 <xs:complexType name="AckRequestedType">
791     <xs:sequence>
792         <xs:element ref="wsrm:Identifier"/>
793         <xs:any namespace="##other" processContents="lax" minOccurs="0"
794 maxOccurs="unbounded"/>
795     </xs:sequence>
796     <xs:anyAttribute namespace="##other" processContents="lax"/>
797 </xs:complexType>
798 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
799 <xs:element name="Identifier">
800     <xs:complexType>
801         <xs:annotation>
802             <xs:documentation>
803                 This type is for elements whose [children] is an anyURI and can have
804 arbitrary attributes.
805             </xs:documentation>
806         </xs:annotation>
807         <xs:simpleContent>
808             <xs:extension base="xs:anyURI">
809                 <xs:anyAttribute namespace="##other" processContents="lax"/>
810             </xs:extension>
811         </xs:simpleContent>
812     </xs:complexType>
813 </xs:element>
814 <xs:simpleType name="MessageNumberType">
815     <xs:restriction base="xs:unsignedLong">
816         <xs:minInclusive value="1"/>
817         <xs:maxInclusive value="9223372036854775807"/>
818     </xs:restriction>
819 </xs:simpleType>
820 <!-- Fault Container and Codes -->
821 <xs:simpleType name="FaultCodes">
822     <xs:restriction base="xs:QName">
823         <xs:enumeration value="wsrm:SequenceTerminated"/>

```

```

761     <xs:enumeration value="wsrm:UnknownSequence"/>
762     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
763     <xs:enumeration value="wsrm:MessageNumberRollover"/>
764     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
765     <xs:enumeration value="wsrm:SequenceClosed"/>
766     <xs:enumeration value="wsrm:WSRMRequired"/>
767   </xs:restriction>
768 </xs:simpleType>
769 <xs:complexType name="SequenceFaultType">
770   <xs:sequence>
771     <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
772     <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
773     <xs:any namespace="##other" processContents="lax" minOccurs="0"
774 maxOccurs="unbounded"/>
775   </xs:sequence>
776   <xs:anyAttribute namespace="##other" processContents="lax"/>
777 </xs:complexType>
778 <xs:complexType name="DetailType">
779   <xs:sequence>
780     <xs:any namespace="##other" processContents="lax" minOccurs="0"
781 maxOccurs="unbounded"/>
782   </xs:sequence>
783   <xs:anyAttribute namespace="##other" processContents="lax"/>
784 </xs:complexType>
785 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
786 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
787 <xs:element name="CreateSequenceResponse"
788 type="wsrm:CreateSequenceResponseType"/>
789 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
790 <xs:element name="CloseSequenceResponse"
791 type="wsrm:CloseSequenceResponseType"/>
792 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
793 <xs:element name="TerminateSequenceResponse"
794 type="wsrm:TerminateSequenceResponseType"/>
795 <xs:complexType name="CreateSequenceType">
796   <xs:sequence>
797     <xs:element ref="wsrm:AcksTo"/>
798     <xs:element ref="wsrm:Expires" minOccurs="0"/>
799     <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
800     <xs:any namespace="##other" processContents="lax" minOccurs="0"
801 maxOccurs="unbounded">
802       <xs:annotation>
803         <xs:documentation>
804           It is the authors intent that this extensibility be used to
805 transfer a Security Token Reference as defined in WS-Security.
806         </xs:documentation>
807       </xs:annotation>
808     </xs:any>
809   </xs:sequence>
810   <xs:anyAttribute namespace="##other" processContents="lax"/>
811 </xs:complexType>
812 <xs:complexType name="CreateSequenceResponseType">
813   <xs:sequence>
814     <xs:element ref="wsrm:Identifier"/>
815     <xs:element ref="wsrm:Expires" minOccurs="0"/>
816     <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>
817     <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
818     <xs:any namespace="##other" processContents="lax" minOccurs="0"
819 maxOccurs="unbounded"/>
820   </xs:sequence>
821   <xs:anyAttribute namespace="##other" processContents="lax"/>
822 </xs:complexType>
823 <xs:complexType name="CloseSequenceType">

```

```

761     <xs:sequence>
762         <xs:element ref="wsrm:Identifier"/>
763         <xs:any namespace="##other" processContents="lax" minOccurs="0"
764 maxOccurs="unbounded"/>
765     </xs:sequence>
766     <xs:anyAttribute namespace="##other" processContents="lax"/>
767 </xs:complexType>
768 <xs:complexType name="CloseSequenceResponseType">
769     <xs:sequence>
770         <xs:element ref="wsrm:Identifier"/>
771         <xs:any namespace="##other" processContents="lax" minOccurs="0"
772 maxOccurs="unbounded"/>
773     </xs:sequence>
774     <xs:anyAttribute namespace="##other" processContents="lax"/>
775 </xs:complexType>
776 <xs:complexType name="TerminateSequenceType">
777     <xs:sequence>
778         <xs:element ref="wsrm:Identifier"/>
779         <xs:any namespace="##other" processContents="lax" minOccurs="0"
780 maxOccurs="unbounded"/>
781     </xs:sequence>
782     <xs:anyAttribute namespace="##other" processContents="lax"/>
783 </xs:complexType>
784 <xs:complexType name="TerminateSequenceResponseType">
785     <xs:sequence>
786         <xs:element ref="wsrm:Identifier"/>
787         <xs:any namespace="##other" processContents="lax" minOccurs="0"
788 maxOccurs="unbounded"/>
789     </xs:sequence>
790     <xs:anyAttribute namespace="##other" processContents="lax"/>
791 </xs:complexType>
792 <xs:element name="AcksTo"

```

```

761     type="wsa:EndpointReferenceType"/>
762     <xs:complexType name="OfferType">
763         <xs:sequence>
764             <xs:element ref="wsrm:Identifier"/>
765             <xs:element ref="wsrm:Expires" minOccurs="0"/>
766             <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
767             <xs:any namespace="##other" processContents="lax" minOccurs="0"
768 maxOccurs="unbounded"/>
769         </xs:sequence>
770         <xs:anyAttribute namespace="##other" processContents="lax"/>
771     </xs:complexType>
772     <xs:complexType name="AcceptType">
773         <xs:sequence>
774             <xs:element ref="wsrm:AcksTo"/>
775             <xs:any namespace="##other" processContents="lax" minOccurs="0"
776 maxOccurs="unbounded"/>
777         </xs:sequence>
778         <xs:anyAttribute namespace="##other" processContents="lax"/>
779     </xs:complexType>
780     <xs:element name="Expires">
781         <xs:complexType>
782             <xs:simpleContent>
783                 <xs:extension base="xs:duration">
784                     <xs:anyAttribute namespace="##other" processContents="lax"/>
785                 </xs:extension>
786             </xs:simpleContent>
787         </xs:complexType>
788     </xs:element>
789     <xs:element name="AcknowledgementInterval">
790         <xs:complexType>
791             <xs:sequence/>
792             <xs:attribute name="Milliseconds" type="xs:unsignedLong"
793 use="required"/>
794             <xs:anyAttribute namespace="##other" processContents="lax"/>
795         </xs:complexType>
796     </xs:element>
797 </xs:schema>

```

## B. Message Examples

### B.1 Create Sequence

#### Create Sequence

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:MessageID>
      http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
    </wsa:MessageID>
    <wsa:To>http://example.com/serviceB/123</wsa:To>
    <wsa:Action>http://docs.oasis-open.org/ws-
rx/wsmr/200604/CreateSequence</wsa:Action>
    <wsa:ReplyTo>
      <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
    </wsa:ReplyTo>
  </S:Header>
  <S:Body>
    <wsmr:CreateSequence>
      <wsmr:AcksTo>
        <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
      </wsmr:AcksTo>
    </wsmr:CreateSequence>
  </S:Body>
</S:Envelope>
```

#### Create Sequence Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:To>http://Business456.com/serviceA/789</wsa:To>
    <wsa:RelatesTo>
      http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
    </wsa:RelatesTo>
    <wsa:Action>
      http://docs.oasis-open.org/ws-rx/wsmr/200604/CreateSequenceResponse
    </wsa:Action>
  </S:Header>
  <S:Body>
    <wsmr:CreateSequenceResponse>
      <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
    </wsmr:CreateSequenceResponse>
  </S:Body>
</S:Envelope>
```

### B.2 Initial Transmission

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

## 761 Message 1

```
761 <?xml version="1.0" encoding="UTF-8"?>
761 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
761 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
761 xmlns:wsa="http://www.w3.org/2005/08/addressing">
761   <S:Header>
761     <wsa:MessageID>
761       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
761     </wsa:MessageID>
761     <wsa:To>http://example.com/serviceB/123</wsa:To>
761     <wsa:From>
761       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
761     </wsa:From>
761     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
761     <wsmr:Sequence>
761       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
761       <wsmr:MessageNumber>1</wsmr:MessageNumber>
761     </wsmr:Sequence>
761   </S:Header>
761   <S:Body>
761     <!-- Some Application Data -->
761   </S:Body>
761 </S:Envelope>
```

## 761 Message 2

```
761 <?xml version="1.0" encoding="UTF-8"?>
761 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
761 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
761 xmlns:wsa="http://www.w3.org/2005/08/addressing">
761   <S:Header>
761     <wsa:MessageID>
761       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
761     </wsa:MessageID>
761     <wsa:To>http://example.com/serviceB/123</wsa:To>
761     <wsa:From>
761       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
761     </wsa:From>
761     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
761     <wsmr:Sequence>
761       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
761       <wsmr:MessageNumber>2</wsmr:MessageNumber>
761     </wsmr:Sequence>
761   </S:Header>
761   <S:Body>
761     <!-- Some Application Data -->
761   </S:Body>
761 </S:Envelope>
```

## 761 Message 3

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

```

761 </wsa:From>
761 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
761 <wsrm:Sequence>
761 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
761 <wsrm:MessageNumber>3</wsrm:MessageNumber>
761 </wsrm:Sequence>
761 <wsrm:AckRequested>
761 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
761 </wsrm:AckRequested>
761 </S:Header>
761 <S:Body>
761 <!-- Some Application Data -->
761 </S:Body>
761 </S:Envelope>

```

## 761 B.3 First Acknowledgement

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

```

761 <?xml version="1.0" encoding="UTF-8"?>
761 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
761 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
761 xmlns:wsa="http://www.w3.org/2005/08/addressing">
761 <S:Header>
761 <wsa:MessageID>
761 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
761 </wsa:MessageID>
761 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
761 <wsa:From>
761 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
761 </wsa:From>
761 <wsa:Action>
761 http://docs.oasis-open.org/ws-rx/wsr/200604/SequenceAcknowledgement
761 </wsa:Action>
761 <wsrm:SequenceAcknowledgement>
761 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
761 <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
761 <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
761 </wsrm:SequenceAcknowledgement>
761 </S:Header>
761 <S:Body/>
761 </S:Envelope>

```

## 761 B.4 Retransmission

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

```

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

```

```

761 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
761 <wsrm:Sequence>
761 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
761 <wsrm:MessageNumber>2</wsrm:MessageNumber>
761 </wsrm:Sequence>
761 <wsrm:AckRequested>
761 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
761 </wsrm:AckRequested>
761 </S:Header>
761 <S:Body>
761 <!-- Some Application Data -->
761 </S:Body>
761 </S:Envelope>

```

## 761 B.5 Termination

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

```

761 <?xml version="1.0" encoding="UTF-8"?>
761 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
761 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
761 xmlns:wsa="http://www.w3.org/2005/08/addressing">
761 <S:Header>
761 <wsa:MessageID>
761 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
761 </wsa:MessageID>
761 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
761 <wsa:From>
761 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
761 </wsa:From>
761 <wsa:Action>
761 http://docs.oasis-open.org/ws-rx/wsrn/200604/SequenceAcknowledgement
761 </wsa:Action>
761 <wsrm:SequenceAcknowledgement>
761 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
761 <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
761 </wsrm:SequenceAcknowledgement>
761 </S:Header>
761 <S:Body/>
761 </S:Envelope>

```

## 761 Terminate Sequence

```

761 <?xml version="1.0" encoding="UTF-8"?>
761 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
761 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
761 xmlns:wsa="http://www.w3.org/2005/08/addressing">
761 <S:Header>
761 <wsa:MessageID>
761 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
761 </wsa:MessageID>
761 <wsa:To>http://example.com/serviceB/123</wsa:To>
761 <wsa:Action>
761 http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequence
761 </wsa:Action>
761 <wsa:From>
761 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
761 </wsa:From>
761 </S:Header>
761 <S:Body>
761 <wsrm:TerminateSequence>

```

```

761     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
761   </wsrm:TerminateSequence>
761 </S:Body>
761 </S:Envelope>

```

## 761 Terminate Sequence Response

```

761 <?xml version="1.0" encoding="UTF-8"?>
761 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
761   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
761   xmlns:wsa="http://www.w3.org/2005/08/addressing">
761   <S:Header>
761     <wsa:MessageID>
761       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
761     </wsa:MessageID>
761     <wsa:To>http://example.com/serviceA/789</wsa:To>
761     <wsa:Action>
761       http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequenceResponse
761     </wsa:Action>
761     <wsa:RelatesTo>
761       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
761     </wsa:RelatesTo>
761     <wsa:From>
761       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
761     </wsa:From>
761   </S:Header>
761   <S:Body>
761     <wsrm:TerminateSequenceResponse>
761       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
761     </wsrm:TerminateSequenceResponse>
761   </S:Body>
761 </S:Envelope>

```

## 761 C. WSDL

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

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

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

```

761 <?xml version="1.0" encoding="utf-8"?>
762 <!--
763 OASIS takes no position regarding the validity or scope of any intellectual
764 property or other rights that might be claimed to pertain to the
765 implementation or use of the technology described in this document or the
766 extent to which any license under such rights might or might not be available;
767 neither does it represent that it has made any effort to identify any such
768 rights. Information on OASIS's procedures with respect to rights in OASIS
769 specifications can be found at the OASIS website. Copies of claims of rights
770 made available for publication and any assurances of licenses to be made
771 available, or the result of an attempt made to obtain a general license or
772 permission for the use of such proprietary rights by implementors or users of
773 this specification, can be obtained from the OASIS Executive Director.
774 OASIS invites any interested party to bring to its attention any copyrights,
775 patents or patent applications, or other proprietary rights which may cover
776 technology that may be required to implement this specification. Please
777 address the information to the OASIS Executive Director.
778 Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
779 This document and translations of it may be copied and furnished to others,
780 and derivative works that comment on or otherwise explain it or assist in its
781 implementation may be prepared, copied, published and distributed, in whole or
782 in part, without restriction of any kind, provided that the above copyright
783 notice and this paragraph are included on all such copies and derivative
784 works. However, this document itself does not be modified in any way, such as
785 by removing the copyright notice or references to OASIS, except as needed for
786 the purpose of developing OASIS specifications, in which case the procedures
787 for copyrights defined in the OASIS Intellectual Property Rights document must
788 be followed, or as required to translate it into languages other than English.
789 The limited permissions granted above are perpetual and will not be revoked by
790 OASIS or its successors or assigns.
791 This document and the information contained herein is provided on an "AS
792 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
793 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
794 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
795 FITNESS FOR A PARTICULAR PURPOSE.
796 -->
797 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
798 xmlns:xs="http://www.w3.org/2001/XMLSchema"
799 xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
800 open.org/ws-rx/wsr/200604" xmlns:tns="http://docs.oasis-open.org/ws-
801 rx/wsr/200604/wsdl" targetNamespace="http://docs.oasis-open.org/ws-
802 rx/wsr/200604/wsdl">
803
804   <wsdl:types>
805     <xs:schema
806       <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsr/200604"
807       schemaLocation="http://docs.oasis-open.org/ws-rx/wsr/200604/wsr-1.1-schema-
808       200604.xsd"/>
809     </xs:schema>
810   </wsdl:types>
811
812   <wsdl:message name="CreateSequence">
813     <wsdl:part name="create" element="rm:CreateSequence"/>
814   </wsdl:message>
815   <wsdl:message name="CreateSequenceResponse">
816     <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
817   </wsdl:message>
818   <wsdl:message name="CloseSequence">
819     <wsdl:part name="close" element="rm:CloseSequence"/>
820   </wsdl:message>
821   <wsdl:message name="CloseSequenceResponse">
822     <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
823   </wsdl:message>

```

```

761     <wsdl:message name="TerminateSequence">
762         <wsdl:part name="terminate" element="rm:TerminateSequence"/>
763     </wsdl:message>
764     <wsdl:message name="TerminateSequenceResponse">
765         <wsdl:part name="terminateResponse"
766 element="rm:TerminateSequenceResponse"/>
767     </wsdl:message>

768     <wsdl:portType name="SequenceAbstractPortType">
769         <wsdl:operation name="CreateSequence">
770             <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
771 open.org/ws-rx/wsrn/200604/CreateSequence"/>
772             <wsdl:output message="tns:CreateSequenceResponse"
773 wsa:Action="http://docs.oasis-open.org/ws-
774 rx/wsrn/200604/CreateSequenceResponse"/>
775         </wsdl:operation>
776         <wsdl:operation name="CloseSequence">
777             <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
778 open.org/ws-rx/wsrn/200604/CloseSequence"/>
779             <wsdl:output message="tns:CloseSequenceResponse"
780 wsa:Action="http://docs.oasis-open.org/ws-
781 rx/wsrn/200604/CloseSequenceResponse"/>
782         </wsdl:operation>
783         <wsdl:operation name="TerminateSequence">
784             <wsdl:input message="tns:TerminateSequence"
785 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequence"/>
786             <wsdl:output message="tns:TerminateSequenceResponse"
787 wsa:Action="http://docs.oasis-open.org/ws-
788 rx/wsrn/200604/TerminateSequenceResponse"/>
789         </wsdl:operation>
790     </wsdl:portType>

791 </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
<b>Create Sequence</b>	<i>Transmit Create Sequence</i> Connecting	N/A	N/A	N/A	N/A	N/A	N/A	N/A
<b>Create Sequence Response</b>	N/A	Connected	N/A	N/A	N/A	N/A	N/A	N/A
<b>Create Sequence Refused Fault</b>	N/A	Terminated	N/A	N/A	N/A	N/A	N/A	N/A
<b>New Message</b>	N/A	N/A	<i>Transmit message</i> Connected	<i>Inhibited</i>	<i>Inhibited?</i> Closing	N/A	N/A	N/A
<b>Retransmit of unack message</b>	N/A	N/A	<i>Transmit message</i> Connected	<i>Transmit message</i> Rollover	<i>Trasmit message?</i> Closing	<i>Transmit message</i> Closed	N/A	N/A
<b>SeqAck (non-final)</b>	N/A	N/A	Connected	Rollover	Closing	Closed	<i>Ignore?</i>	<i>Transmit Unknown Sequence Fault</i> Terminated
<b>Nack</b>	N/A	N/A	<i>Transmit message</i> Connected	<i>Transmit message</i> Rollover	<i>Transmit message?</i> Closing	<i>Transmit message?</i> Closed	<i>Ignore?</i>	<i>Transmit Unknown Sequence fault</i> Terminated
<b>Reached max msg number</b>	N/A	N/A	Rollover	Rollover	N/A	N/A	N/A	N/A

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

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

761 Table 3 RM Destination State Transition Table

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

761 In Table 3 above, the rows consists of events that occur at the RM Destination throughout the lifetime of  
762 an RM Sequence and the columns consists of various RM Destination states. Each cell in the table above  
763 lists the action that the RM Destination takes on occurrence of a particular event and the next state that it  
764 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 <a href="http://docs.oasis-open.org/wsrn/200510/">http://docs.oasis-open.org/wsrn/200510/</a> )
wd-05	2005-09-30	Anish Karmarkar	i045 (Include SecureConversation as a reference and move it to non-normative citation)
wd-05	2005-09-30	Anish Karmarkar	i046 (change the type of wsrn:FaultCode element)
wd-06	2005-11-02	Gilbert Pilz	Start wd-06 by changing title page from cd-01.
wd-06	2005-11-03	Gilbert Pilz	i047 (Reorder spec sections)
wd-07	2005-11-17	Gilbert Pilz	Start wd-07
wd-07	2005-11-28	Doug Davis	i071 – except for period in Appendix headings
wd-07	2005-11-28	Doug Davis	i10
wd-07	2005-11-28	Doug Davis	i030
wd-07	2005-11-28	Doug Davis	i037
wd-07	2005-11-28	Doug Davis	i038
wd-07	2005-11-28	Doug Davis	i041
wd-07	2005-11-28	Doug Davis	i043
wd-07	2005-11-28	Doug Davis	i044

Rev	Date	By Whom	What
wd-07	2005-11-28	Doug Davis	i048
wd-07	2005-11-28	Doug Davis	i051
wd-07	2005-11-28	Doug Davis	i053
wd-07	2005-11-28	Doug Davis	i059
wd-07	2005-11-28	Doug Davis	i062
wd-07	2005-11-28	Doug Davis	i063
wd-07	2005-11-28	Doug Davis	i065
wd-07	2005-11-28	Doug Davis	i067
wd-07	2005-11-28	Doug Davis	i068
wd-07	2005-11-28	Doug Davis	i069
wd-07	2005-11-28	Doug Davis	Fix bulleted list (#2) in section 2.3
wd-07	2005-11-29	Gilbert Pilz	i074 (Use of [tcShortName] in artifact locations namespaces, etc)
wd-07	2005-11-29	Gilbert Pilz	i071 – Fixed styles and formatting for TOC. Fixed styles of the appendix headings.
wd-07	2005-11-30	Doug Davis	Removed dup definition of "Receive"
wd-07	2005-11-30	Gilbert Pilz	Fixed lost formatting from heading for Namespace section. Fixed style of text body elements to match OASIS example documents. Fixed tables to match OASIS example documents.
wd-07	2005-12-01	Gilbert Pilz	Updated fix for i074 to eliminate trailing '/'. Added corresponding text around action IRI composition.
wd-07	2005-12-01	Gilbert Pilz	Use non-fixed fields for date values on both title page and body footers.
wd-07	2005-12-01	Doug Davis	Alphabetize the glossary
wd-07	2005-12-02	Doug Davis	i064
wd-07	2005-12-02	Doug Davis	i066
wd-08	2005-12-15	Doug Davis	Add back in RM Source to glossary
wd-08	2005-12-15	Steve Winkler	Doug added Steve's editorial nits
wd-08	2005-12-21	Doug Davis	i050
wd-08	2005-12-21	Doug Davis	i081
wd-08	2005-12-21	Doug Davis	i080 – but i050 negates the need for any changes
wd-08	2005-12-21	Doug Davis	i079
wd-08	2005-12-21	Doug Davis	i076 – didn't add text about "replies" since the RMD to RMS sequence could be used for any message not just replies
wd-08	2005-12-21	Umit Yalcinalp	Action Su03: removed wsse from Table 1
wd-08	2005-12-21	Umit Yalcinalp	i057 per Sunnyvale F2F 2005, Cleaned up some formatting errors in contributors
wd-08	2005-12-27	Doug Davis	i060
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".
<a href="#">wd-13</a>	<a href="#">2006-05-08</a>	<a href="#">Gilbert Pilz</a>	<a href="#">i093 part 1; more work needed</a>

## 762 G. Notices

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

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

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

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

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

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