



# Web Services ReliableMessaging (WS-Reliable Messaging)

Committee Draft 02, January 16, 2006

# ~~Web Services Reliable Messaging (WS-Reliable Messaging)~~

~~Working Draft 06, November 3<sup>rd</sup> 2005~~

## Document identifier:

wsrm-1.1-spec-cd-02wd-06

## Location:

<http://docs.oasis-open.org/ws-rx/wsrn/200510/wsrn-1.1-spec-cd-02.pdf>

## Editors:

Gilbert Pilz, BEA <[gilbert.pilz@bea.com](mailto:gilbert.pilz@bea.com)>

Doug Davis, IBM <[dug@us.ibm.com](mailto:dug@us.ibm.com)>

Anish Karmarkar, Oracle <[Anish.Karmarkar@oracle.com](mailto:Anish.Karmarkar@oracle.com)>

~~Steve Winkler, SAP <[steve.winkler@sap.com](mailto:steve.winkler@sap.com)>~~

~~Ümit Yalçınalp, SAP <[umit.yalcinalp@sap.com](mailto:umit.yalcinalp@sap.com)>~~

~~TBD~~

## Contributors:

TBD

## Abstract:

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

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

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

**Status:**

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

This document was last revised or approved by the OASIS WS-RX Technical Committee on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document.

For information on whether any patents have been disclosed that may be essential to implementing this specification and any offers of patent licensing terms please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/ws-rx/ipr.php>).

## Table of Contents

46		
47	1 Introduction.....	4
48	1.1 Goals and Requirements.....	4
49	1.1.1 Requirements.....	4
50	1.2 Notational Conventions.....	4
51	1.3 Namespace.....	4
52	1.4 Compliance.....	5
53	2 Reliable Messaging Model.....	6
54	2.1 Glossary.....	6
55	2.2 Protocol Preconditions.....	7
56	2.3 Protocol Invariants.....	7
57	2.4 Example Message Exchange.....	7
58	3 RM Protocol Elements.....	10
59	3.1 Sequence Creation.....	10
60	3.2 Closing A Sequence.....	13
61	3.3 Sequence Termination.....	14
62	3.4 Sequences.....	15
63	3.5 Request Acknowledgement.....	16
64	3.6 Sequence Acknowledgement.....	16
65	4 Faults.....	20
66	4.1 SequenceFault Element.....	21
67	4.2 Sequence Terminated.....	22
68	4.3 Unknown Sequence.....	22
69	4.4 Invalid Acknowledgement.....	22
70	4.5 Message Number Rollover.....	23
71	4.6 Create Sequence Refused.....	23
72	4.7 Sequence Closed.....	23
73	5 Security Considerations.....	24
74	6 References.....	26
75	6.1 Normative.....	26
76	6.2 Non-Normative.....	26
77	A. Schema.....	28
78	B. Message Examples.....	33
79	B.1 Create Sequence.....	33
80	B.2 Initial Transmission.....	33
81	B.3 First Acknowledgement.....	35
82	B.4 Retransmission.....	35
83	B.5 Termination.....	36

84	C. WSDL.....	38
85	D. Acknowledgments.....	40
86	E. Revision History.....	41
87	F. Notices.....	44

|

# 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~~message delivery. It defines a messaging protocol to identify, track, and manage the reliable delivery of messages between exactly two parties,~~ a source and a destination. It also defines a SOAP binding that is required for interoperability. Additional bindings may be defined.

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

## 1.1 Goals and Requirements

### 1.1.1 Requirements

## 1.2 Notational Conventions

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

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

- The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- Characters are appended to elements and attributes to indicate cardinality:
  - "?" (0 or 1)
  - "\*" (0 or more)
  - "+" (1 or more)
- The character "|" is used to indicate a choice between alternatives.
- The characters "[" and "]" are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content specified in this document. Additional children elements and/or attributes MAY be added at the indicated extension points but they, or attribute, content. 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 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/200510-rn/200510/>~~

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

The following namespaces are used in this document:

Table 1

Prefix	Namespace
S	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>
S11	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>
wsrn	<a href="http://docs.oasis-open.org/ws-rx/wsrn/200510">http://docs.oasis-open.org/ws-rx/wsrn/200510</a>
wsa	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

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

<http://docs.oasis-open.org/ws-rx/wsrn/200510/wsrn-1.1.xsd>

Prefix	Namespace
S	<del><a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a></del>
S11	<del><a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a></del>
wsrn	<del><a href="http://docs.oasis-open.org/wsrn/200510/">http://docs.oasis-open.org/wsrn/200510/</a></del>
wsa	<del><a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a></del>
wsse	<del><a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</a></del>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

~~The normative schema for WS-Reliable Messaging can be found at:~~

~~<http://docs.oasis-open.org/wsrn/200510/wsrn.xsd>~~

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

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

~~<http://docs.oasis-open.org/ws-rx/wsrn/200510/SequenceAcknowledgement>~~

## 1.4 Compliance

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

145 ~~An implementation is not compliant with this specification if it fails to satisfy one or more of~~  
146 ~~the MUST or REQUIRED level requirements defined herein. A SOAP Node MUST NOT use the~~  
147 ~~XML namespace identifier for this specification (listed in SectionNamespace) within SOAP~~  
148 ~~Envelopes unless it is compliant with this specification.~~  
149 Normative text within this specification takes precedence over normative outlines, which in turn take  
150 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. In circumstances, accurately determine the disposition of each message transmitted as perceived by the RM Destination, so as to resolve any in-doubt status.

The protocol supports reliability features which include ordered delivery, duplicate elimination, and guaranteed receipt for the RMD. It is expected that the AD and RMD will implement as many of these or as few of these characteristics as necessary to implement the AD. In any case the wire protocol does not change. In addition, The protocol allows the RM Source and RM Destination to provide their respective Application Source and Application Destination a guarantee that a message that is sent by an Application Source will be delivered to the Application Destination.

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 guarantee is specified as a delivery assurance. It is the responsibility of the RM Source and RM Destination to fulfill the delivery assurances on behalf of their respective Application counterparts, or raise an error. The protocol defined here allows endpoints to meet this guarantee for the delivery assurances defined below. However, the means by which these delivery assurances are manifested by either the RM Source or RM Destination roles is an implementation concern, and is out of scope of this specification.

Note that the underlying protocol defined in this specification remains the same regardless of the delivery assurance.

Persistence considerations related to an endpoint's ability to satisfy the delivery assurances defined below are the responsibility of the implementation and do not affect the wire protocol. As such, they are out of scope of this specification.

There are four basic delivery assurances that endpoints can provide:

**AtMostOnce** Messages will be delivered at most once without duplication or an error will be raised on at least one endpoint. It is possible that some messages in a sequence may not be delivered.

**AtLeastOnce** Every message sent will be delivered or an error will be raised on at least one endpoint. Some messages may be delivered more than once.

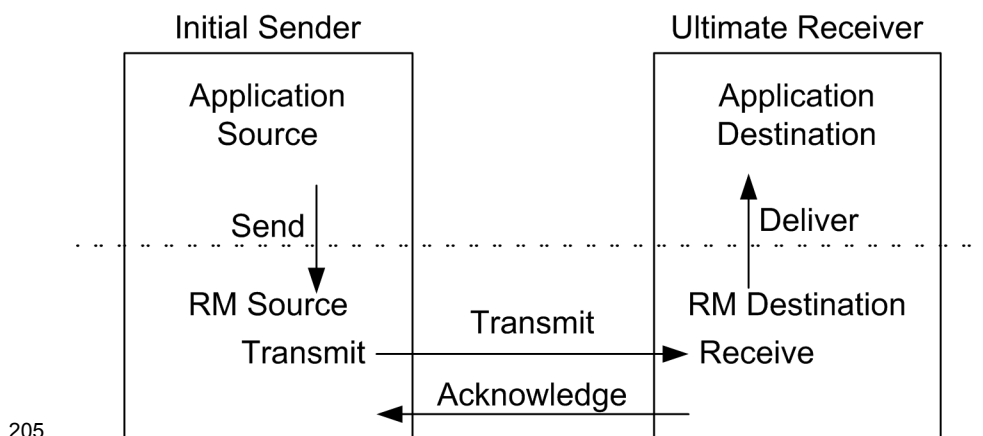
**ExactlyOnce** Every message sent will be delivered without duplication or an error will be raised on at least one endpoint. This delivery assurance is the logical "and" of the two prior delivery assurances.

**InOrder** Messages will be delivered in the order that they were sent. This delivery assurance may be combined with any of the above delivery assurances. It requires that the



195 messages within a Sequence will be delivered in an order so that the message numbers are  
 196 monotonically increasing. Note that this assurance says nothing about duplications or  
 197 omissions. Note also that it is only applicable to messages in the same Sequence. Cross-  
 198 Sequence ordering of messages is not in the scope of this specification.

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



206 Figure 1: Reliable Messaging Model

## 207 2.1 Glossary

208 The following definitions are used throughout this specification:

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

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

212 **Endpoint:** A referencable entity, processor, or resource where Web service messages are  
 213 originated or targeted.

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

215 **Deliver:** The act of transferring a message from the RM Destination to the Application Destination. The  
 216 reliability guarantee is fulfilled at this point.  
 217 **Application Destination:** The endpoint to which a message is Delivered.

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

221 **Delivery Assurance:** The guarantee that the messaging infrastructure provides on the  
 222 delivery of a message.

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

225 **RM Destination:** ~~For any one reliable sent message the endpoint that receives the message.~~  
226 **RM Source:** The endpoint that transmits the message.  
227 **RM Destination:** ~~The endpoint that receives the message.~~  
228 **Send:** The act of submitting a message to the RM Source for reliable delivery. The reliability guarantee  
229 begins at this point.  
230 **Deliver:** ~~The act of transferring a message from the RM Destination to the Application~~  
231 ~~Destination. The reliability guarantee is fulfilled at this point.~~  
232 **Transmit:** The act of writing a message to a network connection.  
233 **Receive:** ~~The act of reading a message from a network connection.~~  
234 **Acknowledgement:** ~~The communication from the RM Destination to the RM Source~~  
235 ~~indicating the successful receipt of a message.~~

## 236 2.2 Protocol Preconditions

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

- 239 • ~~For any single message exchange the RM Source **MUST** have an endpoint reference that uniquely~~  
240 ~~identifies the RM Destination endpoint~~~~The RM Source **MUST** have an endpoint reference that~~  
241 ~~uniquely identifies the RM Destination endpoint; correlations across messages addressed to the~~  
242 ~~unique endpoint **MUST** be meaningful.~~
- 243 • The RM Source **MUST** have knowledge of the destination's policies, if any, and the RM Source  
244 **MUST** be capable of formulating messages that adhere to this policy.

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

## 247 2.3 Protocol Invariants

248 During the lifetime of ~~a Sequence~~~~the protocol~~, two invariants are **REQUIRED** for correctness:

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

## 257 2.4 Example Message Exchange

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



259 Figure 2: The WS-ReliableMessaging Protocol

- 260 1. The protocol preconditions are established. These include policy exchange, endpoint resolution,  
 261 establishing trust.
- 262 2. The RM Source requests creation of a new Sequence.
- 263 3. The RM Destination creates a Sequence by returning a globally unique identifier.
- 264 4. The RM Source begins sending messages beginning with MessageNumber 1. In the figure above,  
 265 the RM Source sends 3 messages.

- 266 5. Since the 3rd message is the last in this exchange, the RM Source includes a  
267 <wsrm:AckRequested> HeaderLastMessage> token.
- 268 6. The 2nd message is lost in transit.
- 269 7. The RM Destination acknowledges receipt of message numbers 1 and 3 as a result of receiving the  
270 RM Source's <wsrm:AckRequested> Headerin response to the RM Source's-  
271 ~~<wsrm:LastMessage> token.~~
- 272 8. The RM Source retransmits the 2nd message. This is a new message on the underlying transport,  
273 but ~~since~~ it has the same sequence identifier and message number so the RM Destination can  
274 recognize it as equivalent to the earlier message, in case both are received.
- 275 9. The RM Source includes an <wsrm:AckRequested> element so the RM Destination will expedite  
276 an acknowledgement.
- 277 10. The RM Destination receives the second transmission of the message with MessageNumber 2 and  
278 acknowledges receipt of message numbers 1, 2, and 3 ~~which carried the <wsrm:LastMessage>~~-  
279 ~~token.~~
- 280 11. The RM Source receives this acknowledgement and sends a TerminateSequence message to the  
281 RM Destination indicating that the sequence is completed and reclaims any resources associated  
282 with the Sequence.
- 283 12. The RM Destination receives the TerminateSequence message indicating that the RM Source will  
284 not be sending any more messages, and reclaims any resources associated with the Sequence.
- 285 The RM Source will expect to receive acknowledgements from the RM Destination during the course of a  
286 message exchange at occasions described in Section 3 below. Should the acknowledgement not be  
287 received in a timely fashion, the RM Source MUST re-transmit the request since either the request or the  
288 associated acknowledgement may have been lost. Since the nature and dynamic characteristics of the  
289 underlying transport and potential intermediaries are unknown in the general case, the timing of re-  
290 transmissions cannot be specified. Additionally, over-aggressive re-transmissions have been  
291 demonstrated to cause transport or intermediary flooding which are counterproductive to the intention of  
292 providing a reliable exchange of messages. Consequently, implementers are encouraged to utilize  
293 adaptive mechanisms that dynamically adjust re-transmission time and the back-off intervals that are  
294 appropriate to the nature of the transports and intermediaries envisioned. For the case of TCP/IP  
295 transports, a mechanism similar to that described as RTTM in RFC 1323 [RTTM] should be considered.
- 296 Now that the basic model has been outlined, the details of the elements used in this protocol are now  
297 provided in Section 3.

## 3 RM Protocol Elements

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

### 3.1 Sequence Creation

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

~~The RM Destination of the outbound sequence is the WS-Addressing EndpointReference [WS-Addressing] to which `<wsrm:CreateSequence>` is sent. The RM Destination of the inbound sequence is the WS-Addressing `<wsa:ReplyTo>` of the `<wsrm:CreateSequence>`.~~

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

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

/wsrm:CreateSequence

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

/wsrm:CreateSequence/wsrm:AcksTo

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

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

/wsrm:CreateSequence/wsrm:Expires

This element, if present, of type `xs:duration` specifies the RM Source's requested duration for the Sequence. The RM Destination MAY either accept the requested duration or assign a lesser value of its

341 choosing. A value of 'PT0S' indicates that the Sequence will never expire. Absence of the element  
342 indicates an implied value of 'PT0S'.

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

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

346 /wsrm:CreateSequence/wsrm:Offer

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

349 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier

350 This REQUIRED element MUST contain an absolute URI conformant with RFC3982396 that uniquely  
351 identifies the offered Sequence.

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

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

355 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires

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

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

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

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

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

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

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

368 **OPTIONAL** /wsrm:CreateSequence/{any}

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

371 /wsrm:CreateSequence/@{any}

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

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

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

```

379 <wsrm:CreateSequenceResponse ...>
380   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
381   <wsrm:Expires> xs:duration </wsrm:Expires> ?
382   <wsrm:Accept ...>
383     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
384     ...
385   </wsrm:Accept> ?
386   ...
387 </wsrm:CreateSequenceResponse>

```

388 /wsrm:CreateSequenceResponse

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

392 /wsrm:CreateSequenceResponse/wsrm:Identifier

393 This REQUIRED element MUST contain an absolute URI conformant with RFC3982396 of the Sequence  
394 that has been created by the RM Destination.

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

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

398 /wsrm:CreateSequenceResponse/wsrm:Expires

399 This element, if present, of type xs:duration accepts or refines the RM Source's requested duration for  
400 the Sequence. A value of 'PT0S' indicates that the Sequence will never expire. Absence of the element  
401 indicates an implied value of 'PT0S'. This value MUST be equal ~~to or lesser~~ ~~lessor~~ ~~lessor~~ than the value  
402 requested by the RM Source in the corresponding <wsrm:CreateSequence> message.

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

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

406 /wsrm:CreateSequenceResponse/wsrm:Accept

407 This element, if present, enables an RM Destination to accept the offer of a corresponding Sequence for  
408 the reliable exchange of messages transmitted from RM Destination to RM Source. ~~This element MUST-~~  
409 ~~be present if the corresponding <wsrm:CreateSequence> message contained an <wsrm:Offer>~~  
410 ~~element.~~

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

414 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo

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

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

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

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



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

424 /wsrm:CreateSequenceResponse/{any}

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

427 /wsrm:CreateSequenceResponse/@{any}

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

## 430 3.2 Closing A Sequence

431 There may be times during the use of an RM Sequence that the RM Source or RM Destination will wish to  
432 discontinue using a Sequence. Simply terminating the Sequence discards the state managed by the RM  
433 Destination, leaving the RM Source unaware of the final ranges of messages that were successfully  
434 delivered to the RM Destination. To ensure that the Sequence ends with a known final state both the RM  
435 Source and RM Destination may choose to 'close' the Sequence before terminating it even if some of the  
436 messages have not been successfully delivered to the RM Destination.

437 If the RM Source wishes to close the Sequence then it sends a <wsrm:CloseSequence> element, in the  
438 body of a message, to the RM Destination. This message indicates that the RM Destination MUST NOT  
439 receive any new messages for the specified sequence, other than those already received at the time the  
440 <wsrm:CloseSequence> element is interpreted by the RMD. Upon receipt of this message, or  
441 subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST  
442 include a final SequenceAcknowledgement (that MUST include the <wsrm:Final> element) header block  
443 on each message destined to the RM Source, including the CloseSequenceResponse message and on  
444 any Sequence Fault transmitted to the RMS. n the case where the RM Source wishes to discontinue use  
445 of a sequence, while it can send a TerminateSequence to the RM Destination, since this is a one-way  
446 message and due to the possibility of late arriving (or lost) messages and Acknowledgements, this would  
447 leave the RM Source unsure of the final ranges of messages that were successfully delivered to the RM  
448 Destination.

449 To alleviate this, the RM Source can send a <wsrm:CloseSequence> element, in the body of a  
450 message, to the RM Destination to indicate that RM Destination MUST NOT receive any  
451 new messages for the specified sequence, other than those already received at the time the  
452 <wsrm:CloseSequence> element is interpreted by the RMD. Upon receipt of this message  
453 the RM Destination MUST send a SequenceAcknowledgement to the RM Source. Note, this  
454 SequenceAcknowledgement MUST include the <wsrm:Final> element.

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

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



465 The following exemplar defines the CloseSequence syntax:

```
466 <wsrm:CloseSequence wsrm:Identifier="xs:anyURI" .../>
467 .../wsrm:CloseSequence
468 </wsrm:CloseSequence>
```

469 /wsrm:CloseSequence

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

473 -/wsrm:CloseSequence@Identifier

474 This REQUIRED attribute contains an absolute URI conformant with RFC3982396 that uniquely identifies  
475 the sequence.

476 /wsrm:CloseSequence/{any}

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

479 /wsrm:CloseSequence@{any}

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

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

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

```
486 <wsrm:CloseSequenceResponse ...>
487 ...
488 </wsrm:CloseSequenceResponse>
489 -/wsrm:CloseSequenceResponse
```

490 /wsrm:CloseSequenceResponse

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

494 /wsrm:CloseSequenceResponse/{any}

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

497 /wsrm:CloseSequenceResponse@{any}

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

### 500 3.3 Sequence Termination

501 When the RM Source has completed its use of the Sequence, it sends a <wsrm:TerminateSequence>  
502 element, in the body of a message to the RM Destination to indicate that the Sequence is complete, and  
503 that it will not be sending any further messages related to the Sequence. The RM Destination can safely  
504 reclaim any resources associated with the Sequence upon receipt of the <wsrm:TerminateSequence>  
505 message. Note, under normal usage the RM source will complete its use of the sequence when all of the

506 messages in the Sequence have been acknowledged. However, the RM Source is free to Terminate or  
507 Close a Sequence at any time regardless of the acknowledgement state of the messages.

508 The following exemplar defines the TerminateSequence syntax:

```
509 <wsrm:TerminateSequence ...>  
510   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
511   ...  
512 </wsrm:TerminateSequence>
```

513 /wsrm:TerminateSequence

514 This element is sent by an RM Source to indicate it has completed its use of the Sequence, i.e. it MUST  
515 NOT send any additional message to the RM Destination referencing this sequence. It indicates that the  
516 RM Destination can safely reclaim any resources related to the identified Sequence. This element MUST  
517 NOT be sent as a header block.

518 /wsrm:TerminateSequence/wsrm:Identifier

519 This REQUIRED element MUST contain an absolute URI conformant with RFC3982396 of the Sequence  
520 that is being terminated.

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

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

524 /wsrm:TerminateSequence/{any}

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

527 /wsrm:TerminateSequence/@{any}

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

## 530 3.4 Sequences

531 The RM protocol uses a <wsrm:Sequence> header block to track and manage the reliable delivery of  
532 messages. Messages for which a reliable delivery is required MUST contain a <wsrm:Sequence>  
533 header block. Each Sequence MUST have a unique <wsrm:Identifier> element and each message  
534 within a Sequence MUST have a <wsrm:MessageNumber> element that increments by 1 from an initial  
535 value of 1. These values are contained within a <wsrm:Sequence> header block accompanying each  
536 message being delivered in the context of a Sequence~~the delivery assurance applies~~ MUST contain a  
537 <wsrm:Sequence> header block. Each Sequence MUST have a unique <wsrm:Identifier> element  
538 and each message within a Sequence MUST have a <wsrm:MessageNumber> element that increments  
539 by 1 from an initial value of 1. These values are contained within a <wsrm:Sequence> header block  
540 accompanying each message being delivered in the context of a Sequence. In addition to mandatory  
541 <wsrm:Identifier> and <wsrm:MessageNumber> elements, the header MAY include a  
542 <wsrm:LastMessage> element.

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

544 ~~The purpose of the <wsrm:LastMessage> element is to signal to the RM Destination that the~~  
545 ~~message represents the last message in the Sequence.~~

546 A following exemplar defines its syntax:

```

547 <wsrm:Sequence ...>
548   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
549   <wsrm:MessageNumber> xs:unsignedLong </wsrm:MessageNumber>
550   <wsrm:LastMessage/>?
551   ...
552 </wsrm:Sequence>

```

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

554 /wsrm:Sequence

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

559 /wsrm:Sequence/wsrm:Identifier

560 This REQUIRED element MUST contain an absolute URI conformant with RFC~~3982~~396 that uniquely identifies the Sequence.

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

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

565 /wsrm:Sequence/wsrm:MessageNumber

566 This REQUIRED element MUST contain an xs:unsignedLong representing the ordinal position of the message within a Sequence. Sequence MessageNumbers start at 1 and monotonically increase throughout the Sequence. If the message number exceeds the internal limitations of an RM Source or RM Destination or reaches the maximum value of an xs:unsignedLong (18,446,744,073,709,551,615), the RM Source or Destination MUST issue a MessageNumberRollover fault.

571 ~~/wsrm:Sequence/wsrm:LastMessage~~

572 ~~This element MAY be included by the RM Source endpoint. The <wsrm:LastMessage> element has no content.~~

574 /wsrm:Sequence/{any}

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

577 /wsrm:Sequence/@{any}

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

580 ~~A RM Source endpoint MUST include a <wsrm:LastMessage> element in the~~  
581 ~~<wsrm:Sequence> element for the last message in a Sequence. An RM Destination endpoint~~  
582 ~~MUST respond with a <wsrm:SequenceAcknowledgement> upon receipt of a~~  
583 ~~<wsrm:LastMessage> element. A Sequence MUST NOT use a <wsrm:MessageNumber> value~~  
584 ~~greater than that which accompanies a <wsrm:LastMessage> element. An RM Destination~~  
585 ~~MUST generate a LastMessageNumberExceeded (See Section 4.6) fault upon receipt of such~~  
586 ~~a message. In the event that an RM Source needs to close a Sequence and there is no~~  
587 ~~application message, the RM Source MAY send a message with an empty body containing~~  
588 ~~<wsrm:Sequence> header with the <wsrm:LastMessage> element. In this usage, the action~~  
589 ~~URI MUST be:~~

590 ~~http://docs.oasis-open.org/wsrn/200510/LastMessage~~

591 ~~in preference to the pattern defined in Section 1.2.~~

592 The following example illustrates a Sequence header block.

```
593 <wsrm:Sequence>
594   _____ <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
595   _____ <wsrm:MessageNumber>10</wsrm:MessageNumber>
596   _____ <wsrm:LastMessage/>
597 </wsrm:Sequence>
```

## 598 3.5 Request Acknowledgement

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

601 ~~The RM Source may request an acknowledgement message from the RM Destination at any time by~~  
602 ~~including an <wsrm:AckRequested> header block in the message. An RM Destination that receives a~~  
603 ~~message that contains an <wsrm:AckRequested> header block MUST respond with a message~~  
604 ~~containing a <wsrm:SequenceAcknowledgement> header block. If a non-mustUnderstand fault occurs~~  
605 ~~when processing an RM Header that was piggy-backed on another message, a fault MUST be generated,~~  
606 ~~but the processing of the original message MUST NOT be affected~~At any time, the RM Source may  
607 request an acknowledgement message from the RM Destination endpoint using an  
608 ~~<wsrm:AckRequested> header block.~~

609 ~~The RM Source endpoint requests this acknowledgement by including an~~  
610 ~~<wsrm:AckRequested> header block in the message. An RM Destination that receives a~~  
611 ~~message that contains an <wsrm:AckRequested> header block MUST respond with a~~  
612 ~~message containing a <wsrm:SequenceAcknowledgement> header block. If a non-~~  
613 ~~mustUnderstand fault occurs when processing an RM Header that was piggy-backed on~~  
614 ~~another message, a fault MUST be generated, but the processing of the original message~~  
615 ~~MUST NOT be affected.~~

616 The following exemplar defines its syntax:

```
617 <wsrm:AckRequested ...>
618   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
619   _____ <wsrm:MessageNumber>xs:unsignedLong</wsrm:MessageNumber>?
620   ...
621 </wsrm:AckRequested>
```

622 /wsrm:AckRequested

623 This element requests an acknowledgement for the identified sequence.

624 /wsrm:AckRequested/wsrm:Identifier

625 This REQUIRED element MUST contain an absolute URI, conformant with RFC~~3982~~396, that uniquely  
626 identifies the Sequence to which the request applies.

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

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

630 ~~/wsrm:AckRequested/wsrm:MessageNumber~~

631 ~~This OPTIONAL element, if present, MUST contain an xs:unsignedLong representing the highest~~

632 ~~<wsrm:MessageNumber> sent by the RM Source within the Sequence. If present, it MAY be treated as a~~

633 ~~hint to the RM Destination as an optimization to the process of preparing to transmit a~~  
634 ~~<wsrm:SequenceAcknowledgement>.~~

635 /wsrm:AckRequested/{any}

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

638 /wsrm:AckRequested/@{any}

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

## 641 3.6 Sequence Acknowledgement

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

651 The following exemplar defines its syntax:

```
652 <wsrm:SequenceAcknowledgement ...>
653   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
654   [ [ <wsrm:AcknowledgementRange ...
655       Upper="xs:unsignedLong"
656       Lower="xs:unsignedLong" /> +
657       <wsrm:Final/> ? }
658       | <wsrm:None/> | <wsrm:Nack> xs:unsignedLong </wsrm:Nack> +
659       <wsrm:Final/> ? <wsrm:None/> }
660       | <wsrm:Nack> xs:unsignedLong </wsrm:Nack> + ]
661   ]
662   ...
663 </wsrm:SequenceAcknowledgement>
```

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

665 /wsrm:SequenceAcknowledgement

666 This element contains the Sequence acknowledgement information.

667 /wsrm:SequenceAcknowledgement/wsrm:Identifier

668 This REQUIRED element MUST contain an absolute URI conformant with RFC3986 that uniquely  
669 identifies the Sequence. A message MUST NOT contain multiple <SequenceAcknowledgement> header  
670 blocks that share the same value for <Identifier>. 2396 that uniquely identifies the Sequence.

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

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

674 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange

675 This OPTIONAL element, if present, can occur 1 or more times. It contains a range of Sequence  
676 MessageNumbers successfully received by the RM Destination. The ranges SHOULD NOT overlap. This  
677 element MUST NOT be present if a sibling <wsrm:Nack> or <wsrm:None> element is  
678 message-  
679 Sequence MessageNumbers successfully received by the receiving endpoint manager. The ranges-  
680 SHOULD NOT overlap. This element MUST NOT be present if either the <wsrm:Nack> or  
<wsrm:None> elements are also present as a child of <wsrm:SequenceAcknowledgement>.

681 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Upper

682 This REQUIRED attribute contains an xs:unsignedLong representing the <wsrm:MessageNumber> of  
683 the highest contiguous message in a Sequence range received by the RM Destination.

684 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Lower

685 This REQUIRED attribute contains an xs:unsignedLong representing the <wsrm:MessageNumber> of  
686 the lowest contiguous message in a Sequence range received by the RM Destination.

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

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

690 /wsrm:SequenceAcknowledgement/wsrm:Final

691 This OPTIONAL element, if present, indicates that the RM Destination is not receiving new messages for  
692 the specified Sequence. The RM Source can be assured that the ranges of messages acknowledged by  
693 this SequenceAcknowledgement header block will not change in the future. This element MUST be  
694 present when the Sequence is no longer receiving new message for the specified sequence. Note: this  
695 element MUST NOT be used when sending a Nack, it can only be used when sending  
696 AcknowledgementRanges or <wsrm:None>.

697 /wsrm:SequenceAcknowledgement/wsrm:Nack

698 This OPTIONAL element, if present, MUST contain an xs:unsignedLong representing the  
699 <wsrm:MessageNumber> of an unreceived message in a Sequence. This element permits the gap  
700 analysis of the <wsrm:AcknowledgementRange> elements to be performed at the RM Destination  
701 rather than at the RM Source which may yield performance benefits in certain environments. The  
702 <wsrm:Nack> element MUST NOT be present if a sibling <wsrm:AcknowledgementRange> or  
703 <wsrm:None> element is  
704 either the <wsrm:AcknowledgementRange> or <wsrm:None> elements are-  
705 also present as a child of <wsrm:SequenceAcknowledgement>. Upon the receipt of a Nack, an RM  
706 Source SHOULD retransmit the message identified by the Nack. The RM Destination MUST NOT issue a  
707 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that it has previously  
708 acknowledged within a <wsrm:AcknowledgementRange>. The RM Source SHOULD ignore a  
709 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that has previously  
709 been acknowledged within a <wsrm:AcknowledgementRange>.

710 /wsrm:SequenceAcknowledgement/wsrm:None

711 This OPTIONAL element, if present, MUST be used when the RM Destination has not received any  
712 messages for the specified sequence. The <wsrm:None> element MUST NOT be present if a sibling  
713 <wsrm:AcknowledgementRange> or <wsrm:Nack> element is  
714 either the  
715 <wsrm:AcknowledgementRange> or <wsrm:Nack> elements are also present as a child of the  
715 <wsrm:SequenceAcknowledgement>.

716 /wsrm:SequenceAcknowledgement/{any}

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

719 /wsrm:SequenceAcknowledgement/@{any}

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

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

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

```
724 <wsrm:SequenceAcknowledgement>  
725     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
726     <wsrm:AcknowledgementRange Upper="10" Lower="1"/>  
727 </wsrm:SequenceAcknowledgement>
```

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

```
730 <wsrm:SequenceAcknowledgement>  
731     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
732     <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
733     <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
734     <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
735 </wsrm:SequenceAcknowledgement>
```

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

```
737 <wsrm:SequenceAcknowledgement>  
738     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
739     <wsrm:Nack>3</wsrm:Nack>  
740 </wsrm:SequenceAcknowledgement>
```



## 4 Faults

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

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

WS-ReliableMessaging faults MUST include as the [action] property the default fault action [URI](http://schemas.xmlsoap.org/ws/2004/08/addressing/fault) defined in the version of WS-Addressing used in the message. The value from the current version is below for informational purposes:

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

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

The definitions of faults use the following properties:

[Code] The fault code.

[Subcode] The fault subcode.

[Reason] The English language reason element.

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

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

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

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

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



```

779     <S:Value> [Subcode] </S:Value>
780     </S:Subcode>
781   </S:Code>
782   <S:Reason>
783     <S:Text xml:lang="en"> [Reason] </S:Text>
784   </S:Reason>
785   <S:Detail>
786     [Detail]
787     ...
788   </S:Detail>
789 </S:Fault>
790 </S:Body>
791 </S:Envelope>

```

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

```

794 <S11:Envelope>
795   <S11:Header>
796     <wsrm:SequenceFault>
797       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
798       ...
799     </wsrm:SequenceFault>
800     <!-- Headers elided for clarity. -->
801   </S11:Header>
802   <S11:Body>
803     <S11:Fault>
804       <faultcode> [Code] </faultcode>
805       <faultstring> [Reason] </faultstring>
806     </S11:Fault>
807   </S11:Body>
808 </S11:Envelope>

```

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

```

811 <S11:Envelope>
812   <S11:Body>
813     <S11:Fault>
814       <faultcode> [Subcode] </faultcode>
815       <faultstring xml:lang="en"> [Reason] </faultstring>
816     </S11:Fault>
817   </S11:Body>
818 </S11:Envelope>

```

## 819 4.1 SequenceFault Element

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

824 The following exemplar defines its syntax:

```

825 <wsrm:SequenceFault ...>
826   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
827   ...
828 </wsrm:SequenceFault>

```

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

830 `/wsrm:SequenceFault`

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

832 `/wsrm:SequenceFault/wsrm:FaultCode`

833 This element, if present, MUST contain a qualified name from the set of fault [\[Subcodes\]codes](#) defined  
834 below.

835 `/wsrm:SequenceFault/{any}`

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

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

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

## 841 **4.2 Sequence Terminated**

842 This fault is sent by either the RM Source or the RM Destination to indicate that [it has either encountered](#)  
843 [an unrecoverable condition, or has detected a violation of the protocol and as a consequence, has chosen](#)  
844 [to terminate the sequence. The endpoint that generates this fault should make every reasonable effort to](#)  
845 [notify the corresponding endpoint of this decision.](#)~~the endpoint that generated the fault has either-~~  
846 ~~encountered an unrecoverable condition, or has detected a violation of the protocol and as a~~  
847 ~~consequence, has chosen to terminate the sequence. The endpoint that generates this fault should make-~~  
848 ~~every reasonable effort to notify the corresponding endpoint of this decision.~~

849 Properties:

850 [Code] Sender or Receiver

851 [Subcode] `wsrm:SequenceTerminated`

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

853 [Detail]-

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

## 855 **4.3 Unknown Sequence**

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

858 Properties:

859 [Code] Sender

860 [Subcode] `wsrm:UnknownSequence`

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

862 [Detail]

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

## 4.4 Invalid Acknowledgement

This fault is sent by the RM Source in response to a `<wsrm: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] `wsrm:InvalidAcknowledgement`

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

[Detail]-

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

## 4.5 Message Number Rollover

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

Properties:

[Code] Sender

[Subcode] `wsrm:MessageNumberRollover`

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

[Detail]

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

## 4.6 Last Message Number Exceeded

~~This fault is sent by an RM Destination to indicate that it has received a message that has a `<wsrm:MessageNumber>` within a Sequence that exceeds the value of the `<wsrm:MessageNumber>` element that accompanied a `<wsrm:LastMessage>` element for the Sequence.~~

~~Properties:~~

~~[Code] Sender~~

~~[Subcode] `wsrm:LastMessageNumberExceeded`~~

~~[Reason] The value for `wsrm:MessageNumber` exceeds the value of the `MessageNumber` accompanying a `LastMessage` element in this Sequence.~~

~~[Detail]~~

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

## 4.7 Create Sequence Refused

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

Properties:

896 [Code] Sender

897 [Subcode] wsrn:CreateSequenceRefused

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

899 [Detail] ~~empty~~

900 ~~xs:any~~

## 901 **4.8 Sequence Closed**

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

905 Properties:

906 [Code] Sender

907 [Subcode] wsrn:SequenceClosed

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

909 [Detail] ~~<wsrn:Identifier...> xs:anyURI </wsrn:Identifier>~~

910 ~~<wsrn:Identifier...> xs:anyURI </wsrn:Identifier>~~

## 5 Security Considerations

It is strongly recommended that the communication between services be secured using the mechanisms described in WS-Security [WS-Security]. ~~In order to properly secure messages, the body and all relevant headers need to be included in the signature. Specifically, the <wsrm:Sequence> header needs to be signed with the body in order to "bind" the two together. 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 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. If a Sequence is bound to a specific endpoint, then the security context needs to be established or shared with the endpoint 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. 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. 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). 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.
- **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by comparing secured policies – see WS-Policy and WS-SecurityPolicy).
- **Authentication** – Authentication is established using the mechanisms described in WS-Security and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- **Accountability** – Accountability is a function of the type of and string of the key and algorithms being used. ~~In many cases, a strong symmetric key provides sufficient accountability. In many cases, a strong symmetric key provides sufficient accountability.~~ However, in some environments, strong PKI signatures are required.
- **Availability** – All reliable messaging services are subject to a variety of availability attacks. ~~Replay detection is a common attack and it is recommended that this be addressed by the mechanisms described in WS-Security. (Note that because of legitimate message replays, detection should include a differentiator besides message id such as a timestamp). Other attacks, such as network-level denial of service attacks are harder to avoid and are outside the scope of this specification. Replay detection is a common attack and it is recommended that this be addressed by the mechanisms described in WS-Security. (Note that because of legitimate message replays, detection should include a differentiator besides message id such as a timestamp). Other attacks, such as network-level denial of service attacks are harder to avoid and are outside the scope of this specification.~~ That said, care should be taken to ensure that minimal state is saved prior to any authenticating 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]URI-

~~W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging Framework" June 2003T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.~~

#### [URI]XML-ns-

~~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]1-

W3C Recommendation, "XML Schema Part 1: Structures," 2 May 2001.

#### [XML-Schema Part2]2-

W3C Recommendation, "XML Schema Part 2: Datatypes," 2 May 2001.

#### [WSDL 1.1Security]

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

#### [Tanenbaum]

~~"Computer Networks," Andrew S. Tanenbaum, Prentice Hall PTR, 2003.~~

#### [WSDL]

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

#### [WS-Addressing]

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

### 6.2 Non-Normative

#### [WS-Policy]

- 1023 D. Box, et al, "[Web Services Policy Framework \(WS-Policy\)](#)," September 2004.
- 1024 **-[WS-PolicyAttachment]**
- 1025 D. ~~D.~~Box, et al, "[Web Services Policy Attachment \(WS-PolicyAttachment\)](#)," September 2004.
- 1026 **[WS-Security]**
- 1027 [Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "OASIS Web Services Security:](#)
- 1028 [SOAP Message Security 1.0 \(WS-Security 2004\)". OASIS Standard 200401, March 2004.](#)
- 1029 **[RTTM]**
- 1030 [V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", RFC 1323, May](#)
- 1031 [1992.](#)
- 1032 **[SecurityPolicy]**
- 1033 G. Della-Libra, [et. al. "Web Services Security Policy Language \(WS-SecurityPolicy\)", July 2005](#)~~"Web-~~
- 1034 ~~Services Security Policy Language (WS-SecurityPolicy)," December 2002.~~
- 1035 **-[SecureConversation]**
- 1036 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," May 2004.



## 1037 **A. Schema**

1038 The normative schema that is defined for WS-ReliableMessaging using [XML-Schema Part1] and [XML-  
1039 Schema Part2] is located at:

1040 <http://docs.oasis-open.org/ws-rx/wsrn/200510/wsrn-1.1-schema-200510.xsd>

1041 -

## Appendix A. Schema

The normative schema for WS-ReliableMessaging is located at:

<http://docs.oasis-open.org/wsrn/200510/wsrn.xsd>

The following copy is provided for reference.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
OASIS takes no position regarding the validity or scope of any intellectual
property or other rights that might be claimed to pertain to the
implementation or use of the technology described in this document or the
extent to which any license under such rights might or might not be available;
neither does it represent that it has made any effort to identify any such
rights. Information on OASIS's procedures with respect to rights in OASIS
specifications can be found at the OASIS website. Copies of claims of rights
made available for publication and any assurances of licenses to be made
available, or the result of an attempt made to obtain a general license or
permission for the use of such proprietary rights by implementors or users of
this specification, can be obtained from the OASIS Executive Director.
OASIS invites any interested party to bring to its attention any copyrights,
patents or patent applications, or other proprietary rights which may cover
technology that may be required to implement this specification. Please
address the information to the OASIS Executive Director.
Copyright © OASIS Open 2002-2006. All Rights Reserved.
This document and translations of it may be copied and furnished to others,
and derivative works that comment on or otherwise explain it or assist in its
implementation may be prepared, copied, published and distributed, in whole or
in part, without restriction of any kind, provided that the above copyright
notice and this paragraph are included on all such copies and derivative
works. However, this document itself does not be modified in any way, such as
by removing the copyright notice or references to OASIS, except as needed for
the purpose of developing OASIS specifications, in which case the procedures
for copyrights defined in the OASIS Intellectual Property Rights document must
be followed, or as required to translate it into languages other than English.
The limited permissions granted above are perpetual and will not be revoked by
OASIS or its successors or assigns.
This document and the information contained herein is provided on an "AS
IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
FITNESS FOR A PARTICULAR PURPOSE.
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200510"
targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200510"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import
namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
  <!-- Protocol Elements -->
  <xs:complexType name="SequenceType">
    <xs:sequence>
      <xs:element ref="wsrm:Identifier"/>
      <xs:element name="MessageNumber"
type="xs:unsignedLong"/>
      <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

1099         <xs:anyAttribute namespace="##other" processContents="lax"/>
1100     </xs:complexType>
1101     <xs:element name="Sequence" type="wsrm:SequenceType"/>
1102     <xs:element name="SequenceAcknowledgement">
1103         <xs:complexType>
1104             <xs:sequence>
1105                 <xs:element ref="wsrm:Identifier"/>
1106                 <xs:choice>
1107                     <xs:sequence>
1108                         <xs:choice>
1109                             <xs:element
1110 name="AcknowledgementRange" maxOccurs="unbounded">
1111             <xs:complexType>
1112             <xs:sequence/>
1113             <xs:attribute name="Upper" type="xs:unsignedLong" use="required"/>
1114             <xs:attribute name="Lower" type="xs:unsignedLong" use="required"/>
1115             <xs:anyAttribute namespace="##other" processContents="lax"/>
1116         </xs:complexType>
1117     </xs:element>
1118     <xs:element
1119 name="None" minOccurs="0">
1120     <xs:complexType>
1121     <xs:sequence/>
1122 </xs:complexType>
1123 </xs:element>
1124 </xs:choice>
1125 <xs:element name="Final"
1126 minOccurs="0">
1127 <xs:complexType>
1128 <xs:sequence/>
1129 </xs:complexType>
1130 </xs:element>
1131 </xs:sequence>
1132 <xs:element name="Nack"
1133 type="xs:unsignedLong" maxOccurs="unbounded"/>
1134 </xs:choice>
1135 <xs:any namespace="##other"
1136 processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
1137 </xs:sequence>
1138 <xs:anyAttribute namespace="##other"
1139 processContents="lax"/>
1140 </xs:complexType>
1141 </xs:element>
1142 <xs:complexType name="AckRequestedType">
1143     <xs:sequence>
1144         <xs:element ref="wsrm:Identifier"/>
1145         <xs:any namespace="##other" processContents="lax"
1146 minOccurs="0" maxOccurs="unbounded"/>
1147     </xs:sequence>
1148     <xs:anyAttribute namespace="##other" processContents="lax"/>
1149 </xs:complexType>
1150 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>

```

```

1151     <xs:element name="Identifier">
1152         <xs:complexType>
1153             <xs:annotation>
1154                 <xs:documentation>
1155                     This type is for elements whose [children] is an anyURI and can have
1156                     arbitrary attributes.
1157                 </xs:documentation>
1158             </xs:annotation>
1159             <xs:simpleContent>
1160                 <xs:extension base="xs:anyURI">
1161                     <xs:anyAttribute namespace="##other"
1162                     processContents="lax"/>
1163                 </xs:extension>
1164             </xs:simpleContent>
1165         </xs:complexType>
1166     </xs:element>
1167     <!-- Fault Container and Codes -->
1168     <xs:simpleType name="FaultCodes">
1169         <xs:restriction base="xs:QName">
1170             <xs:enumeration value="wsrm:UnknownSequence"/>
1171             <xs:enumeration value="wsrm:SequenceTerminated"/>
1172             <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1173             <xs:enumeration value="wsrm:MessageNumberRollover"/>
1174             <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1175         </xs:restriction>
1176     </xs:simpleType>
1177     <xs:complexType name="SequenceFaultType">
1178         <xs:sequence>
1179             <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1180             <xs:any namespace="##any" processContents="lax"
1181             minOccurs="0" maxOccurs="unbounded"/>
1182         </xs:sequence>
1183         <xs:anyAttribute namespace="##any" processContents="lax"/>
1184     </xs:complexType>
1185     <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1186     <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1187     <xs:element name="CreateSequenceResponse"
1188     type="wsrm:CreateSequenceResponseType"/>
1189     <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1190     <xs:element name="CloseSequenceResponse"
1191     type="wsrm:CloseSequenceResponseType"/>
1192     <xs:element name="TerminateSequence"
1193     type="wsrm:TerminateSequenceType"/>
1194     <xs:complexType name="CreateSequenceType">
1195         <xs:sequence>
1196             <xs:element ref="wsrm:AcksTo"/>
1197             <xs:element ref="wsrm:Expires" minOccurs="0"/>
1198             <xs:element name="Offer" type="wsrm:OfferType"
1199             minOccurs="0"/>
1200             <xs:any namespace="##other" processContents="lax"
1201             minOccurs="0" maxOccurs="unbounded"/>
1202         </xs:sequence>
1203     </xs:complexType>
1204     <xs:annotation>
1205         <xs:documentation>
1206             It is the authors intent that this extensibility be used to
1207             transfer a Security Token Reference as defined in WS-Security.
1208         </xs:documentation>
1209     </xs:annotation>
1210 </xs:sequence>
1211 <xs:anyAttribute namespace="##other" processContents="lax"/>
1212 </xs:complexType>
<xs:complexType name="CreateSequenceResponseType">

```

```

1213         <xs:sequence>
1214             <xs:element ref="wsrm:Identifier"/>
1215             <xs:element ref="wsrm:Expires" minOccurs="0"/>
1216             <xs:element name="Accept" type="wsrm:AcceptType"
1217 minOccurs="0"/>
1218             <xs:any namespace="##other" processContents="lax"
1219 minOccurs="0" maxOccurs="unbounded"/>
1220         </xs:sequence>
1221         <xs:anyAttribute namespace="##other" processContents="lax"/>
1222     </xs:complexType>
1223     <xs:complexType name="CloseSequenceType">
1224         <xs:sequence>
1225             <xs:any namespace="##other" processContents="lax"
1226 minOccurs="0" maxOccurs="unbounded"/>
1227         </xs:sequence>
1228         <xs:attribute name="Identifier" type="xs:anyURI"
1229 use="required"/>
1230         <xs:anyAttribute namespace="##other" processContents="lax"/>
1231     </xs:complexType>
1232     <xs:complexType name="CloseSequenceResponseType">
1233         <xs:sequence>
1234             <xs:any namespace="##other" processContents="lax"
1235 minOccurs="0" maxOccurs="unbounded"/>
1236         </xs:sequence>
1237         <xs:anyAttribute namespace="##other" processContents="lax"/>
1238     </xs:complexType>
1239     <xs:complexType name="TerminateSequenceType">
1240         <xs:sequence>
1241             <xs:element ref="wsrm:Identifier"/>
1242             <xs:any namespace="##other" processContents="lax"
1243 minOccurs="0" maxOccurs="unbounded"/>
1244         </xs:sequence>
1245         <xs:anyAttribute namespace="##other" processContents="lax"/>
1246     </xs:complexType>
1247     <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
1248     <xs:complexType name="OfferType">
1249         <xs:sequence>
1250             <xs:element ref="wsrm:Identifier"/>
1251             <xs:element ref="wsrm:Expires" minOccurs="0"/>
1252             <xs:any namespace="##other" processContents="lax"
1253 minOccurs="0" maxOccurs="unbounded"/>
1254         </xs:sequence>
1255         <xs:anyAttribute namespace="##other" processContents="lax"/>
1256     </xs:complexType>
1257     <xs:complexType name="AcceptType">
1258         <xs:sequence>
1259             <xs:element ref="wsrm:AcksTo"/>
1260             <xs:any namespace="##other" processContents="lax"
1261 minOccurs="0" maxOccurs="unbounded"/>
1262         </xs:sequence>
1263         <xs:anyAttribute namespace="##other" processContents="lax"/>
1264     </xs:complexType>
1265     <xs:element name="Expires">
1266         <xs:complexType>
1267             <xs:simpleContent>
1268                 <xs:extension base="xs:duration">
1269                     <xs:anyAttribute namespace="##other"
1270 processContents="lax"/>
1271                 </xs:extension>
1272             </xs:simpleContent>
1273         </xs:complexType>
1274     </xs:element>

```

```

1275 </xs:schema>
1276
1277 <xs:schema targetNamespace="http://docs.oasis-open.org/wsrn/200510/"
1278 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1279 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1280 xmlns:wsrm="http://docs.oasis-open.org/wsrn/200510/"
1281 elementFormDefault="qualified" attributeFormDefault="unqualified"
1282 <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1283 schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
1284 <!-- Protocol Elements -->
1285 <xs:complexType name="SequenceType">
1286 <xs:sequence>
1287 <xs:element ref="wsrm:Identifier"/>
1288 <xs:element name="MessageNumber" type="xs:unsignedLong"/>
1289 <xs:element name="LastMessage" minOccurs="0">
1290 <xs:complexType>
1291 <xs:sequence/>
1292 </xs:complexType>
1293 </xs:element>
1294 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1295 maxOccurs="unbounded"/>
1296 </xs:sequence>
1297 <xs:anyAttribute namespace="##other" processContents="lax"/>
1298 </xs:complexType>
1299 <xs:element name="Sequence" type="wsrm:SequenceType"/>
1300 <xs:element name="SequenceAcknowledgement">
1301 <xs:complexType>
1302 <xs:sequence>
1303 <xs:element ref="wsrm:Identifier"/>
1304 <xs:choice>
1305 <ws:sequence>
1306 <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
1307 <xs:complexType>
1308 <xs:sequence/>
1309 <xs:attribute name="Upper" type="xs:unsignedLong"
1310 use="required"/>
1311 <xs:attribute name="Lower" type="xs:unsignedLong"
1312 use="required"/>
1313 <xs:anyAttribute namespace="##other" processContents="lax"/>
1314 </xs:complexType>
1315 </xs:element>
1316 <ws:element name="Final" minOccurs="0">
1317 <xs:complexType>
1318 <xs:sequence/>
1319 </xs:complexType>
1320 </ws:element>
1321 </ws:sequence>
1322 <xs:element name="Nack" type="xs:unsignedLong"
1323 maxOccurs="unbounded"/>
1324 <xs:element name="None" minOccurs="0">
1325 <xs:complexType>
1326 <xs:sequence/>
1327 </xs:complexType>
1328 </xs:element>
1329 </xs:choice>
1330 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1331 maxOccurs="unbounded"/>
1332 </xs:sequence>
1333 <xs:anyAttribute namespace="##other" processContents="lax"/>
1334 </xs:complexType>
1335 </xs:element>
1336 <xs:complexType name="AckRequestedType">

```

```

1336 <xs:sequence>
1337 <xs:element ref="wsrm:Identifier"/>
1338 <xs:element name="MessageNumber" type="xs:unsignedLong" minOccurs="0"/>
1339 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1340 maxOccurs="unbounded"/>
1341 </xs:sequence>
1342 <xs:anyAttribute namespace="##other" processContents="lax"/>
1343 </xs:complexType>
1344 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1345 <xs:element name="Identifier">
1346 <xs:complexType>
1347 <xs:annotation>
1348 <xs:documentation>
1349 This type is for elements whose [children] is an anyURI and can have arbitrary
1350 attributes.
1351 </xs:documentation>
1352 </xs:annotation>
1353 <xs:simpleContent>
1354 <xs:extension base="xs:anyURI">
1355 <xs:anyAttribute namespace="##other" processContents="lax"/>
1356 </xs:extension>
1357 </xs:simpleContent>
1358 </xs:complexType>
1359 </xs:element>
1360 <!-- Fault Container and Codes -->
1361 <xs:simpleType name="FaultCodes">
1362 <xs:restriction base="xs:QName">
1363 <xs:enumeration value="wsrm:UnknownSequence"/>
1364 <xs:enumeration value="wsrm:SequenceTerminated"/>
1365 <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1366 <xs:enumeration value="wsrm:MessageNumberRollover"/>
1367 <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1368 <xs:enumeration value="wsrm:LastMessageNumberExceeded"/>
1369 </xs:restriction>
1370 </xs:simpleType>
1371 <xs:complexType name="SequenceFaultType">
1372 <xs:sequence>
1373 <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1374 <xs:any namespace="##any" processContents="lax" minOccurs="0"
1375 maxOccurs="unbounded"/>
1376 </xs:sequence>
1377 <xs:anyAttribute namespace="##any" processContents="lax"/>
1378 </xs:complexType>
1379 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1380 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1381 <xs:element name="CreateSequenceResponse"
1382 type="wsrm:CreateSequenceResponseType"/>
1383 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1384 <xs:element name="CloseSequenceResponse"
1385 type="wsrm:CloseSequenceResponseType"/>
1386 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1387 <xs:complexType name="CreateSequenceType">
1388 <xs:sequence>
1389 <xs:element ref="wsrm:AcksTo"/>
1390 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1391 <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1392 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1393 maxOccurs="unbounded"/>
1394 <xs:annotation>
1395 <xs:documentation>
1396 It is the authors intent that this extensibility be used to transfer a
1397 Security Token Reference as defined in WS-Security.
1398 </xs:documentation>

```



```

1399 </xs:annotation>
1400 </xs:any>
1401 </xs:sequence>
1402 <xs:anyAttribute namespace="##other" processContents="lax"/>
1403 </xs:complexType>
1404 <xs:complexType name="CreateSequenceResponseType">
1405 <xs:sequence>
1406 <xs:element ref="wsrm:Identifier"/>
1407 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1408 <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1409 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1410 maxOccurs="unbounded"/>
1411 </xs:sequence>
1412 <xs:anyAttribute namespace="##other" processContents="lax"/>
1413 </xs:complexType>
1414 <xs:complexType name="CloseSequenceType">
1415 <xs:sequence>
1416 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1417 maxOccurs="unbounded"/>
1418 </xs:sequence>
1419 <xs:attribute name="Identifier" type="xs:anyURI" use="required"/>
1420 <xs:anyAttribute namespace="##other" processContents="lax"/>
1421 </xs:complexType>
1422 <xs:complexType name="CloseSequenceResponseType">
1423 <xs:sequence>
1424 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1425 maxOccurs="unbounded"/>
1426 </xs:sequence>
1427 <xs:anyAttribute namespace="##other" processContents="lax"/>
1428 </xs:complexType>
1429 <xs:complexType name="TerminateSequenceType">
1430 <xs:sequence>
1431 <xs:element ref="wsrm:Identifier"/>
1432 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1433 maxOccurs="unbounded"/>
1434 </xs:sequence>
1435 <xs:anyAttribute namespace="##other" processContents="lax"/>
1436 </xs:complexType>
1437 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
1438 <xs:complexType name="OfferType">
1439 <xs:sequence>
1440 <xs:element ref="wsrm:Identifier"/>
1441 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1442 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1443 maxOccurs="unbounded"/>
1444 </xs:sequence>
1445 <xs:anyAttribute namespace="##other" processContents="lax"/>
1446 </xs:complexType>
1447 <xs:complexType name="AcceptType">
1448 <xs:sequence>
1449 <xs:element ref="wsrm:AcksTo"/>
1450 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1451 maxOccurs="unbounded"/>
1452 </xs:sequence>
1453 <xs:anyAttribute namespace="##other" processContents="lax"/>
1454 </xs:complexType>
1455 <xs:element name="Expires">
1456 <xs:complexType>
1457 <xs:simpleContent>
1458 <xs:extension base="xs:duration">
1459 <xs:anyAttribute namespace="##other" processContents="lax"/>
1460 </xs:extension>
1461 </xs:simpleContent>

```



1462	<del>&lt;/xs:complexType&gt;</del>
1463	<del>&lt;/xs:element&gt;</del>
1464	<del>&lt;/xs:schema&gt;</del>

## 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/200510-rm/200510/"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
  <S:Header>
    <wsa:MessageID>
      http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
    </wsa:MessageID>
    <wsa:To>http://example.com/serviceB/123</wsa:To>
    <wsa:Action>http://docs.oasis-open.org/ws-rx/wsmr/200510/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

#### 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/200510-rm/200510/"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
  <S:Header>
    <wsa:To>http://Business456.com/serviceA/789</wsa:To>
    <wsa:RelatesTo>
      http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
    </wsa:RelatesTo>
    <wsa:Action>
      http://docs.oasis-open.org/ws-rx/wsmr/200510/CreateSequenceResponse
    </wsa:Action>
  </S:Header>
  <S:Body>
    <wsmr:CreateSequenceResponse>
      <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
    </wsmr:CreateSequenceResponse>
  </S:Body>
</S:Envelope>
```

## B.2 -Initial Transmission

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

### Message 1

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200510rm/200510/"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
  <S:Header>
    <wsa:MessageID>
      http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
    </wsa:MessageID>
    <wsa:To>http://example.com/serviceB/123</wsa:To>
    <wsa:From>
      <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
    </wsa:From>
    <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
    <wsm:Sequence>
      <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
      <wsm:MessageNumber>1</wsm:MessageNumber>
    </wsm:Sequence>
  </S:Header>
  <S:Body>
    <!-- Some Application Data -->
  </S:Body>
</S:Envelope>
```

### Message 2

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200510rm/200510/"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
  <S:Header>
    <wsa:MessageID>
      http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
    </wsa:MessageID>
    <wsa:To>http://example.com/serviceB/123</wsa:To>
    <wsa:From>
      <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
    </wsa:From>
    <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
    <wsm:Sequence>
      <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
      <wsm:MessageNumber>2</wsm:MessageNumber>
    </wsm:Sequence>
  </S:Header>
  <S:Body>
    <!-- Some Application Data -->
  </S:Body>
</S:Envelope>
```

### Message 3

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200510rm/200510/"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
```

```

1567 <S:Header>
1568   <wsa:MessageID>
1569     http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1570   </wsa:MessageID>
1571   <wsa:To>http://example.com/serviceB/123</wsa:To>
1572   <wsa:From>
1573     <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1574   </wsa:From>
1575   <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1576   <wsrm:Sequence>
1577     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1578     <wsrm:MessageNumber>3</wsrm:MessageNumber>
1579   <del wsrm:LastMessage />
1580   </wsrm:Sequence>
1581   <del wsrm:AckRequested />
1582   <del wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1583   <del /wsrm:AckRequested />
1584 </S:Header>
1585 <S:Body>
1586   <!-- Some Application Data -->
1587 </S:Body>
1588 </S:Envelope>

```

### 1589 B.3 First Acknowledgement

1590 Message number 2 has not been received by the RM Destination due to some transmission error so it  
1591 responds with an acknowledgement for messages 1 and 3:

```

1592 <?xml version="1.0" encoding="UTF-8"?>
1593 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1594   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200510rm/200510/"
1595   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1596   <S:Header>
1597     <wsa:MessageID>
1598       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1599     </wsa:MessageID>
1600     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1601     <wsa:From>
1602       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1603     </wsa:From>
1604     <wsa:Action>
1605       http://docs.oasis-open.org/ws-rx/wsrm/200510/SequenceAcknowledgement
1606     </wsa:Action>
1607     <wsrm:SequenceAcknowledgement>
1608       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1609       <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
1610       <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
1611     </wsrm:SequenceAcknowledgement>
1612   </S:Header>
1613   <S:Body/>
1614 </S:Envelope>

```

### 1615 B.4 Retransmission

1616 The ~~RM Sourcesending endpoint~~ discovers that message number 2 was not received so it resends the  
1617 message and requests an acknowledgement:

```

1618 <?xml version="1.0" encoding="UTF-8"?>
1619 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1620   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200510rm/200510/"
1621   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">

```

```

1622 <S:Header>
1623   <wsa:MessageID>
1624     http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1625   </wsa:MessageID>
1626   <wsa:To>http://example.com/serviceB/123</wsa:To>
1627   <wsa:From>
1628     <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1629   </wsa:From>
1630   <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1631   <wsrm:Sequence>
1632     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1633     <wsrm:MessageNumber>2</wsrm:MessageNumber>
1634   </wsrm:Sequence>
1635   <wsrm:AckRequested>
1636     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1637   </wsrm:AckRequested>
1638 </S:Header>
1639 <S:Body>
1640   <!-- Some Application Data -->
1641 </S:Body>
1642 </S:Envelope>

```

## 1643 B.5 Termination

1644 The RM Destination now responds with an acknowledgement for the complete sequence which can then  
 1645 be terminated:

```

1646 <?xml version="1.0" encoding="UTF-8"?>
1647 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1648   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200510rm/200510/"
1649   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1650   <S:Header>
1651     <wsa:MessageID>
1652       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1653     </wsa:MessageID>
1654     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1655     <wsa:From>
1656       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1657     </wsa:From>
1658     <wsa:Action>
1659       http://docs.oasis-open.org/ws-rx/wsr/200510/SequenceAcknowledgement
1660     </wsa:Action>
1661     <wsrm:SequenceAcknowledgement>
1662       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1663       <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
1664     </wsrm:SequenceAcknowledgement>
1665   </S:Header>
1666   <S:Body/>
1667 </S:Envelope>

```

## 1668 Terminate Sequence

```

1669 <?xml version="1.0" encoding="UTF-8"?>
1670 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1671   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200510rm/200510/"
1672   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1673   <S:Header>
1674     <wsa:MessageID>
1675       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1676     </wsa:MessageID>
1677     <wsa:To>http://example.com/serviceB/123</wsa:To>
1678     <wsa:Action>

```

```
1679      http://docs.oasis-open.org/ws-rx/wsrm/200510/TerminateSequence
1680      </wsa:Action>
1681      <wsa:From>
1682        <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1683      </wsa:From>
1684    </S:Header>
1685    <S:Body>
1686      <wsrm:TerminateSequence>
1687        <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1688      </wsrm:TerminateSequence>
1689    </S:Body>
1690  </S:Envelope>
```

## C. WSDL

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

<http://docs.oasis-open.org/ws-rx/wsrn/200510/wsd/wsrn-1.1-wsd-200510-rm/200510/wsd/wsrn.wsd>

The following non-normative copy is provided for reference.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
OASIS takes no position regarding the validity or scope of any intellectual
property or other rights that might be claimed to pertain to the
implementation or use of the technology described in this document or the
extent to which any license under such rights might or might not be available;
neither does it represent that it has made any effort to identify any such
rights. Information on OASIS's procedures with respect to rights in OASIS
specifications can be found at the OASIS website. Copies of claims of rights
made available for publication and any assurances of licenses to be made
available, or the result of an attempt made to obtain a general license or
permission for the use of such proprietary rights by implementors or users of
this specification, can be obtained from the OASIS Executive Director.
OASIS invites any interested party to bring to its attention any copyrights,
patents or patent applications, or other proprietary rights which may cover
technology that may be required to implement this specification. Please
address the information to the OASIS Executive Director.
Copyright © OASIS Open 2002-2006. All Rights Reserved.
This document and translations of it may be copied and furnished to others,
and derivative works that comment on or otherwise explain it or assist in its
implementation may be prepared, copied, published and distributed, in whole or
in part, without restriction of any kind, provided that the above copyright
notice and this paragraph are included on all such copies and derivative
works. However, this document itself does not be modified in any way, such as
by removing the copyright notice or references to OASIS, except as needed for
the purpose of developing OASIS specifications, in which case the procedures
for copyrights defined in the OASIS Intellectual Property Rights document must
be followed, or as required to translate it into languages other than English.
The limited permissions granted above are perpetual and will not be revoked by
OASIS or its successors or assigns.
This document and the information contained herein is provided on an "AS
IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
BUT NOT LIMITED TO WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
FITNESS FOR A PARTICULAR PURPOSE.
-->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:rm="http://docs.oasis-open.org/ws-rx/wsrn/200510"
xmlns:tns="http://docs.oasis-open.org/ws-rx/wsrn/200510/wsd"
targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200510/wsd">
  <wsdl:types>
    <xs:schema>
      <xs:import namespace="http://docs.oasis-open.org/ws-
rx/wsrn/200510" schemaLocation="http://docs.oasis-open.org/ws-
rx/wsrn/200510/wsrn-1.1-schema-200510.xsd"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="CreateSequence">
    <wsdl:part name="create" element="rm:CreateSequence"/>
  </wsdl:message>
</wsdl:definitions>
```

```

1747     </wsdl:message>
1748     <wsdl:message name="CreateSequenceResponse">
1749         <wsdl:part name="createResponse"
1750 element="rm:CreateSequenceResponse"/>
1751     </wsdl:message>
1752     <wsdl:message name="CloseSequence">
1753         <wsdl:part name="close" element="rm:CloseSequence"/>
1754     </wsdl:message>
1755     <wsdl:message name="CloseSequenceResponse">
1756         <wsdl:part name="closeResponse"
1757 element="rm:CloseSequenceResponse"/>
1758     </wsdl:message>
1759     <wsdl:message name="TerminateSequence">
1760         <wsdl:part name="terminate" element="rm:TerminateSequence"/>
1761     </wsdl:message>
1762     <wsdl:portType name="SequenceAbstractPortType">
1763         <wsdl:operation name="CreateSequence">
1764             <wsdl:input message="tns:CreateSequence"
1765 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200510/CreateSequence"/>
1766             <wsdl:output message="tns:CreateSequenceResponse"
1767 wsa:Action="http://docs.oasis-open.org/ws-
1768 rx/wsrn/200510/CreateSequenceResponse"/>
1769         </wsdl:operation>
1770         <wsdl:operation name="CloseSequence">
1771             <wsdl:input message="tns:CloseSequence"
1772 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200510/CloseSequence"/>
1773             <wsdl:output message="tns:CloseSequenceResponse"
1774 wsa:Action="http://docs.oasis-open.org/ws-
1775 rx/wsrn/200510/CloseSequenceResponse"/>
1776         </wsdl:operation>
1777         <wsdl:operation name="TerminateSequence">
1778             <wsdl:input message="tns:TerminateSequence"
1779 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200510/TerminateSequence"/>
1780         </wsdl:operation>
1781     </wsdl:portType>
1782 </wsdl:definitions>

1783 <del><wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
1784 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1785 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1786 xmlns:rm="http://docs.oasis-open.org/wsrn/200510/"
1787 xmlns:tns="http://docs.oasis-open.org/wsrn/200510/wsdl"
1788 targetNamespace="http://docs.oasis-open.org/wsrn/200510/wsdl"><wsdl:types>
1789     <xs:schema
1790     <del><xs:import namespace="http://docs.oasis-open.org/wsrn/200510/"
1791 schemaLocation="http://docs.oasis-open.org/wsrn/200510/wsrn.xsd"/>
1792     <del><xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1793 schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
1794     <del></xs:schema>
1795     <del></wsdl:types>
1796     <del><wsdl:message name="CreateSequence">
1797     <del><wsdl:part name="create" element="rm:CreateSequence"/>
1798     <del></wsdl:message>
1799     <del><wsdl:message name="CreateSequenceResponse">
1800     <del><wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
1801     <del></wsdl:message>
1802     <del><wsdl:message name="CloseSequence">
1803     <del><wsdl:part name="close" element="rm:CloseSequence"/>
1804     <del></wsdl:message>
1805     <del><wsdl:message name="CloseSequenceResponse">
1806     <del><wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
1807     <del></wsdl:message>

```



```

1808 <wsdl:message name="TerminateSequence">
1809 <wsdl:part name="terminate" element="rm:TerminateSequence"/>
1810 </wsdl:message>
1811 <wsdl:portType name="SequenceAbstractPortType">
1812 <wsdl:operation name="CreateSequence">
1813 <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
1814 open.org/wsrn/200510/CreateSequence"/>
1815 <wsdl:output message="tns:CreateSequenceResponse"
1816 wsa:Action="http://docs.oasis-open.org/wsrn/200510/CreateSequenceResponse"/>
1817 </wsdl:operation>
1818 <wsdl:operation name="CloseSequence">
1819 <wsdl:input name="tns:CloseSequence" wsa:Action="http://docs.oasis-
1820 open.org/wsrn/200510/CloseSequence"/>
1821 <wsdl:output name="tns:CloseSequenceResponse"
1822 wsa:Action="http://docs.oasis-open.org/wsrn/200510/CloseSequenceResponse"/>
1823 </wsdl:operation>
1824 <wsdl:operation name="TerminateSequence">
1825 <wsdl:input message="tns:TerminateSequence"
1826 wsa:Action="http://docs.oasis-open.org/wsrn/200510/CreateSequenceResponse"/>
1827 </wsdl:operation>
1828 </wsdl:portType>
1829 </wsdl:definitions>

```

## D. 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~~

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*

## E. Revision History

Rev	Date	By Whom	What
<a href="#">wd-01</a>	<a href="#">2005-07-07</a>	<a href="#">Christopher Ferris</a>	<a href="#">Initial version created based on submission by the authors.</a>
<a href="#">ws-02</a>	<a href="#">2005-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">I011 (PTOS) added</a>
<a href="#">wd-02</a>	<a href="#">2005-08-16</a>	<a href="#">Anish Karmarkar</a>	<a href="#">Trivial editorial changes</a>
<a href="#">ws-03</a>	<a href="#">2005-09-15</a>	<a href="#">Doug Davis</a>	<a href="#">I019 and i028 (CloseSeq) added</a>
<a href="#">wd-05</a>	<a href="#">2005-09-26</a>	<a href="#">Gilbert Pilz</a>	<a href="#">i005 (Source resend of nacks messages when ack already received) added.</a>
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i027 (InOrder delivery assurance spanning multiple sequences) added</a>
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i020 (Semantics of "At most once" Delivery Assurance) added</a>
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i034 (Fault while processing a piggy-backed RM header) added</a>
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i033 (Processing model of NACKs) added</a>
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i031 (AckRequested schema inconsistency) added</a>
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i025 (SeqAck/None) added</a>
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i029 (Remove dependency on WS-Security) added</a>
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i039 (What does 'have a mU attribute' mean) added</a>
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i040 (Change 'optiona'/'required' to 'OPTIONAL'/'REQUIRED') added</a>
<a href="#">wd-05</a>	<a href="#">2005-09-30</a>	<a href="#">Anish Karmarkar</a>	<a href="#">i017 (Change NS to <a href="http://docs.oasis-open.org/wsrn/200510/">http://docs.oasis-open.org/wsrn/200510/</a>)</a>
<a href="#">wd-05</a>	<a href="#">2005-09-30</a>	<a href="#">Anish Karmarkar</a>	<a href="#">i045 (Include SecureConversation as a reference and move it to non-normative citation)</a>
<a href="#">wd-05</a>	<a href="#">2005-09-30</a>	<a href="#">Anish Karmarkar</a>	<a href="#">i046 (change the type of wsrn:FaultCode element)</a>
<a href="#">wd-06</a>	<a href="#">2005-11-02</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Start wd-06 by changing title page from cd-01.</a>
<a href="#">wd-06</a>	<a href="#">2005-11-03</a>	<a href="#">Gilbert Pilz</a>	<a href="#">i047 (Reorder spec sections)</a>
<a href="#">wd-07</a>	<a href="#">2005-11-17</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Start wd-07</a>
<a href="#">wd-07</a>	<a href="#">2005-11-28</a>	<a href="#">Doug Davis</a>	<a href="#">i071 – except for period in Appendix headings</a>
<a href="#">wd-07</a>	<a href="#">2005-11-28</a>	<a href="#">Doug Davis</a>	<a href="#">i10</a>
<a href="#">wd-07</a>	<a href="#">2005-11-28</a>	<a href="#">Doug Davis</a>	<a href="#">i030</a>
<a href="#">wd-07</a>	<a href="#">2005-11-28</a>	<a href="#">Doug Davis</a>	<a href="#">i037</a>
<a href="#">wd-07</a>	<a href="#">2005-11-28</a>	<a href="#">Doug Davis</a>	<a href="#">i038</a>
<a href="#">wd-07</a>	<a href="#">2005-11-28</a>	<a href="#">Doug Davis</a>	<a href="#">i041</a>
<a href="#">wd-07</a>	<a href="#">2005-11-28</a>	<a href="#">Doug Davis</a>	<a href="#">i043</a>
<a href="#">wd-07</a>	<a href="#">2005-11-28</a>	<a href="#">Doug Davis</a>	<a href="#">i044</a>

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

Rev	Date	By Whom	What
			OpenDocument Text format. Changed line numbers to be a single style.
<a href="#">wd-08</a>	<a href="#">2005-12-28</a>	<a href="#">Anish Karmarkar</a>	Included a section link to <a href="#">c:\temp\wsrm-1.1-schema-200510.xsd</a> and to <a href="#">c:\temp\wsrm-1.1-wsdl-200510.wsdl</a>
<a href="#">wd-08</a>	<a href="#">2006-01-04</a>	<a href="#">Gilbert Pilz</a>	Fixed formatting for included sections.
<a href="#">wd-08</a>	<a href="#">2006-01-05</a>	<a href="#">Gilbert Pilz</a>	Created links for unused references. Fixed exemplars for <a href="#">CloseSequence</a> and <a href="#">CloseSequenceResponse</a> .
<a href="#">wd-09</a>	<a href="#">2006-01-11</a>	<a href="#">Doug Davis</a>	Minor tweaks to text/typos.
<a href="#">cd-02</a>	<a href="#">2006-01-13</a>	<a href="#">Gilbert Pilz</a>	Titles, boilerplate, etc. for <a href="#">cd-02</a>
Rev	Date	By Whom	What
<a href="#">wd-01</a>	<a href="#">2005-07-07</a>	<a href="#">Christopher Ferris</a>	Initial version created based on submission by the authors.
<a href="#">ws-02</a>	<a href="#">2005-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">i011 (PTOS)</a> added
<a href="#">wd-02</a>	<a href="#">2005-08-16</a>	<a href="#">Anish Karmarkar</a>	Trivial editorial changes
<a href="#">ws-03</a>	<a href="#">2005-09-15</a>	<a href="#">Doug Davis</a>	<a href="#">i019</a> and <a href="#">i028 (CloseSeq)</a> added
<a href="#">wd-05</a>	<a href="#">2005-09-26</a>	<a href="#">Gilbert Pilz</a>	<a href="#">i005 (Source resend of nacks messages when ack already received)</a> added.
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i027 (InOrder delivery assurance spanning multiple sequences)</a> added
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i020 (Semantics of "At most once" Delivery Assurance)</a> added
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i034 (Fault while processing a piggy-backed RM header)</a> added
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i033 (Processing model of NACKs)</a> added
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i031 (AckRequested schema inconsistency)</a> added
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i025 (SeqAck/None)</a> added
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i029 (Remove dependency on WS-Security)</a> added
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i039 (What does 'have a mU attribute' mean)</a> added
<a href="#">wd-05</a>	<a href="#">2005-09-27</a>	<a href="#">Doug Davis</a>	<a href="#">i040 (Change 'optiona'/'required' to 'OPTIONAL'/'REQUIRED')</a> added
<a href="#">wd-05</a>	<a href="#">2005-09-30</a>	<a href="#">Anish Karmarkar</a>	<a href="#">i017 (Change NS to <a href="#">http://docs.oasis-open.org/wsrm/200510/</a>)</a>
<a href="#">wd-05</a>	<a href="#">2005-09-30</a>	<a href="#">Anish Karmarkar</a>	<a href="#">i045 (Include SecureConversation as a reference and move it to non-normative-citation)</a>
<a href="#">wd-05</a>	<a href="#">2005-09-30</a>	<a href="#">Anish Karmarkar</a>	<a href="#">i046 (change the type of <a href="#">wsrm:FaultCode</a> element)</a>

Rev	Date	By Whom	What
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)

## F. Notices

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

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

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

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

Copyright (C) OASIS Open (2005). All Rights Reserved.

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

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

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

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