



1 **Web Services Reliable Messaging**  
2 **(WS-ReliableMessaging**  
3 **(WS-Reliable-Messaging)**

4 **Committee Draft 04, August 11, 2006 ~~Draft 03, August 17,~~**  
5 **2006**

6 **Document identifier:**

7 wsrn-1.1-spec-cd-043

8 **Location:**

9 <http://docs.oasis-open.org/ws-rx/wsrn/200608/wsrn-1.1-spec-cd-042/wsrn-1.1-spec-ed-03.pdf>

10 **Editors:**

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

16 **Contributors:**

17 [See the Acknowledgments \(Appendix E\).](#)  
18 **TBD**

19 **Abstract:**

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

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

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

34 **Status:**

35 This document [was last revised or approved by the WS-RX on the above date. The level of approval is](#)  
36 [also listed above. Check the current location noted above for possible later revisions of this document.](#)  
37 [This document is updated periodically on no particular schedule. Technical Committee members should](#)

38 [send comments on this specification to the Technical Committee's email list. Others should send](#)  
39 [comments to the Technical Committee by using the "Send A Comment" button on the Technical](#)  
40 [Committee's web page at http://www.oasis-open.org/committees/ws-rx. For information on](#)  
41 [whether any patents have been disclosed that may be essential to implementing this specification,](#)  
42 [and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of](#)  
43 [the Technical Committee web page \(http://www.oasis-open.org/committees/ws-rx/ipr.php\). The](#)  
44 [non-normative errata page for this specification is located at http://www.oasis-](#)  
45 [open.org/committees/ws-rx](#). ~~is a work in progress and will be updated to reflect issues as they are~~  
46 ~~resolved by the Web Services Reliable Exchange (WS-RX) Technical Committee.~~

# 47 **Table of Contents**

48	1 Introduction.....	4
49	1.1 Notational Conventions.....	4
50	1.2 Namespace.....	5
51	1.3 Compliance.....	5
52	2 Reliable Messaging Model.....	6
53	2.1 Glossary.....	6
54	2.2 Protocol Preconditions.....	7
55	2.3 Protocol Invariants.....	7
56	2.4 Example Message Exchange.....	8
57	3 RM Protocol Elements.....	10
58	3.1 Considerations on the Use of Extensibility Points.....	10
59	3.2 Considerations on the Use of "Piggy-Backing".....	10
60	3.3 Composition with WS-Addressing.....	10
61	3.4 Sequence Creation.....	10
62	3.5 Closing A Sequence.....	15
63	3.6 Sequence Termination.....	16
64	3.7 Sequences.....	18
65	3.8 Request Acknowledgement.....	19
66	3.9 Sequence Acknowledgement.....	20
67	3.10 MakeConnection.....	22
68	3.11 MessagePending.....	24
69	4 Faults.....	25
70	4.1 SequenceFault Element.....	26
71	4.2 Sequence Terminated.....	27
72	4.3 Unknown Sequence.....	27
73	4.4 Invalid Acknowledgement.....	28
74	4.5 Message Number Rollover.....	28
75	4.6 Create Sequence Refused.....	29
76	4.7 Sequence Closed.....	29
77	4.8 WSRM Required.....	30
78	4.9 Unsupported Selection.....	30
79	5 Security Threats and Countermeasures.....	32
80	5.1 Threats and Countermeasures.....	32
81	5.1.1 Integrity Threats.....	32
82	5.1.1.1 Countermeasures.....	32
83	5.1.2 Resource Consumption Threats.....	33
84	5.1.2.1 Countermeasures.....	33

85	5.1.3 Sequence Spoofing Threats.....	33
86	5.1.3.1 Sequence Hijacking.....	33
87	5.1.3.2 Countermeasures.....	33
88	5.2 Security Solutions and Technologies.....	34
89	5.2.1 Transport Layer Security.....	34
90	5.2.1.1 Model.....	34
91	5.2.1.2 Countermeasure Implementation.....	35
92	5.2.2 SOAP Message Security.....	36
93	5.2.2.1 Model.....	36
94	5.2.2.2 Countermeasure Implementation.....	36
95	6 Securing Sequences.....	38
96	6.1 Securing Sequences Using WS-Security.....	38
97	6.2 Securing Sequences Using SSL/TLS.....	39
98	7 References.....	41
99	7.1 Normative.....	41
100	7.2 Non-Normative.....	41
101	Appendix A. Schema.....	43
102	Appendix B. WSDL.....	48
103	Appendix C. Message Examples.....	50
104	Appendix C.1 Create Sequence.....	50
105	Appendix C.2 Initial Transmission.....	50
106	Appendix C.3 First Acknowledgement.....	52
107	Appendix C.4 Retransmission.....	52
108	Appendix C.5 Termination.....	53
109	Appendix C.6 MakeConnection.....	54
110	Appendix D. State Tables.....	58
111	Appendix E. Acknowledgments.....	63
112	Appendix F. Revision History.....	64
113	Appendix G. Notices.....	70

# 114 1 Introduction

115 It is often a requirement for two Web services that wish to communicate to do so reliably in the presence  
116 of software component, system, or network failures. The primary goal of this specification is to create a  
117 modular mechanism for reliable [transfer of messages. It defines a messaging protocol to identify, track,  
118 and manage the reliable transfer of messages between a source and a destination. It also defines a  
119 SOAP binding that is required for interoperability. Additional bindings candelivery of messages. It defines  
120 a messaging protocol to identify, track, and manage the reliable delivery of messages between a source  
121 and a destination. It also defines a SOAP binding that is required for interoperability. Additional bindings  
122 may](#) be defined.

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

## 127 1.1 Goals and Requirements

### 128 1.1.1 Requirements

### 129 1.1 Notational Conventions

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

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

- 134 • The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- 135 • Characters are appended to elements and attributes to indicate cardinality:
  - 136 ○ "?" (0 or 1)
  - 137 ○ "\*" (0 or more)
  - 138 ○ "+" (1 or more)
- 139 • The character "|" is used to indicate a choice between alternatives.
- 140 • The characters "[" and "]" are used to indicate that contained items are to be treated as a group  
141 with respect to cardinality or choice.
- 142 • An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content  
143 specified in this document. Additional children elements and/or attributes MAY be added at the  
144 indicated extension points but they MUST NOT contradict the semantics of the parent and/or  
145 owner, respectively. If an extension is not recognized it SHOULD be ignored.
- 146 • XML namespace prefixes (See Section [1.2 Namespace](#)) are used to indicate the namespace of  
147 the element being defined.

148 [Elements and Attributes defined by this specification are referred to in the text of this document using  
149 XPath 1.0 \[XPATH 1.0\] expressions. Extensibility points are referred to using an extended version of this  
150 syntax:](#)

- 151 • [An element extensibility point is referred to using {any} in place of the element name. This](#)  
152 [indicates that any element name can be used, from any namespace other than the wsrml:](#)  
153 [namespace.](#)
- 154 • [An attribute extensibility point is referred to using @{any} in place of the attribute name. This](#)  
155 [indicates that any attribute name can be used, from any namespace other than the wsrml:](#)  
156 [namespace.](#)

## 157 1.2 Namespace

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

159 <http://docs.oasis-open.org/ws-rx/wsrml/2006082>

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

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

164 ~~The following namespaces are used in this document:~~

165 Table 1

Prefix	Namespace
S	(Either SOAP 1.1 or 1.2)
S11	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>
S12	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>
wsrml	<a href="http://docs.oasis-open.org/ws-rx/wsrml/200608">http://docs.oasis-open.org/ws-rx/wsrml/200608</a>
wsa	<a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a>
wsaw	<a href="http://www.w3.org/2006/05/addressing/wsdl">http://www.w3.org/2006/05/addressing/wsdl</a>
wsse	<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>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

166 ~~The normative schema for WS-ReliableMessaging can be found linked from the namespace document~~  
167 ~~that is located at the namespace URI specified above.~~

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>
wsrml	<a href="http://docs.oasis-open.org/ws-rx/wsrml/200602">http://docs.oasis-open.org/ws-rx/wsrml/200602</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>

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

169 ~~<http://docs.oasis-open.org/ws-rx/wsrml/200602/wsrml-1.1.xsd>~~

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

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

174

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

### 175 **1.3 Compliance**

176 An implementation is not compliant with this specification if it fails to satisfy one or more of the MUST or  
177 REQUIRED level requirements defined herein. A SOAP Node MUST NOT use the XML namespace  
178 identifier for this specification (listed in Section [1.2Namespace](#)) within SOAP Envelopes unless it is  
179 compliant with this specification.

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

## 2 Reliable Messaging Model

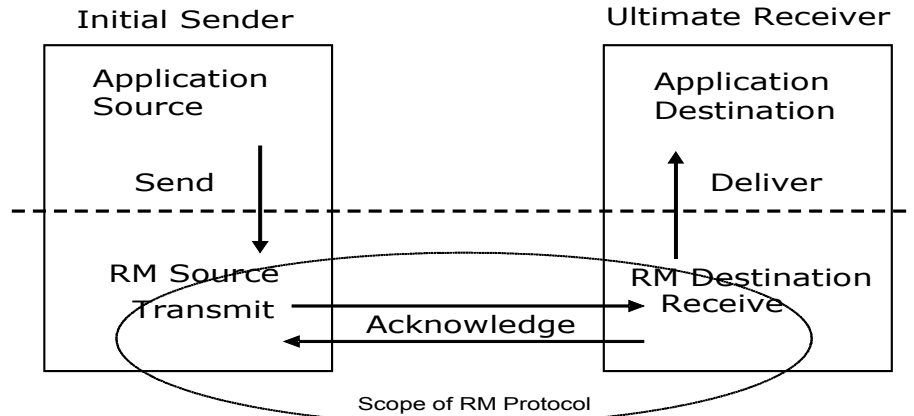
182  
183 Many errors ~~can interrupt a conversation. Messages can be lost, duplicated or reordered. Further the host~~  
184 ~~systems canmay interrupt a conversation. Messages may be lost, duplicated or reordered. Further the-~~  
185 ~~host systems may~~ experience failures and lose volatile state.

186 The WS-ReliableMessaging specification defines an interoperable protocol that ~~enables a Reliable~~  
187 ~~Messaging (RM) Source to accurately determine the disposition of each message it Transmits as~~  
188 ~~perceived by the RM Destination, so as to allow it to resolve any in-doubt status regarding receipt of the~~  
189 ~~message Transmitted. The protocol also enables an RM Destination to efficiently determine which of~~  
190 ~~those messages it Receives have been previously Received, enabling it to filter out duplicate message~~  
191 ~~transmissions caused by the retransmission, by the RM Source, of unacknowledged message. It also~~  
192 ~~enables an RM Destination to Deliver the messages it Receives to the Application Destination in the order~~  
193 ~~in which they were sent by an Application Source, in the event that they are Received out of order. Note~~  
194 ~~that this specification places no restriction on the scope of the RM Source or RM Destination entities. For~~  
195 ~~example, either can span multiple WSDL Ports or Erequires a Reliable Messaging (RM) Source and~~  
196 ~~Reliable Messaging (RM) Destination to ensure that each message transmitted by the RM Source is-~~  
197 ~~successfully received by an RM Destination, or barring successful receipt, that an RM Source can, except~~  
198 ~~in the most extreme circumstances, accurately determine the disposition of each message transmitted as-~~  
199 ~~perceived by the RM Destination, so as to resolve any in-doubt status. Note that this specification makes-~~  
200 ~~no restriction on the scope of the RM Source or RM Destination entities. For example, either may span~~  
201 ~~multiple WSDL Ports or endpoints.~~

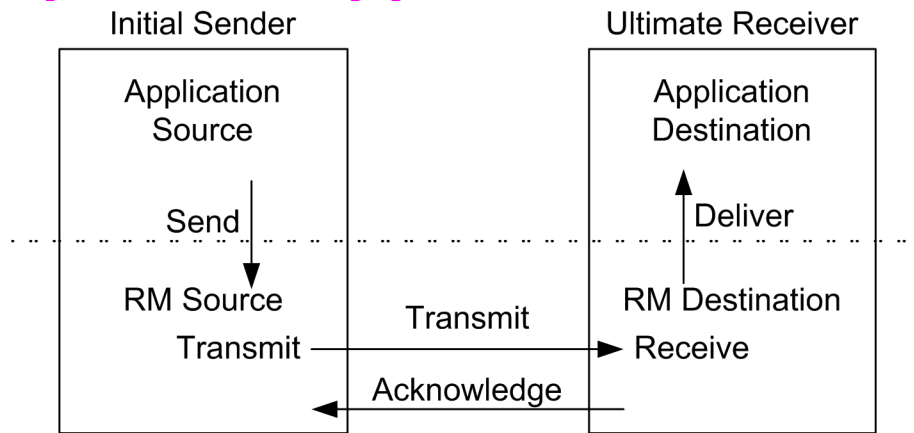
202 The protocol ~~enables the implementation of a broad range of reliability features which include ordered~~  
203 ~~Delivery, duplicate elimination, and guaranteed receipt. The protocol can also be implemented with a~~  
204 ~~range of robustness characteristics ranging from in-memory persistence that is scoped to a single process~~  
205 ~~lifetime, to replicated durable storage that is recoverable in all but the most extreme circumstances. It is~~  
206 ~~expected that the Endpoints will implement as many or as few of these reliability characteristics as~~  
207 ~~necessary for the correct operation of the application using the protocol. Regardless of which of the~~  
208 ~~reliability features is enabled, supports reliability features which include ordered delivery, duplicate-~~  
209 ~~elimination, and guaranteed receipt for the RMD. It is expected that the AD and RMD will implement as~~  
210 ~~many of these or as few of these characteristics as necessary to implement the AD. In any case the wire~~  
211 ~~protocol does not change.~~

212 Figure 1 below illustrates the entities and events in a simple reliable exchange of messages. First, the  
213 Application Source Sends a message for reliable ~~transfer. The Reliable Messaging Source accepts the~~  
214 ~~message and Transmits it one or more times. After accepting the message, the RM Destination~~  
215 ~~Acknowledges it. Finally, the RM Destination Ddelivery. The Reliable Messaging (RM) Source accepts the-~~  
216 ~~message and Transmits it one or more times. After receiving the message, the RM Destination-~~  
217 ~~Aeknowledges it. Finally, the RM Destination delivers the message to the Application Destination. The~~  
218 exact roles the entities play and the complete meaning of the events will be defined throughout this  
219 specification.





220 [Figure 1: Reliable Messaging Model](#)



221 [Figure 1: Reliable Messaging Model](#)

## 222 2.1 Glossary

223 The following definitions are used throughout this specification:

224 **Accept:** [The act of qualifying a message by the RM Destination such that it becomes eligible for Delivery](#)  
 225 [and acknowledgement.](#)

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

228 **Acknowledgement Message:** [A message containing a SequenceAcknowledgement header block.](#)

229 [Acknowledgement Messages may or may not contain a SOAP body](#)**application-Destination:** The endpoint  
 230 [to which a message is Delivered.](#)

231 **Acknowledgement Request:** [A message containing a AckRequested header. Acknowledgement](#)

232 [Requests may or may not contain a SOAP body](#)**application-Source:** The endpoint that Sends a message.

233 **Application Destination:** The Endpoint to which a message is Delivered**Deliver:** The act of transferring a  
 234 [message from the RM Destination to the Application Destination. The reliability guarantee is fulfilled at this](#)  
 235 [point.](#)

236 **Application Source:** [The Endpoint that Sends a message.](#)

237 **Deliver:** [The act of transferring a message from the RM Destination to the Application Destination.](#)

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

241 **Receive:** The act of reading a message from a network connection and accepting it.

242 **RM Destination:** The Endpoint that Receives messages Transmitted reliably from an RM Source.

243 **RM Protocol Header Block:** One of Sequence, SequenceAcknowledgement, or AckRequested.

244 **RM Source:** The Endpoint that Transmits messages reliably to an RM Destination.

245 **Send:** The act of transferring a message from the Application Source to the RM Source for reliable  
246 transfer.

247 **Sequence Lifecycle Message:** A message that contains one of: CreateSequence,  
248 CreateSequenceResponse, CloseSequence, CloseSequenceResponse, TerminateSequence,  
249 TerminateSequenceResponse as the child element of the SOAP body element.

250 **Sequence Traffic Message:** A message containing a Sequence header block.

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

254 ~~Receive:~~ The act of reading a message from a network connection and qualifying it as relevant to RM-  
255 ~~Destination functions.~~

256 ~~RM Destination:~~ For any one reliable sent message the endpoint that receives the message.

257 ~~RM Source:~~ The endpoint that transmits the message.

258 ~~Send:~~ The act of submitting a message to the RM Source for reliable delivery. The reliability guarantee  
259 ~~begins at this point.~~

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

## 261 2.2 Protocol Preconditions

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

- 264 • For any single message exchange the RM Source MUST have an endpoint reference that uniquely  
265 identifies the RM Destination Eendpoint.
- 266 • The RM Source MUST have successfully created a Sequence with the RM Destination, knowledge of  
267 the destination's policies, if any, and the RM Source MUST be capable of formulating messages that  
268 adhere to this policy.
- 269 • The RM Source MUST be capable of formulating messages that adhere to the RM Destination's  
270 policiesIf a secure exchange of messages is required, then the RM Source and RM Destination  
271 MUST have a security context.
- 272 • If a secure exchange of messages is REQUIRED, then the RM Source and RM Destination MUST  
273 have a security context.

## 274 2.3 Protocol Invariants

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

- 276 • The RM Source MUST assign each message within a Sequence a message number (defined
- 277 below) beginning at 1 and increasing by exactly 1 for each subsequent message. These numbers
- 278 MUST be assigned in the same order in which messages are sent by the Application Source.
- 279 • Within every Acknowledgement Message it issues, the RM Destination MUST include one or more
- 280 AcknowledgementRange child elements that contain, in their collective ranges, the message
- 281 number of every message accepted by the RM Destination. The RM Destination MUST exclude, in
- 282 the AcknowledgementRange elements, the message numbers of any messages it has not
- 283 acceptEvery acknowledgement issued by the RM Destination MUST include within an
- 284 acknowledgement range or ranges the sequence number of every message successfully received
- 285 by the RM Destination and MUST exclude sequence numbers of any messages not yet received.

## 286 2.4 Example Message Exchange

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

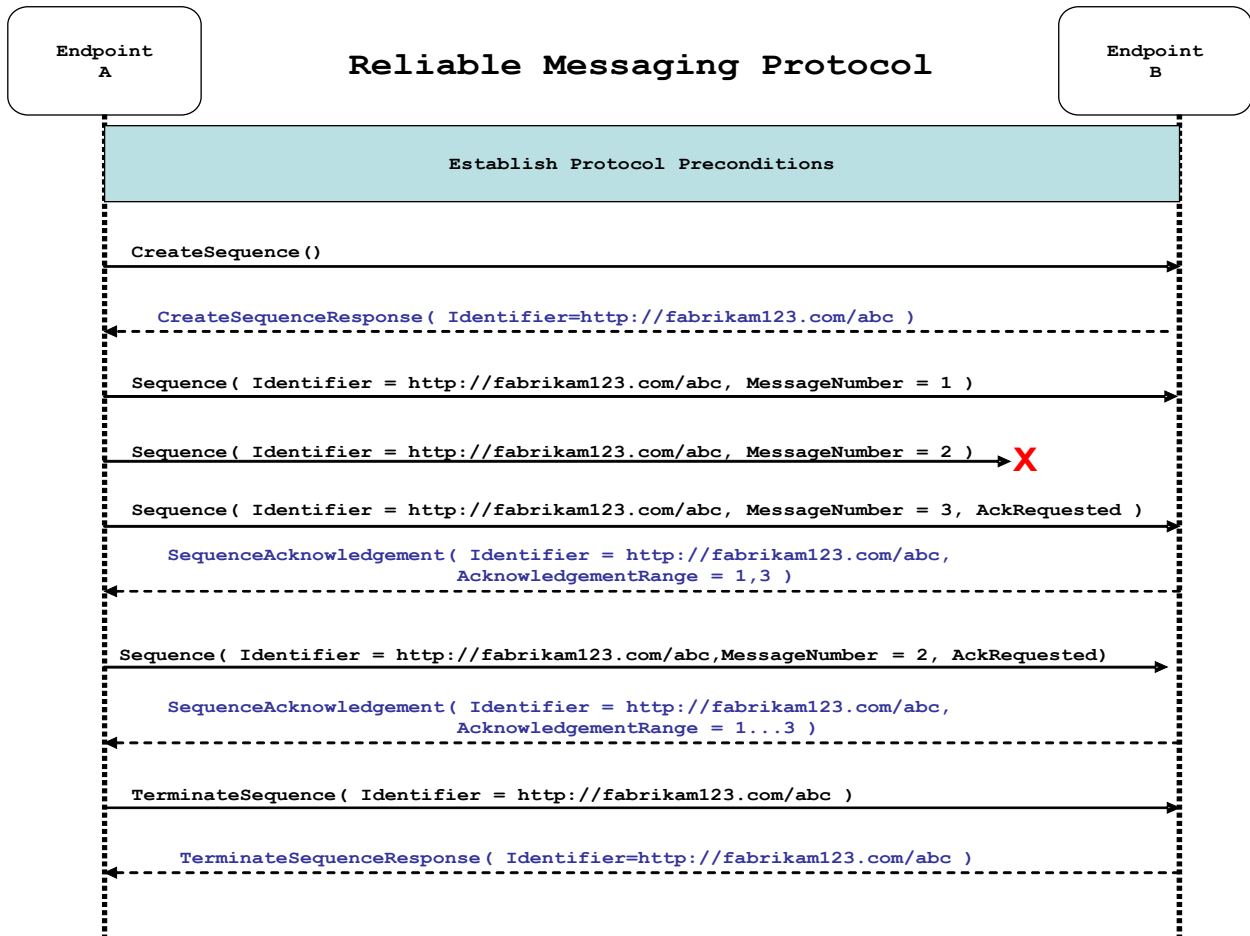


Figure 2: The WS-ReliableMessaging Protocol

288 ~~Figure 2: The WS-ReliableMessaging Protocol~~

- 289 1. The protocol preconditions are established. These include policy exchange, endpoint resolution,
- 290 and establishing trust.
- 291 2. The RM Source requests creation of a new Sequence.

- 292 3. The RM Destination creates a new Sequence and returns its unique identifier.  
293 Sequence by returning a globally unique identifier.
- 294 4. The RM Source begins Transmitting messages in the Sequence beginning with MessageNumber 1.  
295 In the figure above, the RM Source sends 3 messages in the Sequence.  
296 sending messages beginning with MessageNumber 1. In the figure above, the RM Source sends 3 messages.
- 297 5. The 2<sup>nd</sup> message in the Sequence is lost in transit.  
298 Since the 3<sup>rd</sup> message is the last in this exchange, the RM Source includes a <wsm:AckRequested> Header.
- 299 6. The 3<sup>rd</sup> message is the last in this Sequence and the RM Source includes an AckRequested  
300 header to ensure that it gets a timely SequenceAcknowledgement for the Sequence.  
301 2<sup>nd</sup> message is lost in transit.
- 302 7. The RM Destination acknowledges receipt of message numbers 1 and 3 as a result of receiving the  
303 RM Source's AckRequested Header.
- 304 8. The RM Source retransmits the unacknowledged message with MessageNumber 2. This is a new  
305 message from the perspective of the underlying transport, but it has the same Sequence Identifier  
306 and MessageNumber so the RM Destination can recognize it as a duplicate of the earlier message.  
307 in case the original and retransmitted messages are both Received. The RM Source includes an  
308 AckRequested header in the retransmitted message so the RM Destination will expedite an  
309 acknowledgement.  
310 2<sup>nd</sup> message. This is a new message on the underlying transport, but it has the  
311 same sequence identifier and message number so the RM Destination can recognize it as  
equivalent to the earlier message, in case both are received.
- 312 9. The RM Destination Receives the second transmission of the message with MessageNumber 2  
313 and acknowledges receipt of message numbers 1, 2, and 3.  
314 Source includes an <wsm:AckRequested> element so the RM Destination will expedite an acknowledgement.
- 315 10. The RM Source Receives this Acknowledgement and sends a TerminateSequence message to the  
316 RM Destination indicating that the Sequence is completed and reclaims any resources associated  
317 with the Sequence.  
318 Destination receives the second transmission of the message with  
MessageNumber 2 and acknowledges receipt of message numbers 1, 2, and 3.
- 319 11. The RM Destination Receives the TerminateSequence message indicating that the RM Source will  
320 not be sending any more messages. The RM Destination sends a TerminateSequenceResponse  
321 message to the RM Source.  
322 Source receives this acknowledgement and sends a  
323 TerminateSequence message to the RM Destination indicating that the sequence is completed and  
reclaims any resources associated with the Sequence.

324 The RM Source will expect to Receive Acknowledgements from the RM Destination during the course of a  
325 message exchange at occasions described in Section 3 below. Should an Acknowledgement not be  
326 Received in a timely fashion, the RM Source MUST re-transmit the message since either the message or  
327 the associated Acknowledgement might have been lost. Since the nature and dynamic characteristics of  
328 the underlying transport and potential intermediaries are unknown in the general case, the timing of re-  
329 transmissions cannot be specified. Additionally, over-aggressive re-transmissions have been  
330 demonstrated to cause transport or intermediary flooding which are counterproductive to the intention of  
331 providing a reliable exchange of messages. Consequently, implementers are encouraged to utilize  
332 adaptive mechanisms that dynamically adjust re-transmission time and the back-off intervals that are  
333 appropriate to the nature of the transports and intermediaries envisioned. For the case of TCP/IP  
334 transports, a mechanism similar to that described as RTTM in RFC 1323 [RTTM] SHOULD be  
335 considered.  
336 Destination receives the TerminateSequence message indicating that the RM Source will not  
337 be sending any more messages. The RM Destination sends a TerminateSequenceResponse message to the  
RM Source and reclaims any resources associated with the Sequence.

338 The RM Source will expect to receive acknowledgements from the RM Destination during the course of a  
339 message exchange at occasions described in Section 3 below. Should an acknowledgement not be  
340 received in a timely fashion, the RM Source MUST re-transmit the request since either the request or the

341 ~~associated acknowledgement may have been lost. Since the nature and dynamic characteristics of the~~  
342 ~~underlying transport and potential intermediaries are unknown in the general case, the timing of re-~~  
343 ~~transmissions cannot be specified. Additionally, over-aggressive re-transmissions have been~~  
344 ~~demonstrated to cause transport or intermediary flooding which are counterproductive to the intention of~~  
345 ~~providing a reliable exchange of messages. Consequently, implementers are encouraged to utilize~~  
346 ~~adaptive mechanisms that dynamically adjust re-transmission time and the back-off intervals that are~~  
347 ~~appropriate to the nature of the transports and intermediaries envisioned. For the case of TCP/IP~~  
348 ~~transports, a mechanism similar to that described as RTTM in RFC 1323 [RTTM] should be considered.~~

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

## 3 RM Protocol Elements

The following sub-sections define the various RM protocol elements, and prescribe their usage by a conformant implementation. The following sub-sections define the various RM protocol elements, and prescribe their usage by a conformant implementation. 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 Considerations on the Use of Extensibility Points

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

### 3.2 Considerations on the Use of "Piggy-Backing"

Some RM header blocks may be added to messages that happen to be targeted to the same Endpoint to which those headers are to be sent (a concept often referred to as "piggy-backing"), thus saving the overhead of an additional message exchange. Reference parameters MUST be considered when determining whether two EPRs are targeted to the same Endpoint.

### 3.3 Composition with WS-Addressing

When the RM protocol, defined in this specification, is composed with the WS-Addressing specification, the following rules prescribe the constraints on the value of the `wsa:Action` header:

1. When an Endpoint generates a message that carries an RM protocol element, that is defined in section 3 below, in the body of a SOAP envelope that Endpoint MUST include in that envelope a `wsa:Action` SOAP header block whose value is an IRI that is a concatenation of the WS-RM namespace URI, followed by a "/", followed by the value of the local name of the child element of the SOAP body. For example, for a Sequence creation request message as described in section 3.1 below, the value of the `wsa:Action` IRI would be:

```
http://docs.oasis-open.org/ws-rx/wsrn/200608/CreateSequence
```

2. When an Endpoint generates an Acknowledgement Message that has no element content in the SOAP body, then the value of the `wsa:Action` IRI MUST be:

```
http://docs.oasis-open.org/ws-rx/wsrn/200608/SequenceAcknowledgement
```

3. When an Endpoint generates an Acknowledgement Request that has no element content in the SOAP body, then the value of the `wsa:Action` IRI MUST be:

```
http://docs.oasis-open.org/ws-rx/wsrn/200608/AckRequested
```

4. When an Endpoint generates an RM fault as defined in section 4 below, the value of the `wsa:Action` IRI MUST be as defined in section 4 below.

### 3.4 Sequence Creation

The RM Source MUST request creation of an outbound Sequence by sending a `CreateSequence` element in the body of a message to the RM Destination which in turn responds either with a message containing `CreateSequenceResponse` or a `CreateSequenceRefused` fault. The RM Source MAY

389 include an offer to create an inbound Sequence within the CreateSequence message. This offer is  
 390 either accepted or rejected by the RM Destination in the CreateSequenceResponse  
 391 messag~~<wsmr:CreateSequence>~~ element in the body of a message to the RM Destination which in-  
 392 turn responds either with a message containing <wsmr:CreateSequenceResponse> or a  
 393 CreateSequenceRefused fault.. <wsmr:CreateSequence> MAY carry an offer to create an inbound-  
 394 sequence which is either accepted or rejected in the <wsmr:CreateSequenceResponse>. Note that  
 395 offering a Sequence within the <wsmr:CreateSequence> element is simply a protocol optimization.-  
 396 There is no semantic difference between offering a Sequence, and choosing not to offer one and  
 397 subsequently creating a new Sequence to carry messages from the RM Destination to the RM Source.

398 The SOAP version used for the CreateSequence message SHOULD be used for all subsequent  
 399 messages in or for that Sequence, sent by either the RM Source or the RM Destination, following exemplar-  
 400 defines the <wsmr:CreateSequence> syntax:

401 The following exemplar defines the CreateSequence syntax:

```

402 <wsmr:CreateSequence ...>
403   <wsmr:AcksTo ...> wsa:EndpointReferenceType </wsmr:AcksTo>
404   <wsmr:Expires ...> xs:duration </wsmr:Expires> ?
405   <wsmr:Offer ...>
406     <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>
407     <wsmr:Endpoint> wsa:EndpointReferenceType </wsmr:Endpoint>
408     <wsmr:Expires ...> xs:duration </wsmr:Expires> ?
409     <wsmr:IncompleteSequenceBehavior>
410       wsmr:IncompleteSequenceBehaviorType
411     </wsmr:IncompleteSequenceBehavior> ?
412     ...
413   </wsmr:Offer> ?
414   ...
415 </wsmr:CreateSequence>
  
```

416 /wsmr:CreateSequence

417 This element requests creation of a new Sequence between the RM Source that sends it, and the RM  
 418 Destination to which it is sent. The RM Source MUST NOT send this element as a header block. The RM  
 419 Destination MUST respond either with a CreateSequenceResponseis element MUST NOT be-  
 420 sent as a header block. The RM Destination MUST respond either with a-  
 421 <wsmr:CreateSequenceResponse> response message or a CreateSequenceRefused fault.

422 /wsmr:CreateSequence/wsmr:AcksTo

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

430 Implementations MUST NOT use an endpoint reference in the AcksTo element that would prevent the  
 431 sending of Sequence Acknowledgements back to the RM Source. For example, using the WS-Addressing  
 432 "http://www.w3.org/2005/08/addressing/none" IRI would make it impossible for the RM Destination to ever  
 433 send Sequence Acknowledgements.none" IRI would make it impossible for the RM Destination to ever  
 434 send Sequence Acknowledgements.-

435 /wsmr:CreateSequence/wsmr:Expires

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

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

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

444 `/wsrm:CreateSequence/wsrm:Offer`

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

447 `/wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier`

448 ~~The RM Source MUST set the value of this element to an absolute URI (conformant with RFC3986  
449 [URI])is REQUIRED element MUST contain an absolute URI conformant with RFC3986~~ that uniquely  
450 identifies the offered Sequence.

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

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

454 `/wsrm:CreateSequence/wsrm:Offer/wsrm:Endpoint`

455 ~~An RM Source MUST include this element, of type `wsa:EndpointReferenceType` (as specified by  
456 WS-Addressing). This element specifies the endpoint reference to which Sequence Lifecycle Messages,  
457 Sequence Traffic Messages, Acknowledgement Requests, and fault messages related to the offered  
458 Sequence are to be sent.~~

459 ~~Implementations MUST NOT use an endpoint reference in the Endpoint element that would prevent the  
460 sending of Sequence Lifecycle Message, Sequence Traffic Message, etc. For example, using the WS-  
461 Addressing "http://www.w3.org/2005/08/addressing/none" IRI would make it impossible for the RM  
462 Destination to ever send Sequence Lifecycle Messages (e.g. `TerminateSequence`) to the RM Source  
463 for the Offered Sequence. Implementations MAY use the WS-RM anonymous URI template and doing so  
464 implies that messages will be retrieved using a mechanism such as the `MakeConnection` message (see  
465 section 3.7).~~

466 `/wsrm:CreateSequence/wsrm:Offer/wsrm:Expires`

467 This element, if present, of type `xs:duration` specifies the duration for the ~~offered Sequence. A value of  
468 "PT0S" indicates that the offered Sequence will never expire. Absence of the element indicates an implied  
469 value of "PT0S".~~~~Sequence. A value of 'PT0S' indicates that the Sequence will never expire. Absence of the  
470 element indicates an implied value of 'PT0S'.~~

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

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

474 `/wsrm:CreateSequence/wsrm:Offer/wsrm:IncompleteSequenceBehavior`



475 This element, if present, specifies the behavior that the destination will exhibit upon the closure or  
476 termination of an incomplete Sequence. For the purposes of defining the values used, the term "discard"  
477 refers to behavior equivalent to the Application Destination never processing a particular message.

478 A value of "DiscardEntireSequence" indicates that the entire Sequence MUST be discarded if the  
479 Sequence is closed, or terminated, when there are one or more gaps in the final  
480 SequenceAcknowledgement.

481 A value of "DiscardFollowingFirstGap" indicates that messages in the Sequence beyond the first gap  
482 MUST be discarded when there are one or more gaps in the final SequenceAcknowledgement.

483 The default value of "NoDiscard" indicates that no acknowledged messages in the Sequence will be  
484 discarded.

485 /wsmr:CreateSequence/wsmr:Offer/{any}

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

488 /wsmr:CreateSequence/wsmr:Offer/@{any}

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

491 /wsmr:CreateSequence/{any}

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

494 /wsmr:CreateSequence/@{any}

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

497 A [CreateSequenceResponse](#) is sent in the body of a response message by an RM Destination in  
498 response to receipt of a [CreateSequence](#) request message. It carries the [Identifier](#) of the created  
499 Sequence and indicates that the RM Source can~~<wsmr:CreateSequenceResponse> is sent in the~~  
500 ~~body of a response message by an RM Destination in response to receipt of a~~  
501 ~~<wsmr:CreateSequence> request message. It carries the <wsmr:Identifier> of the created~~  
502 Sequence and indicates that the RM Source may begin sending messages in the context of the identified  
503 Sequence.

504 The following exemplar defines the [CreateSequenceResponse](#)~~<wsmr:CreateSequenceResponse>~~  
505 syntax:

```
506 <wsmr:CreateSequenceResponse ...>
507   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>
508   <wsmr:Expires ...> xs:duration </wsmr:Expires> ?
509   <wsmr:IncompleteSequenceBehavior>AcknowledgementInterval
510   Milliseconds="xs:unsignedLong" ... /> ?
511   wsmr:IncompleteSequenceBehaviorType
512 </wsmr:IncompleteSequenceBehavior> ?
513   <wsmr:Accept ...>
514     <wsmr:AcksTo ...> wsa:EndpointReferenceType </wsmr:AcksTo>
515     ...
516   </wsmr:Accept> ?
517   ...
518 </wsmr:CreateSequenceResponse>
```

519 /wsmr:CreateSequenceResponse

520 This element is sent in the body of the response message in response to a [CreateSequence request message](#). It indicates that the RM Destination has created a new Sequence at the request of the RM  
521 Source. The RM Destination MUST NOT send this element ~~in a [wsmr:CreateSequence](#) request message~~.  
522 It indicates that the RM Destination has created a new Sequence at the request of the RM Source. This  
523 element MUST NOT be sent as a header block.  
524

525 `/wsmr:CreateSequenceResponse/wsmr:Identifier`

526 The RM Destination MUST include this element within any [CreateSequenceResponse](#) message it sends.  
527 The RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986)  
528 that uniquely identifies ~~REQUIRED~~ element MUST contain an absolute URI conformant with RFC3986  
529 of the Sequence that has been created by the RM Destination.

530 `/wsmr:CreateSequenceResponse/wsmr:Identifier/@{any}`

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

533 `/wsmr:CreateSequenceResponse/wsmr:Expires`

534 This element, if present, of type `xs:duration` accepts or refines the RM Source's requested duration for  
535 the Sequence. It specifies the amount of time after which any resources associated with the Sequence  
536 SHOULD be reclaimed thus causing the Sequence to be silently terminated. At the RM Destination this  
537 duration is measured from a point proximate to Sequence creation and at the RM Source this duration is  
538 measured from a point approximate to the successful processing of the [CreateSequenceResponse](#). A  
539 value of "PT0S" indicates that the Sequence will never expire. Absence of the element indicates an  
540 implied value of "PT0S". The RM Destination MUST set the value of this element to be equal to or less  
541 than the value requested by the RM Source in the corresponding [CreateSequence](#) message. A value of  
542 'PT0S' indicates that the Sequence will never expire. Absence of the element indicates an implied value of  
543 'PT0S'. This value MUST be equal to or less than the value requested by the RM Source in the  
544 corresponding ~~`<wsmr:CreateSequence>` message~~.

545 `/wsmr:CreateSequenceResponse/wsmr:Expires/@{any}`

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

548 `/wsmr:CreateSequenceResponse/wsmr:IncompleteSequenceBehaviorAcknowledgementInter  
549 val`

550 This element, if present, specifies the behavior that the destination will exhibit upon the closure or  
551 termination of an incomplete Sequence. For the purposes of defining the values used, the term "discard"  
552 refers to behavior equivalent to the Application Destination never processing a particular message  
553 after which the RM Destination will transmit an acknowledgement. If omitted, there is no implied value.

554 A value of "DiscardEntireSequence" indicates that the entire Sequence MUST be discarded if the  
555 Sequence is closed, or terminated, when there are one or more gaps in the final  
556 [SequenceAcknowledgement](#).

557 A value of "DiscardFollowingFirstGap" indicates that messages in the Sequence beyond the first gap  
558 MUST be discarded when there are one or more gaps in the final [SequenceAcknowledgement](#).

559 The default value of "NoDiscard" indicates that no acknowledged messages in the Sequence will be  
560 discarded.

561 `/wsmr:CreateSequenceResponse/wsmr:AcknowledgementInterval/@Milliseconds`

562 The acknowledgement interval, specified in milliseconds.

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

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

566 `/wsrm:CreateSequenceResponse/wsrm:Accept`

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

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

574 `/wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo`

575 The RM Destination MUST include this element, of type `wsa:EndpointReferenceType` (as specified  
576 by WS-Addressing). It specifies the endpoint reference to which messages containing  
577 `SequenceAcknowledgement` header blocks and faults related to the created Sequence are to be sent,  
578 unless otherwise noted in this specification (for example, see Section 3.2) is REQUIRED element, of type  
579 `wsa:EndpointReferenceType` as specified by WS-Addressing [WS-Addressing], specifies the endpoint-  
580 reference to which `<wsrm:SequenceAcknowledgement>` messages related to the accepted Sequence  
581 are to be sent.

582 Implementations MUST NOT use an endpoint reference in the `AcksTo` element that would prevent the  
583 sending of Sequence Acknowledgements back to the RM Source. For example, using the WS-Addressing  
584 "<http://www.w3.org/2005/08/addressing/none>" IRI would make it impossible for the RM Destination to ever  
585 send Sequence Acknowledgements.

586 `/wsrm:CreateSequenceResponse/wsrm:Accept/{any}`

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

589 `/wsrm:CreateSequenceResponse/wsrm:Accept/@{any}`

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

592 `/wsrm:CreateSequenceResponse/{any}`

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

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

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

### 598 **3.5 Closing A Sequence**

599 There are times during the use of an RM Sequence that the RM Source or RM Destination will wish to  
600 discontinue using a Sequence. Simply terminating the Sequence discards the state managed by the RM  
601 Destination, leaving the RM Source unaware of the final ranges of messages that were successfully  
602 transferred to the RM Destination. To ensure that the Sequence ends with a known final state either the

603 ~~RM Source or RM Destination MAY choose to close~~ may be times during the use of an RM Sequence that  
604 ~~the RM Source or RM Destination will wish to discontinue using a Sequence. Simply terminating the~~  
605 ~~Sequence discards the state managed by the RM Destination, leaving the RM Source unaware of the final~~  
606 ~~ranges of messages that were successfully delivered to the RM Destination. To ensure that the~~  
607 ~~Sequence ends with a known final state both the RM Source and RM Destination may choose to 'close'~~  
608 ~~the Sequence before terminating it.~~

609 ~~If the RM Source wishes to close the Sequence, then it sends a CloseSequence element, in the body of~~  
610 ~~a message, to the RM Destination. This message indicates that the RM Destination MUST NOT accept~~  
611 ~~any new messages for the specified Sequence, other than those already accepted at the time the~~  
612 ~~CloseSequence element is interpreted by the RM Destination. Upon receipt of this message, or~~  
613 ~~subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST~~  
614 ~~include a final SequenceAcknowledgement (within which the RM Destination MUST include the Final~~  
615 ~~element) header block on any messages associated with the Sequence destined to the RM Source,~~  
616 ~~including the CloseSequenceResponse message or on any Sequence fault Transmitted to the RM~~  
617 ~~Source, then it sends a <wsrm:CloseSequence> element, in the body of a message, to the RM~~  
618 ~~Destination. This message indicates that the RM Destination MUST NOT receive any new messages for~~  
619 ~~the specified sequence, other than those already received at the time the <wsrm:CloseSequence>~~  
620 ~~element is interpreted by the RMD. Upon receipt of this message, or subsequent to the RM Destination~~  
621 ~~closing the Sequence of its own volition, the RM Destination MUST include a final~~  
622 ~~SequenceAcknowledgement (that MUST include the <wsrm:Final> element) header block on each~~  
623 ~~message destined to the RM Source, including the CloseSequenceResponse message and on any~~  
624 ~~Sequence Fault transmitted to the RMS.~~

625 ~~While the RM Destination MUST NOT accept any new messages for the specified Sequence it MUST still~~  
626 ~~process Sequence Lifecycle Messages and Acknowledgement Requests. For example, it MUST respond to~~  
627 ~~AckRequested, TerminateSequence as well as CloseSequence messages. Note, subsequent~~  
628 ~~CloseSequence messages have no effect on the state of the Sreceive any new messages for the~~  
629 ~~specified sequence it MUST still process RM protocol messages. For example, it MUST respond to~~  
630 ~~AckRequested, TerminateSequence as well as CloseSequence messages. Note, subsequent~~  
631 ~~CloseSequence messages have no effect on the state of the sequence.~~

632 ~~In the case where the RM Destination wishes to discontinue use of a Sequence it is RECOMMENDED~~  
633 ~~that it close the Sequence. Please see Final and the SequenceClosed fault. Whenever possible the~~  
634 ~~SequenceClosed fault SHOULD be used in place of the SequenceTerminated fault to allow the RM~~  
635 ~~Source to still Rsequence it may 'close' the sequence itself. Please see <wsrm:Final> above and the~~  
636 ~~SequenceClosed fault below. Note, the SequenceClosed Fault SHOULD be used in place of the~~  
637 ~~SequenceTerminated Fault, whenever possible, to allow the RM Source to still receive~~  
638 ~~Acknowledgements.~~

639 The following exemplar defines the CloseSequence syntax:

```
640 <wsrm:CloseSequence ...>  
641   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
642   ...  
643 </wsrm:CloseSequence>
```

644 /wsrm:CloseSequence

645 This element is sent by an RM Source to indicate that the RM Destination MUST NOT accept any new  
646 messages for this Sequence. A SequenceClosed fault MUST be generated by the RM Destination when it  
647 Receives a message for a Sequence that is already receive any new messages for this sequence. A  
648 SequenceClosed fault MUST be generated by the RM Destination when it receives a message for a  
649 sequence that is closed.

650 /wsmr:CloseSequence/wsmr:Identifier

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

655 /wsmr:CloseSequence/wsmr:Identifier/@{any}

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

658 /wsmr:CloseSequence/{any}

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

661 /wsmr:CloseSequence@{any}

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

664 A CloseSequenceResponse is sent in the body of a response message by an RM Destination in  
665 response to receipt of a CloseSequence request message. It indicates that the RM Destination has  
666 closed the S<wsmr:CloseSequenceResponse> ~~is sent in the body of a response message by an RM-~~  
667 ~~Destination in response to receipt of a <wsmr:CloseSequence> request message. It indicates that the~~  
668 ~~RM Destination has closed the sequence.~~

669 The following exemplar defines the CloseSequenceResponse<wsmr:CloseSequenceResponse>  
670 syntax:

```
671 <wsmr:CloseSequenceResponse ...>  
672   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
673   ...  
674 </wsmr:CloseSequenceResponse>
```

675 /wsmr:CloseSequenceResponse

676 This element is sent in the body of a response message by an RM Destination in response to receipt of a  
677 CloseSequence request message. It indicates that the RM Destination has closed the  
678 S<wsmr:CloseSequence> ~~request message. It indicates that the RM Destination has closed the~~  
679 ~~sequence.~~

680 /wsmr:CloseSequenceResponse/wsmr:Identifier

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

685 /wsmr:CloseSequenceResponse/wsmr:Identifier/@{any}

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

688 /wsmr:CloseSequenceResponse/{any}

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

691 /wsmr:CloseSequenceResponse@{any}

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

## 694 3.6 Sequence Termination

695 When the RM Source has completed its use of the Sequence it sends a TerminateSequence element,  
696 in the body of a message, to the RM Destination to indicate that the Sequence is complete and that it will  
697 not be sending any further messages related to the Sequence. The RM Destination can safely reclaim any  
698 resources associated with the Sequence upon receipt of the TerminateSequence message. Under  
699 normal usage the RM Source will complete its use of the S, it sends a <wsmr:TerminateSequence>  
700 element, in the body of a message to the RM Destination to indicate that the Sequence is complete, and  
701 that it will not be sending any further messages related to the Sequence. The RM Destination can safely  
702 reclaim any resources associated with the Sequence upon receipt of the <wsmr:TerminateSequence>  
703 message. Note, under normal usage the RM source will complete its use of the sequence when all of the  
704 messages in the Sequence have been acknowledged. However, the RM Source is free to Terminate or  
705 Close a Sequence at any time regardless of the acknowledgement state of the messages.

706 The following exemplar defines the TerminateSequence syntax:

```
707 <wsmr:TerminateSequence ...>  
708   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
709   ...  
710 </wsmr:TerminateSequence>
```

711 /wsmr:TerminateSequence

712 This element is sent by an RM Source to indicate it has completed its use of the Sequence. It indicates  
713 that the RM Destination can safely reclaim any resources related to the identified Sequence. The RM  
714 Source MUST NOT send this element as a header block. The RM Source MAY retransmit this element.  
715 Once this element is sent, other than this element, the RM Source MUST NOT send any additional  
716 message to the RM Destination referencing this S-is element MUST NOT be sent as a header block. The  
717 RM Source MAY retransmit this element. Once this element is sent, other than this element, the RM  
718 Source MUST NOT send any additional message to the RM Destination referencing this sequence.

719 /wsmr:TerminateSequence/wsmr:Identifier

720 The RM Source MUST include this element in any TerminateSequence message it sends. The RM  
721 Source MUST set the value of this element to the absolute URI (conformant with RFC3986) of the  
722 Sequence that is being terminated. is-REQUIRED-element MUST contain an absolute URI conformant with  
723 RFC3986 of the Sequence that is being terminated.

724 /wsmr:TerminateSequence/wsmr:Identifier/@{any}

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

727 /wsmr:TerminateSequence/{any}

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

730 /wsmr:TerminateSequence/@{any}

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

733 A [TerminateSequenceResponse](#) is sent in the body of a response message by an RM Destination in  
734 response to receipt of a [TerminateSequence](#) request message. It indicates that the RM Destination has  
735 terminated the S<wsm:TerminateSequenceResponse> is sent in the body of a response message by  
736 an RM Destination in response to receipt of a <wsm:TerminateSequence> request message. It  
737 indicates that the RM Destination has terminated the sequence.

738 The following exemplar defines the

739 [TerminateSequenceResponse](#)<wsm:TerminateSequenceResponse> syntax:

```
740 <wsm:TerminateSequenceResponse ...>  
741   <wsm:Identifier ...> xs:anyURI </wsm:Identifier>  
742   ...  
743 </wsm:TerminateSequenceResponse>
```

744 /wsm:TerminateSequenceResponse

745 This element is sent in the body of a response message by an RM Destination in response to receipt of a  
746 [TerminateSequence](#) request message. It indicates that the RM Destination has terminated the  
747 [Sequence](#). The RM Destination MUST NOT send this elem<wsm:TerminateSequence> request-  
748 message. It indicates that the RM Destination has terminated the sequence. This element MUST NOT be  
749 sent as a header block.

750 /wsm:TerminateSequenceResponse/wsm:Identifier

751 The RM Destination MUST include this element in any [TerminateSequenceResponse](#) message it  
752 sends. The RM Destination MUST set the value of this element to the absolute URI (conformant with  
753 [RFC3986](#))is REQUIRED element MUST contain an absolute URI conformant with [RFC3986](#) of the  
754 Sequence that is being terminated.

755 /wsm:TerminateSequenceResponse/wsm:Identifier/@{any}

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

758 /wsm:TerminateSequenceResponse/{any}

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

761 /wsm:TerminateSequenceResponse/@{any}

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

764 On receipt of a [TerminateSequence](#) message an RM Destination MUST respond with a corresponding  
765 [TerminateSequenceResponse](#) message or generate a fault [UnknownSequenceFault](#) if the  
766 [Sequence](#) is not known<wsm:TerminateSequence> message an RM Destination MUST respond with  
767 a corresponding <wsm:TerminateSequenceResponse> message or generate a fault.

## 768 3.7 Sequences

769 The RM protocol uses a [Sequence](#) header block to track and manage the reliable transfer of messages.  
770 The RM Source MUST include a [Sequence](#) header block in all messages for which reliable transfer is  
771 REQUIRED. The RM Source MUST identify Sequences with unique Identifier elements and the RM  
772 Source MUST assign each message within a Sequence a [MessageNumber](#) element that increments by 1  
773 from an initial value of 1. These values are contained within a [Sequence](#) header block accompanying  
774 each message being transfer<wsm:Sequence> header block to track and manage the reliable delivery

775 of messages. Messages for which a reliable delivery is required MUST contain a ~~<wsrm:Sequence>~~  
776 header block. Each Sequence MUST have a unique ~~<wsrm:Identifier>~~ element and each message  
777 within a Sequence MUST have a ~~<wsrm:MessageNumber>~~ element that increments by 1 from an initial  
778 value of 1. These values are contained within a ~~<wsrm:Sequence>~~ header block accompanying each  
779 message being delivered in the context of a Sequence.

780 The RM Source MUST NOT include more than one ~~Sequence~~ ~~MUST be no more than one~~  
781 ~~<wsrm:Sequence>~~ header block in any message.

782 A following exemplar defines its syntax:

```
783 <wsrm:Sequence ...>  
784   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
785   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>  
786   ...  
787 </wsrm:Sequence>
```

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

789 /wsrm:Sequence

790 This ~~protocol element associates the message in which it is contained with a previously established RM~~  
791 ~~Sequence. It contains the Sequence's unique identifier and the containing message's ordinal position~~  
792 ~~within that Sequence. The RM Destination MUST understand the Sequence header block. The RM~~  
793 ~~Source MUST assign a mustUnderstand attribute with a value 1/true (from the namespace~~  
794 ~~corresponding to the version of SOAP to which the Sequence SOAP header block is bound) to the~~  
795 ~~Sequence header block element is the element containing Sequence information for WS-~~  
796 ~~ReliableMessaging. The <wsrm:Sequence> element MUST be understood by the RM-Destination. The~~  
797 ~~<wsrm:Sequence> element MUST have a mustUnderstand attribute with a value 1/true from the~~  
798 ~~namespace corresponding to the version of SOAP to which the <wsrm:Sequence> SOAP header block~~  
799 ~~is bound.~~

800 /wsrm:Sequence/wsrm:Identifier

801 ~~An RM Source that includes a Sequence header block in a SOAP envelope MUST include this element in~~  
802 ~~that header block. The RM Source MUST set the value of this element to the absolute URI (conformant~~  
803 ~~with RFC3986) that uniquely identifies the Sequence.~~

804 ~~This REQUIRED element MUST contain an absolute URI conformant with RFC3986 that uniquely~~  
805 ~~identifies the Sequence.~~

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

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

809 /wsrm:Sequence/wsrm:MessageNumber

810 ~~The RM Source MUST include this element within any Sequence headers it creates. This element is of~~  
811 ~~type MessageNumberType. It represents the ordinal position of the message within a Sequence.~~  
812 ~~Sequence message numbers start at 1 and monotonically increase by 1 throughout the Sequence. See~~  
813 ~~Section 4.5 for Message Number is REQUIRED element MUST contain a wsrm:MessageNumberType~~  
814 ~~representing the ordinal position of the message within a Sequence. Sequence MessageNumbers start at~~  
815 ~~1 and monotonically increase throughout the Sequence. If the message number exceeds the internal~~  
816 ~~limitations of an RM Source or RM Destination or reaches the maximum value of~~  
817 ~~9,223,372,036,854,775,807 the RM Source or Destination MUST issue a MessageNumberRollover fault.~~

818 /wsrm:Sequence/{any}



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

821 /wsrm:Sequence/@{any}

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

824 The following example illustrates a Sequence header block.

```
825 <wsrm:Sequence>  
826   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
827   <wsrm:MessageNumber>10</wsrm:MessageNumber>  
828 </wsrm:Sequence>
```

### 829 3.8 Request Acknowledgement

830 The purpose of the [AckRequested](#) header block is to signal to the RM Destination that the RM Source is  
831 [requesting that a SequenceAcknowledgement be sent](#).  
832 ~~signal to the RM Destination that the RM Source is requesting that a~~  
833 ~~<wsrm:SequenceAcknowledgement> be returned.~~

834 The RM Source [MAY request an Acknowledgement Message from the RM Destination at any time by](#)  
835 [including an AckRequested header block in any message targeted to the RM Destination. An RM](#)  
836 [Destination that Receives a message that contains an AckRequested header block MUST send a](#)  
837 [message containing a SequenceAcknowledgement header block to the AcksTo endpoint reference](#)  
838 [\(see Section 3.1\) for a known Sequence or else generate an UnknownSequence fault. If a non-](#)  
839 [mustUnderstand fault occurs when processing an RM header that was piggy-backed on another](#)  
840 [message, a fault MUST be generated, but the processing of the original message MUST NOT be](#)  
841 [affected. It is RECOMMENDED that the RM Destination return a AcknowledgementRange Or None](#)  
842 [element instead of a Nack element \(see Section 3.6\).](#)  
843 ~~the RM Destination at any time by including an <wsrm:AckRequested> header block in the message.~~  
844 ~~An RM Destination that receives a message that contains an <wsrm:AckRequested> header block~~  
845 ~~MUST respond with a message containing a <wsrm:SequenceAcknowledgement> header block. If a~~  
846 ~~non-mustUnderstand fault occurs when processing an RM Header that was piggy-backed on another~~  
847 ~~message, a fault MUST be generated, but the processing of the original message MUST NOT be~~  
848 ~~affected.~~

849 The following exemplar defines its syntax:

```
850 <wsrm:AckRequested ...>  
851   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
852   ...  
853 </wsrm:AckRequested>
```

854 /wsrm:AckRequested

855 This element requests an [Acknowledgement for the identified Sacknowledgement for the identified](#)  
856 [sequence](#).

857 /wsrm:AckRequested/wsrm:Identifier

858 [An RM Source that includes a AckRequested header block in a SOAP envelope MUST include this](#)  
859 [element in that header block. The RM Source MUST set the value of this element to the absolute URI,](#)  
860 [\(conformant with RFC3986\), that uniquely identifies the Sequence to which the request applies.](#)

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

863 /wsmr:AckRequested/wsmr:Identifier/@{any}

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

866 /wsmr:AckRequested/{any}

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

869 /wsmr:AckRequested/@{any}

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

### 872 **3.9 Sequence Acknowledgement**

873 The RM Destination informs the RM Source of successful message receipt using a  
874 SequenceAcknowledgement header block. The RM Destination MAY Transmit the  
875 SequenceAcknowledgement header block independently or it MAY include the  
876 SequenceAcknowledgement header block on any message targeted to the AcksTo EPR.  
877 Acknowledgements can be explicitly requested using the AckRequested directive (see Section 3.5). If a  
878 non-mustUnderstand fault occurs when processing an RM h<wsmr:SequenceAcknowledgement>  
879 header block. The <wsmr:SequenceAcknowledgement> header block MAY be transmitted-  
880 independently or included on return messages. The RM Destination MAY send a-  
881 <wsmr:SequenceAcknowledgement> header block at any point during which the sequence is valid.  
882 The timing of acknowledgements can be advertised using policy and acknowledgements can be explicitly  
883 requested using the <wsmr:AckRequested> directive (see Section Request Acknowledgement). If a  
884 non-mustUnderstand fault occurs when processing an RM Header that was piggy-backed on another  
885 message, a fault MUST be generated, but the processing of the original message MUST NOT be  
886 affected.

887 A RM Destination MAY include a SequenceAcknowledgement header block on any SOAP envelope  
888 targetted to the endpoint referenced by the AcksTo EPR.

889 During creation of a Sequence the RM Source MAY specify the WS-Addressing anonymous IRI as the  
890 address of the AcksTo EPR for that Sequence. When the RM Source specifies the WS-Addressing  
891 anonymous IRI as the address of the AcksTo EPR, the RM Destination MUST Transmit any  
892 SequenceAcknowledgement headers for the created Sequence in a SOAP envelope to be Transmitted  
893 on the protocol binding-specific channel. Such a channel is provided by the context of a Received  
894 message containing a SOAP envelope that contains a Sequence header block and/or a AckRequested  
895 header block for that same Sequence identifier.

896 The following exemplar defines its syntax:

```
897 <wsmr:SequenceAcknowledgement ...>
898   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>
899   [ [ | <wsmr:AcknowledgementRange ...
900     __Upper="wsmr:MessageNumberType"
901     __Lower="wsmr:MessageNumberType"/> +
902     | <wsmr:None/> ]
903     <wsmr:Final/> ?|<wsmr+None/> ]
904   <wsmr+Final/> ?
```

```

905 | <wsrm:Nack> wsrm:MessageNumberType </wsrm:Nack> + ]
906
907 . . .
908 </wsrm:SequenceAcknowledgement>

```

909 The following describes the content model of the

910 [SequenceAcknowledgement](#)<wsrm:SequenceAcknowledgement> header block.

911 /wsrm:SequenceAcknowledgement

912 This element contains the Sequence [A](#)cknowledgement information.

913 /wsrm:SequenceAcknowledgement/wsrm:Identifier

914 [An RM Destination that includes a SequenceAcknowledgement header block in a SOAP envelope MUST include this element in that header block. The RM Destination MUST set the value of this element to the absolute URI \(conformant with RFC3986\) that uniquely identifies the Sequence. The RM Destination MUST NOT include multiple SequenceAcknowledgement header blocks that share the same value for Identifier within the same SOAP envelope.](#)

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

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

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

925 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange

926 [The RM Destination MAY include one or more instances of this element within a SequenceAcknowledgement header block. It contains a range of Sequence MessageNumbers successfully accepted by the RM Destination. The ranges SHOULD NOT overlap. The RM Destination MUST NOT include this element if a sibling Nack or None element is also present as a child of SequenceAcknowledgement.](#)

930 ~~SequenceAcknowledgement is OPTIONAL element, if present, can occur 1 or more times. It contains a range of Sequence MessageNumbers successfully received~~

932 ~~by the RM Destination. The ranges SHOULD NOT overlap. This element MUST NOT~~

933 ~~be present if a sibling <wsrm:Nack> or <wsrm:None> element is also present as~~

934 ~~a child of <wsrm:SequenceAcknowledgement>.~~

935 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Upper

936 [The RM Destination MUST set the value of this attribute equal to the message number of the highest contiguous message in a Sequence range acceptis-REQUIRED-attribute contains a wsrm:MessageNumberType representing the <wsrm:MessageNumber> of the highest contiguous message in a Sequence range received by the RM Destination.](#)

940 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Lower

941 [The RM Destination MUST set the value of this attribute equal to the message number of the lowest contiguous message in a Sequence range acceptis-REQUIRED-attribute contains a wsrm:MessageNumberType representing the <wsrm:MessageNumber> of the lowest contiguous message in a Sequence range received by the RM Destination.](#)

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

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

948 /wsrm:SequenceAcknowledgement/wsrm:None

949 The RM Destination MUST include this element within a SequenceAcknowledgement header block if  
950 the RM Destination has not accepted any messages for the specified Sequence. The RM Destination  
951 MUST NOT include this element if a sibling AcknowledgementRange or Nack element is also present  
952 as a child of the SequenceAcknowledgement.

953 /wsrm:SequenceAcknowledgement/wsrm:Final

954 The RM Destination MAY include this element within a SequenceAcknowledgement header block. This  
955 element indicates that the RM Destination is not receiving new messages for the specified Sequence. The  
956 RM Source can be assured that the ranges of messages acknowledged by this  
957 SequenceAcknowledgement header block will not change in the future. The RM Destination MUST  
958 include this element when the Sequence is closed. The RM Destination MUST NOT include this element  
959 when sending a Nack; it can only be used when sending AcknowledgementRange elements or a  
960 Noneis OPTIONAL element, if present, indicates that the RM Destination is not  
961 receiving new messages for the specified Sequence. The RM Source can be  
962 assured that the ranges of messages acknowledged by this  
963 SequenceAcknowledgement header block will not change in the future. This  
964 element MUST be present when the Sequence is no longer receiving new message  
965 for the specified sequence. Note: this element MUST NOT be used when sending  
966 a Nack, it can only be used when sending AcknowledgementRanges or <wsrm:None>.

967 /wsrm:SequenceAcknowledgement/wsrm:Nack

968 The RM Destination MAY include this element within a SequenceAcknowledgement header block. If  
969 used, the RM Destination MUST set the value of this element to a MessageNumberType representing  
970 the MessageNumber of an unreceived message in a Sequence. The RM Destination MUST NOT include  
971 a Nack element if a sibling AcknowledgementRange or None element is also present as a child of  
972 SequenceAcknowledgement. Upon the receipt of a Nack, an RM Source SHOULD retransmit the  
973 message identified by the Nack. The RM Destination MUST NOT issue a SequenceAcknowledgement  
974 containing a Nack for a message that it has previously acknowledged within a  
975 AcknowledgementRange. The RM Source SHOULD ignore a SequenceAcknowledgement containing  
976 a Nack for a message that has previously been acknowledged within a AcknowledgementRangeis  
977 OPTIONAL element, if present, MUST contain a wsrm:MessageNumberType  
978 representing the <wsrm:MessageNumber> of an unreceived message in a Sequence.  
979 This element permits the gap analysis of the <wsrm:AcknowledgementRange>  
980 elements to be performed at the RM Destination rather than at the RM Source  
981 which may yield performance benefits in certain environments. The <wsrm:Nack>  
982 element MUST NOT be present if a sibling <wsrm:AcknowledgementRange> or  
983 <wsrm:None> element is also present as a child of  
984 <wsrm:SequenceAcknowledgement>. Upon the receipt of a Nack, an RM Source  
985 SHOULD retransmit the message identified by the Nack. The RM Destination MUST  
986 NOT issue a <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a  
987 message that it has previously acknowledged within a  
988 <wsrm:AcknowledgementRange>. The RM Source SHOULD ignore a  
989 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that  
990 has previously been acknowledged within a <wsrm:AcknowledgementRange>.

991 /wsrm:SequenceAcknowledgement/wsrm:None

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

996 `/wsrm:SequenceAcknowledgement/{any}`

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

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

1000 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
1001 element.

1002 The following examples illustrate `SequenceAcknowledgement` ~~`<wsrm:SequenceAcknowledgement>`~~  
1003 elements:

- 1004 • Message numbers 1..10 inclusive in a Sequence have been `acceptreceived` by the RM Destination.

```
1005 <wsrm:SequenceAcknowledgement>  
1006   _____<wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
1007   _____<wsrm:AcknowledgementRange Upper="10" Lower="1"/>  
1008 </wsrm:SequenceAcknowledgement>
```

- 1009 • Message numbers 1..2, 4..6, and 8..10 inclusive in a Sequence have been `accepted by the RM`  
1010 `Destination, messages 3 and 7 have not been acceptreceived by the RM Destination, messages 3`  
1011 `and 7 have not been received.`

```
1012 <wsrm:SequenceAcknowledgement>  
1013   _____<wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
1014   _____<wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
1015   _____<wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
1016   _____<wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
1017 </wsrm:SequenceAcknowledgement>
```

- 1018 • Message number 3 in a Sequence has not been `acceptreceived` by the RM Destination.

```
1019 <wsrm:SequenceAcknowledgement>  
1020   _____<wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
1021   _____<wsrm:Nack>3</wsrm:Nack>  
1022 </wsrm:SequenceAcknowledgement>
```

### 1023 **3.10 MakeConnection**

1024 When an Endpoint is not directly addressable (e.g. behind a firewall or not able to allow incoming  
1025 connections), an anonymous URI in the EPR address property can indicate such an Endpoint. The WS-  
1026 Addressing anonymous URI is one such anonymous URI. This specification defines a URI template (the  
1027 WS-RM anonymous URI) which may be used to uniquely identify anonymous Endpoints.

```
1028 http://docs.oasis-open.org/ws-rx/wsrn/200608/anonymous?id={uuid}
```

1029 This URI template in an EPR indicates a protocol-specific back-channel will be established through a  
1030 mechanism such as `MakeConnection`, defined below. When using this URI template, "{uuid}" MUST be  
1031 replaced by a UUID value as defined by RFC4122[UUID]. This UUID value uniquely distinguishes the  
1032 Endpoint. A sending Endpoint SHOULD Transmit messages at Endpoints identified with the URI template  
1033 using a protocol-specific back-channel, including but not limited to those established with a  
1034 `MakeConnection` message. Note, this URI is semantically similar to the WS-Addressing anonymous  
1035 URI if a protocol-specific back-channel is available.

1036 The `MakeConnection` is a one-way operation that establishes a contextualized back-channel for the  
1037 transmission of messages according to matching criteria (defined below). In the non-faulting case, if no  
1038 matching message is available then no SOAP envelopes will be returned on the back-channel. A common  
1039 usage will be a client RM Destination sending `MakeConnection` to a server RM Source for the purpose  
1040 of receiving asynchronous response messages.

1041 The following exemplar defines the `MakeConnection` syntax:

```
1042 <wsm:MakeConnection ...>  
1043   <wsm:Identifier ...> xs:anyURI </wsm:Identifier> ?  
1044   <wsm:Address ...> xs:anyURI </wsm:Address> ?  
1045   ...  
1046 </wsm:MakeConnection>
```

1047 `/wsm:MakeConnection`

1048 This element allows the sender to create a transport-specific back-channel that can be used to return a  
1049 message that matches the selection criteria. Endpoints MUST NOT send this element as a header block.

1050 `/wsm:MakeConnection/wsm:Identifier`

1051 This element specifies the WS-RM Sequence Identifier that establishes the context for the transport-  
1052 specific back-channel. The Sequence Identifier should be compared with the Sequence Identifiers  
1053 associated with the messages held by the sending Endpoint, and if there is a matching message it will be  
1054 returned. If this element is omitted from the message then the `Address` MUST be included in the  
1055 message.

1056 `/wsm:MakeConnection/wsm:Identifier/@{any}`

1057 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
1058 element.

1059 `/wsm:MakeConnection/wsm:Address`

1060 This element specifies the URI (`wsa:Address`) of the initiating Endpoint. Endpoints MUST NOT return  
1061 messages on the transport-specific back-channel unless they have been addressed to this URI. This  
1062 `Address` property and a message's WS-Addressing destination property are considered identical when  
1063 they are exactly the same character-for-character. Note that URIs which are not identical in this sense  
1064 may in fact be functionally equivalent. Examples include URI references which differ only in case, or  
1065 which are in external entities which have different effective base URIs. If this element is omitted from the  
1066 message then the `Identifier` MUST be included in the message.

1067 `/wsm:MakeConnection/wsm:Address/@{any}`

1068 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
1069 element.

1070 `/wsm:MakeConnection/{any}`

1071 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,  
1072 to be passed. This allows fine-tuning of the messages to be returned, additional selection criteria included  
1073 here are logically ANDed with the `Address` and/or `Identifier`. If an extension is not supported by the  
1074 Endpoint then it should return a `UnsupportedSelection` fault.

1075 `/wsm:MakeConnection/@{any}`

1076 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
1077 element.

1078 If both `Identifier` and `Address` are present, then the Endpoint processing the `MakeConnection`  
1079 message MUST insure that any SOAP Envelope flowing on the backchannel MUST be associated with  
1080 the given Sequence and MUST be addressed to the given URI.

1081 The management of messages that are awaiting the establishment of a back-channel to their receiving  
1082 Endpoint is an implementation detail that is outside the scope of this specification. Note, however, that  
1083 these messages form a class of asynchronous messages that is not dissimilar from "ordinary"  
1084 asynchronous messages that are waiting for the establishment of a connection to their destination  
1085 Endpoints.

1086 This specification places no constraint on the types of messages that can be returned on the transport-  
1087 specific back-channel. As in an asynchronous environment, it is up to the recipient of the  
1088 `MakeConnection` message to decide which messages are appropriate for transmission to any particular  
1089 Endpoint. However, the Endpoint processing the `MakeConnection` message MUST insure that the  
1090 messages match the selection criteria as specified by the child elements of the `MakeConnection`  
1091 element.

### 1092 **3.11 MessagePending**

1093 When `MakeConnection` is used, and a message is returned on the transport-specific back-channel, the  
1094 `MessagePending` header SHOULD be included on the returned message as an indicator whether there  
1095 are additional messages waiting to be retrieved using the same selection criteria that was specified in the  
1096 `MakeConnection` element.

1097 The following exemplar defines the `MessagePending` syntax:

```
1098 <wsrm:MessagePending pending="xs:boolean" ...>  
1099 ...  
1100 </wsrm:MessagePending>
```

1101 `/wsrm:MessagePending`

1102 This element indicates whether additional messages are waiting to be retrieved.

1103 `/wsrm:MessagePending@pending`

1104 This attribute, when set to "true", indicates that there is at least one message waiting to be retrieved.

1105 When this attribute is set to "false" it indicates there are currently no messages waiting to be retrieved.

1106 `/wsrm:MessagePending/{any}`

1107 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,  
1108 to be passed.

1109 `/wsrm:MessagePending/@{any}`

1110 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
1111 element.

1112 The absence of the `MessagePending` header has no implication as to whether there are additional  
1113 messages waiting to be retrieved.

## 1114 4 Faults

1114 Faults for the `CreateSequence` message exchange are treated as defined in WS-Addressing. `Create`  
1115 `Sequence Refused` is a possible fault reply for this operation. `Unknown Sequence` is a fault generated by  
1116 `Endpoints` when messages carrying RM header blocks targeted at unrecognized or terminated Sequences  
1117 are detected. `WSRM Required` is a fault generated an RM Destination that requires the use of WS-RM on  
1118 a `Received` message that did not use the protocol. All other faults in this section relate to known  
1119 Sequences. RM Destinations that generate Sequence faults SHOULD send those faults to the same  
1120 [destination] as `Acknowledgement`. MThe fault definitions defined in this section reference certain abstract  
1121 properties, such as [fault endpoint], that are defined in section 3 of the WS-Addressing [WS-Addressing]  
1122 specification. Endpoints compliant with this specification MUST include required Message Addressing  
1123 Properties on all fault messages.

1114 Entities that generate WS-ReliableMessaging faults MUST include as the [action] property the default fault  
1115 action IRI defined below. The value from the W3C Recommendation is below for informational purposes:

1114 <http://docs.oasis-open.org/ws-rx/wsrn/200608/fault>

1114 The faults defined in this section are generated if the condition stated in the preamble is met. Fault  
1115 handling rules are defined in section 6 of WS-Addressing SOAP Binding.

1114 ~~Faults for this operation are treated as defined in WS-Addressing. `CreateSequenceRefused` is a possible~~  
1115 ~~fault reply for this operation. `UnknownSequence` is a fault generated by endpoints when messages~~  
1116 ~~carrying RM header blocks targeted at unrecognized or terminated sequences are detected, these faults~~  
1117 ~~are also treated as defined in WS-Addressing. All other faults in this section relate to the processing of RM~~  
1118 ~~header blocks targeted at known sequences and are collectively referred to as sequence faults. Sequence~~  
1119 ~~faults SHOULD be sent to the same [destination] as `<wsrm:SequenceAcknowledgement>` messages.~~  
1120 ~~These faults are correlated using the Sequence identifier carried in the detail.~~

1114 ~~WS-ReliableMessaging faults MUST include as the [action] property the default fault action IRI defined in~~  
1115 ~~the version of WS-Addressing used in the message. The value from the current version is below for~~  
1116 ~~informational purposes:~~

1114 ~~<http://schemas.xmlsoap.org/ws/2004/08/addressing/fault>~~

1114 ~~The faults defined in this section are generated if the condition stated in the preamble is met. Fault~~  
1115 ~~handling rules are defined in section 4 of WS-Addressing.~~

1114 The definitions of faults use the following properties:

1114 [Code] The fault code.

1114 [Subcode] The fault subcode.

1114 [Reason] The English language reason element.

1114 [Detail] The detail element(s). If absent, no detail element is defined for the fault. If more than one detail  
1115 element is defined for a fault, implementations MUST include the elements in the order that they are  
1116 specified. If absent, no detail element is defined for the fault.

1114 Entities that generate WS-ReliableMessaging faults MUST set the [Code] property toThe [Code] property  
1115 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



1114 The properties above bind to a SOAP 1.2 fault as follows:

```
1115 <S:Envelope>
1116   <S:Header>
1117     <wsa:Action>
1118       http://docs.oasis-open.org/ws-
1119       rx/wsrn/200608/schemas.xmlsoap.org/ws/2004/08/addressing/fault
1120     </wsa:Action>
1121     <!-- Headers elided for clarity. -->
1122   </S:Header>
1123   <S:Body>
1124     <S:Fault>
1125       <S:Code>
1126         <S:Value> [Code] </S:Value>
1127         <S:Subcode>
1128           <S:Value> [Subcode] </S:Value>
1129         </S:Subcode>
1130       </S:Code>
1131       <S:Reason>
1132         <S:Text xml:lang="en"> [Reason] </S:Text>
1133       </S:Reason>
1134       <S:Detail>
1135         [Detail]
1136       ...
1137     </S:Detail>
1138   </S:Fault>
1139 </S:Body>
1140 </S:Envelope>
```

1141 The properties above bind to a SOAP 1.1 fault as follows when the fault is triggered by processing an RM  
1142 header block:

```
1143 <S11:Envelope>
1144   <S11:Header>
1145     <wsrm:SequenceFault>
1146       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
1147       <wsrm:Detail> [Detail] </wsrm:Detail>
1148     ...
1149   </wsrm:SequenceFault>
1150   <!-- Headers elided for clarity. -->
1151 </S11:Header>
1152 <S11:Body>
1153   <S11:Fault>
1154     <faultcode> [Code] </faultcode>
1155     <faultstring> [Reason] </faultstring>
1156   </S11:Fault>
1157 </S11:Body>
1158 </S11:Envelope>
```

1159 The properties bind to a SOAP 1.1 fault as follows when the fault is generated as a result of processing a  
1160 [CreateSequence](#) request message:

```
1161 <S11:Envelope>
1162   <S11:Body>
1163     <S11:Fault>
1164       <faultcode> [Subcode] </faultcode>
1165       <faultstring xml:lang="en"> [Reason] </faultstring>
1166     </S11:Fault>
1167   </S11:Body>
1168 </S11:Envelope>
```

## 1169 4.1 SequenceFault Element

1170 The purpose of the [SequenceFault](#) element is to carry the specific details of a fault generated during  
1171 the reliable messaging specific processing of a message belonging to a Sequence. WS-  
1172 ReliableMessaging nodes MUST use the [SequenceFault](#) container only in conjunction with the SOAP  
1173 1.1 fault mechanism. WS-ReliableMessaging nodes MUST NOT use the [SequenceFault](#) container in  
1174 conjunction with the SOAP `<wsrm:SequenceFault>` element is to carry the specific details of a fault  
1175 generated during the reliable messaging specific processing of a message belonging to a Sequence. The  
1176 `<wsrm:SequenceFault>` container MUST only be used in conjunction with the SOAP1.1 fault-  
1177 mechanism. It MUST NOT be used in conjunction with the SOAP1.2 binding.

1178 The following exemplar defines its syntax:

```
1179 <wsrm:SequenceFault ...>  
1180   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>  
1181   <wsrm:Detail> ... </wsrm:Detail> ?  
1182   ...  
1183 </wsrm:SequenceFault>
```

1184 The following describes the content model of the `SequenceFault` element.

1185 `/wsrm:SequenceFault`

1186 This is the element containing Sequence information for WS-ReliableMessaging

1187 `/wsrm:SequenceFault/wsrm:FaultCode`

1188 [WS-ReliableMessaging nodes that generate a `SequenceFault` MUST set the value of this element to a  
1189 qualified name from the set of fault \[Subcodes\] defined below.](#)

1190 [/`wsrm:SequenceFault/wsrm:Detail`](#)

1191 [This element, if present, carries application specific error information related to the fault being described.](#)

1192 [/`wsrm:SequenceFault/wsrm:Detail/{any}`](#)

1193 [The application specific error information related to the fault being described.](#)

1194 [/`wsrm:SequenceFault/wsrm:Detail/@{any}`](#)

1195 [The application specific error information related to the fault being described.](#)

1196 [This element, if present, MUST contain a qualified name from the set of fault \[Subcodes\] defined below.](#)

1197 `/wsrm:SequenceFault/{any}`

1198 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,  
1199 to be passed.

1200 `/wsrm:SequenceFault/@{any}`

1201 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
1202 element.

## 1203 4.2 Sequence Terminated

1204 The [Endpoint that generates this fault SHOULD make every reasonable effort to notify the corresponding  
1205 Eis fault is sent by either the RM Source or the RM Destination to indicate that it has either encountered  
1206 an unrecoverable condition, or has detected a violation of the protocol and as a consequence, has chosen](#)

1207 ~~to terminate the sequence. The endpoint that generates this fault should make every reasonable effort to~~  
 1208 ~~notify the corresponding~~ endpoint of this decision.

1209 Properties:

1210 [Code] Sender or Receiver

1211 [Subcode] wsrn:SequenceTerminated

1212 [Reason] The Sequence has been terminated due to an unrecoverable error.

1213 [Detail]

1214 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`

<u>Generated by</u>	<u>Condition</u>	<u>Action Upon Generation</u>	<u>Action Upon Receipt</u>
<u>RM Source or RM Destination.</u>	<u>Encountering an unrecoverable condition or detection of violation of the protocol.</u>	<u>Sequence termination.</u>	<u>MUST terminate the Sequence if not otherwise terminated.</u>

### 1215 4.3 Unknown Sequence

1216 ~~This fault is sent by either the RM Source or the RM Destination in response to a message containing an~~  
 1217 ~~unknown or terminated sequence identifier.~~

1218 Properties:

1219 [Code] Sender

1220 [Subcode] wsrn:UnknownSequence

1221 [Reason] The value of wsrn:Identifier is not a known Sequence identifier.

1222 [Detail]

1223 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`

<u>Generated by</u>	<u>Condition</u>	<u>Action Upon Generation</u>	<u>Action Upon Receipt</u>
<u>RM Source or RM Destination.</u>	<u>In response to a message containing an unknown or terminated Sequence identifier.</u>	<u>None.</u>	<u>MUST terminate the Sequence if not otherwise terminated.</u>

### 1224 4.4 Invalid Acknowledgement

1225 ~~An example of when this fault is generated is when a message is Received by the RM Source~~  
 1226 ~~containing~~This fault is sent by the RM Source in response to a ~~<wsrm:SequenceAcknowledgement>~~  
 1227 ~~that violates the cumulative acknowledgement invariant. An example of such a violation would be a~~  
 1228 ~~SequenceAcknowledgement~~ covering messages that have not been sent.

1229 [Code] Sender

1230 [Subcode] wsrn:InvalidAcknowledgement

1231 [Reason] The SequenceAcknowledgement violates the cumulative **A**acknowledgement invariant.

1232 [Detail]

1233 `<wsrm:SequenceAcknowledgement ...> ... </wsrm:SequenceAcknowledgement>`

<u>Generated by</u>	<u>Condition</u>	<u>Action Upon Generation</u>	<u>Action Upon Receipt</u>
<u>RM Source.</u>	<u>In response to a SequenceAcknowledgement that violate the invariants stated in 2.3 or any of the requirements in 3.6 about valid combinations of AckRange, Nack and None in a single SequenceAcknowledgement element or with respect to already Received such elements.</u>	<u>Unspecified.</u>	<u>Unspecified.</u>

## 1234 **4.5 Message Number Rollover**

1235 If the condition listed below is reached, the RM Destination MUST generate this fault~~This fault is sent to-~~  
1236 indicate that message numbers for a sequence have been exhausted.

1237 Properties:

1238 [Code] Sender

1239 [Subcode] wsrn:MessageNumberRollover

1240 [Reason] The maximum value for wsrn:MessageNumber has been exceeded.

1241 [Detail]

1242 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`  
1243 `<wsrm:MaxMessageNumber> wsrn:MessageNumberType </wsrm:MaxMessageNumber>`

<u>Generated by</u>	<u>Condition</u>	<u>Action Upon Generation</u>	<u>Action Upon Receipt</u>
RM Destination.	Message number in <u>/wsrm:Sequence/wsr</u> <u>m:MessageNumber</u> of a Received message exceeds the internal limitations of an RM Destination or reaches the maximum value of <u>9,223,372,036,854,775,8</u> <u>07.</u>	RM Destination SHOULD continue to accept undelivered messages until the Sequence is closed or terminated.	RM Source SHOULD continue to retransmit undelivered messages until the Sequence is closed or terminated.

## 1244 4.6 Create Sequence Refused

1245 This fault is sent in response to a create sequence request that cannot be satisfied.

1246 Properties:

1247 [Code] Sender

1248 [Subcode] wsrm:CreateSequenceRefused

1249 [Reason] The create Sequence request has been refused by the RM Destination.

1250 [Detail]

1251 `xs:any`

<u>Generated by</u>	<u>Condition</u>	<u>Action Upon Generation</u>	<u>Action Upon Receipt</u>
RM Destination.	In response to a <u>CreateSequence</u> message when the RM Destination does not wish to create a new Sequence.	Unspecified.	Sequence terminated.

## 1252 4.7 Sequence Closed

1253 This fault is generated by an RM Destination to indicate that the specified Sequence has been closed.

1254 This fault MUST be generated when an RM Destination is asked to accept a message for a Sequence that  
1255 is closed or when an RM Destination is asked to close a Sequence that is already sent by an RM-

1256 Destination to indicate that the specified sequence has been closed. This fault MUST be generated when-  
1257 an RM Destination is asked to receive a message for a sequence that is closed.

1258 Properties:

1259 [Code] Sender

1260 [Subcode] wsrm:SequenceClosed

1261 [Reason] The Sequence is closed and can not accept~~sequence is closed and can not receive~~ new  
1262 messages.

1263 [Detail]

1264 `<wsrm:Identifier...> xs:anyURI </wsrm:Identifier>`

<u>Generated by</u>	<u>Condition</u>	<u>Action Upon Generation</u>	<u>Action Upon Receipt</u>
<u>RM Destination.</u>	<u>In response to a message that belongs to a Sequence that is already closed.</u>	<u>Unspecified.</u>	<u>Sequence closed.</u>

## 1265 **4.8 WSRM Required**

1266 If an RM Destination requires the use of WS-RM, this fault is generated when it **R**receives an incoming  
1267 message that did not use this protocol.

1268 Properties:

1269 [Code] Sender

1270 [Subcode] wsrm:WSRMRequired

1271 [Reason] The RM Destination requires the use of WSRM.

1272 [Detail]

1273 `xs:any`

1274 **2 Security Considerations**

<u>Generated by</u>	<u>Condition</u>	<u>Action Upon Generation</u>	<u>Action Upon Receipt</u>
<u>RM Destination.</u>	<u>On receipt of a message that does not use this protocol and for which this protocol is required.</u>	<u>Unspecified.</u>	<u>Unspecified.</u>

1275 **4.9 Unsupported Selection**

1276 The QName of the unsupported element(s) are included in the detail.

1277 Properties:

1278 [Code] Receiver

1279 [Subcode] wsrn:UnsupportedSelection

1280 [Reason] The extension element used in the message selection is not supported by the RM Source

1281 [Detail]

1282 <wsrm:UnsupportedElement> xs:QName </wsrm:UnsupportedElement>+

<u>Generated by</u>	<u>Condition</u>	<u>Action Upon Generation</u>	<u>Action Upon Receipt</u>
<u>RM Source or RM Destination.</u>	<u>In response to a MakeConnection message containing a selection criteria in the extensibility section of the message that is not supported</u>	<u>Unspecified.</u>	<u>Unspecified.</u>

## 1283 **5 Security Threats and Countermeasures**

1284 This specification considers two sets of security requirements, those of the applications that use the WS-  
1285 RM protocol and those of the protocol itself.

1286 This specification makes no assumptions about the security requirements of the applications that use WS-  
1287 RM. However, once those requirements have been satisfied within a given operational context, the  
1288 addition of WS-RM to this operational context should not undermine the fulfillment of those requirements;  
1289 the use of WS-RM should not create additional attack vectors within an otherwise secure system.

1290 There are many other security concerns that one may need to consider when implementing or using this  
1291 protocol. The material below should not be considered as a "check list". Implementers and users of this  
1292 protocol are urged to perform a security analysis to determine their particular threat profile and the  
1293 appropriate responses to those threats.

1294 Implementers are also advised that there is a core tension between security and reliable messaging that  
1295 can be problematic if not addressed by implementations; one aspect of security is to prevent message  
1296 replay but one of the invariants of this protocol is to resend messages until they are acknowledged.  
1297 Consequently, if the security sub-system processes a message but a failure occurs before the reliable  
1298 messaging sub-system Receives that message, then it is possible (and likely) that the security sub-system  
1299 will treat subsequent copies as replays and discard them. At the same time, the reliable messaging sub-  
1300 system will likely continue to expect and even solicit the missing message(s). Care should be taken to  
1301 avoid and prevent this condition.

### 1302 **5.1 Threats and Countermeasures**

1303 The primary security requirement of this protocol is to protect the specified semantics and protocol  
1304 invariants against various threats. The following sections describe several threats to the integrity and  
1305 operation of this protocol and provide some general outlines of countermeasures to those threats.  
1306 Implementers and users of this protocol should keep in mind that all threats are not necessarily applicable  
1307 to all operational contexts.

#### 1308 **5.1.1 Integrity Threats**

1309 In general, any mechanism which allows an attacker to alter the information in a Sequence Traffic  
1310 Message, Sequence Lifecycle Message, Acknowledgement Messages, Acknowledgement Request, or  
1311 Sequence-related fault, or which allows an attacker to alter the correlation of a RM Protocol Header Block  
1312 to its intended message represents a threat to the WS-RM protocol.

1313 For example, if an attacker is able to swap `Sequence` headers on messages in transit between the RM  
1314 Source and RM Destination then they have undermined the implementation's ability to guarantee the first  
1315 invariant described in Section 2.3. The result is that there is no way of guaranteeing that messages will be  
1316 Delivered to the Application Destination in the same order that they were sent by the Application Source.

##### 1317 **5.1.1.1 Countermeasures**

1318 Integrity threats are generally countered via the use of digital signatures some level of the communication  
1319 protocol stack. Note that, in order to counter header swapping attacks, the signature SHOULD include  
1320 both the SOAP body and any relevant SOAP headers (e.g. `Sequence` header). Because some headers  
1321 (`AckRequested`, `SequenceAcknowledgement`) are independent of the body of the SOAP message in which  
1322 they occur, implementations MUST allow for signatures that cover only these headers.



## 1323 **5.1.2 Resource Consumption Threats**

1324 The creation of a Sequence with an RM Destination consumes various resources on the systems used to  
1325 implement that RM Destination. These resources can include network connections, database tables,  
1326 message queues, etc. This behavior can be exploited to conduct denial of service attacks against an RM  
1327 Destination. For example, a simple attack is to repeatedly send `CreateSequence` messages to an RM  
1328 Destination. Another attack is to create a Sequence for a service that is known to require in-order  
1329 message Delivery and use this Sequence to send a stream of very large messages to that service,  
1330 making sure to omit message number “1” from that stream.

### 1331 **5.1.2.1 Countermeasures**

1332 There are a number of countermeasures against the described resource consumption threats. The  
1333 technique advocated by this specification is for the RM Destination to restrict the ability to create a  
1334 Sequence to a specific set of entities/principals. This reduces the number of potential attackers and, in  
1335 some cases, allows the identity of any attackers to be determined.

1336 The ability to restrict Sequence creation depends, in turn, upon the RM Destination's ability identify and  
1337 authenticate the RM Source that issued the `CreateSequence` message.

## 1338 **5.1.3 Sequence Spoofing Threats**

1339 Sequence spoofing is a class of threats in which the attacker uses knowledge of the `Identifier` for a  
1340 particular Sequence to forge Sequence Lifecycle or Traffic Messages. For example the attacker creates a  
1341 fake `TerminateSequence` message that references the target Sequence and sends this message to the  
1342 appropriate RM Destination. Some sequence spoofing attacks also require up-to-date knowledge of the  
1343 current `MessageNumber` for their target Sequence.

1344 In general any Sequence Lifecycle Message, RM Protocol Header Block, or sequence-correlated SOAP  
1345 fault (e.g. `InvalidAcknowledgement`) can be used by someone with knowledge of the Sequence identifier  
1346 to attack the Sequence. These attacks are “two-way” in that an attacker may choose to target the RM  
1347 Source by, for example, inserting a fake `SequenceAcknowledgement` header into a message that it sends  
1348 to the `AcksTo` EPR of an RM Source.

### 1349 **5.1.3.1 Sequence Hijacking**

1350 Sequence hijacking is a specific case of a sequence spoofing attack. The attacker attempts to inject  
1351 Sequence Traffic Messages into an existing Sequence by inserting fake `Sequence` headers into those  
1352 messages.

1353 Note that “sequence hijacking” should not be equated with “security session hijacking”. Although a  
1354 Sequence may be bound to some form of a security session in order to counter the threats described in  
1355 this section, applications MUST NOT rely on WS-RM-related information to make determinations about  
1356 the identity of the entity that created a message; applications SHOULD rely only upon information that is  
1357 established by the security infrastructure to make such determinations. Failure to observe this rule  
1358 creates, among other problems, a situation in which the absence of WS-RM may deprive an application of  
1359 the ability to authenticate its peers even though the necessary security processing has taken place.

### 1360 **5.1.3.2 Countermeasures**

1361 There are a number of countermeasures against sequence spoofing threats. The technique advocated by  
1362 this specification is to consider the Sequence to be a shared resource that is jointly owned by the RM.

1363 Source that initiated its creation (i.e. that sent the `CreateSequence` message) and the RM Destination that  
1364 serves as its terminus (i.e. that sent the `CreateSequenceResponse` message). To counter sequence  
1365 spoofing attempts the RM Destination SHOULD ensure that every message or fault that it Receives that  
1366 refers to a particular Sequence originated from the RM Source that jointly owns the referenced Sequence.  
1367 For its part the RM Source SHOULD ensure that every message or fault that it Receives that refers to a  
1368 particular Sequence originated from the RM Destination that jointly owns the referenced Sequence.  
  
1369 For the RM Destination to be able to identify its sequence peer it MUST be able to identify and  
1370 authenticate the entity that sent the `CreateSequence` message. Similarly for the RM Source to identify its  
1371 sequence peer it MUST be able to identify and authenticate the entity that sent the  
1372 `CreateSequenceResponse` message. For either the RM Destination or the RM Source to determine if a  
1373 message was sent by its sequence peer it MUST be able to identify and authenticate the initiator of that  
1374 message and, if necessary, correlate this identity with the sequence peer identity established at sequence  
1375 creation time.

## 1376 **5.2 Security Solutions and Technologies**

1377 The security threats described in the previous sections are neither new nor unique. The solutions that  
1378 have been developed to secure other SOAP-based protocols can be used to secure WS-RM as well. This  
1379 section maps the facilities provided by common web services security solutions against countermeasures  
1380 described in the previous sections.

1381 Before continuing this discussion, however, some examination of the underlying requirements of the  
1382 previously described countermeasures is necessary. Specifically it should be noted that the technique  
1383 described in Section 5.1.2.1 has two components. Firstly, the RM Destination identifies and authenticates  
1384 the issuer of a `CreateSequence` message. Secondly, the RM Destination to performs an authorization  
1385 check against this authenticated identity and determines if the RM Source is permitted to create  
1386 Sequences with the RM Destination. Since the facilities for performing this authorization check (runtime  
1387 infrastructure, policy frameworks, etc.) lie completely within the domain of individual implementations, any  
1388 discussion of such facilities is considered to be beyond the scope of this specification.

### 1389 **5.2.1 Transport Layer Security**

1390 This section describes how the the facilities provided by SSL/TLS [RFC 4346] can be used to implement  
1391 the countermeasures described in the previous sections. The use of SSL/TLS is subject to the constraints  
1392 defined in Section 4 of the Basic Security Profile 1.0 [BSP 1.0].

1393 The description provided here is general in nature and is not intended to serve as a complete definition on  
1394 the use of SSL/TLS to protect WS-RM. In order to interoperate implementations need to agree on the  
1395 choice of features as well as the manner in which they will be used. The mechanisms described in the  
1396 Web Services Security Policy Language [SecurityPolicy] MAY be used by services to describe the  
1397 requirements and constraints of the use of SSL/TLS.

#### 1398 **5.2.1.1 Model**

1399 The basic model for using SSL/TLS is as follows:

- 1400 1. The RM Source establishes an SSL/TLS session with the RM Destination.
- 1401 2. The RM Source uses this SSL/TLS session to send a `CreateSequence` message to the RM  
1402 Destination.

- 1403 3. The RM Destination establishes an SSL/TLS session with the RM Source and sends an  
1404 asynchronous CreateSequenceResponse using this session. Alternately it may respond with a  
1405 synchronous CreateSequenceResponse using the session established in (1).
- 1406 4. For the lifetime of the Sequence the RM Source uses the SSL/TLS session from (1) to Transmit  
1407 any and all messages or faults that refer to that Sequence.
- 1408 5. For the lifetime of the Sequence the RM Destination either uses the SSL/TLS session established  
1409 in (3) to Transmit any and all messages or faults that refer to that Sequence or, for synchronous  
1410 exchanges, the RM Destination uses the SSL/TLS session established in (1).

### 1411 **5.2.1.2 Countermeasure Implementation**

1412 Used in its simplest fashion (without relying upon any authentication mechanisms), SSL/TLS provides the  
1413 necessary integrity qualities to counter the threats described in Section 5.1.1. Note, however, that the  
1414 nature of SSL/TLS limits the scope of this integrity protection to a single transport level session. If  
1415 SSL/TLS is the only mechanism used to provide integrity, any intermediaries between the RM Source and  
1416 the RM Destination MUST be trusted to preserve the integrity of the messages that flow through them.

1417 As noted, the technique described in Sections 5.1.2.1 involves the use of authentication. This specification  
1418 advocates either of two mechanisms for authenticating entities using SSL/TLS. In both of these methods  
1419 the SSL/TLS server (the party accepting the SSL/TLS connection) authenticates itself to the SSL/TLS  
1420 client using an X.509 certificate that is exchanged during the SSL/TLS handshake.

- 1421 • **HTTP Basic Authentication:** This method of authentication presupposes that a SOAP/HTTP  
1422 binding is being used as part of the protocol stack beneath WS-RM. Subsequent to the  
1423 establishment of the the SSL/TLS session, the sending party authenticates itself to the receiving  
1424 party using HTTP Basic Authentication [RFC 2617]. For example, a RM Source might  
1425 authenticate itself to a RM Destination (e.g. when transmitting a Sequence Traffic Message) using  
1426 BasicAuth. Similarly the RM Destination might authenticate itself to the RM Source (e.g. when  
1427 sending an Acknowledgement) using BasicAuth.
- 1428 • **SSL/TLS Client Authentication:** In this method of authentication, the party initiating the  
1429 connection authenticates itself to the party accepting the connection using an X.509 certificate  
1430 that is exchanged during the SSL/TLS handshake.

1431 To implement the countermeasures described in section 5.1.2.1 the RM Source must authenticate itself  
1432 using one the above mechanisms. The authenticated identity can then be used to determine if the RM  
1433 Source is authorized to create a Sequence with the RM Destination.

1434 This specification advocates implementing the countermeasures described in section 5.1.3.2 by requiring  
1435 an RM node's Sequence peer to be equivalent to their SSL/TLS session peer. This allows the  
1436 authorization decisions described in section 5.1.3.2 to be based on SSL/TLS session identity rather than  
1437 on authentication information. For example, an RM Destination can determine that a Sequence Traffic  
1438 Message rightfully belongs to its referenced Sequence if that message arrived over the same SSL/TLS  
1439 session that was used to carry the CreateSequence message for that Sequence. Note that requiring a  
1440 one-to-one relationship between SSL/TLS session peer and Sequence peer constrains the lifetime of a  
1441 SSL/TLS-protected Sequence to be less than or equal to the lifetime of the SSL/TLS session that is used  
1442 to protect that Sequence.

1443 This specification does not preclude the use of other methods of using SSL/TLS to implement the  
1444 countermeasures (such as associating specific authentication information with a Sequence) although such  
1445 methods are not covered by this document.

1446 Issues specific to the life-cycle management of SSL/TLS sessions (such as the resumption of a SSL/TLS  
1447 session) are outside the scope of this specification.

## 1448 **5.2.2 SOAP Message Security**

1449 The mechanisms described in WS-Security may be used in various ways to implement the  
1450 countermeasures described in the previous sections. This specification advocates using the protocol  
1451 described by WS-SecureConversation [SecureConversation] (optionally in conjunction with WS-Trust  
1452 [Trust]) as a mechanism for protecting Sequences. The use of WS-Security (as an underlying component  
1453 of WS-SecureConversation) is subject to the constraints defined in the Basic Security Profile 1.0.

1454 The description provided here is general in nature and is not intended to serve as a complete definition on  
1455 the use of WS-SecureConversation/WS-Trust to protect WS-RM. In order to interoperate implementations  
1456 need to agree on the choice of features as well as the manner in which they will be used. The  
1457 mechanisms described in the Web Services Security Policy Language MAY be used by services to  
1458 describe the requirements and constraints of the use of WS-SecureConversation.

### 1459 **5.2.2.1 Model**

1460 The basic model for using WS-SecureConversation is as follows:

- 1461 1. The RM Source and the RM Destination create a WS-SecureConversation security context. This  
1462 may involve the participation of third parties such as a security token service. The tokens  
1463 exchanged may contain authentication claims (e.g. X.509 certificates or Kerberos service tickets).
- 1464 2. During the CreateSequence exchange, the RM Source SHOULD explicitly identify the security  
1465 context that will be used to protect the Sequence. This is done so that, in cases where the  
1466 CreateSequence message is signed by more than one security context, the RM Source can  
1467 indicate which security context should be used to protect the newly created Sequence.
- 1468 3. For the lifetime of the Sequence the RM Source and the RM Destination use the session key(s)  
1469 associated with the security context to sign (as defined by WS-Security) at least the body and any  
1470 relevant WS-RM-defined headers of any and all messages or faults that refer to that Sequence.

### 1471 **5.2.2.2 Countermeasure Implementation**

1472 Without relying upon any authentication information, the per-message signatures provide the necessary  
1473 integrity qualities to counter the threats described in Section 5.1.1.

1474 To implement the countermeasures described in section 5.1.2.1 some mutually agreed upon form of  
1475 authentication claims must be provided by the RM Source to the RM Destination during the establishment  
1476 of the Security Context. These claims can then be used to determine if the RM Source is authorized to  
1477 create a Sequence with the RM Destination.

1478 This specification advocates implementing the countermeasures described in section 5.1.3.2 by requiring  
1479 an RM node's Sequence peer to be equivalent to their security context session peer. This allows the  
1480 authorization decisions described in section 5.1.3.2 to be based on the identity of the message's security  
1481 context rather than on any authentication claims that may have been established during security context  
1482 initiation. Note that other methods of using WS-SecurityConversation to implement the countermeasures  
1483 (such as associating specific authentication claims to a Sequence) are possible but not covered by this  
1484 document.

1485 As with transport security, the requisite equivalence of a security context peer and with a Sequence peer  
1486 limits the lifetime of a Sequence to the lifetime of the protecting security context. Unlike transport security,

1487 the association between a Sequence and its protecting security context cannot always be established  
1488 implicitly at Sequence creation time. This is due to the fact that the `CreateSequence` and  
1489 `CreateSequenceResponse` messages may be signed by more than one security context.  
1490 Issues specific to the life-cycle management of WS-SecurityConversation security contexts (such as  
1491 amending or renewing contexts) are outside the scope of this specification.

## 1492 **6 Securing Sequences**

1493 As noted in Section 5, the RM Source and RM Destination should be able to protect their shared  
1494 Sequences against the threat of Sequence Spoofing attacks. There are a number of OPTIONAL means of  
1495 achieving this objective depending upon the underlying security infrastructure.

### 1496 **6.1 Securing Sequences Using WS-Security**

1497 One mechanism for protecting a Sequence is to include a security token using a  
1498 wsse:SecurityTokenReference element from WS-Security (see section 9 in WS-  
1499 SecureConversation) in the CreateSequence element. This establishes an association between the  
1500 created (and, if present, offered) Sequence(s) and the referenced security token, such that the RM Source  
1501 and Destination MUST use the security token as the basis for authorization of all subsequent interactions  
1502 related to the Sequence(s). The wsse:SecurityTokenReference explicitly identifies the token as  
1503 there may be more than one token on a CreateSequence message or inferred from the communication  
1504 context (e.g. transport protection).

1505 It is RECOMMENDED that a message independent referencing mechanism be used to identify the token,  
1506 if the token being referenced supports such mechanism.

1507 The following exemplar defines the CreateSequence syntax when extended to include a  
1508 wsse:SecurityTokenReference:

```
1509 <wsrm:CreateSequence ...>  
1510   <wsrm:AcksTo> wsa:EndpointReferenceType </wsrm:AcksTo>  
1511   <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
1512   <wsrm:Offer ...>  
1513     <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
1514     <wsrm:Endpoint> wsa:EndpointReferenceType </wsrm:Endpoint>  
1515     <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
1516     <wsrm:IncompleteSequenceBehavior>  
1517       wsrml:IncompleteSequenceBehaviorType  
1518     </wsrm:IncompleteSequenceBehavior> ?  
1519     ...  
1520   </wsrm:Offer> ?  
1521   ...  
1522   <wsse:SecurityTokenReference>  
1523     ...  
1524   </wsse:SecurityTokenReference> ?  
1525   ...  
1526 </wsrm:CreateSequence>
```

1527 /wsrm:CreateSequence/wsse:SecurityTokenReference

1528 This element uses the extensibility mechanism defined for the CreateSequence element (defined in  
1529 section 3.1) to communicate an explicit reference to the security token, using a  
1530 wsse:SecurityTokenReference as documented in WS-Security, that the RM Source and Destination  
1531 MUST use to authorize messages for the created (and, if present, the offered) Sequence(s). All  
1532 subsequent messages related to the created (and, if present, the offered) Sequence(s) MUST  
1533 demonstrate proof-of-possession of the secret associated with the token (e.g., by using or deriving from a  
1534 private or secret key).

1535 When a RM Source Transmits a CreateSequence that has been extended to include a  
1536 wsse:SecurityTokenReference it SHOULD ensure that the RM Destination both understands and  
1537 will conform with the requirements listed above. In order to achieve this, the RM Source SHOULD include  
1538 the UsesSequenceSTR element as a SOAP header block within the CreateSequence message. This  
1539 element MUST include a soap:mustUnderstand attribute with a value of 'true'. Thus the RM Source

1540 can be assured that a RM Destination that responds with a `CreateSequenceResponse` understands  
1541 and conforms with the requirements listed above. Note that an RM Destination understanding this header  
1542 does not mean that it has processed and understood any WS-Security headers, the fault behavior defined  
1543 in WS-Security still applies.

1544 The following exemplar defines the `UsesSequenceSTR` syntax:

```
1545 <wsm:UsesSequenceSTR ... />
```

1546 /wsm:UsesSequenceSTR

1547 This element SHOULD be included as a SOAP header block in `CreateSequence` messages that use the  
1548 extensibility mechanism described above in this section. The `soap:mustUnderstand` attribute value  
1549 MUST be 'true'. The receiving RM Destination MUST understand and correctly implement the extension  
1550 described above or else generate a `soap:MustUnderstand` fault, thus aborting the requested  
1551 Sequence creation.

1552 The following is an example of a `CreateSequence` message using the  
1553 `wsse:SecurityTokenReference` extension and the `UsesSequenceSTR` header block:

```
1554 <soap:Envelope ...>  
1555   <soap:Header>  
1556     ...  
1557     <wsm:UsesSequenceSTR soap:mustUnderstand='true' />  
1558     ...  
1559   </soap:Header>  
1560   <soap:Body>  
1561     <wsm:CreateSequence>  
1562       <wsm:AcksTo>  
1563         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>  
1564       </wsm:AcksTo>  
1565       <wsse:SecurityTokenReference>  
1566         ...  
1567       </wsse:SecurityTokenReference>  
1568     </wsm:CreateSequence>  
1569   </soap:Body>  
1570 </soap:Envelope>
```

## 1571 **6.2 Securing Sequences Using SSL/TLS**

1572 One mechanism for protecting a Sequence is to bind the Sequence to the underlying SSL/TLS session(s).  
1573 The RM Source indicates to the RM Destination that a Sequence is to be bound to the underlying  
1574 SSL/TLS session(s) via the `UsesSequenceSSL` header block. If the RM Source wishes to bind a  
1575 Sequence to the underlying SSL/TLS sessions(s) it MUST include the `UsesSequenceSSL` element as a  
1576 SOAP header block within the `CreateSequence` message.

1577 The following exemplar defines the `UsesSequenceSSL` syntax:

```
1578 <wsm:UsesSequenceSSL soap:mustUnderstand="true" ... />
```

1579 /wsm:UsesSequenceSSL

1580 The RM Source MAY include this element as a SOAP header block of a `CreateSequence` message to  
1581 indicate to the RM Destination that the resulting Sequence is to be bound to the SSL/TLS session that was  
1582 used to carry the `CreateSequence` message. If included, the RM Source MUST mark this header with a  
1583 `soap:mustUnderstand` attribute with a value of 'true'. The receiving RM Destination MUST understand  
1584 and correctly implement the functionality described in Section 5.2.1 or else generate a  
1585 `soap:MustUnderstand` fault, thus aborting the requested Sequence creation.

1586 Note that the use inclusion of the above header by the RM Source implies that all Sequence-related  
1587 information (Sequence Lifecycle or Acknowledgment messages or Sequence-related faults) flowing from  
1588 the RM Destination to the RM Source will be bound to the SSL/TLS session that is used to carry the  
1589 CreateSequenceResponse message.

1590 It is strongly recommended that the communication between services be secured using the mechanisms  
1591 described in WS-Security [WS-Security]. In order to properly secure messages, the body and all relevant  
1592 headers need to be included in the signature. Specifically, the <wsm:Sequence> header needs to be  
1593 signed with the body in order to "bind" the two together. The <wsm:SequenceAcknowledgement>  
1594 header may be signed independently because a reply independent of the message is not a security-  
1595 concern.

1596 Because Sequences are expected to exchange a number of messages, it is recommended that a security  
1597 context be established using the mechanisms described in WS-Trust[Trust] and WS-SecureConversation  
1598 [SecureConversation]. If a Sequence is bound to a specific destination, then the security context needs to  
1599 be established or shared with the destination servicing the Sequence. While the context can be  
1600 established at any time, it is critical that the messages establishing the Sequence be secured even if they  
1601 precede security context establishment. However, it is recommended that the security context be  
1602 established first. Security contexts are independent of reliable messaging Sequences. Consequently,  
1603 security contexts can come and go independent of the lifetime of the Sequence. In fact, it is  
1604 recommended that the lifetime of a security context be less than the lifetime of the Sequence unless the  
1605 Sequence is very short-lived.

1606 It is common for message Sequences to exchange a number of messages (or a large amount of data). As  
1607 a result, the usage profile of a Sequence is such that it is susceptible to key attacks. For this reason it is  
1608 strongly recommended that the keys be changed frequently. This "re-keying" can be effected a number of  
1609 ways. The following list outlines four common techniques:

- 1610 • Closing and re-establishing a security context
- 1611 • Exchanging new secrets between the parties
- 1612 • Using a derived key sequence and switch "generations"
- 1613 • Attaching a nonce to each message and using it in a derived key function with the shared secret

1614 The security context may be re-established using the mechanisms described in WS-Trust and WS-  
1615 SecureConversation. Similarly, secrets can be exchanged using the mechanisms described in WS-Trust.  
1616 Note, however, that the current shared secret should not be used to encrypt the new shared secret.  
1617 Derived keys, the preferred solution from this list, can be specified using the mechanisms described in  
1618 WS-SecureConversation.

1619 There is a core tension between security and reliable messaging that can be problematic if not considered  
1620 in implementations. That is, one aspect of security is to prevent message replay and the core tenet of  
1621 reliable messaging is to replay messages until they are acknowledged. Consequently, if the security sub-  
1622 system processes a message but a failure occurs before the reliable messaging sub-system records the  
1623 message (or the message is considered "processed"), then it is possible (and likely) that the security sub-  
1624 system will treat subsequent copies as replays and discard them. At the same time, the reliable  
1625 messaging sub-system will likely continue to expect and even solicit the missing message(s). Care should  
1626 be taken to avoid and prevent this rare condition.

1627 The following list summarizes common classes of attacks that apply to this protocol and identifies the  
1628 mechanism to prevent/mitigate the attacks:

- 1629 • **Message alteration**—Alteration is prevented by including signatures of the message information  
1630 using WS-Security.



- 1631 • **Message disclosure**—Confidentiality is preserved by encrypting sensitive data using WS-Security.
- 1632 • **Key integrity**—Key integrity is maintained by using the strongest algorithms possible (by comparing
- 1633 secured policies—see WS-Policy and WS-SecurityPolicy).
- 1634 • **Authentication**—Authentication is established using the mechanisms described in WS-Security
- 1635 and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- 1636 • **Accountability**—Accountability is a function of the type of and string of the key and algorithms
- 1637 being used. In many cases, a strong symmetric key provides sufficient accountability. However, in
- 1638 some environments, strong PKI signatures are required.
- 1639 • **Availability**—All reliable messaging services are subject to a variety of availability attacks. Replay
- 1640 detection is a common attack and it is recommended that this be addressed by the mechanisms
- 1641 described in WS-Security. (Note that because of legitimate message replays, detection should
- 1642 include a differentiator besides message id such as a timestamp). Other attacks, such as network-
- 1643 level denial of service attacks are harder to avoid and are outside the scope of this specification.
- 1644 That said, care should be taken to ensure that minimal state is saved prior to any authenticating
- 1645 sequences.

## 1646 **7 References**

### 1647 **7.1 Normative**

#### 1648 **[KEYWORDS]**

1649 S. Bradner, "[Key words for use in RFCs to Indicate Requirement Levels](#)," RFC 2119, Harvard University,  
1650 March 1997

#### 1651 **[SOAP 1.1]**

1652 W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May 2000.

#### 1653 **[SOAP 1.2]**

1654 W3C Recommendation, "[SOAP Version 1.2 Part 1: Messaging Framework](#)" June 2003.

#### 1655 **[URI]**

1656 T. Berners-Lee, R. Fielding, L. Masinter, "[Uniform Resource Identifiers \(URI\): Generic Syntax](#)," RFC 3986,  
1657 MIT/LCS, U.C. Irvine, Xerox Corporation, January 2005.

#### 1658 **[UUID]**

1659 [P. Leach, M. Mealling, R. Salz, "A Universally Unique Identifier \(UUID\) URN Namespace," RFC 4122, Microsoft, Refactored Networks - LLC, DataPower Technology Inc, July 2005](#)

#### 1661 **[XML]**

1662 W3C Recommendation, "[Extensible Markup Language \(XML\) 1.0 \(Second Edition\)](#)" "~~Extensible Markup Language (XML) 1.0 (Second Edition)~~", October 2000.

#### 1664 **[XML-ns]**

1665 W3C Recommendation, "[Namespaces in XML](#)," 14 January 1999.

#### 1666 **[XML-Schema Part1]**

1667 W3C Recommendation, "[XML Schema Part 1: Structures](#)," 2 May 2001.

#### 1668 **[XML-Schema Part2]**

1669 W3C Recommendation, "[XML Schema Part 2: Datatypes](#)," 2 May 2001.

#### 1670 **[XPath 1.0]**

1671 [W3C Recommendation, "XML Path Language \(XPath\) Version 1.0," 16 November 1999.](#)

#### 1672 **[WSDL 1.1]**

1673 W3C Note, "[Web Services Description Language \(WSDL 1.1\)](#)," 15 March 2001.

#### 1674 **[WS-Addressing]**

1675 [W3C Recommendation, "Web Services Addressing 1.0 - Core", May 2006](#)~~D. Box, et al, "Web Services Addressing (WS-Addressing)," August 2004.~~

1677 [W3C Recommendation, "Web Services Addressing 1.0 – SOAP Binding", May 2006.](#)

### 1678 **7.2 Non-Normative**

#### 1679 **[BSP 1.0]**

- 1680 [WS-I Working Group Draft. "Basic Security Profile Version 1.0," March 2006](#)
- 1681 **[RDDL 2.0]**
- 1682 Johnathan Borden, Tim Bray, eds. "[Resource Directory Description Language \(RDDL\) 2.0,](#)" January 2004
- 1683 **[RFC 2617]**
- 1684 [J. Franks, P. Hallam-Baker, J. Hostettler, S. Lawrence, P. Leach, A. Loutonen, L. Stewart, "HTTP](#)
- 1685 [Authentication: Basic and Digest Access Authentication," June 1999.](#)
- 1686 **[RFC 4346]**
- 1687 [T. Dierks, E. Rescorla, "The Transport Layer Security \(TLS\) Protocol Version 1.1," April 2006.](#)
- 1688 **[WS-Policy]**
- 1689 [W3C Member Submission, "Web Services Policy Framework \(WS-Policy\)," April 2006](#)
- 1690 [D. Box, et al, "Web Services Policy Framework \(WS-Policy\)," September 2004.](#)
- 1691 **[WS-PolicyAttachment]**
- 1692 [W3C Member Submission, "Web Services Policy Attachment \(WS-PolicyAttachment\)," April 2006](#)
- 1693 [Box, et al, "Web Services Policy Attachment \(WS-PolicyAttachment\)," September 2004.](#)
- 1694 **[WS-Security]**
- 1695 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "[OASIS Web Services Security:](#)
- 1696 [SOAP Message Security 1.0 \(WS-Security 2004\)," OASIS Standard 200401, March 2004.](#)
- 1697 [Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "OASIS Web Services Security:](#)
- 1698 [SOAP Message Security 1.1 \(WS-Security 2004\)," OASIS Standard 200602, February 2006.](#)
- 1699 **[RTTM]**
- 1700 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May
- 1701 1992.
- 1702 **[SecurityPolicy]**
- 1703 G. Della-Libra, et. al. "[Web Services Security Policy Language \(WS-SecurityPolicy\)"](#)"~~"Web Services-~~
- 1704 ~~Security Policy Language (WS-SecurityPolicy)"~~, July 2005
- 1705 **[SecureConversation]**
- 1706 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\),](#)" February
- 1707 2005.
- 1708 **[Trust]**
- 1709 S. Anderson, et al, "[Web Services Trust Language \(WS-Trust\),](#)" February 2005.

## 1710 Appendix A. Schema

1711 The normative schema that is defined for WS-ReliableMessaging using [XML-Schema Part1] and [XML-  
1712 Schema Part2] is located at:

1713 [http://docs.oasis-open.org/ws-rx/wsrn/200608/wsrn-1.1-schema-2006082/wsrn-1.1-schema-  
200602.xsd](http://docs.oasis-open.org/ws-rx/wsrn/200608/wsrn-1.1-schema-2006082/wsrn-1.1-schema-<br/>1714 200602.xsd)

1715 The following copy is provided for reference.

```
1716 <?xml version="1.0" encoding="UTF-8"?>
1717 <!--
1718 OASIS takes no position regarding the validity or scope of any intellectual
1719 property or other rights that might be claimed to pertain to the
1720 implementation or use of the technology described in this document or the
1721 extent to which any license under such rights might or might not be available;
1722 neither does it represent that it has made any effort to identify any such
1723 rights. Information on OASIS's procedures with respect to rights in OASIS
1724 specifications can be found at the OASIS website. Copies of claims of rights
1725 made available for publication and any assurances of licenses to be made
1726 available, or the result of an attempt made to obtain a general license or
1727 permission for the use of such proprietary rights by implementors or users of
1728 this specification, can be obtained from the OASIS Executive Director.
1729 OASIS invites any interested party to bring to its attention any copyrights,
1730 patents or patent applications, or other proprietary rights which may cover
1731 technology that may be required to implement this specification. Please
1732 address the information to the OASIS Executive Director.
1733 Copyright © OASIS Open 2002-2006. All Rights Reserved.
1734 This document and translations of it may be copied and furnished to others,
1735 and derivative works that comment on or otherwise explain it or assist in its
1736 implementation may be prepared, copied, published and distributed, in whole or
1737 in part, without restriction of any kind, provided that the above copyright
1738 notice and this paragraph are included on all such copies and derivative
1739 works. However, this document itself does not be modified in any way, such as
1740 by removing the copyright notice or references to OASIS, except as needed for
1741 the purpose of developing OASIS specifications, in which case the procedures
1742 for copyrights defined in the OASIS Intellectual Property Rights document must
1743 be followed, or as required to translate it into languages other than English.
1744 The limited permissions granted above are perpetual and will not be revoked by
1745 OASIS or its successors or assigns.
1746 This document and the information contained herein is provided on an "AS IS"
1747 basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
1748 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT
1749 INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS
1750 FOR A PARTICULAR PURPOSE.
1751 -->
1752 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1753 xmlns:wsa="http://www.w3.org/2005/08/addressing"
1754 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200608"
1755 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200608"
1756 elementFormDefault="qualified" attributeFormDefault="unqualified">
1757 <xs:import namespace="http://www.w3.org/2005/08/addressing"
1758 schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
1759 <!-- Protocol Elements -->
1760 <xs:complexType name="SequenceType">
1761 <xs:sequence>
1762 <xs:element ref="wsrm:Identifier"/>
1763 <xs:element name="MessageNumber" type="wsrm:MessageNumberType"/>
1764 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1765 maxOccurs="unbounded"/>

```

```

1766     </xs:sequence>
1767     <xs:anyAttribute namespace="##other" processContents="lax"/>
1768 </xs:complexType>
1769 <xs:element name="Sequence" type="wsrm:SequenceType"/>
1770 <xs:element name="SequenceAcknowledgement">
1771   <xs:complexType>
1772     <xs:sequence>
1773       <xs:element ref="wsrm:Identifier"/>
1774       <xs:choice>
1775         <xs:sequence>
1776           <xs:choice>
1777             <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
1778               <xs:complexType>
1779                 <xs:sequence/>
1780                 <xs:attribute name="Upper" type="xs:unsignedLong"
1781 use="required"/>
1782                 <xs:attribute name="Lower" type="xs:unsignedLong"
1783 use="required"/>
1784             <xs:anyAttribute namespace="##other" processContents="lax"/>
1785           </xs:choice>
1786         </xs:sequence>
1787         <xs:element name="None">
1788           <xs:complexType>
1789             <xs:sequence/>
1790           </xs:complexType>
1791         </xs:element>
1792       </xs:choice>
1793       <xs:element name="Final" minOccurs="0">
1794         <xs:complexType>
1795           <xs:sequence/>
1796         </xs:complexType>
1797       </xs:element>
1798     </xs:sequence>
1799     <xs:element name="Nack" type="xs:unsignedLong"
1800 maxOccurs="unbounded"/>
1801   </xs:choice>
1802   <xs:any namespace="##other" processContents="lax" minOccurs="0"
1803 maxOccurs="unbounded"/>
1804 </xs:sequence>
1805 <xs:anyAttribute namespace="##other" processContents="lax"/>
1806 </xs:complexType>
1807 </xs:element>
1808 <xs:complexType name="AckRequestedType">
1809   <xs:sequence>
1810     <xs:element ref="wsrm:Identifier"/>
1811     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1812 maxOccurs="unbounded"/>
1813   </xs:sequence>
1814   <xs:anyAttribute namespace="##other" processContents="lax"/>
1815 </xs:complexType>
1816 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1817 <xs:complexType name="MessagePendingType">
1818   <xs:sequence>
1819     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1820 maxOccurs="unbounded"/>
1821   </xs:sequence>
1822   <xs:attribute name="pending" type="xs:boolean"/>
1823   <xs:anyAttribute namespace="##other" processContents="lax"/>
1824 </xs:complexType>
1825 <xs:element name="MessagePending" type="wsrm:MessagePendingType"/>
1826 <xs:element name="Identifier">
1827   <xs:complexType>
1828     <xs:annotation>

```

```

1829     <xs:documentation>
1830         This type is for elements whose [children] is an anyURI and can have
1831         arbitrary attributes.
1832     </xs:documentation>
1833 </xs:annotation>
1834 <xs:simpleContent>
1835     <xs:extension base="xs:anyURI">
1836         <xs:anyAttribute namespace="##other" processContents="lax"/>
1837     </xs:extension>
1838 </xs:simpleContent>
1839 </xs:complexType>
1840 </xs:element>
1841 <xs:element name="Address">
1842     <xs:complexType>
1843         <xs:simpleContent>
1844             <xs:extension base="xs:anyURI">
1845                 <xs:anyAttribute namespace="##other" processContents="lax"/>
1846             </xs:extension>
1847         </xs:simpleContent>
1848     </xs:complexType>
1849 </xs:element>
1850 <xs:complexType name="MakeConnectionType">
1851     <xs:sequence>
1852         <xs:element ref="wsrm:Identifier" minOccurs="0" maxOccurs="1"/>
1853         <xs:element ref="wsrm:Address" minOccurs="0" maxOccurs="1"/>
1854         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1855         maxOccurs="unbounded"/>
1856     </xs:sequence>
1857     <xs:anyAttribute namespace="##other" processContents="lax"/>
1858 </xs:complexType>
1859 <xs:element name="MakeConnection" type="wsrm:MakeConnectionType"/>
1860 <xs:simpleType name="MessageNumberType">
1861     <xs:restriction base="xs:unsignedLong">
1862         <xs:minInclusive value="1"/>
1863         <xs:maxInclusive value="9223372036854775807"/>
1864     </xs:restriction>
1865 </xs:simpleType>
1866 <!-- Fault Container and Codes -->
1867 <xs:simpleType name="FaultCodes">
1868     <xs:restriction base="xs:QName">
1869         <xs:enumeration value="wsrm:SequenceTerminated"/>
1870         <xs:enumeration value="wsrm:UnknownSequence"/>
1871         <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1872         <xs:enumeration value="wsrm:MessageNumberRollover"/>
1873         <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1874         <xs:enumeration value="wsrm:SequenceClosed"/>
1875         <xs:enumeration value="wsrm:WSRMRequired"/>
1876         <xs:enumeration value="wsrm:UnsupportedSelection"/>
1877     </xs:restriction>
1878 </xs:simpleType>
1879 <xs:complexType name="SequenceFaultType">
1880     <xs:sequence>
1881         <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1882         <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
1883         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1884         maxOccurs="unbounded"/>
1885     </xs:sequence>
1886     <xs:anyAttribute namespace="##other" processContents="lax"/>
1887 </xs:complexType>
1888 <xs:complexType name="DetailType">
1889     <xs:sequence>
1890         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1891         maxOccurs="unbounded"/>

```

```

1892     </xs:sequence>
1893     <xs:anyAttribute namespace="##other" processContents="lax" />
1894 </xs:complexType>
1895 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1896 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1897 <xs:element name="CreateSequenceResponse"
1898 type="wsrm:CreateSequenceResponseType"/>
1899 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1900 <xs:element name="CloseSequenceResponse"
1901 type="wsrm:CloseSequenceResponseType"/>
1902 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1903 <xs:element name="TerminateSequenceResponse"
1904 type="wsrm:TerminateSequenceResponseType"/>
1905 <xs:complexType name="CreateSequenceType">
1906 <xs:sequence>
1907 <xs:element ref="wsrm:AcksTo"/>
1908 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1909 <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1910 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1911 maxOccurs="unbounded">
1912 <xs:annotation>
1913 <xs:documentation>
1914 It is the authors intent that this extensibility be used to
1915 transfer a Security Token Reference as defined in WS-Security.
1916 </xs:documentation>
1917 </xs:annotation>
1918 </xs:any>
1919 </xs:sequence>
1920 <xs:anyAttribute namespace="##other" processContents="lax" />
1921 </xs:complexType>
1922 <xs:complexType name="CreateSequenceResponseType">
1923 <xs:sequence>
1924 <xs:element ref="wsrm:Identifier"/>
1925 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1926 <xs:element name="IncompleteSequenceBehavior"
1927 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
1928 <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1929 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1930 maxOccurs="unbounded"/>
1931 </xs:sequence>
1932 <xs:anyAttribute namespace="##other" processContents="lax" />
1933 </xs:complexType>
1934 <xs:complexType name="CloseSequenceType">
1935 <xs:sequence>
1936 <xs:element ref="wsrm:Identifier"/>
1937 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1938 maxOccurs="unbounded"/>
1939 </xs:sequence>
1940 <xs:anyAttribute namespace="##other" processContents="lax" />
1941 </xs:complexType>
1942 <xs:complexType name="CloseSequenceResponseType">
1943 <xs:sequence>
1944 <xs:element ref="wsrm:Identifier"/>
1945 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1946 maxOccurs="unbounded"/>
1947 </xs:sequence>
1948 <xs:anyAttribute namespace="##other" processContents="lax" />
1949 </xs:complexType>
1950 <xs:complexType name="TerminateSequenceType">
1951 <xs:sequence>
1952 <xs:element ref="wsrm:Identifier"/>
1953 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1954 maxOccurs="unbounded"/>

```

```

1955 </xs:sequence>
1956 <xs:anyAttribute namespace="##other" processContents="lax"/>
1957 </xs:complexType>
1958 <xs:complexType name="TerminateSequenceResponseType">
1959 <xs:sequence>
1960 <xs:element ref="wsrm:Identifier"/>
1961 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1962 maxOccurs="unbounded"/>
1963 </xs:sequence>
1964 <xs:anyAttribute namespace="##other" processContents="lax"/>
1965 </xs:complexType>
1966 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
1967 <xs:complexType name="OfferType">
1968 <xs:sequence>
1969 <xs:element ref="wsrm:Identifier"/>
1970 <xs:element name="Endpoint" type="wsa:EndpointReferenceType"/>
1971 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1972 <xs:element name="IncompleteSequenceBehavior"
1973 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
1974 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1975 maxOccurs="unbounded"/>
1976 </xs:sequence>
1977 <xs:anyAttribute namespace="##other" processContents="lax"/>
1978 </xs:complexType>
1979 <xs:complexType name="AcceptType">
1980 <xs:sequence>
1981 <xs:element ref="wsrm:AcksTo"/>
1982 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1983 maxOccurs="unbounded"/>
1984 </xs:sequence>
1985 <xs:anyAttribute namespace="##other" processContents="lax"/>
1986 </xs:complexType>
1987 <xs:element name="Expires">
1988 <xs:complexType>
1989 <xs:simpleContent>
1990 <xs:extension base="xs:duration">
1991 <xs:anyAttribute namespace="##other" processContents="lax"/>
1992 </xs:extension>
1993 </xs:simpleContent>
1994 </xs:complexType>
1995 </xs:element>
1996 <xs:simpleType name="IncompleteSequenceBehaviorType">
1997 <xs:restriction base="xs:string">
1998 <xs:enumeration value="DiscardEntireSequence"/>
1999 <xs:enumeration value="DiscardFollowingFirstGap"/>
2000 <xs:enumeration value="NoDiscard"/>
2001 </xs:restriction>
2002 </xs:simpleType>
2003 <xs:element name="UsesSequenceSTR">
2004 <xs:sequence/>
2005 <xs:anyAttribute namespace="##other" processContents="lax"/>
2006 </xs:element>
2007 <xs:element name="UsesSequenceSSL">
2008 <xs:sequence/>
2009 <xs:anyAttribute namespace="##other" processContents="lax"/>
2010 </xs:element>
2011 <xs:element name="UnsupportedElement">
2012 <xs:simpleType>
2013 <xs:restriction base="xs:QName"/>
2014 </xs:simpleType>
2015 </xs:element>
2016 </xs:schema>

```



## 2017 Appendix B. WSDL

2018 The normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

2019 <http://docs.oasis-open.org/ws-rx/wsrn/200608/wsd/wsrn-1.1-wsd-200608.wsd>

2020 The following non-normative copy is provided for reference.

```
2021 <?xml version="1.0" encoding="utf-8"?>
2022 <!--
2023 OASIS takes no position regarding the validity or scope of any intellectual
2024 property or other rights that might be claimed to pertain to the
2025 implementation or use of the technology described in this document or the
2026 extent to which any license under such rights might or might not be available;
2027 neither does it represent that it has made any effort to identify any such
2028 rights. Information on OASIS's procedures with respect to rights in OASIS
2029 specifications can be found at the OASIS website. Copies of claims of rights
2030 made available for publication and any assurances of licenses to be made
2031 available, or the result of an attempt made to obtain a general license or
2032 permission for the use of such proprietary rights by implementors or users of
2033 this specification, can be obtained from the OASIS Executive Director.
2034 OASIS invites any interested party to bring to its attention any copyrights,
2035 patents or patent applications, or other proprietary rights which may cover
2036 technology that may be required to implement this specification. Please
2037 address the information to the OASIS Executive Director.
2038 Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
2039 This document and translations of it may be copied and furnished to others,
2040 and derivative works that comment on or otherwise explain it or assist in its
2041 implementation may be prepared, copied, published and distributed, in whole or
2042 in part, without restriction of any kind, provided that the above copyright
2043 notice and this paragraph are included on all such copies and derivative
2044 works. However, this document itself does not be modified in any way, such as
2045 by removing the copyright notice or references to OASIS, except as needed for
2046 the purpose of developing OASIS specifications, in which case the procedures
2047 for copyrights defined in the OASIS Intellectual Property Rights document must
2048 be followed, or as required to translate it into languages other than English.
2049 The limited permissions granted above are perpetual and will not be revoked by
2050 OASIS or its successors or assigns.
2051 This document and the information contained herein is provided on an "AS IS"
2052 basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
2053 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT
2054 INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS
2055 FOR A PARTICULAR PURPOSE.
2056 -->
2057 <wSDL:definitions xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
2058 xmlns:xs="http://www.w3.org/2001/XMLSchema"
2059 xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
2060 open.org/ws-rx/wsrn/200608" xmlns:tns="http://docs.oasis-open.org/ws-
2061 rx/wsrn/200608/wsd" targetNamespace="http://docs.oasis-open.org/ws-
2062 rx/wsrn/200608/wsd">
2063   <wSDL:types>
2064     <xs:schema
2065       <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsrn/200608"
2066       schemaLocation="http://docs.oasis-open.org/ws-rx/wsrn/200608/wsrn-1.1-schema-
2067       200608.xsd"/>
2068     </xs:schema>
2069   </wSDL:types>
2070   <wSDL:message name="CreateSequence">
2071     <wSDL:part name="create" element="rm:CreateSequence"/>
```

```

2072 </wsdl:message>
2073 <wsdl:message name="CreateSequenceResponse">
2074 <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
2075 </wsdl:message>
2076 <wsdl:message name="CloseSequence">
2077 <wsdl:part name="close" element="rm:CloseSequence"/>
2078 </wsdl:message>
2079 <wsdl:message name="CloseSequenceResponse">
2080 <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
2081 </wsdl:message>
2082 <wsdl:message name="TerminateSequence">
2083 <wsdl:part name="terminate" element="rm:TerminateSequence"/>
2084 </wsdl:message>
2085 <wsdl:message name="TerminateSequenceResponse">
2086 <wsdl:part name="terminateResponse"
2087 element="rm:TerminateSequenceResponse"/>
2088 </wsdl:message>
2089 <wsdl:message name="MakeConnection">
2090 <wsdl:part name="makeConnection" element="rm:MakeConnection"/>
2091 </wsdl:message>
2092 <wsdl:portType name="SequenceAbstractPortType">
2093 <wsdl:operation name="CreateSequence">
2094 <wsdl:input message="tns:CreateSequence" wsaw:Action="http://docs.oasis-
2095 open.org/ws-rx/wsrn/200608/CreateSequence"/>
2096 <wsdl:output message="tns:CreateSequenceResponse"
2097 wsaw:Action="http://docs.oasis-open.org/ws-
2098 rx/wsrn/200608/CreateSequenceResponse"/>
2099 </wsdl:operation>
2100 <wsdl:operation name="CloseSequence">
2101 <wsdl:input message="tns:CloseSequence" wsaw:Action="http://docs.oasis-
2102 open.org/ws-rx/wsrn/200608/CloseSequence"/>
2103 <wsdl:output message="tns:CloseSequenceResponse"
2104 wsaw:Action="http://docs.oasis-open.org/ws-
2105 rx/wsrn/200608/CloseSequenceResponse"/>
2106 </wsdl:operation>
2107 <wsdl:operation name="TerminateSequence">
2108 <wsdl:input message="tns:TerminateSequence"
2109 wsaw:Action="http://docs.oasis-open.org/ws-rx/wsrn/200608/TerminateSequence"/>
2110 <wsdl:output message="tns:TerminateSequenceResponse"
2111 wsaw:Action="http://docs.oasis-open.org/ws-
2112 rx/wsrn/200608/TerminateSequenceResponse"/>
2113 </wsdl:operation>
2114 <wsdl:operation name="MakeConnection">
2115 <wsdl:input message="tns:MakeConnection" wsaw:Action="http://docs.oasis-
2116 open.org/ws-rx/wsrn/200608/MakeConnection"/>
2117 </wsdl:operation>
2118 </wsdl:portType>
2119 </wsdl:definitions>

```

2072 <?xml version="1.0" encoding="UTF-8"?>

2073 <!--

2074 OASIS takes no position regarding the validity or scope of any intellectual-  
2075 property or other rights that might be claimed to pertain to the  
2076 implementation or use of the technology described in this document or the  
2077 extent to which any license under such rights might or might not be  
2078 available; neither does it represent that it has made any effort to identify  
2079 any such rights. Information on OASIS's procedures with respect to rights in  
2080 OASIS specifications can be found at the OASIS website. Copies of claims of  
2081 rights made available for publication and any assurances of licenses to be

2082 ~~made available, or the result of an attempt made to obtain a general license~~  
2083 ~~or permission for the use of such proprietary rights by implementors or users~~  
2084 ~~of this specification, can be obtained from the OASIS Executive Director.~~  
2085 ~~OASIS invites any interested party to bring to its attention any copyrights,~~  
2086 ~~patents or patent applications, or other proprietary rights which may cover~~  
2087 ~~technology that may be required to implement this specification. Please~~  
2088 ~~address the information to the OASIS Executive Director.~~  
2089 ~~Copyright © OASIS Open 2002-2006. All Rights Reserved.~~  
2090 ~~This document and translations of it may be copied and furnished to others,~~  
2091 ~~and derivative works that comment on or otherwise explain it or assist in its~~  
2092 ~~implementation may be prepared, copied, published and distributed, in whole~~  
2093 ~~or in part, without restriction of any kind, provided that the above~~  
2094 ~~copyright notice and this paragraph are included on all such copies and~~  
2095 ~~derivative works. However, this document itself does not be modified in any~~  
2096 ~~way, such as by removing the copyright notice or references to OASIS, except~~  
2097 ~~as needed for the purpose of developing OASIS specifications, in which case~~  
2098 ~~the procedures for copyrights defined in the OASIS Intellectual Property~~  
2099 ~~Rights document must be followed, or as required to translate it into~~  
2100 ~~languages other than English.~~  
2101 ~~The limited permissions granted above are perpetual and will not be revoked~~  
2102 ~~by OASIS or its successors or assigns.~~  
2103 ~~This document and the information contained herein is provided on an "AS~~  
2104 ~~IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING~~  
2105 ~~BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL~~  
2106 ~~NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR~~  
2107 ~~FITNESS FOR A PARTICULAR PURPOSE.~~  
2108 ~~→~~  
2109 ~~<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"~~  
2110 ~~xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"~~  
2111 ~~xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200602"~~  
2112 ~~targetNamespace="http://docs.oasis-open.org/ws-rx/wsm/200602"~~  
2113 ~~elementFormDefault="qualified" attributeFormDefault="unqualified">~~  
2114 ~~<xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"~~  
2115 ~~schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>~~  
2116 ~~<!-- Protocol Elements -->~~  
2117 ~~<xs:complexType name="SequenceType">~~  
2118 ~~<xs:sequence>~~  
2119 ~~<xs:element ref="wsm:Identifier"/>~~  
2120 ~~<xs:element name="MessageNumber" type="wsm:MessageNumberType"/>~~  
2121 ~~<xs:any namespace="##other" processContents="lax" minOccurs="0"~~  
2122 ~~maxOccurs="unbounded"/>~~  
2123 ~~</xs:sequence>~~  
2124 ~~<xs:anyAttribute namespace="##other" processContents="lax"/>~~  
2125 ~~</xs:complexType>~~  
2126 ~~<xs:element name="Sequence" type="wsm:SequenceType"/>~~  
2127 ~~<xs:element name="SequenceAcknowledgement">~~  
2128 ~~<xs:complexType>~~  
2129 ~~<xs:sequence>~~  
2130 ~~<xs:element ref="wsm:Identifier"/>~~  
2131 ~~<xs:choice>~~

```

2132 <del><xs:sequence>  

2133 <del><xs:choice>  

2134 <del><xs:element name="AcknowledgementRange" maxOccurs="unbounded">  

2135 <del><xs:complexType>  

2136 <del><xs:sequence/>  

2137 <del><xs:attribute name="Upper" type="xs:unsignedLong"  

2138 use="required"/>  

2139 <del><xs:attribute name="Lower" type="xs:unsignedLong"  

2140 use="required"/>  

2141 <del><xs:anyAttribute namespace="##other" processContents="lax"/>  

2142 <del></xs:complexType>  

2143 <del></xs:element>  

2144 <del><xs:element name="None" minOccurs="0">  

2145 <del><xs:complexType>  

2146 <del><xs:sequence/>  

2147 <del></xs:complexType>  

2148 <del></xs:element>  

2149 <del></xs:choice>  

2150 <del><xs:element name="Final" minOccurs="0">  

2151 <del><xs:complexType>  

2152 <del><xs:sequence/>  

2153 <del></xs:complexType>  

2154 <del></xs:element>  

2155 <del></xs:sequence>  

2156 <del><xs:element name="Nack" type="xs:unsignedLong"  

2157 maxOccurs="unbounded"/>  

2158 <del></xs:choice>  

2159 <del><xs:any namespace="##other" processContents="lax" minOccurs="0"  

2160 maxOccurs="unbounded"/>  

2161 <del></xs:sequence>  

2162 <del><xs:anyAttribute namespace="##other" processContents="lax"/>  

2163 <del></xs:complexType>  

2164 <del></xs:element>  

2165 <del><xs:complexType name="AckRequestedType">  

2166 <del><xs:sequence>  

2167 <del><xs:element ref="wsrm:Identifier"/>  

2168 <del><xs:any namespace="##other" processContents="lax" minOccurs="0"  

2169 maxOccurs="unbounded"/>  

2170 <del></xs:sequence>  

2171 <del><xs:anyAttribute namespace="##other" processContents="lax"/>  

2172 <del></xs:complexType>  

2173 <del><xs:element name="AckRequested" type="wsrm:AckRequestedType"/>  

2174 <del><xs:element name="Identifier">  

2175 <del><xs:complexType>  

2176 <del><xs:annotation>  

2177 <del><xs:documentation>  

2178 <del>This type is for elements whose [children] is an anyURI and can  

2179 have arbitrary attributes.  

2180 <del></xs:documentation>  

2181 <del></xs:annotation>

```

```

2182 <del><xs:simpleContent>
2183 <del><xs:extension base="xs:anyURI">
2184 <del><xs:anyAttribute namespace="##other" processContents="lax"/>
2185 <del></xs:extension>
2186 <del></xs:simpleContent>
2187 <del></xs:complexType>
2188 <del></xs:element>
2189 <del><xs:simpleType name="MessageNumberType">
2190 <del><xs:restriction base="xs:unsignedLong">
2191 <del><xs:minInclusive value="1"/>
2192 <del><xs:maxInclusive value="9223372036854775807"/>
2193 <del></xs:restriction>
2194 <del></xs:simpleType>
|
2195 <del><!-- Fault Container and Codes -->
2196 <del><xs:simpleType name="FaultCodes">
2197 <del><xs:restriction base="xs:QName">
2198 <del><xs:enumeration value="wsrm:SequenceTerminated"/>
2199 <del><xs:enumeration value="wsrm:UnknownSequence"/>
2200 <del><xs:enumeration value="wsrm:InvalidAcknowledgement"/>
2201 <del><xs:enumeration value="wsrm:MessageNumberRollover"/>
2202 <del><xs:enumeration value="wsrm:CreateSequenceRefused"/>
2203 <del><xs:enumeration value="wsrm:SequenceClosed"/>
2204 <del><xs:enumeration value="wsrm:WSRMRequired"/>
2205 <del></xs:restriction>
2206 <del></xs:simpleType>
2207 <del><xs:complexType name="SequenceFaultType">
2208 <del><xs:sequence>
2209 <del><xs:element name="FaultCode" type="wsrm:FaultCodes"/>
2210 <del><xs:any namespace="##any" processContents="lax" minOccurs="0"
2211 maxOccurs="unbounded"/>
2212 <del></xs:sequence>
2213 <del><xs:anyAttribute namespace="##any" processContents="lax"/>
2214 <del></xs:complexType>
2215 <del><xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
2216 <del><xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
2217 <del><xs:element name="CreateSequenceResponse"
2218 type="wsrm:CreateSequenceResponseType"/>
2219 <del><xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
2220 <del><xs:element name="CloseSequenceResponse"
2221 type="wsrm:CloseSequenceResponseType"/>
2222 <del><xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
2223 <del><xs:element name="TerminateSequenceResponse"
2224 type="wsrm:TerminateSequenceResponseType"/>
|
2225 <del><xs:complexType name="CreateSequenceType">
2226 <del><xs:sequence>
2227 <del><xs:element ref="wsrm:AcksTo"/>
2228 <del><xs:element ref="wsrm:Expires" minOccurs="0"/>
2229 <del><xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>

```

```

2230 -----<xs:any namespace="##other" processContents="lax" minOccurs="0"
2231 maxOccurs="unbounded">
2232 -----<xs:annotation>
2233 -----<xs:documentation>
2234 -----It is the authors intent that this extensibility be used to
2235 transfer a Security Token Reference as defined in WS Security.
2236 -----</xs:documentation>
2237 -----</xs:annotation>
2238 -----</xs:any>
2239 -----</xs:sequence>
2240 -----<xs:anyAttribute namespace="##other" processContents="lax"/>
2241 -----</xs:complexType>
2242 -----<xs:complexType name="CreateSequenceResponseType">
2243 -----<xs:sequence>
2244 -----<xs:element ref="wsrm:Identifier"/>
2245 -----<xs:element ref="wsrm:Expires" minOccurs="0"/>
2246 -----<xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>
2247 -----<xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
2248 -----<xs:any namespace="##other" processContents="lax" minOccurs="0"
2249 maxOccurs="unbounded"/>
2250 -----</xs:sequence>
2251 -----<xs:anyAttribute namespace="##other" processContents="lax"/>
2252 -----</xs:complexType>
2253 -----<xs:complexType name="CloseSequenceType">
2254 -----<xs:sequence>
2255 -----<xs:element ref="wsrm:Identifier"/>
2256 -----<xs:any namespace="##other" processContents="lax" minOccurs="0"
2257 maxOccurs="unbounded"/>
2258 -----</xs:sequence>
2259 -----<xs:anyAttribute namespace="##other" processContents="lax"/>
2260 -----</xs:complexType>
2261 -----<xs:complexType name="CloseSequenceResponseType">
2262 -----<xs:sequence>
2263 -----<xs:element ref="wsrm:Identifier"/>
2264 -----<xs:any namespace="##other" processContents="lax" minOccurs="0"
2265 maxOccurs="unbounded"/>
2266 -----</xs:sequence>
2267 -----<xs:anyAttribute namespace="##other" processContents="lax"/>
2268 -----</xs:complexType>
2269 -----<xs:complexType name="TerminateSequenceType">
2270 -----<xs:sequence>
2271 -----<xs:element ref="wsrm:Identifier"/>
2272 -----<xs:any namespace="##other" processContents="lax" minOccurs="0"
2273 maxOccurs="unbounded"/>
2274 -----</xs:sequence>
2275 -----<xs:anyAttribute namespace="##other" processContents="lax"/>
2276 -----</xs:complexType>
2277 -----<xs:complexType name="TerminateSequenceResponseType">
2278 -----<xs:sequence>
2279 -----<xs:element ref="wsrm:Identifier"/>

```

```

2280 <xs:any namespace="##other" processContents="lax"
2281 minOccurs="0" maxOccurs="unbounded"/>
2282 </xs:sequence>
2283 <xs:anyAttribute namespace="##other" processContents="lax"/>
2284 </xs:complexType>
2285 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
2286 <xs:complexType name="OfferType">
2287 <xs:sequence>
2288 <xs:element ref="wsrm:Identifier"/>
2289 <xs:element ref="wsrm:Expires" minOccurs="0"/>
2290 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2291 maxOccurs="unbounded"/>
2292 -
2293 </xs:sequence>
2294 <xs:anyAttribute namespace="##other" processContents="lax"/>
2295 </xs:complexType>
2296 <xs:complexType name="AcceptType">
2297 <xs:sequence>
2298 <xs:element ref="wsrm:AcksTo"/>
2299 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2300 maxOccurs="unbounded"/>
2301 </xs:sequence>
2302 <xs:anyAttribute namespace="##other" processContents="lax"/>
2303 </xs:complexType>
2304 <xs:element name="Expires">
2305 <xs:complexType>
2306 <xs:simpleContent>
2307 <xs:extension base="xs:duration">
2308 <xs:anyAttribute namespace="##other" processContents="lax"/>
2309 </xs:extension>
2310 </xs:simpleContent>
2311 </xs:complexType>
2312 </xs:element>
2313 <xs:element name="AcknowledgementInterval">
2314 <xs:complexType>
2315 <xs:sequence/>
2316 <xs:attribute name="Milliseconds" type="xs:unsignedLong"
2317 use="required"/>
2318 <xs:anyAttribute namespace="##other" processContents="lax"/>
2319 </xs:complexType>
2320 </xs:element>
2321 </xs:schema>

```

## 2322 Appendix C. Message Examples

### 2323 Appendix C.1 Create Sequence

#### 2324 Create Sequence

```
2325 <?xml version="1.0" encoding="UTF-8"?>
2326 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2327 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/2006082"
2328 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
2329   <S:Header>
2330     <wsa:MessageID>
2331       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
2332     </wsa:MessageID>
2333     <wsa:To>http://example.com/serviceB/123</wsa:To>
2334     <wsa:Action>http://docs.oasis-open.org/ws-
2335 rx/wsmr/2006082/CreateSequence</wsa:Action>
2336     <wsa:ReplyTo>
2337       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2338     </wsa:ReplyTo>
2339   </S:Header>
2340   <S:Body>
2341     <wsmr:CreateSequence>
2342       <wsmr:AcksTo>
2343         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2344       </wsmr:AcksTo>
2345     </wsmr:CreateSequence>
2346   </S:Body>
2347 </S:Envelope>
```

#### 2348 Create Sequence Response

```
2349 <?xml version="1.0" encoding="UTF-8"?>
2350 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2351 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200608"
2352 xmlns:wsa="http://www.w3.org/20052"
2353 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
2354   <S:Header>
2355     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
2356     <wsa:RelatesTo>
2357       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
2358     </wsa:RelatesTo>
2359     <wsa:Action>
2360       http://docs.oasis-open.org/ws-rx/wsmr/2006082/CreateSequenceResponse
2361     </wsa:Action>
2362   </S:Header>
2363   <S:Body>
2364     <wsmr:CreateSequenceResponse>
2365       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2366     </wsmr:CreateSequenceResponse>
2367   </S:Body>
2368 </S:Envelope>
```

### 2369 Appendix C.2 Initial Transmission

2370 The following example WS-ReliableMessaging headers illustrate the message exchange in the above  
2371 figure. The three messages have the following headers; the third message is identified as the last  
2372 message in the **S**sequence:



2373 **Message 1**

```
2374 <?xml version="1.0" encoding="UTF-8"?>
2375 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2376 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/2006082"
2377 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
2378   <S:Header>
2379     <wsa:MessageID>
2380       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
2381     </wsa:MessageID>
2382     <wsa:To>http://example.com/serviceB/123</wsa:To>
2383     <wsa:From>
2384       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2385     </wsa:From>
2386     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2387     <wsmr:Sequence>
2388       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2389       <wsmr:MessageNumber>1</wsmr:MessageNumber>
2390     </wsmr:Sequence>
2391   </S:Header>
2392   <S:Body>
2393     <!-- Some Application Data -->
2394   </S:Body>
2395 </S:Envelope>
```

2396 **Message 2**

```
2397 <?xml version="1.0" encoding="UTF-8"?>
2398 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2399 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/2006082"
2400 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
2401   <S:Header>
2402     <wsa:MessageID>
2403       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
2404     </wsa:MessageID>
2405     <wsa:To>http://example.com/serviceB/123</wsa:To>
2406     <wsa:From>
2407       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2408     </wsa:From>
2409     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2410     <wsmr:Sequence>
2411       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2412       <wsmr:MessageNumber>2</wsmr:MessageNumber>
2413     </wsmr:Sequence>
2414   </S:Header>
2415   <S:Body>
2416     <!-- Some Application Data -->
2417   </S:Body>
2418 </S:Envelope>
```

2419 **Message 3**

```
2420 <?xml version="1.0" encoding="UTF-8"?>
2421 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2422 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/2006082"
2423 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
2424   <S:Header>
2425     <wsa:MessageID>
2426       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
2427     </wsa:MessageID>
2428     <wsa:To>http://example.com/serviceB/123</wsa:To>
2429     <wsa:From>
2430       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
```

```

2431 </wsa:From>
2432 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2433 <wsrm:Sequence>
2434 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2435 <wsrm:MessageNumber>3</wsrm:MessageNumber>
2436 </wsrm:Sequence>
2437 <wsrm:AckRequested>
2438 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2439 </wsrm:AckRequested>
2440 </S:Header>
2441 <S:Body>
2442 <!-- Some Application Data -->
2443 </S:Body>
2444 </S:Envelope>

```

## 2445 Appendix C.3 First Acknowledgement

2446 Message number 2 has not been [accepted by the RM Destination due to some transmission error so it](#)  
2447 [responds with an Areceived by the RM Destination due to some transmission error so it responds with an](#)  
2448 [acknowledgement for messages 1 and 3:](#)

```

2449 <?xml version="1.0" encoding="UTF-8"?>
2450 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2451 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/2006082"
2452 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
2453 <S:Header>
2454 <wsa:MessageID>
2455 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
2456 </wsa:MessageID>
2457 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
2458 <wsa:From>
2459 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
2460 </wsa:From>
2461 <wsa:Action>
2462 http://docs.oasis-open.org/ws-rx/wsrn/2006082/SequenceAcknowledgement
2463 </wsa:Action>
2464 <wsrm:SequenceAcknowledgement>
2465 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2466 <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
2467 <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
2468 </wsrm:SequenceAcknowledgement>
2469 </S:Header>
2470 <S:Body/>
2471 </S:Envelope>

```

## 2472 Appendix C.4 Retransmission

2473 The RM Sourcediscovers that message number 2 was not [accepted so it resends the message and](#)  
2474 [requests an Areceived so it resends the message and requests an](#) acknowledgement:

```

2475 <?xml version="1.0" encoding="UTF-8"?>
2476 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2477 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/2006082"
2478 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
2479 <S:Header>
2480 <wsa:MessageID>
2481 http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
2482 </wsa:MessageID>
2483 <wsa:To>http://example.com/serviceB/123</wsa:To>
2484 <wsa:From>

```

```

2485     <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2486     </wsa:From>
2487     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2487     <wsrm:Sequence>
2487     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2487     <wsrm:MessageNumber>2</wsrm:MessageNumber>
2487     </wsrm:Sequence>
2487     <wsrm:AckRequested>
2487     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2487     </wsrm:AckRequested>
2487     </S:Header>
2487     <S:Body>
2487     <!-- Some Application Data -->
2487     </S:Body>
2487     </S:Envelope>

```

## 2487 Appendix C.5 Termination

2487 The RM Destination now responds with an [Acknowledgement for the complete Sacknowledgement for the](#)  
2488 [complete](#) sequence which can then be terminated:

```

2487 <?xml version="1.0" encoding="UTF-8"?>
2487 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2487 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/2006082"
2487 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
2487   <S:Header>
2487     <wsa:MessageID>
2487       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
2487     </wsa:MessageID>
2487     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
2487     <wsa:From>
2487       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
2487     </wsa:From>
2487     <wsa:Action>
2487       http://docs.oasis-open.org/ws-rx/wsrn/2006082/SequenceAcknowledgement
2487     </wsa:Action>
2487     <wsrm:SequenceAcknowledgement>
2487     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2487     <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
2487     </wsrm:SequenceAcknowledgement>
2487   </S:Header>
2487   <S:Body/>
2487 </S:Envelope>

```

### 2487 Terminate Sequence

```

2487 <?xml version="1.0" encoding="UTF-8"?>
2487 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2487 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/2006082"
2487 xmlns:wsa="http://www.w3.org/2005schemas.xmlsoap.org/ws/2004/08/addressing">
2487   <S:Header>
2487     <wsa:MessageID>
2487       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
2487     </wsa:MessageID>
2487     <wsa:To>http://example.com/serviceB/123</wsa:To>
2487     <wsa:Action>
2487       http://docs.oasis-open.org/ws-rx/wsrn/2006082/TerminateSequence
2487     </wsa:Action>
2487     <wsa:From>
2487       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2487     </wsa:From>
2487   </S:Header>

```

```

2488 <S:Body>
2488 <wsrm:TerminateSequence>
2488 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2488 </wsrm:TerminateSequence>
2488 </S:Body>
2488 </S:Envelope>

```

## 2488 Terminate Sequence Response

```

2488 <?xml version="1.0" encoding="UTF-8"?>
2488 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2488 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrmp/2006082"
2488 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2488 <S:Header>
2488 <wsa:MessageID>
2488 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
2488 </wsa:MessageID>
2488 <wsa:To>http://example.com/serviceA/789</wsa:To>
2488 <wsa:Action>
2488 http://docs.oasis-open.org/ws-rx/wsrmp/2006082/TerminateSequenceResponse
2488 </wsa:Action>
2488 <wsa:RelatesTo>
2488 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
2488 </wsa:RelatesTo>
2488 <wsa:From>
2488 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2488 </wsa:From>
2488 </S:Header>
2488 <S:Body>
2488 <wsrm:TerminateSequenceResponse>
2488 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2488 </wsrm:TerminateSequenceResponse>
2488 </S:Body>
2488 </S:Envelope>

```

## 2488 Appendix C.6 MakeConnection

2488 To illustrate how a MakeConnection message exchange can be used to deliver messages to an  
2489 Endpoint that is not addressable, consider the case of a pub/sub scenario in which the Endpoint to which  
2490 notifications are to be delivered (the "event consumer") is not addressable by the notification sending  
2491 Endpoint (the "event producer"). In this scenario the event consumer must initiate the connections in order  
2492 for the notifications to be delivered. One possible set of message exchanges (using HTTP) that  
2493 demonstrate how this can be achieved using MakeConnection is shown below.

2488 **Step 1** – During a "subscribe" operation, the event consumer's EPR specifies the RM anonymous URI  
2489 and the RM Policy Assertion to indicate whether or not RM is required:

```

2488 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2488 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrmp/200608"
2488 xmlns:wsrmp="http://docs.oasis-open.org/ws-rx/wsrmp/200608"
2488 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2488 <S:Header>
2488 <wsa:To> http://example.org/subscriptionService </wsa:To>
2488 <wsa:MessageID> http://client456.org/id-a6d8-a7c2eb546813</wsa:MessageID>
2488 <wsa:ReplyTo>
2488 <wsa:To> http://client456.org/response </wsa:To>
2488 </wsa:ReplyTo>
2488 </S:Header>
2488 <S:Body>
2488 <sub:Subscribe xmlns:sub="http://example.org/subscriptionService">
2488 <!-- subscription service specific data -->

```

```

2489     <targetEPR>
2489         <wsa:Address>http://docs.oasis-open.org/ws-
2490 rx/wsrmp/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:Address>
2489     <wsa:Metadata>
2489         <wsp:Policy wsu:Id="MyPolicy">
2489             <wsrmp:RMAssertion/>
2489         </wsp:Policy>
2489     </wsa:Metadata>
2489 </targetEPR>
2489 </sub:Subscribe>
2489 </S:Body>
2489 </S:Envelope>

```

2489 In this example the `subscribe` and `targetEPR` elements are simply examples of what a subscription request message might contain. Note: the `wsa:Address` element contains the RM anonymous URI indicating that the notification producer needs to queue the messages until they are requested using the `MakeConnection` message exchange. The EPR also contains the RM Policy Assertion indicating the RM must be used when notifications related to this subscription are sent.

2489 **Step 2** – Once the subscription is established, the event consumer checks for a pending message:

```

2489 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2489 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrmp/200608"
2489 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2489     <S:Header>
2489         <wsa:Action>http://docs.oasis-open.org/ws-
2490 rx/wsrmp/200608/MakeConnection</wsa:Action>
2489         <wsa:To> http://example.org/subscriptionService </wsa:To>
2489     </S:Header>
2489     <S:Body>
2489         <wsrm:MakeConnection>
2489             <wsrm:Address>http://docs.oasis-open.org/ws-
2490 rx/wsrmp/200608/anonymous?id=550e8400-e29b-11d4-a716-
2491 446655440000</wsrm:Address>
2489         </wsrm:MakeConnection>
2489     </S:Body>
2489 </S:Envelope>

```

2489 **Step 3** – If there are messages waiting to be delivered then a message will be returned back to the event consumer. However, because WS-RM is being used to deliver the messages, the first message returned is a `CreateSequence`:

```

2489 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2489 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrmp/200608"
2489 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2489     <S:Header>
2489         <wsa:Action>http://docs.oasis-open.org/ws-
2490 rx/wsrmp/200608/CreateSequence</wsa:Action>
2489         <wsa:To>http://docs.oasis-open.org/ws-
2490 rx/wsrmp/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:To>
2489         <wsa:ReplyTo> http://example.org/subscriptionService </wsa:ReplyTo>
2489         <wsa:MessageID> http://example.org/id-123-456 </wsa:MessageID>
2489     </S:Header>
2489     <S:Body>
2489         <wsrm:CreateSequence>
2489             <wsrm:AcksTo>
2489                 <wsa:Address> http://example.org/subscriptionService </wsa:Address>
2489             </wsrm:AcksTo>
2490         </wsrm:CreateSequence>

```

```
2491 </S:Body>
2491 </S:Envelope>
```

2492 Notice from the perspective of how the RM Source on the event producer interacts with the RM  
2493 Destination of those messages, nothing new is introduced by the use of the `MakeConnection`, the use  
2494 of RM protocol is the same as the case where the event consumer is addressable.

2495 **Step 4** – The event consumer will respond with a `CreateSequenceResponse` message per normal WS-  
2496 Addressing rules:

```
2497 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2498 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200608"
2499 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2500 <S:Header>
2501 <wsa:Action>http://docs.oasis-open.org/ws-
2502 rx/wsmr/200608/CreateSequenceResponse</wsa:Action>
2503 <wsa:To> http://example.org/subscriptionService </wsa:To>
2504 <wsa:RelatesTo> http://example.org/id-123-456 </wsa:RelatesTo>
2505 </S:Header>
2506 <S:Body>
2507 <wsmr:CreateSequenceResponse>
2508 <wsmr:Identifier> http://example.org/rmid-456 </wsmr:Identifier>
2509 </wsmr:CreateSequenceResponse>
2510 </S:Body>
2511 </S:Envelope>
```

2512 Note, this message is carried on an HTTP request directed to the `wsa:ReplyTo` EPR, and the HTTP  
2513 response will be an HTTP 202.

2514 **Step 5** – The event consumer checks for another message pending:

```
2515 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2516 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200608"
2517 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2518 <S:Header>
2519 <wsa:Action>http://docs.oasis-open.org/ws-
2520 rx/wsmr/200608/MakeConnection</wsa:Action>
2521 <wsa:To> http://example.org/subscriptionService </wsa:To>
2522 </S:Header>
2523 <S:Body>
2524 <wsmr:MakeConnection>
2525 <wsmr:Address>http://docs.oasis-open.org/ws-
2526 rx/wsmr/200608/anonymous?id=550e8400-e29b-11d4-a716-
2527 446655440000</wsmr:Address>
2528 </wsmr:MakeConnection>
2529 </S:Body>
2530 </S:Envelope>
```

2531 Notice this is the same message as the one sent in step 2.

2532 **Step 6** – If there is a message pending for this destination then it is returned on the HTTP response:

```
2533 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2534 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200608"
2535 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2536 <S:Header>
2537 <wsa:Action> http://example.org/eventType1 </wsa:Action>
```

```

2538 <wsa:To>http://docs.oasis-open.org/ws-
2539 rx/wsrn/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:To>
2540 <wsrm:Sequence>
2541 <wsrm:Identifier> http://example.org/rmid-456 </wsrm:Identifier>
2542 </wsrm:Sequence>
2543 <wsrm:MessagePending pending="true"/>
2544 </S:Header>
2545 <S:Body>
2546 <!-- event specific data -->
2547 </S:Body>
2548 </S:Envelope>

```

2549 As noted in step 3, the use of the RM protocol does not change when using `MakeConnection`. The  
2550 format of the messages, the order of the messages sent and the timing of when to send it remains the  
2551 same.

2552 **Step 7** – At some later interval, or immediately due to the `MessagePending` header's "pending"  
2553 attribute being set to "true", the event consumer will poll again:

```

2554 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2555 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200608"
2556 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2557 <S:Header>
2558 <wsa:Action>http://docs.oasis-open.org/ws-
2559 rx/wsrn/200608/MakeConnection</wsa:Action>
2560 <wsa:To> http://example.org/subscriptionService </wsa:To>
2561 </S:Header>
2562 <S:Body>
2563 <wsrm:MakeConnection>
2564 <wsrm:Address>http://docs.oasis-open.org/ws-
2565 rx/wsrn/200608/anonymous?id=550e8400-e29b-11d4-a716-
2566 446655440000</wsrm:Address>
2567 </wsrm:MakeConnection>
2568 </S:Body>
2569 </S:Envelope>

```

2570 Notice this is the same message as the one sent in steps 2 and 5. As in steps 3 and 6, the response to  
2571 the `MakeConnection` can be any message destined to the specified Endpoint. This allows the event  
2572 producer to send not only application messages but RM protocol messages (e.g. `CloseSequence`,  
2573 `TerminateSequence` or even additional `CreateSequences`) as needed.

2574 **Step 8** – If at any point in time there are no messages pending, in response to a `MakeConnection` the  
2575 event producer returns an HTTP 202 back to the event consumer. The process then repeats (back to step  
2576 7) until the subscription ends.

## 2577 **Appendix D. WSDL**

2578 The non-normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

2579 <http://docs.oasis-open.org/ws-rx/wsrn/200602/wsd/wsrn-1.1-wsd-200602.wsd>

2580 The following non-normative copy is provided for reference.

2581 ~~<?xml version="1.0" encoding="utf-8"?>~~

2582 ~~<!--~~

2583 ~~OASIS takes no position regarding the validity or scope of any intellectual-~~

2584 ~~property or other rights that might be claimed to pertain to the-~~

2585 ~~implementation or use of the technology described in this document or the-~~

2586 ~~extent to which any license under such rights might or might not be-~~

2587 ~~available; neither does it represent that it has made any effort to identify-~~

2588 ~~any such rights. Information on OASIS's procedures with respect to rights in-~~

2589 ~~OASIS specifications can be found at the OASIS website. Copies of claims of-~~

2590 ~~rights made available for publication and any assurances of licenses to be-~~

2591 ~~made available, or the result of an attempt made to obtain a general license-~~

2592 ~~or permission for the use of such proprietary rights by implementors or users-~~

2593 ~~of this specification, can be obtained from the OASIS Executive Director.~~

2594 ~~OASIS invites any interested party to bring to its attention any copyrights,~~

2595 ~~patents or patent applications, or other proprietary rights which may cover-~~

2596 ~~technology that may be required to implement this specification. Please-~~

2597 ~~address the information to the OASIS Executive Director.~~

2598 ~~Copyright © OASIS Open 2002-2006. All Rights Reserved.~~

2599 ~~This document and translations of it may be copied and furnished to others,~~

2600 ~~and derivative works that comment on or otherwise explain it or assist in its-~~

2601 ~~implementation may be prepared, copied, published and distributed, in whole-~~

2602 ~~or in part, without restriction of any kind, provided that the above-~~

2603 ~~copyright notice and this paragraph are included on all such copies and-~~

2604 ~~derivative works. However, this document itself does not be modified in any-~~

2605 ~~way, such as by removing the copyright notice or references to OASIS, except-~~

2606 ~~as needed for the purpose of developing OASIS specifications, in which case-~~

2607 ~~the procedures for copyrights defined in the OASIS Intellectual Property-~~

2608 ~~Rights document must be followed, or as required to translate it into-~~

2609 ~~languages other than English.~~

2610 ~~The limited permissions granted above are perpetual and will not be revoked-~~

2611 ~~by OASIS or its successors or assigns.~~

2612 ~~This document and the information contained herein is provided on an "AS-~~

2613 ~~IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING-~~

2614 ~~BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL-~~

2615 ~~NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR-~~

2616 ~~FITNESS FOR A PARTICULAR PURPOSE.~~

2617 ~~—>~~

2618 ~~<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"~~

2619 ~~xmlns:xs="http://www.w3.org/2001/XMLSchema"~~

2620 ~~xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"~~

2621 ~~xmlns:rm="http://docs.oasis-open.org/ws-rx/wsrn/200602"~~



```

2622 xmlns:tns="http://docs.oasis-open.org/ws-rx/wsrn/200602/wsdl"
2623 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200602/wsdl">
2624 <wsdl:types>
2625 <xs:schema
2626 <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsrn/200602"
2627 schemaLocation="http://docs.oasis-open.org/ws-rx/wsrn/200602/wsrn-1.1-schema-
2628 200602.xsd"/>
2629 </xs:schema>
2630 </wsdl:types>
2631 <wsdl:message name="CreateSequence">
2632 <wsdl:part name="create" element="rm:CreateSequence"/>
2633 </wsdl:message>
2634 <wsdl:message name="CreateSequenceResponse">
2635 <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
2636 </wsdl:message>
2637 <wsdl:message name="CloseSequence">
2638 <wsdl:part name="close" element="rm:CloseSequence"/>
2639 </wsdl:message>
2640 <wsdl:message name="CloseSequenceResponse">
2641 <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
2642 </wsdl:message>
2643 <wsdl:message name="TerminateSequence">
2644 <wsdl:part name="terminate" element="rm:TerminateSequence"/>
2645 </wsdl:message>
2646 <wsdl:message name="TerminateSequenceResponse">
2647 <wsdl:part name="terminateResponse"
2648 element="rm:TerminateSequenceResponse"/>
2649 </wsdl:message>
2650 <wsdl:portType name="SequenceAbstractPortType">
2651 <wsdl:operation name="CreateSequence">
2652 <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
2653 open.org/ws-rx/wsrn/200602/CreateSequence"/>
2654 <wsdl:output message="tns:CreateSequenceResponse"
2655 wsa:Action="http://docs.oasis-open.org/ws-
2656 rx/wsrn/200602/CreateSequenceResponse"/>
2657 </wsdl:operation>
2658 <wsdl:operation name="CloseSequence">
2659 <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
2660 open.org/ws-rx/wsrn/200602/CloseSequence"/>
2661 <wsdl:output message="tns:CloseSequenceResponse"
2662 wsa:Action="http://docs.oasis-open.org/ws-
2663 rx/wsrn/200602/CloseSequenceResponse"/>
2664 </wsdl:operation>
2665 <wsdl:operation name="TerminateSequence">
2666 <wsdl:input message="tns:TerminateSequence"
2667 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200602/TerminateSequence"/>
2668 <wsdl:output message="tns:TerminateSequenceResponse"
2669 wsa:Action="http://docs.oasis-open.org/ws-
2670 rx/wsrn/200602/TerminateSequenceResponse"/>
2671 </wsdl:operation>

```

2672 </wsdl:portType>  
2673 </wsdl:definitions>

## 2674 Appendix E. State Tables

2674 This appendix specifies the non-normative state transition tables for RM Source and RM Destination.

2674 The state tables describe the lifetime of a sequence in both the RM Source and the RM Destination

2674 Legend:

2674 The first column of these tables contains the motivating event and has the following format:

<b>Event</b>
<i>Event name</i> [source] {ref}

2674 Where:

2674 ● Event Name: indicates the name of the event. Event Names surrounded by "<>" are optional as  
2675 described by the specification.

2674 ● [source]: indicates the source of the event; one of:

2674 ● [msg] a Received message

2674 ● [int]: an internal event such as the firing of a timer

2674 ● [app]: the application

2674 ● [unspec]: the source is unspecified

2674 Each event / state combination cell in the tables in this appendix has the following format:

<b>State Name</b>
<i>Action to take</i> [next state] {ref}

2674 Where:

2674 ● action to take: indicates that the state machine performs the following action. Actions surrounded  
2675 by "<>" are optional as described by the specification. "Xmit" is used as a short form for the word  
2676 "Transmit"

2674 ● [next state]: indicates the state to which the state machine will advance upon the performance of  
2675 the action. For ease of reading the next state "same" indicates that the state does not change.

2674 ● {ref} is a reference to the document section describing the behavior in this cell

2674 "N/A" in a cell indicates a state / event combination self-inconsistent with the state machine; should these  
2675 conditions occur, it would indicate an implementation error. A blank cell indicates that the behavior is not  
2676 described in this specification and does not indicate normal protocol operation. Implementations MAY  
2677 generate a Sequence Terminated fault (see section 4.2) in these circumstances. Robust implementations  
2678 MUST be able to operate in a stable manner despite the occurrence of unspecified event / state  
2679 combinations.

2680 Table 1 RM Source Sequence State Transition Table

Events	Sequence States					
	None	Creating	Created	Closing	Closed	Terminating
<b>Create Sequence</b> [unspec] {3.1}	Xmit Create Sequence [Creating] {3.1}	N/A	N/A	N/A	N/A	N/A
<b>Create Sequence Response</b> [msg] {3.1}	-	Process Create Sequence Response [Created] {3.1}	-	-	-	-
<b>Create Sequence Refused Fault</b> [msg] {3.1}	-	No action [None] {4.6}	-	-	-	-
<b>Send message</b> [app] {2.1}	N/A	N/A	Xmit message [Same] {2}	No action [Same] {2}	N/A	N/A
<b>Retransmit of un-ack'd message</b> [int]	N/A	N/A	Xmit message [Same] {2.4}	Xmit message [Same] {2.4}	N/A	N/A
<b>SeqAck (non-final)</b> [msg] {3.6}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Process Ack ranges [Same] {3.6}	Process Ack ranges [Same] {3.6}	Process Ack ranges [Same] {3.6}	Process Ack ranges [Same] {3.6}
<b>Nack</b> [msg] {3.6}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	<Xmit message(s)> [Same] {3.6}	<Xmit message(s)> [Same] {3.6}	No action [Same]	No action [Same]
<b>Message Number Rollover Fault</b> [msg]	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	No action [Rollover]	No action [Same]	No action [Same]	No action [Same]
<b>&lt;Close Sequence&gt;</b> [int] {3.2}	N/A	-	Xmit Close Sequence [Closing] {3.2}	N/A	N/A	N/A
<b>Close Sequence Response</b> [msg] {3.2}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	-	No action [Closed] {3.2}	No action [Same] {3.2}	No action [Same] {3.2}
<b>SeqAck (final)</b> [msg] {3.6}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Process Ack ranges [Closed] {3.6}	Process Ack ranges [Closed] {3.6}	Process Ack ranges [Same]	Process Ack ranges [Same]
<b>Sequence Closed Fault</b> [msg]	Generate Unknown Sequence Fault	Generate Unknown Sequence Fault	No action [Closed] {4.7}	No action [Closed] {4.7}	No action [Same]	No action [Same]

Events	Sequence States					
	None	Creating	Created	Closing	Closed	Terminating
{4.7}	[Same] {4.3}	[Same] {4.3}				
<u>Unknown Sequence Fault</u> [msg] {4.3}	-	-	<u>Terminate Sequence</u> [None] {4.3}	<u>Terminate Sequence</u> [None] {4.3}	<u>Terminate Sequence</u> [None] {4.3}	<u>Terminate Sequence</u> [None] {4.3}
<u>Sequence Terminated Fault</u> [msg] {4.2}	N/A	-	<u>Terminate Sequence</u> [None] {4.2}	<u>Terminate Sequence</u> [None] {4.2}	<u>Terminate Sequence</u> [None] {4.2}	<u>Terminate Sequence</u> [None] {4.2}
<u>Terminate Sequence</u> [int]	N/A	No action [None] {unspec}	<u>Xmit Terminate Sequence</u> [Terminating]	<u>Xmit Terminate Sequence</u> [Terminating]	<u>Xmit Terminate Sequence</u> [Terminating]	N/A
<u>Terminate Sequence Response</u> [msg]	<u>Generate Unknown Sequence Fault</u> [Same] {4.3}	<u>Generate Unknown Sequence Fault</u> [Same] {4.3}	-	-	-	<u>Terminate Sequence</u> [None] {3.3}
<u>Expires exceeded</u> [int]	N/A	<u>Terminate Sequence</u> [None] {3.4}	<u>Terminate Sequence</u> [None] {3.4}	<u>Terminate Sequence</u> [None] {3.4}	<u>Terminate Sequence</u> [None] {3.4}	<u>Terminate Sequence</u> [None] {3.4}
<u>Invalid Acknowledgment</u> [msg] {4.4}	<u>Generate Unknown Sequence Fault</u> [Same] {4.3}	<u>Generate Unknown Sequence Fault</u> [Same] {4.3}	<u>Generate Invalid Acknowledgment Fault</u> [Same] {4.4}	<u>Generate Invalid Acknowledgment Fault</u> [Same] {4.4}	<u>Generate Invalid Acknowledgment Fault</u> [Same] {4.4}	<u>Generate Invalid Acknowledgment Fault</u> [Same] {4.4}

2680 Each cell in the tables in this appendix uses the following convention:

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

2680 Table 2-RM Source State Transition Table

Events	States							
	None	Connecting	Connected	Rollover	Closing	Closed	Terminating	Terminated
<u>Create Sequence</u>	<i>Transmit Create Sequence</i> Connecting	N/A	N/A	N/A	N/A	N/A	N/A	N/A
<u>Create Sequence Response</u>	N/A	Connected	N/A	N/A	N/A	N/A	N/A	N/A
<u>Create</u>	N/A		N/A	N/A	N/A	N/A	N/A	N/A

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

Events	States							
	None	Connecting	Connected	Rollover	Closing	Closed	Terminating	Terminated
Unknown-Sequence-Fault	N/A	N/A	Terminated?	Terminated?	Terminated?	Terminated?	Terminated?	Ignored
Sequence-Terminated-Fault	N/A	Terminated?	Terminated?	Terminated?	Terminated?	Terminated?	Terminated?	Ignored
Terminate-sequence	N/A	N/A	Transmit-Terminate-Sequence Terminating	Transmit-Terminate-Sequence Terminating	Transmit-Terminate-Sequence Terminating	Transmit-Terminate-Sequence Terminating	Transmit-Terminate-Sequence Terminating	N/A
Terminate-Sequence-Response	N/A	N/A	N/A	N/A	N/A	N/A	Terminated	Terminated
Elapse-Expires-duration	N/A	N/A	Terminated	Terminated	Terminated	Terminated	Terminated?	N/A

2680 Table 2 RM Destination Sequence State Transition Table

Events	Sequence States		
	None	Created	Closed
CreateSequence (successful) [msg/int] {3.1}	Xmit Create Sequence Response [Created] {3.1}	N/A	N/A
CreateSequence (unsuccessful) [msg/int] {3.1}	Generate Create Sequence Refused Fault [None] {3.1}	N/A	N/A
Message (with message number within range) [msg]	Generate Unknown Sequence Fault [Same] {4.3}	Accept Message: <Xmit SeqAck> [Same]	Generate Sequence Closed Fault (with SeqAck+Final) [Same] {3.2}
Message (with message number outside of range) [msg]	Generate Unknown Sequence Fault [Same] {4.3}	Xmit Message Number Rollover Fault [Same] {3.4}{4.5}	Generate Sequence Closed Fault (with SeqAck+Final) [Same] {3.2}
<AckRequested> [msg] {3.5}	Generate Unknown Seq Fault [Same] {4.3}	Xmit SeqAck [Same] {3.5}	Xmit SeqAck+Final [Same] {3.6}
CloseSequence [msg] {3.2}	Generate Unknown Sequence Fault [Same] {4.3}	Xmit CloseSequence Response with SeqAck+Final [Closed] {3.2}	Generate Sequence Closed Fault [Same] {4.7}
<CloseSequence autonomously> [int]	N/A	No Action [Closed]	N/A
TerminateSequence [msg] {3.3}	Generate Unknown Sequence Fault [Same]	Xmit Terminate Sequence Response [None]	Xmit Terminate Sequence Response [None]

Events	Sequence States		
	None	Created	Closed
	{4.3}	{3.3}	{3.3}
<b>UnknownSequence Fault</b> [msg] {4.3}	-	Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}
<b>SequenceTerminated Fault</b> [msg] {4.2}	-	Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.2}
<b>Invalid Acknowledgement Fault</b> [msg] {4.4}	N/A	-	-
<b>Expires exceeded</b> [int]	N/A	Terminate Sequence [None] {3.4}	Terminate Sequence [None] {3.4}
<b>&lt;Seq Acknowledgement autonomously&gt;</b> [int] {3.6}	N/A	Xmit SeqAck [Same] {3.6}	Xmit SeqAck+Final [Same] {3.6}
<b>Non WSRM message when WSRM required</b> [msg] {4.8}	Generate WSRMRequired Fault [Same] {4.8}	Generate WSRMRequired Fault [Same] {4.8}	Generate WSRMRequired Fault [Same] {4.8}

2680 In Table 2 above, the rows consists of events that occur at the RM Source throughout the lifetime of an  
2681 RM Sequence and the columns consists of various RM Source states. Each cell in the table above lists  
2682 the action that the RM Source takes on occurrence of a particular event and the next state that it  
2683 transitions.

2680 The following two tables apply only if the `MakeConnection` mechanism is utilized.

2680 Table 3 Sending Endpoint Message Transfer Engine

Event	None	Queued n=1	Queued, n>1
<b>Message destined to anon Endpoint when channel unavailable</b> [int] {3.7}	Queue message [Queued n=1]	Queue message [Queued n>1]	Queue message [Queued n>1]
<b>MakeConnection</b> [msg] {3.7}	-	Send message [none]	Xmit message with MessagePending [if n=2 then (Queued n=1) else (Queued n>1)]

2680 Table 4 Receiving Endpoint Message Transfer Engine 3-RM-Destination State Transition Table

Event	None	Polling
<b>Expectation of unreceived message</b>	No Action [Polling]	No Action [Same]



[int_unspecified]							
Polling trigger [int_unspecified]	-					Xmit MakeConnection [Polling] (3.7)	
<b>Events</b>	<b>States</b>						
	<b>None</b>	<b>Connecting</b>	<b>Connected</b>	<b>Rollover</b>	<b>Rollover-Closed</b>	<b>Closed</b>	<b>Terminated</b>
<b>Creation-request-not-satisfied</b>	N/A	Send-Create-Sequence-Refused-Fault Terminated	N/A	N/A	N/A	N/A	
<b>Unrecoverable-error-on-creation</b>	N/A	Send-Sequence-Terminated-Fault? Terminated	N/A	N/A	N/A	N/A	
<b>New-message</b>	N/A	N/A	Send-SequenceAck Connection	Send-Message-Number-Rollover-Fault Rollover	Send-Message-Number-Rollover-or-Sequence-Closed-Fault?(with-SeqAck+Final) Rollover-Closed	Send-Sequence-Closed-Fault-(with-SeqAck+Final) Closed	Send-Unknown-Seq-Fault? Terminated
<b>Retransmitted-message</b>	N/A	N/A	Send-SequenceAck Connected	Send-SequenceAck Rollover	Send-SeqAck+Final Rollover-Closed	Send-SeqAck+Final Closed	Send-Unknown-Seq-Fault Terminated
<b>Ack-requested</b>	N/A	N/A	Send-SequenceAck Connected	Send-SequenceAck Rollover	Send-SeqAck+Final Rollover-Closed	Send-SeqAck+Final Closed	Send-Unknown-Seq-Fault Terminated
<b>Reach-max-message-number</b>	N/A	N/A	Rollover	Rollover	Rollover-Closed	N/A	N/A
<b>Message-Number-Rollover-Fault</b>	N/A	N/A	Rollover	Rollover	Rollover-Closed	Closed?	Send-Unknown-Sequence-Fault Terminated
<b>Close-sequence</b>	N/A	N/A	Send-CloseSequenceResponse-with-SequenceAck(Final) Close	Send-CloseSequenceResponse-with-SequenceAck-Final Rollover-Closed	Send-Close-Sequence-Response-with-SeqAck+Final Rollover-Closed	Send-Close-Sequence-Response-with-SeqAck+Final Closed	Send-Unknown-Sequence-Fault Terminated

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

2680 In Table 3 above, the rows consists of events that occur at the RM Destination throughout the lifetime of  
2681 an RM Sequence and the columns consists of various RM Destination states. Each cell in the table above  
2682 lists the action that the RM Destination takes on occurrence of a particular event and the next state that it  
2683 transitions.

## 2680 Appendix F. Acknowledgments

2680 This document is based on initial contribution to OASIS WS-RX Technical Committee by the following  
2681 authors:

2680 [Ruslan Bilorusets\(BEA\)](#), [Don Box\(Microsoft\)](#), [Luis Felipe Cabrera\(Microsoft\)](#), [Doug Davis\(IBM\)](#),  
2681 [Donald Ferguson\(IBM\)](#), [Christopher Ferris\(Editor\)\(BM\)](#), [Tom Freund\(IBM\)](#), [Mary Ann Hondo\(IBM\)](#),  
2682 [John Ibbotson\(IBM\)](#), [Lei Jin\(BEA\)](#), [Chris Kaler\(Microsoft\)](#), [David Langworthy-Editor\(Microsoft\)](#),  
2683 [Amelia Lewis\(TIBCO Software\)](#), [Rodney Limprecht\(Microsoft\)](#), [Steve Lucco\(Microsoft\)](#), [Don](#)  
2684 [Mullen\(TIBCO Software\)](#), [Anthony Nadalin\(IBM\)](#), [Mark Nottingham\(BEA\)](#), [David Orchard\(BEA\)](#),  
2685 [Jamie Roots\(IBM\)](#), [Shivajee Samdarshi\(TIBCO Software\)](#), [John Shewchuk\(Microsoft\)](#), [Tony](#)  
2686 [Storey\(IBM\)](#), ~~BEA, Don Box, Microsoft, Luis Felipe Cabrera, Microsoft, Doug Davis, IBM, Donald~~  
2687 ~~Ferguson, IBM, Christopher Ferris, IBM (Editor), Tom Freund, IBM, Mary Ann Hondo, IBM, John~~  
2688 ~~Ibbotson, IBM, Lei Jin, BEA, Chris Kaler, Microsoft, David Langworthy, Microsoft (Editor), Amelia~~  
2689 ~~Lewis, TIBCO Software, Rodney Limprecht, Microsoft, Steve Lucco, Microsoft, Don Mullen,~~  
2690 ~~TIBCO Software, Anthony Nadalin, IBM, Mark Nottingham, BEA, David Orchard, BEA, Jamie~~  
2691 ~~Roots, IBM, Shivajee Samdarshi, TIBCO Software, John Shewchuk, Microsoft, Tony Storey, IBM.~~

2680 The following individuals have provided invaluable input into the initial contribution:

2680 [Keith Ballinger\(Microsoft\)](#), [Stefan Batres\(Microsoft\)](#), [Rebecca Bergersen\(Iona\)](#), [Allen](#)  
2681 [Brown\(Microsoft\)](#), [Michael Conner\(IBM\)](#), [George Copeland\(Microsoft\)](#), [Francisco Curbera\(IBM\)](#),  
2682 [Paul Fremantle\(IBM\)](#), [Steve Graham\(IBM\)](#), [Pat Helland\(Microsoft\)](#), [Rick Hill\(Microsoft\)](#), [Scott](#)  
2683 [Hinkelman\(IBM\)](#), [Tim Holloway\(IBM\)](#), [Efim Hudis\(Microsoft\)](#), [David Ingham\(Microsoft\)](#), [Gopal](#)  
2684 [Kakivaya\(Microsoft\)](#), [Johannes Klein\(Microsoft\)](#), [Frank Leymann\(IBM\)](#), [Martin Nally\(IBM\)](#), [Peter](#)  
2685 [Niblett\(IBM\)](#), [Jeffrey Schlimmer\(Microsoft\)](#), [James Snell\(IBM\)](#), [Keith Stobie\(Microsoft\)](#), [Satish](#)  
2686 [Thatte\(Microsoft\)](#), [Stephen Todd\(IBM\)](#), [Sanjiva Weerawarana\(IBM\)](#), [Roger Wolter\(Microsoft\)](#),  
2687 ~~Microsoft, Stefan Batres, Microsoft, Allen Brown, Microsoft, Michael Conner, IBM, George~~  
2688 ~~Gopeland, Microsoft, Francisco Curbera, IBM, Paul Fremantle, IBM, Steve Graham, IBM, Pat~~  
2689 ~~Helland, Microsoft, Rick Hill, Microsoft, Scott Hinkelman, IBM, Tim Holloway, IBM, Efim Hudis,~~  
2690 ~~Microsoft, Gopal Kakivaya, Microsoft, Johannes Klein, Microsoft, Frank Leymann, IBM, Martin~~  
2691 ~~Nally, IBM, Peter Niblett, IBM, Jeffrey Schlimmer, Microsoft, James Snell, IBM, Keith Stobie,~~  
2692 ~~Microsoft, Satish Thatte, Microsoft, Stephen Todd, IBM, Sanjiva Weerawarana, IBM, Roger~~  
2693 ~~Wolter, Microsoft.~~

2680 The following individuals were members of the committee during the development of this specification:

2680 [Abbie Barbir\(Nortel\)](#), [Charlton Barreto\(Adobe\)](#), [Stefan Batres\(Microsoft\)](#), [Hamid Ben](#)  
2681 [Malek\(Fujitsu\)](#), [Andreas Bjarlestam\(Ericsson\)](#), [Toufic Boubez\(Layer 7\)](#), [Doug Bunting\(Sun\)](#), [Lloyd](#)  
2682 [Burch\(Novell\)](#), [Steve Carter\(Novell\)](#), [Martin Chapman\(Oracle\)](#), [Dave Chappell\(Sonic\)](#), [Paul](#)  
2683 [Cotton\(Microsoft\)](#), [Glen Daniels\(Sonic\)](#), [Doug Davis\(IBM\)](#), [Blake Dournaee\(Intel\)](#), [Jacques](#)  
2684 [Durand\(Fujitsu\)](#), [Colleen Evans\(Microsoft\)](#), [Christopher Ferris\(IBM\)](#), [Paul Fremantle\(WSO2\)](#),  
2685 [Robert Freund\(Hitachi\)](#), [Peter Furniss\(Erebor\)](#), [Marc Goodner\(Microsoft\)](#), [Alastair](#)  
2686 [Green\(Choreology\)](#), [Mike Grogan\(Sun\)](#), [Ondrej Hrebicek\(Microsoft\)](#), [Kazunori Iwasa\(Fujitsu\)](#),  
2687 [Chamikara Jayalath\(WSO2\)](#), [Lei Jin\(BEA\)](#), [Ian Jones\(BTplc\)](#), [Anish Karmarkar\(Oracle\)](#), [Paul](#)  
2688 [Knight\(Nortel\)](#), [Dan Leshchiner\(Tibco\)](#), [Mark Little\(JBoss\)](#), [Lily Liu\(webMethods\)](#), [Matt](#)  
2689 [Lovett\(IBM\)](#), [Ashok Malhotra\(Oracle\)](#), [Jonathan Marsh\(Microsoft\)](#), [Daniel Millwood\(IBM\)](#), [Jeff](#)  
2690 [Mischkinsky\(Oracle\)](#), [Nilo Mitra\(Ericsson\)](#), [Peter Niblett\(IBM\)](#), [Duane Nickull\(Adobe\)](#), [Eisaku](#)  
2691 [Nishiyama\(Hitachi\)](#), [Dave Orchard\(BEA\)](#), [Chouthri Palanisamy\(NEC\)](#), [Sanjay Patil\(SAP\)](#), [Gilbert](#)  
2692 [Pilz\(BEA\)](#), [Martin Raeppe\(SAP\)](#), [Eric Rajkovic\(Oracle\)](#), [Stefan Rossmannith\(SAP\)](#), [Tom](#)  
2693 [Rutt\(Fujitsu\)](#), [Rich Salz\(IBM\)](#), [Shivajee Samdarshi\(Tibco\)](#), [Vladimir Videlov\(SAP\)](#), [Claus von](#)

2680 [Riegen\(SAP\), Pete Wenzel\(Sun\), Steve Winkler\(SAP\), Ümit Yalçinalp\(SAP\), Nobuyuki](#)  
2681 [Yamamoto\(Hitachi\).](#)  
2680 *TBD*

## Appendix G. 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>

<u>Rev</u>	<u>Date</u>	<u>By Whom</u>	<u>What</u>
<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>

<u>Rev</u>	<u>Date</u>	<u>By Whom</u>	<u>What</u>
			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 c:\temp\wsrm-1.1-schema-200510.xsd and to c:\temp\wsrm-1.1-wsdl-200510.wsdl
<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 CloseSequence and CloseSequenceResponse.
<a href="#">wd-09</a>	<a href="#">2006-01-11</a>	<a href="#">Doug Davis</a>	Minor tweaks to text/typos.
<a href="#">wd-10</a>	<a href="#">2006-01-23</a>	<a href="#">Doug Davis</a>	Accept all changes from wd-09  Make some minor editorial tweaks from Marc's comments.
<a href="#">wd-10</a>	<a href="#">2006-02-14</a>	<a href="#">Doug Davis</a>	Issue 082 resolution
<a href="#">wd-10</a>	<a href="#">2006-02-14</a>	<a href="#">Doug Davis</a>	Issue 083 resolution
<a href="#">wd-10</a>	<a href="#">2006-02-14</a>	<a href="#">Doug Davis</a>	Issue 085 resolution
<a href="#">wd-10</a>	<a href="#">2006-02-14</a>	<a href="#">Doug Davis</a>	Issues 086, 087 resolutions  Defined MessageNumberType
<a href="#">wd-10</a>	<a href="#">2006-02-15</a>	<a href="#">Doug Davis</a>	Issue 078 resolution
<a href="#">wd-10</a>	<a href="#">2006-02-15</a>	<a href="#">Doug Davis</a>	Issue 094 resolution
<a href="#">wd-10</a>	<a href="#">2006-02-15</a>	<a href="#">Doug Davis</a>	Issue 095 resolution
<a href="#">wd-10</a>	<a href="#">2006-02-15</a>	<a href="#">Gilbert Pilz</a>	Issue 088 – added namespace URI link to namespace URI; added text explaining that this URI could be dereferenced to produce the RDDDL doc; added non-normative reference to RDDDL 2.0
<a href="#">wd-10</a>	<a href="#">2006-02-17</a>	<a href="#">Anish Karmarkar</a>	Namespace changed to 200602 for both WSDL and XSD docs.
<a href="#">wd-10</a>	<a href="#">2006-02-17</a>	<a href="#">Anish Karmarkar</a>	Issue i087 as it applies to WSRM spec.
<a href="#">wd-10</a>	<a href="#">2006-02-17</a>	<a href="#">Anish Karmarkar</a>	Added titles and minor text for state table (issue i058).
<a href="#">wd-11</a>	<a href="#">2006-02-22</a>	<a href="#">Doug Davis</a>	Accept all changes for new WD  Minor typos fixed
<a href="#">wd-11</a>	<a href="#">2006-02-23</a>	<a href="#">Doug Davis</a>	s'/close'/close/g – per Marc Goodner  Added first ref to [URI] – per Marc G again
<a href="#">wd-11</a>	<a href="#">2006-02-27</a>	<a href="#">Doug Davis</a>	Issue i061 applied
<a href="#">wd-11</a>	<a href="#">2006-02-28</a>	<a href="#">Doug Davis</a>	Fixed typo around the use of "above" and "below"
<a href="#">wd-11</a>	<a href="#">2006-03-01</a>	<a href="#">Doug Davis</a>	Minor typos found by Marc Goodner
<a href="#">wd-11</a>	<a href="#">2006-03-02</a>	<a href="#">Doug Davis</a>	Minor typos found by Matt Lovett
<a href="#">wd-11</a>	<a href="#">2006-03-08</a>	<a href="#">Doug Davis</a>	Issue 091 applied
<a href="#">wd-11</a>	<a href="#">2006-03-08</a>	<a href="#">Doug Davis</a>	Issue 092 applied
<a href="#">wd-11</a>	<a href="#">2006-03-08</a>	<a href="#">Doug Davis</a>	Issue 100 applied

Rev	Date	By Whom	What
wd-12	2006-03-20	Doug Davis	Added space in "SOAP1.x" – PaulCotton
wd-12	2006-04-11	Doug Davis	Issue 007 applied
wd-12	2006-04-11	Doug Davis	Issue 090 applied
wd-12	2006-04-11	Doug Davis	Issue 098 applied
wd-12	2006-04-11	Doug Davis	Issue 099 applied
wd-12	2006-04-11	Doug Davis	Issue 101 applied
wd-12	2006-04-11	Doug Davis	Issue 103 applied
wd-12	2006-04-11	Doug Davis	Issue 104 applied
wd-12	2006-04-11	Doug Davis	Issue 105 applied
wd-12	2006-04-11	Doug Davis	Issue 107 applied
wd-12	2006-04-11	Doug Davis	Issue 109 applied
wd-12	2006-04-11	Doug Davis	Issue 110 applied
wd-12	2006-04-12	Doug Davis	Used "generated" instead of "issue" or "send" when talking about faults.
wd-12	2006-04-24	Gilbert Pilz	Update references to WS-Addressing to the Proposed Recommendations; update WS-RM namespace to "200604".
wd-13	2006-05-08	Gilbert Pilz	i093 part 1; more work needed
wd-13	2006-05-10	Doug Davis	Issue 096 applied
wd-13	2006-05-26	Gilbert Pilz	i093 part 2; reflects decisions from 2006-05-25 meeting
wd-13	2006-05-28	Gilbert Pilz	Issue 106 applied
wd-13	2006-05-29	Gilbert Pilz	Issue 118 applied
wd-13	2006-05-29	Gilbert Pilz	Issue 120 applied
wd-13	2006-05-30	Gilbert Pilz	Issue 114 applied
wd-13	2006-05-30	Gilbert Pilz	Issue 116 applied
wd-14	2006-06-05	Gilbert Pilz	Accept all changes; bump WD number
wd-14	2006-06-07	Doug Davis	Applied lots of minor edits from Marc Goodner
wd-14	2006-06-07	Doug Davis	Change a couple of period/sp/sp to period/sp
wd-14	2006-06-07	Doug Davis	Added a space in "URI]of" – per Marc Goodner
wd-14	2006-06-07	Doug Davis	Issue 131 applied
wd-14	2006-06-07	Doug Davis	Issue 132 applied
wd-14	2006-06-07	Doug Davis	Issue 119 applied
wd-14	2006-06-07	Doug Davis	Applied lots of minor edits from Doug Davis
wd-14	2006-06-07	Doug Davis	s/"none"/"full-uri"/ - per Marc Goodner
wd-14	2006-06-12	Doug Davis	Complete i106
wd-14	2006-06-12	Doug Davis	Issues 089 applied
wd-14	2006-06-12	Doug Davis	Fix for several RFC2119 keywords – per Anish
wd-15	2006-06-12	Doug Davis	Accept all changed, dump WD number
wd-15	2006-06-12	Doug Davis	Move WSDL after Schema
wd-15	2006-06-12	Doug Davis	Nits – remove tabs, extra [yyy]'s ...
wd-15	2006-06-14	Doug Davis	Remove extra "OPTIONAL"s – Matt Lovett



Rev	Date	By Whom	What
<a href="#">wd-15</a>	<a href="#">2006-06-14</a>	<a href="#">Doug Davis</a>	<a href="#">Remove blank rows/columns from state table. Fix italics in state table</a>
<a href="#">wd-15</a>	<a href="#">2006-06-15</a>	<a href="#">Doug Davis</a>	<a href="#">Typo – section D was empty</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 125 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 126 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 127 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 133 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 136 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 138 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 135 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-20</a>	<a href="#">Doug Davis</a>	<a href="#">Added all TC members to the ack list</a>
<a href="#">wd-15</a>	<a href="#">2006-06-22</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 129 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-22</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 130 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-22</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 137 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-26</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 111 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-26</a>	<a href="#">Doug Davis</a>	<a href="#">Missed a part of issue 129</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Fixed a typo in schema</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 141 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 142 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 148 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 149 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 150 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-06</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 121 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 139 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 144 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 147 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">Issues 122-124 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-27</a>	<a href="#">Doug Davis</a>	<a href="#">Updated list of oasis TC members (i134)</a>
<a href="#">wd-15</a>	<a href="#">2006-07-27</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 140 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-27</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 145 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-27</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 143 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-28</a>	<a href="#">Doug Davis</a>	<a href="#">Lots of minor typos found by Matt L.</a>
<a href="#">wd-15</a>	<a href="#">2006-07-28</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 113 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Update old namespaces – found by PaulC</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 150 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Minor typos – found by PeterN</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Verify all [refs]</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Change namespace to 2006/08</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 148 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-08-07</a>	<a href="#">Doug Davis</a>	<a href="#">Add some new glossary terms – per GilP</a>
<a href="#">cd-04</a>	<a href="#">2006-08-10</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Formatting changes for better HTML rendering.</a>

Rev	Date	By Whom	What
cd-04	2006-08-11	Doug Davis	Issue 158 applied
cd-04	2006-08-11	Doug Davis	Issue 153 applied
cd-04	2006-08-11	Doug Davis	Issue 156 applied
cd-04	2006-08-15	Gilbert Pilz	More formatting changes for better HTML rendering.
Rev	Date	By Whom	What
wd-01	2005-07-07	Christopher Ferris	Initial version created based on submission by the authors.
ws-02	2005-07-21	Doug Davis	i011 (PTOS) added
wd-02	2005-08-16	Anish Karmarkar	Trivial editorial changes
ws-03	2005-09-15	Doug Davis	i019 and i028 (CloseSeq) added
wd-05	2005-09-26	Gilbert Pilz	i005 (Source resend of nacks messages when ack already received) added.
wd-05	2005-09-27	Doug Davis	i027 (InOrder delivery assurance spanning multiple sequences) added
wd-05	2005-09-27	Doug Davis	i020 (Semantics of "At most once" Delivery Assurance) added
wd-05	2005-09-27	Doug Davis	i034 (Fault while processing a piggy-backed RM header) added
wd-05	2005-09-27	Doug Davis	i033 (Processing model of NACKs) added
wd-05	2005-09-27	Doug Davis	i031 (AckRequested schema inconsistency) added
wd-05	2005-09-27	Doug Davis	i025 (SeqAck/None) added
wd-05	2005-09-27	Doug Davis	i029 (Remove dependency on WS-Security) added
wd-05	2005-09-27	Doug Davis	i039 (What does 'have a mU attribute' mean) added
wd-05	2005-09-27	Doug Davis	i040 (Change 'optional'/'required' to 'OPTIONAL'/'REQUIRED') added
wd-05	2005-09-30	Anish Karmarkar	i017 (Change NS to http://docs.oasis-open.org/wsrn/200510/-)
wd-05	2005-09-30	Anish Karmarkar	i045 (Include SecureConversation as a reference and move it to non-normative-citation)
wd-05	2005-09-30	Anish Karmarkar	i046 (change the type of wsrn:FaultCode element)
wd-06	2005-11-02	Gilbert Pilz	Start wd-06 by changing title page from cd-01.
wd-06	2005-11-03	Gilbert Pilz	i047 (Reorder spec sections)
wd-07	2005-11-17	Gilbert Pilz	Start wd-07
wd-07	2005-11-28	Doug Davis	i071 — except for period in Appendix headings
wd-07	2005-11-28	Doug Davis	i10
wd-07	2005-11-28	Doug Davis	i030
wd-07	2005-11-28	Doug Davis	i037
wd-07	2005-11-28	Doug Davis	i038
wd-07	2005-11-28	Doug Davis	i041

Rev	Date	By Whom	What
wd-07	2005-11-28	Doug Davis	i043
wd-07	2005-11-28	Doug Davis	i044
wd-07	2005-11-28	Doug Davis	i048
wd-07	2005-11-28	Doug Davis	i051
wd-07	2005-11-28	Doug Davis	i053
wd-07	2005-11-28	Doug Davis	i059
wd-07	2005-11-28	Doug Davis	i062
wd-07	2005-11-28	Doug Davis	i063
wd-07	2005-11-28	Doug Davis	i065
wd-07	2005-11-28	Doug Davis	i067
wd-07	2005-11-28	Doug Davis	i068
wd-07	2005-11-28	Doug Davis	i069
wd-07	2005-11-28	Doug Davis	Fix bulleted list (#2) in section 2.3
wd-07	2005-11-29	Gilbert Pilz	i074 (Use of [tcShortName] in artifact locations namespaces, etc)
wd-07	2005-11-29	Gilbert Pilz	i071 — Fixed styles and formatting for TOC. Fixed styles of the appendix headings.
wd-07	2005-11-30	Doug Davis	Removed dup definition of "Receive"
wd-07	2005-11-30	Gilbert Pilz	Fixed lost formatting from heading for Namespace section. Fixed style of text body elements to match OASIS example documents. Fixed tables to match OASIS example documents.
wd-07	2005-12-01	Gilbert Pilz	Updated fix for i074 to eliminate trailing '/'. Added corresponding text around action IRI composition.
wd-07	2005-12-01	Gilbert Pilz	Use non-fixed fields for date values on both title page and body footers.
wd-07	2005-12-01	Doug Davis	Alphabetize the glossary
wd-07	2005-12-02	Doug Davis	i064
wd-07	2005-12-02	Doug Davis	i066
wd-08	2005-12-15	Doug Davis	Add back in RM Source to glossary
wd-08	2005-12-15	Steve Winkler	Doug added Steve's editorial nits
wd-08	2005-12-21	Doug Davis	i050
wd-08	2005-12-21	Doug Davis	i081
wd-08	2005-12-21	Doug Davis	i080 — but i050 negates the need for any changes
wd-08	2005-12-21	Doug Davis	i079
wd-08	2005-12-21	Doug Davis	i076 — didn't add text about "replies" since the RMD to RMS sequence could be used for any message not just replies
wd-08	2005-12-21	Umit Yalcinalp	Action Su03: removed wsse from Table 4
wd-08	2005-12-21	Umit Yalcinalp	i057 per Sunnyvale F2F 2005, Cleaned up some formatting errors in contributors

Rev	Date	By Whom	What
wd-08	2005-12-27	Doug Davis	i060
wd-08	2005-12-27	Gilbert Pilz	Moved schema and WSDL files to their own artifacts. Converted source document to OpenDocument Text format. Changed line numbers to be a single style.
wd-08	2005-12-28	Anish Karmarkar	Included a section link to c:\temp\wsrm-1.1-schema-200510.xsd and to c:\temp\wsrm-1.1-wsdl-200510.wsdl
wd-08	2006-01-04	Gilbert Pilz	Fixed formatting for included sections.
wd-08	2006-01-05	Gilbert Pilz	Created links for unused references. Fixed exemplars for CloseSequence and CloseSequenceResponse.
wd-09	2006-01-11	Doug Davis	Minor tweaks to text/types.
wd-10	2006-01-23	Doug Davis	Accept all changes from wd-09  Make some minor editorial tweaks from Marc's comments.
wd-10	2006-02-14	Doug Davis	Issue 082 resolution
wd-10	2006-02-14	Doug Davis	Issue 083 resolution
wd-10	2006-02-14	Doug Davis	Issue 085 resolution
wd-10	2006-02-14	Doug Davis	Issues 086, 087 resolutions  Defined MessageNumberType
wd-10	2006-02-15	Doug Davis	Issue 078 resolution
wd-10	2006-02-15	Doug Davis	Issue 094 resolution
wd-10	2006-02-15	Doug Davis	Issue 095 resolution
wd-10	2006-02-15	Gilbert Pilz	Issue 088 — added namespace URI link to namespace URI; added text explaining that this URI could be dereferenced to produce the RDDL doc; added non-normative reference to RDDL 2.0
wd-10	2006-02-17	Anish Karmarkar	Namespace changed to 200602 for both WSDL and XSD docs.
wd-10	2006-02-17	Anish Karmarkar	Issue i087 as it applies to WSRM spec.
wd-10	2006-02-17	Anish Karmarkar	Added titles and minor text for state table (issue i058).

## 2680 **Appendix H. Notices**

2681 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that  
2682 might be claimed to pertain to the implementation or use of the technology described in this document or  
2683 the extent to which any license under such rights might or might not be available; neither does it represent  
2684 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to  
2685 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made  
2686 available for publication and any assurances of licenses to be made available, or the result of an attempt  
2687 made to obtain a general license or permission for the use of such proprietary rights by implementors or  
2688 users of this specification, can be obtained from the OASIS Executive Director.

2689 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or  
2690 other proprietary rights which may cover technology that may be required to implement this specification.  
2691 Please address the information to the OASIS Executive Director.

2692 Copyright (C) OASIS Open (2006). All Rights Reserved.

2693 This document and translations of it may be copied and furnished to others, and derivative works that  
2694 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
2695 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and  
2696 this paragraph are included on all such copies and derivative works. However, this document itself may  
2697 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as  
2698 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights  
2699 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it  
2700 into languages other than English.

2701 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors  
2702 or assigns.

2703 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
2704 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY  
2705 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR  
2706 ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.