



1 ~~Web Services Reliable Messaging~~
2 ~~(WS-ReliableMessaging)~~

3 Web Services Reliable Messaging (WS- 4 ReliableMessaging) 1.1

5 Committee Draft 05

6 1 February 2007

7 **Specification URIs:**

8 **This Version:**

9 <http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-spec-cd-05.pdf>

10 <http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-spec-cd-05.html>

11 **Previous Version:**

12 ~~Committee Draft 04, August 11, 2006~~

13 **Document identifier:**

14 ~~wsrn-1.1-spec-cd-04~~

15 **Location:**

16 <http://docs.oasis-open.org/ws-rx/wsrn/200608/wsrn-1.1-spec-cd-04.pdf>

17 <http://docs.oasis-open.org/ws-rx/wsrn/200608/wsrn-1.1-spec-cd-04.html>

18 **Latest Version:**

19 <http://docs.oasis-open.org/ws-rx/wsrn/200702>

20 **Latest Approved Version:**

21 <http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-spec-cd-05.pdf>

22 <http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-spec-cd-05.html>

23 **Technical Committee:**

24 [OASIS Web Services Reliable Exchange \(WS-RX\) TC](#)

25 **Chairs:**

26 [Paul Fremantle <paul@wso2.com>](mailto:paul@wso2.com)

27 [Sanjay Patil <sanjay.patil@sap.com>](mailto:sanjay.patil@sap.com)

28 **Editors:**

29 Doug Davis, IBM <dug@us.ibm.com>

30 Anish Karmarkar, Oracle <Anish.Karmarkar@oracle.com>

31 Gilbert Pilz, BEA <gpilz@bea.com>

32 Steve Winkler, SAP <steve.winkler@sap.com>

33 Ümit Yalçinalp, SAP <umit.yalcinalp@sap.com>

34 **Related WorkContributors:**

35 This specification replaces or supercedes:

- 36 ● WS-ReliableMessaging v1.0

37 **Declared XML Namespaces:**

38 <http://docs.oasis-open.org/ws-rx/wsrn/200702>

39 See the Acknowledgments (Appendix E).

40 **Abstract:**

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

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

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

56 **Status:**

57 This document was last revised or approved by the WS-RX on the above date. The level of
58 approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location
59 noted above for possible later revisions of this document~~current location noted above for possible~~
60 ~~later revisions of this document. This document is updated periodically on no particular schedule.~~
61 ~~Technical Committee members should send comments on this specification to the Technical~~
62 ~~Committee's email list. Others should send comments to the Technical Committee by using the~~
63 ~~"Send A Comment" button on the Technical Committee's web page at [open.org/committees/ws-rx](http://www.oasis-
64 <a href=). For information on whether any patents have been disclosed that~~
65 ~~may be essential to implementing this specification, and any offers of patent licensing terms,~~
66 ~~please refer to the Intellectual Property Rights section of the Technical Committee web page~~
67 ~~(<http://www.oasis-open.org/committees/ws-rx/ipr.php>. The non-normative errata page for this~~
68 ~~specification is located at <http://www.oasis-open.org/committees/ws-rx>.~~

69 Technical Committee members should send comments on this specification to the Technical
70 Committee's email list. Others should send comments to the Technical Committee by using the
71 "Send A Comment" button on the Technical Committee's web page at [open.org/committees/ws-rx](http://www.oasis-</u>
72 <u><a href=).

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

77 The non-normative errata page for this specification is located at [open.org/committees/ws-rx](http://www.oasis-</u>
78 <u><a href=).

79 Notices

80 Copyright © OASIS® 1993–2007. All Rights Reserved. OASIS trademark, IPR and other policies apply.

81 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
82 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

83 This document and translations of it may be copied and furnished to others, and derivative works that
84 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published,
85 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
86 and this section are included on all such copies and derivative works. However, this document itself may
87 not be modified in any way, including by removing the copyright notice or references to OASIS, except as
88 needed for the purpose of developing any document or deliverable produced by an OASIS Technical
89 Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be
90 followed) or as required to translate it into languages other than English.

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

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

98 OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
99 necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to
100 notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such
101 patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced
102 this specification.

103 OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any
104 patent claims that would necessarily be infringed by implementations of this specification by a patent
105 holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR
106 Mode of the OASIS Technical Committee that produced this specification. OASIS may include such
107 claims on its website, but disclaims any obligation to do so.

108 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
109 might be claimed to pertain to the implementation or use of the technology described in this document or
110 the extent to which any license under such rights might or might not be available; neither does it represent
111 that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to
112 rights in any document or deliverable produced by an OASIS Technical Committee can be found on the
113 OASIS website. Copies of claims of rights made available for publication and any assurances of licenses
114 to be made available, or the result of an attempt made to obtain a general license or permission for the
115 use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS
116 Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any
117 information or list of intellectual property rights will at any time be complete, or that any claims in such list
118 are, in fact, Essential Claims.

119 The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be
120 used only to refer to the organization and its official outputs. OASIS welcomes reference to, and
121 implementation and use of, specifications, while reserving the right to enforce its marks against
122 misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

123 **Table of Contents**

124	1 Introduction.....	7
125	1.1 Terminology.....	7
126	1.2 Namespace.....	8
127	1.3 Conformance.....	8
128	2 Reliable Messaging Model.....	9
129	2.1 Glossary.....	9
130	2.2 Protocol Preconditions.....	10
131	2.3 Protocol Invariants.....	11
132	2.4 Delivery Assurances.....	11
133	2.5 Example Message Exchange.....	12
134	3 RM Protocol Elements.....	14
135	3.1 Considerations on the Use of Extensibility Points.....	14
136	3.2 Considerations on the Use of "Piggy-Backing".....	14
137	3.3 Composition with WS-Addressing.....	14
138	3.4 Sequence Creation.....	15
139	3.5 Closing A Sequence.....	19
140	3.6 Sequence Termination.....	21
141	3.7 Sequences.....	23
142	3.8 Request Acknowledgement.....	24
143	3.9 Sequence Acknowledgement.....	25
144	4 Faults.....	28
145	4.1 SequenceFault Element.....	29
146	4.2 Sequence Terminated.....	30
147	4.3 Unknown Sequence.....	30
148	4.4 Invalid Acknowledgement.....	31
149	4.5 Message Number Rollover.....	31
150	4.6 Create Sequence Refused.....	32
151	4.7 Sequence Closed.....	32
152	4.8 WSRM Required.....	33
153	5 Security Threats and Countermeasures.....	34
154	5.1 Threats and Countermeasures.....	34
155	5.2 Security Solutions and Technologies.....	36
156	6 Securing Sequences.....	40
157	6.1 Securing Sequences Using WS-Security.....	40
158	6.2 Securing Sequences Using SSL/TLS.....	41
159	7 References.....	43
160	7.1 Normative.....	43

161	7.2 Non-Normative	44
162	Appendix A. Schema.....	46
163	Appendix B. WSDL.....	51
164	Appendix C. Message Examples.....	53
165	Appendix C.1 Create Sequence.....	53
166	Appendix C.2 Initial Transmission.....	53
167	Appendix C.3 First Acknowledgement.....	55
168	Appendix C.4 Retransmission.....	55
169	Appendix C.5 Termination.....	56
170	Appendix D. State Tables.....	58
171	Appendix E. Acknowledgments.....	63
172	Appendix F. Revision History.....	64

173 1 Introduction

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

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

183 1.1 Terminology

184 1.1 Notational Conventions

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

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

- 189 • The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- 190 • Characters are appended to elements and attributes to indicate cardinality:
 - 191 ○ "?" (0 or 1)
 - 192 ○ "*" (0 or more)
 - 193 ○ "+" (1 or more)
- 194 • The character "|" is used to indicate a choice between alternatives.
- 195 • The characters "[" and "]" are used to indicate that contained items are to be treated as a group
196 with respect to cardinality or choice.
- 197 • An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content
198 specified in this document. Additional children elements and/or attributes MAY be added at the
199 indicated extension points but they MUST NOT contradict the semantics of the parent and/or
200 owner, respectively. If an extension is not recognized it SHOULD be ignored.
- 201 • XML namespace prefixes (See Section 1.2) are used to indicate the namespace of the element
202 being defined.

203 Elements and Attributes defined by this specification are referred to in the text of this document using
204 XPath 1.0 [XPATH 1.0] expressions. Extensibility points are referred to using an extended version of this
205 syntax:

- 206 • An element extensibility point is referred to using {any} in place of the element name. This
207 indicates that any element name can be used, from any namespace other than the wsrm:
208 namespace.

- An attribute extensibility point is referred to using @{any} in place of the attribute name. This indicates that any attribute name can be used, from any namespace other than the wsrn: namespace.

1.2 Namespace

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

<http://docs.oasis-open.org/ws-rx/wsrn/200702608>

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

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

Table 1

Prefix	Namespace
S	(Either SOAP 1.1 or 1.2)
S11	http://schemas.xmlsoap.org/soap/envelope/
S12	http://www.w3.org/2003/05/soap-envelope
wsrn	http://docs.oasis-open.org/ws-rx/wsrn/200702
wsa	http://www.w3.org/2005/08/addressing
wsam	http://www.w3.org/2007/02/addressing/metadata
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
xs	http://www.w3.org/2001/XMLSchema

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

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

1.3 Conformance

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

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

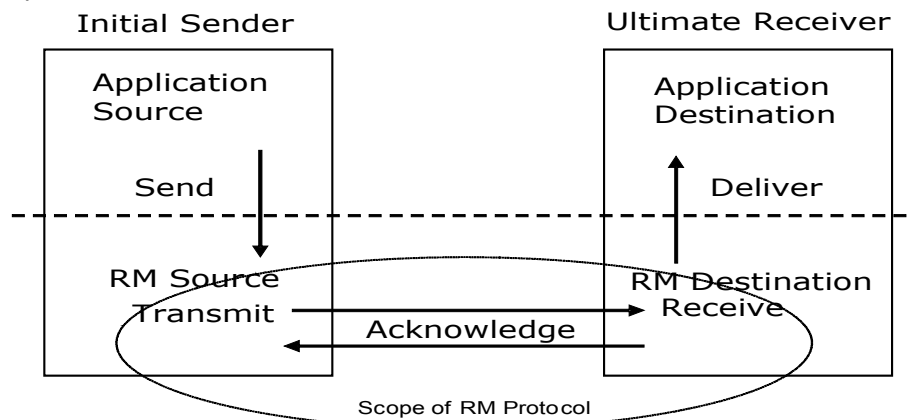
232 2 Reliable Messaging Model

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

235 The WS-ReliableMessaging specification defines an interoperable protocol that enables a Reliable
236 Messaging (RM) Source to accurately determine the disposition of each message it Transmits as
237 perceived by the RM Destination, so as to allow it to resolve any in-doubt status regarding receipt of the
238 message Transmitted. The protocol also enables an RM Destination to efficiently determine which of
239 those messages it Receives have been previously Received, enabling it to filter out duplicate message
240 transmissions caused by the retransmission, by the RM Source, of [an](#) unacknowledged message. It also
241 enables an RM Destination to Deliver the messages it Receives to the Application Destination in the order
242 in which they were sent by an Application Source, in the event that they are Received out of order. Note
243 that this specification places no restriction on the scope of the RM Source or RM Destination entities. For
244 example, either can span multiple WSDL Ports or Endpoints.

245 The protocol enables the implementation of a broad range of reliability features which include ordered
246 Delivery, duplicate elimination, and guaranteed receipt. The protocol can also be implemented with a
247 range of robustness characteristics ranging from in-memory persistence that is scoped to a single process
248 lifetime, to replicated durable storage that is recoverable in all but the most extreme circumstances. It is
249 expected that the Endpoints will implement as many or as few of these reliability characteristics as
250 necessary for the correct operation of the application using the protocol. Regardless of which of the
251 reliability features is enabled, the wire protocol does not change.

252 Figure 1 below illustrates the entities and events in a simple reliable exchange of messages. First, the
253 Application Source Sends a message for reliable transfer. The Reliable Messaging Source accepts the
254 message and Transmits it one or more times. After accepting the message, the RM Destination
255 Acknowledges it. Finally, the RM Destination Delivers the message to the Application Destination. The
256 exact roles the entities play and the complete meaning of the events will be defined throughout this
257 specification.



258 Figure 1: Reliable Messaging Model

259 2.1 Glossary

260 The following definitions are used throughout this specification:

261 **Accept:** The act of qualifying a message by the RM Destination such that it becomes eligible for Delivery
262 and acknowledgement.

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

265 **Acknowledgement Message:** A message containing a `SequenceAcknowledgement` header block.
266 Acknowledgement Messages may or may not contain a SOAP body.

267 **Acknowledgement Request:** A message containing an `AckRequested` header. Acknowledgement
268 Requests may or may not contain a SOAP body.

269 **Application Destination:** The Endpoint to which a message is Delivered.

270 **Application Source:** The Endpoint that Sends a message.

271 **Back-channel:** When the underlying transport provides a mechanism to return a transport-protocol
272 specific response, capable of carrying a SOAP message, without initiating a new connection, this
273 specification refers to this mechanism as a back-channel.~~**Deliver:** The act of transferring a message from~~
274 ~~the RM Destination to the Application Destination.~~

275 ~~**Deliver:** The act of transferring responsibility for a message from the RM Destination to the Application~~
276 ~~Destination.~~

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

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

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

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

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

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

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

289 **Sequence Lifecycle Message:** A message that contains one of: `CreateSequence`,
290 `CreateSequenceResponse`, `CloseSequence`, `CloseSequenceResponse`, `TerminateSequence`,
291 `TerminateSequenceResponse` as the child element of the SOAP body element.

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

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

294 2.2 Protocol Preconditions

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

- 297 • For any single message exchange the RM Source MUST have an endpoint reference that uniquely
298 identifies the RM Destination Endpoint.
- 299 • The RM Source MUST have successfully created a Sequence with the RM Destination.
- 300 • The RM Source MUST be capable of formulating messages that adhere to the RM Destination's
301 policies.

- 302 • If a secure exchange of messages is REQUIRED, then the RM Source and RM Destination MUST
303 have a security context.

304 **2.3 Protocol Invariants**

305 During the lifetime of a Sequence, the following invariants are REQUIRED for correctness:

- 306 • The RM Source MUST assign each message within a Sequence a message number (defined
307 below) beginning at 1 and increasing by exactly 1 for each subsequent message. These numbers
308 MUST be assigned in the same order in which messages are sent by the Application Source.
- 309 • Within every Acknowledgement Message it issues, the RM Destination MUST include one or more
310 `AcknowledgementRange` child elements that contain, in their collective ranges, the message
311 number of every message accepted by the RM Destination. The RM Destination MUST exclude, in
312 the `AcknowledgementRange` elements, the message numbers of any messages it has not
313 accepted. If no messages have been received the RM Destination MUST return `None` instead of an
314 `AcknowledgementRange(s)`. The RM Destination MAY transmit a `Nack` for a specific message
315 or messages instead of an `AcknowledgementRange(s)`.
- 316 • While the Sequence is not closed or terminated, the RM Source SHOULD retransmit
317 unacknowledged messages.

318 **2.4 Delivery Assurances**

319 This section defines a number of Delivery Assurance assertions, which can be supported by RM Sources
320 and RM Destinations. These assertions can be specified as policy assertions using the WS-Policy
321 framework [[WS-Policy]]. For details on this see the WSRM Policy specification [WS-RM Policy].

322 AtLeastOnce

323 Each message is to be delivered at least once, or else an error MUST be raised by the RM Source and/or
324 RM Destination. The requirement on an RM Source is that it SHOULD retry transmission of every
325 message sent by the Application Source until it receives an acknowledgement from the RM Destination.
326 The requirement on the RM Destination is that it SHOULD retry the transfer to the Application Destination
327 of any message that it accepts from the RM Source, until that message has been successfully delivered.
328 There is no requirement for the RM Destination to apply duplicate message filtering.

329 AtMostOnce

330 Each message is to be delivered at most once. The RM Source MAY retry transmission of
331 unacknowledged messages, but is NOT REQUIRED to do so. The requirement on the RM Destination is
332 that it MUST filter out duplicate messages, i.e. that it MUST NOT deliver a duplicate of a message that
333 has already been delivered.

334 ExactlyOnce

335 Each message is to be delivered exactly once: if a message cannot be delivered then an error MUST be
336 raised by the RM Source and/or RM Destination. The requirement on an RM Source is that it SHOULD
337 retry transmission of every message sent by the Application Source until it receives an acknowledgement
338 from the RM Destination. The requirement on the RM Destination is that it SHOULD retry the transfer to
339 the Application Destination of any message that it accepts from the RM Source until that message has
340 been successfully delivered, and that it MUST NOT deliver a duplicate of a message that has already
341 been delivered.

342 InOrder

343 Messages from each individual sequence are to be delivered in the same order they have been sent by
 344 the Application Source. The requirement on an RM Source is that it MUST ensure that the ordinal position
 345 of each message in the sequence (as indicated by a message sequence number) is consistent with the
 346 order in which the messages have been sent from the Application Source. The requirement on the RM
 347 Destination is that it MUST deliver received messages for each sequence in the order indicated by the
 348 message numbering. This DeliveryAssurance can be used in combination with any of the AtLeastOnce,
 349 AtMostOnce or ExactlyOnce assertions, and the requirements of those assertions MUST also be met. In
 350 particular if the AtLeastOnce or ExactlyOnce assertion applies and the RM Destination detects a gap in
 351 the sequence then the RM Destination MUST NOT deliver any subsequent messages from that sequence
 352 until the missing messages are received or until the sequence is closed.

353 2.5 Example Message Exchange

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

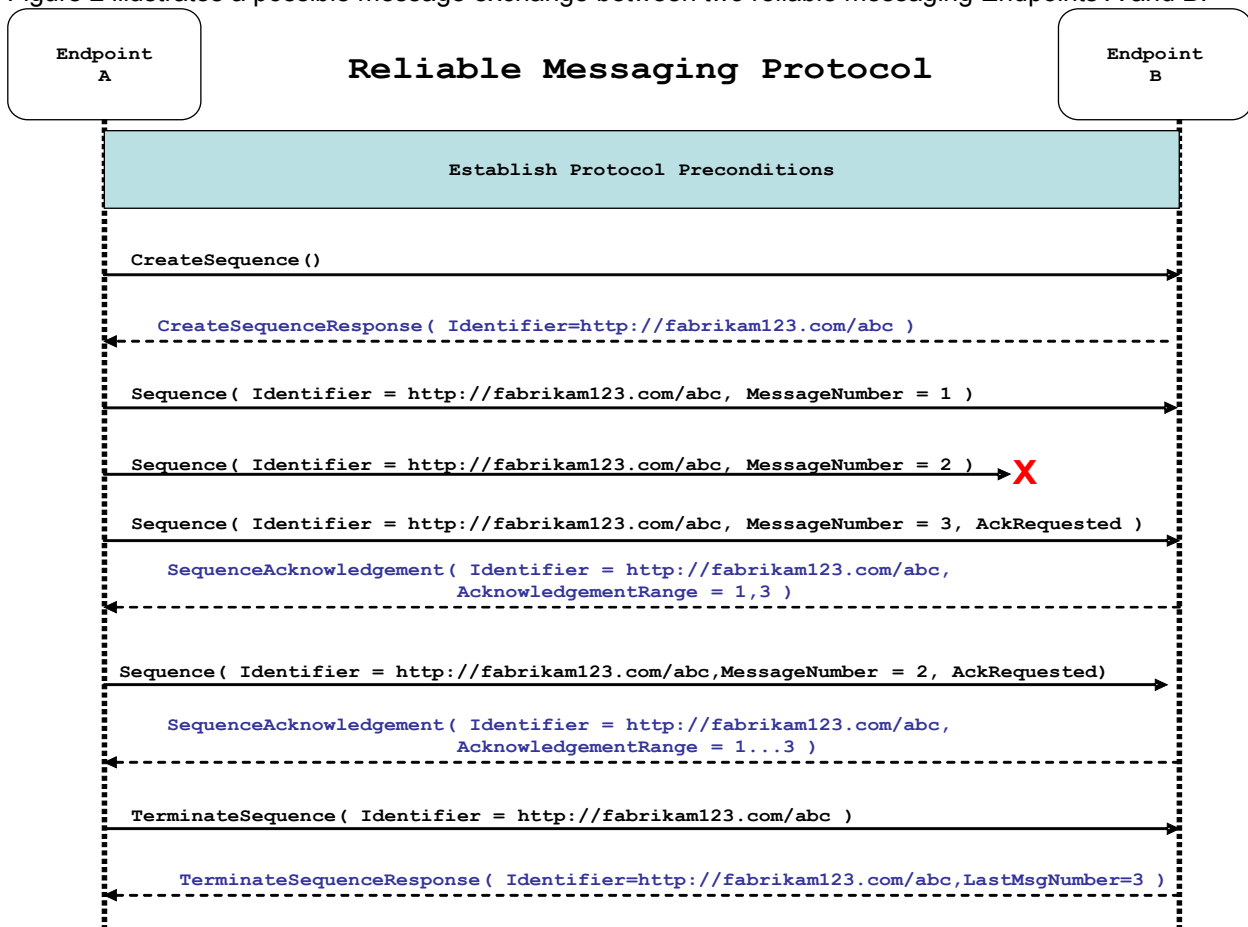


Figure 2: The WS-ReliableMessaging Protocol

- 355 1. The protocol preconditions are established. These include policy exchange, endpoint resolution,
- 356 and establishing trust.
- 357 2. The RM Source requests creation of a new Sequence.
- 358 3. The RM Destination creates a new Sequence and returns its unique identifier.

- 359 4. The RM Source begins Transmitting messages in the Sequence beginning with MessageNumber 1.
360 In the figure above, the RM Source sends 3 messages in the Sequence.
- 361 5. The 2nd message in the Sequence is lost in transit.
- 362 6. The 3rd message is the last in this Sequence and the RM Source includes an `AckRequested`
363 header to ensure that it gets a timely `SequenceAcknowledgement` for the Sequence.
- 364 7. The RM Destination acknowledges receipt of message numbers 1 and 3 as a result of receiving the
365 RM Source's `AckRequested` header.
- 366 8. The RM Source retransmits the unacknowledged message with MessageNumber 2. This is a new
367 message from the perspective of the underlying transport, but it has the same Sequence Identifier
368 and MessageNumber so the RM Destination can recognize it as a duplicate of the earlier message,
369 in case the original and retransmitted messages are both Received. The RM Source includes an
370 `AckRequested` header in the retransmitted message so the RM Destination will expedite an
371 acknowledgement.
- 372 9. The RM Destination Receives the second transmission of the message with MessageNumber 2
373 and acknowledges receipt of message numbers 1, 2, and 3.
- 374 10. The RM Source Receives this Acknowledgement and sends a `TerminateSequence` message to the
375 RM Destination indicating that the Sequence is completed. The `TerminateSequence` message
376 indicates that message number 3 was the last message in the Sequence. The RM Destination then
377 and reclaims any resources associated with the Sequence.
- 378 11. The RM Destination Receives the `TerminateSequence` message indicating that the RM Source will
379 not be sending any more messages. The RM Destination sends a `TerminateSequenceResponse`
380 message to the RM Source and and reclaims any resources associated with the Sequence.

381 The RM Source will expect to Receive Acknowledgements from the RM Destination during the course of a
382 message exchange at occasions described in Section 3 below. Should an Acknowledgement not be
383 Received in a timely fashion, the RM Source MUST re-transmit the message since either the message or
384 the associated Acknowledgement might have been lost. Since the nature and dynamic characteristics of
385 the underlying transport and potential intermediaries are unknown in the general case, the timing of re-
386 transmissions cannot be specified. Additionally, over-aggressive re-transmissions have been
387 demonstrated to cause transport or intermediary flooding which are counterproductive to the intention of
388 providing a reliable exchange of messages. Consequently, implementers are encouraged to utilize
389 adaptive mechanisms that dynamically adjust re-transmission time and the back-off intervals that are
390 appropriate to the nature of the transports and intermediaries envisioned. For the case of TCP/IP
391 transports, a mechanism similar to that described as RTTM in RFC 1323 [RTTM] SHOULD be
392 considered.

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

395 **3 RM Protocol Elements**

396 The following sub-sections define the various RM protocol elements, and prescribe their usage by a
397 conformant implementations.

398 **3.1 Considerations on the Use of Extensibility Points**

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

403 **3.2 Considerations on the Use of "Piggy-Backing"**

404 Some RM Protocol Header Blocks may be added to messages that are targeted to the same Endpoint to
405 which those headers are to be sent (a concept often referred to as "piggy-backing"), thus saving the
406 overhead of an additional message exchange. Reference parameters MUST be considered when
407 determining whether two EPRs are targeted to the same Endpoint. The determination of if and when a
408 Header Block will be piggy-backed onto another message is made by the entity (RM Source or RM
409 Destination) that is sending the header. In order to ensure optimal and successful processing of RM
410 Sequences, endpoints that receive RM-related messages SHOULD be prepared to process RM Protocol
411 Header Blocks that are included in any message it receives. See the sections that define each RM
412 Protocol Header Block to know which ones may be considered for piggy-backing. header blocks may be
413 added to messages that happen to be targeted to the same Endpoint to which those headers are to be
414 sent (a concept often referred to as "piggy-backing"), thus saving the overhead of an additional message
415 exchange. Reference parameters MUST be considered when determining whether two EPRs are targeted
416 to the same Endpoint.

417 **3.3 Composition with WS-Addressing**

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

- 420 1. When an Endpoint generates a message that carries an RM protocol element, that is defined in
421 the following sections, in the body of a SOAP envelope that Endpoint MUST include in that
422 envelope a `wsa:Action` SOAP header block whose value is an IRI that is a concatenation of the
423 WS-RM namespace URI, followed by a "/", followed by the value of the local name of the child
424 element of the SOAP body . For example, for a Sequence creation request message as described
425 in section 3.4section 3 below, in the body of a SOAP envelope that Endpoint MUST include in that
426 envelope a `wsa:Action` SOAP header block whose value is an IRI that is a concatenation of the
427 WS-RM namespace URI, followed by a "/", followed by the value of the local name of the child
428 element of the SOAP body . For example, for a Sequence creation request message as described
429 in section 3.4 below, the value of the `wsa:Action` IRI would be:

```
430 http://docs.oasis-open.org/ws-rx/wsrn/200702608/CreateSequence
```

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

```
433 http://docs.oasis-open.org/ws-rx/wsrn/200702608/SequenceAcknowledgement
```

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

436 `http://docs.oasis-open.org/ws-rx/wsrn/200702698/AckRequested`

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

439 3.4 Sequence Creation

440 The RM Source MUST request creation of an outbound Sequence by sending a `CreateSequence`
441 element in the body of a message to the RM Destination which in turn responds either with a message
442 containing `CreateSequenceResponse` or a `CreateSequenceRefused` fault. The RM Source MAY
443 include an offer to create an inbound Sequence within the `CreateSequence` message. This offer is
444 either accepted or rejected by the RM Destination in the `CreateSequenceResponse` message.

445 The SOAP version used for the `CreateSequence` message SHOULD be used for all subsequent
446 messages in or for that Sequence, sent by either the RM Source or the RM Destination.

447 The following exemplar defines the `CreateSequence` syntax:

```
448 <wsrm:CreateSequence ...>  
449   <wsrm:AcksTo> wsa:EndpointReferenceType </wsrm:AcksTo>  
450   <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
451   <wsrm:Offer ...>  
452     <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
453     <wsrm:Endpoint> wsa:EndpointReferenceType </wsrm:Endpoint>  
454     <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
455     <wsrm:IncompleteSequenceBehavior>  
456       wsrn:IncompleteSequenceBehaviorType  
457     </wsrm:IncompleteSequenceBehavior> ?  
458     ...  
459   </wsrm:Offer> ?  
460   ...  
461 </wsrm:CreateSequence>
```

462 [The following describes the content model of the `CreateSequence` element.](#)

463 `/wsrm:CreateSequence`

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

468 `/wsrm:CreateSequence/wsrm:AcksTo`

469 The RM Source MUST include this element in any `CreateSequence` message it sends. This element is of
470 type `wsa:EndpointReferenceType` (as specified by WS-Addressing). It specifies the endpoint
471 reference to which messages containing `SequenceAcknowledgement` header blocks and faults related
472 to the created Sequence are to be sent, unless otherwise noted in this specification (for example, see
473 Section 3.52).

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

478 /wsmr:CreateSequence/wsmr:Expires

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

483 /wsmr:CreateSequence/wsmr:Expires/@{any}

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

486 /wsmr:CreateSequence/wsmr:Offer

487 This element, if present, enables an RM Source to offer a corresponding Sequence for the reliable
488 exchange of messages Transmitted from RM Destination to RM Source.

489 /wsmr:CreateSequence/wsmr:Offer/wsmr:Identifier

490 The RM Source MUST set the value of this element to an absolute URI (conformant with RFC3986 [URI])
491 that uniquely identifies the offered Sequence.

492 /wsmr:CreateSequence/wsmr:Offer/wsmr:Identifier/@{any}

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

495 /wsmr:CreateSequence/wsmr:Offer/wsmr:Endpoint

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

500 Implementations MUST NOT use an endpoint reference in the Endpoint element that would prevent the
501 sending of Sequence Lifecycle Message, etc. For example, using the WS-Addressing
502 "http://www.w3.org/2005/08/addressing/none" IRI would make it impossible for the RM Destination to ever
503 send Sequence Lifecycle Messages (e.g. TerminateSequence) to the RM Source for the Offered
504 Sequence. Sequence Traffic Message, etc. For example, using the WS-Addressing-
505 "http://www.w3.org/2005/08/addressing/none" IRI would make it impossible for the RM Destination to ever
506 send Sequence Lifecycle Messages (e.g. TerminateSequence) to the RM Source for the Offered-
507 Sequence. Implementations MAY use the WS-RM anonymous URI template and doing so implies that
508 messages will be retrieved using a mechanism such as the MakeConnection message (see section-
509 3.7)-

510 The Offer of an Endpoint containing the "http://www.w3.org/2005/08/addressing/anonymous" IRI as its
511 address is problematic due to the inability of a source to connect to this address and retry
512 unacknowledged messages (as described in Section 2.3). Note that this specification does not define any
513 mechanisms for providing this assurance. In the absence of an extension that addresses this issue, an
514 RM Destination MUST NOT accept (via the /wsmr:CreateSequenceResponse/wsmr:Accept
515 element described below) an Offer that contains the "http://www.w3.org/2005/08/addressing/anonymous"
516 IRI as its address.

517 /wsmr:CreateSequence/wsmr:Offer/wsmr:Expires

518 This element, if present, of type `xs:duration` specifies the duration for the offered Sequence. A value of
519 "PT0S" indicates that the offered Sequence will never expire. Absence of the element indicates an implied
520 value of "PT0S".

521 `/wsmr:CreateSequence/wsmr:Offer/wsmr:Expires/@{any}`

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

524 `/wsmr:CreateSequence/wsmr:Offer/wsmr:IncompleteSequenceBehavior`

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

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

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

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

535 `/wsmr:CreateSequence/wsmr:Offer/{any}`

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

538 `/wsmr:CreateSequence/wsmr:Offer/@{any}`

539 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
540 element~~different (extensible) types of information, based on a schema, to be passed.~~

541 `/wsmr:CreateSequence/{any}`

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

544 `/wsmr:CreateSequence/@{any}`

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

547 A `CreateSequenceResponse` is sent in the body of a response message by an RM Destination in
548 response to receipt of a `CreateSequence` request message. It carries the `Identifier` of the created
549 Sequence and indicates that the RM Source can begin sending messages in the context of the identified
550 Sequence.

551 The following exemplar defines the `CreateSequenceResponse` syntax:

```
552 <wsmr:CreateSequenceResponse ...>  
553   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
554   <wsmr:Expires ...> xs:duration </wsmr:Expires> ?  
555   <wsmr:IncompleteSequenceBehavior>  
556     wsmr:IncompleteSequenceBehaviorType  
557   </wsmr:IncompleteSequenceBehavior> ?  
558   <wsmr:Accept ...>  
559     <wsmr:AcksTo wsa:EndpointReferenceType </wsmr:AcksTo>  
560     ...
```

```
561     </wsrm:Accept> ?
562     ...
563 </wsrm:CreateSequenceResponse>
```

564 [The following describes the content model of the CreateSequenceResponse element.](#)

565 /wsrm:CreateSequenceResponse

566 This element is sent in the body of the response message in response to a `CreateSequence` request
567 message. It indicates that the RM Destination has created a new Sequence at the request of the RM
568 Source. The RM Destination MUST NOT send this element as a header block.

569 /wsrm:CreateSequenceResponse/wsrm:Identifier

570 The RM Destination MUST include this element within any `CreateSequenceResponse` message it sends.
571 The RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986)
572 that uniquely identifies the Sequence that has been created by the RM Destination.

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

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

576 /wsrm:CreateSequenceResponse/wsrm:Expires

577 This element, if present, of type `xs:duration` accepts or refines the RM Source's requested duration for
578 the Sequence. It specifies the amount of time after which any resources associated with the Sequence
579 SHOULD be reclaimed thus causing the Sequence to be silently terminated. At the RM Destination this
580 duration is measured from a point proximate to Sequence creation and at the RM Source this duration is
581 measured from a point approximate to the successful processing of the `CreateSequenceResponse`. A
582 value of "PT0S" indicates that the Sequence will never expire. Absence of the element indicates an
583 implied value of "PT0S". The RM Destination MUST set the value of this element to be equal to or less
584 than the value requested by the RM Source in the corresponding `CreateSequence` message.

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

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

588 /wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior

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

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

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

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

599 /wsrm:CreateSequenceResponse/wsrm:Accept

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

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

605 `/wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo`

606 The RM Destination MUST include this element, of type `wsa:EndpointReferenceType` (as specified
607 by WS-Addressing). It specifies the endpoint reference to which messages containing
608 `SequenceAcknowledgement` header blocks and faults related to the created Sequence are to be sent,
609 unless otherwise noted in this specification (for example, see Section 3.52).

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

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

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

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

618 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
619 element, different (extensible) types of information, based on a schema, to be passed.

620 `/wsrm:CreateSequenceResponse/{any}`

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

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

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

626 3.5 Closing A Sequence

627 There are times during the use of an RM Sequence that the RM Source or RM Destination will wish to
628 discontinue using a Sequence. Simply terminating the Sequence discards the state managed by the RM
629 Destination, leaving the RM Source unaware of the final ranges of messages that were successfully
630 transferred to the RM Destination. To ensure that the Sequence ends with a known final state either the
631 RM Source or RM Destination MAY choose to close the Sequence before terminating it.

632 If the RM Source wishes to close the Sequence, then it sends a `CloseSequence` element, in the body of
633 a message, to the RM Destination. This message indicates that the RM Destination MUST NOT accept
634 any new messages for the specified Sequence, other than those already accepted at the time the
635 `CloseSequence` element is interpreted by the RM Destination. Upon receipt of this message, or
636 subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST
637 include a final `SequenceAcknowledgement` (within which the RM Destination MUST include the `Final`
638 element) header block on any messages associated with the Sequence destined to the RM Source,
639 including the `CloseSequenceResponse` message or on any Sequence fault Transmitted to the RM
640 Source.

641 To allow the RM Destination to determine if it has received all of the messages in a Sequence, the RM
642 Source SHOULD include the `LastMsgNumber` element in any `CloseSequence` messages it sends. The

643 RM Destination can use this information, for example, to implement the behavior indicated by
644 /wsmr:CreateSequenceResponse/wsmr:IncompleteSequenceBehavior. The value of the
645 LastMsgNumber element MUST be the same in all the CloseSequence messages for the closing
646 Sequence.

647 If the RM Destination decides to close a Sequence of its own volition, it MAY inform the RM Source of this
648 event by sending a CloseSequence element, in the body of a message, to the AcksTo EPR of that
649 Sequence. The RM Destination MUST include a final SequenceAcknowledgement (within which the RM
650 Destination MUST include the Final element) header block in this message and any subsequent
651 messages associated with the Sequence destined to the RM Source.

652 While the RM Destination MUST NOT accept any new messages for the specified Sequence it MUST still
653 process Sequence Lifecycle Messages and Acknowledgement Requests. For example, it MUST respond to
654 AckRequested, TerminateSequence as well as CloseSequence messages. Note, subsequent
655 CloseSequence messages have no effect on the state of the Sequence.

656 In the case where the RM Destination wishes to discontinue use of a Sequence it is RECOMMENDED
657 that it close the Sequence. Please see Final and the SequenceClosed fault. Whenever possible the
658 SequenceClosed fault SHOULD be used in place of the SequenceTerminated fault to allow the RM
659 Source to still Receive Acknowledgements.

660 The following exemplar defines the CloseSequence syntax:

```
661 <wsmr:CloseSequence ...>  
662   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
663   <wsmr>LastMsgNumber> wsmr:MessageNumberType </wsmr>LastMsgNumber> ?  
664   ...  
665 </wsmr:CloseSequence>
```

666 The following describes the content model of the CloseSequence element.

667 /wsmr:CloseSequence

668 This element MAY be sent by an RM Source to indicate that the RM Destination MUST NOT accept any
669 new messages for this Sequence This element MAY also be sent by an RM Destination to indicate that it
670 will not accept any new messages for this Sequence is sent by an RM Source to indicate that the RM
671 Destination MUST NOT accept any new messages for this Sequence. A SequenceClosed fault MUST be
672 generated by the RM Destination when it Receives a message for a Sequence that is already closed.

673 /wsmr:CloseSequence/wsmr:Identifier

674 The RM Source or RM Destination MUST include this element in any CloseSequence messages it sends.
675 The RM Source or RM Destination MUST set the value of this element to the absolute URI (conformant
676 with RFC3986) of the closing Sequence MUST include this element in any CloseSequence messages it
677 sends. The RM Source MUST set the value of this element to the absolute URI (conformant with
678 RFC3986) of the Sequence that is being closed.

679 /wsmr:CloseSequence/wsmr>LastMessageNumber

680 The RM Source SHOULD include this element in any CloseSequence message it sends. The
681 LastMsgNumber element specifies the highest assigned message number of all the Sequence Traffic
682 Messages for the closing Sequence.

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

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

686 /wsmr:CloseSequence/{any}

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

689 /wsmr:CloseSequence@{any}

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

692 A `CloseSequenceResponse` is sent in the body of a [message in response to receipt of a](#)
693 [CloseSequence request message. It indicates that the responderresponse message by an RM-](#)
694 [Destination in response to receipt of a CloseSequence request message. It indicates that the RM-](#)
695 [Destination](#) has closed the Sequence.

696 The following exemplar defines the `CloseSequenceResponse` syntax:

```
697 <wsmr:CloseSequenceResponse ...>  
698   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
699   ...  
700 </wsmr:CloseSequenceResponse>
```

701 [The following describes the content model of the CloseSequenceResponse element.](#)

702 /wsmr:CloseSequenceResponse

703 This element is sent in the body of a [message in response to receipt of a CloseSequence request](#)
704 [message. It indicates that the responderresponse message by an RM Destination in response to receipt-](#)
705 [of a CloseSequence request message. It indicates that the RM Destination](#) has closed the Sequence.

706 /wsmr:CloseSequenceResponse/wsmr:Identifier

707 The [responder \(RM Source or RM Destination\) MUST include this element in any](#)
708 [CloseSequenceResponse message it sends. The responder MUST set the value of this element to the](#)
709 [absolute URI \(conformant with RFC3986\) of the closing SequenceRM Destination MUST include this-](#)
710 [element in any CloseSequenceResponse message it sends. The RM Destination MUST set the value of](#)
711 [this element to the absolute URI \(conformant with RFC3986\) of the Sequence that is being closed.](#)

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

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

715 /wsmr:CloseSequenceResponse/{any}

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

718 /wsmr:CloseSequenceResponse@{any}

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

721 **3.6 Sequence Termination**

722 When the RM Source has completed its use of the Sequence it sends a `TerminateSequence` element,
723 in the body of a message, to the RM Destination to indicate that the Sequence is complete and that it will
724 not be sending any further messages related to the Sequence. The RM Destination can safely reclaim any
725 resources associated with the Sequence upon receipt of the `TerminateSequence` message. Under
726 normal usage the RM Source will complete its use of the Sequence when all of the messages in the

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

729 To allow the RM Destination to determine if it has received all of the messages in a Sequence, the RM
730 Source SHOULD include the `LastMsgNumber` element in any `TerminateSequence` messages it sends.
731 The RM Destination can use this information, for example, to implement the behavior indicated by
732 `/wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior`. The value of the
733 `LastMsgNumber` element in the `TerminateSequence` message MUST be equal to the value of the
734 `LastMsgNumber` element in any `CloseSequence` message(s) sent by the RM Source for the same
735 Sequence.

736 If the RM Destination decides to terminate a Sequence of its own volition, it MAY inform the RM Source of
737 this event by sending a `TerminateSequence` element, in the body of a message, to the `AcksTo` EPR for
738 that Sequence. The RM Destination MUST include a final `SequenceAcknowledgement` (within which
739 the RM Destination MUST include the `Final` element) header block in this message.

740 The following exemplar defines the `TerminateSequence` syntax:

```
741 <wsrm:TerminateSequence ...>  
742   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
743   <wsrm>LastMsgNumber> wsrm:MessageNumberType </wsrm>LastMsgNumber> ?  
744   ...  
745 </wsrm:TerminateSequence>
```

746 The following describes the content model of the `TerminateSequence` element.

747 `/wsrm:TerminateSequence`

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

753 This element MAY also be sent by the RM Destination to indicate that it has unilaterally terminated the
754 Sequence. Upon sending this message the RM Destination MUST NOT accept any additional messages
755 (with the exception of the corresponding `TerminateSequenceResponse`) for this Sequence. Upon
756 receipt of a `TerminateSequence` the RM Source MUST NOT send any additional messages (with the
757 exception of the corresponding `TerminateSequenceResponse`) for this Sequence.

758 `/wsrm:TerminateSequence/wsrm:Identifier`

759 The RM Source **or** RM Destination **MUST** include this element in any `TerminateSequence` message it
760 sends. The RM Source **or** RM Destination **MUST** set the value of this element to the absolute URI
761 (conformant with RFC3986) of the terminating Sequence. ~~The RM Source MUST include this element in any~~
762 ~~`TerminateSequence` message it sends. The RM Source MUST set the value of this element to the~~
763 ~~absolute URI (conformant with RFC3986) of the Sequence that is being terminated.~~

764 `/wsrm:TerminateSequence/wsrm>LastMsgNumber`

765 The RM Source **SHOULD** include this element in any `TerminateSequence` message it sends. The
766 `LastMsgNumber` element specifies the highest assigned message number of all the Sequence Traffic
767 Messages for the closing Sequence.

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

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

771 /wsmr:TerminateSequence/{any}

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

774 /wsmr:TerminateSequence/@{any}

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

777 A TerminateSequenceResponse is sent in the body of a message in response to receipt of a
778 TerminateSequence request message. It indicates that responderresponse message by an RM-
779 Destination in response to receipt of a TerminateSequence request message. It indicates that the RM-
780 Destination has terminated the Sequence.

781 The following exemplar defines the TerminateSequenceResponse syntax:

```
782 <wsmr:TerminateSequenceResponse ...>  
783   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
784   ...  
785 </wsmr:TerminateSequenceResponse>
```

786 The following describes the content model of the TerminateSequence element.

787 /wsmr:TerminateSequenceResponse

788 This element is sent in the body of a message in response to receipt of a TerminateSequence request
789 message. It indicates that the responder has terminated the Sequence. The responderresponse message-
790 by an RM Destination in response to receipt of a TerminateSequence request message. It indicates-
791 that the RM Destination has terminated the Sequence. The RM Destination MUST NOT send this element
792 as a header block.

793 /wsmr:TerminateSequenceResponse/wsmr:Identifier

794 The responder (RM Source or RM Destination) MUST include this element in any
795 TerminateSequenceResponse message it sends. The responder MUST set the value of this element
796 to the absolute URI (conformant with RFC3986) of the terminating SequenceRM Destination MUST-
797 include this element in any TerminateSequenceResponse message it sends. The RM Destination-
798 MUST set the value of this element to the absolute URI (conformant with RFC3986) of the Sequence that
799 is being terminated.

800 /wsmr:TerminateSequenceResponse/wsmr:Identifier/@{any}

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

803 /wsmr:TerminateSequenceResponse/{any}

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

806 /wsmr:TerminateSequenceResponse/@{any}

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

809 On receipt of a TerminateSequence message the receiver (RM Source or RM Destination)an RM-
810 Destination MUST respond with a corresponding TerminateSequenceResponse message or generate
811 a fault UnknownSequenceFault if the Sequence is not known.

812 3.7 Sequences

813 The RM protocol uses a Sequence header block to track and manage the reliable transfer of messages.
814 The RM Source MUST include a *Sequence* header block in all messages for which reliable transfer is
815 REQUIRED. The RM Source MUST identify Sequences with unique Identifier elements and the RM
816 Source MUST assign each message within a Sequence a *MessageNumber* element that increments by 1
817 from an initial value of 1. These values are contained within a *Sequence* header block accompanying
818 each message being transferred in the context of a Sequence.

819 The RM Source MUST NOT include more than one *Sequence* header block in any message.

820 A following exemplar defines its syntax:

```
821 <wsrm:Sequence ...>  
822   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
823   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>  
824   ...  
825 </wsrm:Sequence>
```

826 The following describes the content model of the *Sequence* header block.

827 /wsrm:Sequence

828 This protocol element associates the message in which it is contained with a previously established RM
829 Sequence. It contains the Sequence's unique identifier and the containing message's ordinal position
830 within that Sequence. The RM Destination MUST understand the *Sequence* header block. The RM
831 Source MUST assign a *mustUnderstand* attribute with a value 1/true (from the namespace
832 corresponding to the version of SOAP to which the *Sequence* SOAP header block is bound) to the
833 *Sequence* header block element.

834 /wsrm:Sequence/wsrm:Identifier

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

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

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

841 /wsrm:Sequence/wsrm:MessageNumber

842 The RM Source MUST include this element within any Sequence headers it creates. This element is of
843 type *MessageNumberType*. It represents the ordinal position of the message within a Sequence.
844 Sequence message numbers start at 1 and monotonically increase by 1 throughout the Sequence. See
845 Section 4.5 for Message Number Rollover fault.

846 /wsrm:Sequence/{any}

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

849 /wsrm:Sequence/@{any}

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

852 The following example illustrates a *Sequence* header block.

```
853 <wsrm:Sequence>
854   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
855   <wsrm:MessageNumber>10</wsrm:MessageNumber>
856 </wsrm:Sequence>
```

857 3.8 Request Acknowledgement

858 The purpose of the `AckRequested` header block is to signal to the RM Destination that the RM Source is
859 requesting that a `SequenceAcknowledgement` be sent.

860 The RM Source MAY request an Acknowledgement Message from the RM Destination at any time by
861 independently transmitting an `AckRequested` header block (i.e. as a header of a SOAP envelope with an
862 empty body). Alternatively the RM Source MAY include an `AckRequested` header block in any message
863 targeted to the RM Destination. The RM Destination SHOULD process `AckRequested` header blocks
864 that are included in any message it receives. If a non-mustUnderstand fault occurs when processing an
865 `AckRequested` header block that was piggy-backed, a fault MUST be generated, but the processing of
866 the original message MUST NOT be affected, including an `AckRequested` header block in any message
867 targeted to the RM Destination. An RM Destination that Receives a message that contains an
868 `AckRequested` header block MUST send a message containing a `SequenceAcknowledgement`
869 header block to the `AcksTo` endpoint reference (see Section 3.1) for a known Sequence or else generate
870 an `UnknownSequence` fault. If a non-mustUnderstand fault occurs when processing an RM header that
871 was piggy-backed on another message, a fault MUST be generated, but the processing of the original
872 message MUST NOT be affected. It is RECOMMENDED that the RM Destination return a
873 `AcknowledgementRange` or `None` element instead of a `Nack` element (see Section 3.6).

874 An RM Destination that Receives a message that contains an `AckRequested` header block MUST send
875 a message containing a `SequenceAcknowledgement` header block to the `AcksTo` endpoint reference
876 (see Section 3.4) for a known Sequence or else generate an `UnknownSequence` fault. It is
877 RECOMMENDED that the RM Destination return a `AcknowledgementRange` or `None` element instead
878 of a `Nack` element (see Section 3.9).

879 The following exemplar defines its syntax:

```
880 <wsrm:AckRequested ...>
881   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
882   ...
883 </wsrm:AckRequested>
```

884 The following describes the content model of the `AckRequested` header block.

885 `/wsrm:AckRequested`

886 This element requests an Acknowledgement for the identified Sequence.

887 `/wsrm:AckRequested/wsrm:Identifier`

888 An RM Source that includes an `AckRequested` header block in a SOAP envelope MUST include this
889 element in that header block. The RM Source MUST set the value of this element to the absolute URI,
890 (conformant with RFC3986), that uniquely identifies the Sequence to which the request applies.

891 `/wsrm:AckRequested/wsrm:Identifier/@{any}`

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

894 `/wsrm:AckRequested/{any}`

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

897 /wsmr:AckRequested/@{any}

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

900 3.9 Sequence Acknowledgement

901 The RM Destination informs the RM Source of successful message receipt using a
902 SequenceAcknowledgement header block. ~~Acknowledgements can be explicitly requested using the~~
903 ~~AckRequested directive (see Section 3.8)The RM Destination MAY Transmit the~~
904 ~~SequenceAcknowledgement header block independently or it MAY include the~~
905 ~~SequenceAcknowledgement header block on any message targeted to the AcksTo EPR.~~
906 ~~Acknowledgements can be explicitly requested using the AckRequested directive (see Section 3.5). If a~~
907 ~~non-mustUnderstand fault occurs when processing an RM header that was piggy-backed on another~~
908 ~~message, a fault MUST be generated, but the processing of the original message MUST NOT be~~
909 ~~affected.~~

910 ~~The RM Destination MAY Transmit the SequenceAcknowledgement header block independently (i.e.~~
911 ~~As a header of a SOAP envelope with an empty body). Alternatively, an RM Destination MAY include a~~
912 ~~SequenceAcknowledgement header block on any SOAP envelope targeted to the endpoint referenced~~
913 ~~by the AcksTo EPR. The RM Source SHOULD process SequenceAcknowledgement header blocks~~
914 ~~that are included in any message it receives. If a non-mustUnderstand fault occurs when processing a~~
915 ~~SequenceAcknowledgement header that was piggy-backed, a fault MUST be generated, but the~~
916 ~~processing of the original message MUST NOT be affectedA RM Destination MAY include a~~
917 ~~SequenceAcknowledgement header block on any SOAP envelope targetted to the endpoint referenced~~
918 ~~by the AcksTo EPR.~~

919 During creation of a Sequence the RM Source MAY specify the WS-Addressing anonymous IRI as the
920 address of the AcksTo EPR for that Sequence. When the RM Source specifies the WS-Addressing
921 anonymous IRI as the address of the AcksTo EPR, the RM Destination MUST Transmit any
922 SequenceAcknowledgement headers for the created Sequence in a SOAP envelope to be Transmitted
923 on the protocol binding-specific back-channel. Such a channel is provided by the context of a Received
924 message containing a SOAP envelope that contains a Sequence header block and/or an AckRequested
925 header block for that same Sequence identifier. When the RM Destination receives an AckRequested
926 header, and the AckTo EPR for that sequence is the WS-Addressing anonymous IRI, the RM Destination
927 SHOULD respond on the protocol binding-specific back-channel provided by the Received message
928 containing the AckRequested header blockchannel. Such a channel is provided by the context of a
929 Received message containing a SOAP envelope that contains a Sequence header block and/or a
930 AckRequested header block for that same Sequence identifier.

931 The following exemplar defines its syntax:

```
932 <wsmr:SequenceAcknowledgement ...>
933   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>
934   [ [ [ <wsmr:AcknowledgementRange ...
935         Upper="wsmr:MessageNumberType"
936         Lower="wsmr:MessageNumberType"/> +
937         | <wsmr:None/> ]
938     <wsmr:Final/> ? ]
939   | <wsmr:Nack> wsmr:MessageNumberType </wsmr:Nack> + ]
940
```

941
942

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

943 The following describes the content model of the `SequenceAcknowledgement` header block.

944 `/wsrm:SequenceAcknowledgement`

945 This element contains the Sequence Acknowledgement information.

946 `/wsrm:SequenceAcknowledgement/wsrm:Identifier`

947 An RM Destination that includes a `SequenceAcknowledgement` header block in a SOAP envelope
948 MUST include this element in that header block. The RM Destination MUST set the value of this element
949 to the absolute URI (conformant with RFC3986) that uniquely identifies the Sequence. The RM
950 Destination MUST NOT include multiple `SequenceAcknowledgement` header blocks that share the
951 same value for `Identifier` within the same SOAP envelope.

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

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

955 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

956 The RM Destination MAY include one or more instances of this element within a
957 `SequenceAcknowledgement` header block. It contains a range of Sequence message numbers
958 successfully accepted by the RM Destination. The ranges MUST MessageNumbers-successfully-accepted-
959 by the RM Destination. The ranges SHOULD NOT overlap. The RM Destination MUST NOT include this
960 element if a sibling `Nack` or `None` element is also present as a child of `SequenceAcknowledgement`.

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

962 The RM Destination MUST set the value of this attribute equal to the message number of the highest
963 contiguous message in a Sequence range accepted by the RM Destination.

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

965 The RM Destination MUST set the value of this attribute equal to the message number of the lowest
966 contiguous message in a Sequence range accepted by the RM Destination.

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

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

970 `/wsrm:SequenceAcknowledgement/wsrm:None`

971 The RM Destination MUST include this element within a `SequenceAcknowledgement` header block if
972 the RM Destination has not accepted any messages for the specified Sequence. The RM Destination
973 MUST NOT include this element if a sibling `AcknowledgementRange` or `Nack` element is also present
974 as a child of the `SequenceAcknowledgement`.

975 `/wsrm:SequenceAcknowledgement/wsrm:Final`

976 The RM Destination MAY include this element within a `SequenceAcknowledgement` header block. This
977 element indicates that the RM Destination is not receiving new messages for the specified Sequence. The
978 RM Source can be assured that the ranges of messages acknowledged by this
979 `SequenceAcknowledgement` header block will not change in the future. The RM Destination MUST
980 include this element when the Sequence is closed. The RM Destination MUST NOT include this element
981 when sending a `Nack`; it can only be used when sending `AcknowledgementRange` elements or a `None`.

982 /wsmr:SequenceAcknowledgement/wsmr:Nack

983 The RM Destination MAY include this element within a SequenceAcknowledgement header block. If
984 used, the RM Destination MUST set the value of this element to a MessageNumberType representing
985 the MessageNumber of an unreceived message in a Sequence. The RM Destination MUST NOT include
986 a Nack element if a sibling AcknowledgementRange or None element is also present as a child of
987 SequenceAcknowledgement. Upon the receipt of a Nack, an RM Source SHOULD retransmit the
988 message identified by the Nack. The RM Destination MUST NOT issue a SequenceAcknowledgement
989 containing a Nack for a message that it has previously acknowledged within an
990 AcknowledgementRange. The RM Source SHOULD ignore a SequenceAcknowledgement containing
991 a Nack for a message that has previously been acknowledged within an AcknowledgementRange. The
992 RM Source SHOULD ignore a SequenceAcknowledgement containing a Nack for a message that has
993 previously been acknowledged within a AcknowledgementRange.

994 /wsmr:SequenceAcknowledgement/{any}

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

997 /wsmr:SequenceAcknowledgement/@{any}

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

1000 The following examples illustrate SequenceAcknowledgement elements:

- 1001 • Message numbers 1..10 inclusive in a Sequence have been accepted by the RM Destination.

```
1002 <wsmr:SequenceAcknowledgement>  
1003   <wsmr:Identifier>http://example.com/abc</wsmr:Identifier>  
1004   <wsmr:AcknowledgementRange Upper="10" Lower="1"/>  
1005 </wsmr:SequenceAcknowledgement>
```

- 1006 • Message numbers 1..2, 4..6, and 8..10 inclusive in a Sequence have been accepted by the RM
1007 Destination, messages 3 and 7 have not been accepted.

```
1008 <wsmr:SequenceAcknowledgement>  
1009   <wsmr:Identifier>http://example.com/abc</wsmr:Identifier>  
1010   <wsmr:AcknowledgementRange Upper="2" Lower="1"/>  
1011   <wsmr:AcknowledgementRange Upper="6" Lower="4"/>  
1012   <wsmr:AcknowledgementRange Upper="10" Lower="8"/>  
1013 </wsmr:SequenceAcknowledgement>
```

- 1014 • Message number 3 in a Sequence has not been accepted by the RM Destination.

```
1015 <wsmr:SequenceAcknowledgement>  
1016   <wsmr:Identifier>http://example.com/abc</wsmr:Identifier>  
1017   <wsmr:Nack>3</wsmr:Nack>  
1018 </wsmr:SequenceAcknowledgement>
```

1019 1.2 MakeConnection

1020 ~~When an Endpoint is not directly addressable (e.g. behind a firewall or not able to allow incoming~~
1021 ~~connections), an anonymous URI in the EPR address property can indicate such an Endpoint. The WS-~~
1022 ~~Addressing anonymous URI is one such anonymous URI. This specification defines a URI template (the~~
1023 ~~WS-RM anonymous URI) which may be used to uniquely identify anonymous Endpoints.~~

```
1024 http://docs.oasis-open.org/ws-rx/wsmr/200608/anonymous?id={uuid}
```

1025 This URI template in an EPR indicates a protocol-specific back-channel will be established through a
1026 mechanism such as `MakeConnection`, defined below. When using this URI template, “{uuid}” MUST be
1027 replaced by a UUID value as defined by RFC4122[UUID]. This UUID value uniquely distinguishes the
1028 Endpoint. A sending Endpoint SHOULD Transmit messages at Endpoints identified with the URI template
1029 using a protocol-specific back-channel, including but not limited to those established with a
1030 `MakeConnection` message. Note, this URI is semantically similar to the WS-Addressing anonymous-
1031 URI if a protocol-specific back-channel is available.

1032 The `MakeConnection` is a one-way operation that establishes a contextualized back-channel for the
1033 transmission of messages according to matching criteria (defined below). In the non-faulting case, if no
1034 matching message is available then no SOAP envelopes will be returned on the back-channel. A common
1035 usage will be a client RM Destination sending `MakeConnection` to a server RM Source for the purpose
1036 of receiving asynchronous response messages.

1037 The following exemplar defines the `MakeConnection` syntax:

```
1038 <wsm:MakeConnection ...>  
1039   <wsm:Identifier ...> xs:anyURI </wsm:Identifier> ?  
1040   <wsm:Address ...> xs:anyURI </wsm:Address> ?  
1041   ...  
1042 </wsm:MakeConnection>
```

1043 `/wsm:MakeConnection`

1044 This element allows the sender to create a transport-specific back-channel that can be used to return a
1045 message that matches the selection criteria. Endpoints MUST NOT send this element as a header block.

1046 `/wsm:MakeConnection/wsm:Identifier`

1047 This element specifies the WS-RM Sequence Identifier that establishes the context for the transport-
1048 specific back-channel. The Sequence Identifier should be compared with the Sequence Identifiers
1049 associated with the messages held by the sending Endpoint, and if there is a matching message it will be
1050 returned. If this element is omitted from the message then the `Address` MUST be included in the
1051 message.

1052 `/wsm:MakeConnection/wsm:Identifier/@{any}`

1053 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
1054 element.

1055 `/wsm:MakeConnection/wsm:Address`

1056 This element specifies the URI (`wsa:Address`) of the initiating Endpoint. Endpoints MUST NOT return
1057 messages on the transport-specific back-channel unless they have been addressed to this URI. This
1058 `Address` property and a message’s WS-Addressing destination property are considered identical when
1059 they are exactly the same character-for-character. Note that URIs which are not identical in this sense
1060 may in fact be functionally equivalent. Examples include URI references which differ only in case, or
1061 which are in external entities which have different effective base URIs. If this element is omitted from the
1062 message then the `Identifier` MUST be included in the message.

1063 `/wsm:MakeConnection/wsm:Address/@{any}`

1064 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
1065 element.

1066 `/wsm:MakeConnection/{any}`

1067 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
1068 to be passed. This allows fine-tuning of the messages to be returned, additional selection criteria included

1069 ~~here are logically ANDed with the Address and/or Identifier. If an extension is not supported by the~~
1070 ~~Endpoint then it should return a UnsupportedSelection fault.~~

1071 ~~/wsrm:MakeConnection/@{any}~~

1072 ~~This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the~~
1073 ~~element.~~

1074 ~~If both Identifier and Address are present, then the Endpoint processing the MakeConnection~~
1075 ~~message MUST insure that any SOAP Envelope flowing on the backchannel MUST be associated with~~
1076 ~~the given Sequence and MUST be addressed to the given URL.~~

1077 ~~The management of messages that are awaiting the establishment of a back-channel to their receiving~~
1078 ~~Endpoint is an implementation detail that is outside the scope of this specification. Note, however, that~~
1079 ~~these messages form a class of asynchronous messages that is not dissimilar from "ordinary"~~
1080 ~~asynchronous messages that are waiting for the establishment of a connection to their destination~~
1081 ~~Endpoints.~~

1082 ~~This specification places no constraint on the types of messages that can be returned on the transport-~~
1083 ~~specific back-channel. As in an asynchronous environment, it is up to the recipient of the~~
1084 ~~MakeConnection message to decide which messages are appropriate for transmission to any particular~~
1085 ~~Endpoint. However, the Endpoint processing the MakeConnection message MUST insure that the~~
1086 ~~messages match the selection criteria as specified by the child elements of the MakeConnection~~
1087 ~~element.~~

1088 **1.3 MessagePending**

1089 ~~When MakeConnection is used, and a message is returned on the transport-specific back-channel, the~~
1090 ~~MessagePending header SHOULD be included on the returned message as an indicator whether there~~
1091 ~~are additional messages waiting to be retrieved using the same selection criteria that was specified in the~~
1092 ~~MakeConnection element.~~

1093 ~~The following exemplar defines the MessagePending syntax:~~

```
1094 <wsrm:MessagePending pending="xs:boolean" ...>  
1095 ...  
1096 </wsrm:MessagePending>
```

1097 ~~/wsrm:MessagePending~~

1098 ~~This element indicates whether additional messages are waiting to be retrieved.~~

1099 ~~/wsrm:MessagePending@pending~~

1100 ~~This attribute, when set to "true", indicates that there is at least one message waiting to be retrieved.~~
1101 ~~When this attribute is set to "false" it indicates there are currently no messages waiting to be retrieved.~~

1102 ~~/wsrm:MessagePending/{any}~~

1103 ~~This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,~~
1104 ~~to be passed.~~

1105 ~~/wsrm:MessagePending/@{any}~~

1106 ~~This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the~~
1107 ~~element.~~

1108 ~~The absence of the MessagePending header has no implication as to whether there are additional~~
1109 ~~messages waiting to be retrieved.~~

1110 4 Faults

1111 Faults for the `CreateSequence` message exchange are treated as defined in WS-Addressing. `Create`
1112 `Sequence Refused` is a possible fault reply for this operation. `Unknown Sequence` is a fault generated by
1113 Endpoints when messages carrying RM header blocks targeted at unrecognized or terminated Sequences
1114 are detected. ~~WSRMRequired is a fault generated an RM Destination that requires the use of WS-RM on~~
1115 ~~a Received message that did not use the protocol. All other faults in this section relate to known~~
1116 ~~Sequences. Destinations that generate faults related to known sequences SHOULD transmit those faults.~~
1117 ~~If transmitted, such faults MUST be transmitted to the same [destination] as Acknowledgement m~~
1118 ~~Required is a fault generated an RM Destination that requires the use of WS-RM on a Received message-~~
1119 ~~that did not use the protocol. All other faults in this section relate to known Sequences. RM Destinations-~~
1120 ~~that generate Sequence faults SHOULD send those faults to the same [destination] as Acknowledgement~~
1121 ~~Messages.~~

1122 Entities that generate WS-ReliableMessaging faults MUST include as the [action] property the default fault
1123 action IRI defined below. The value from the W3C Recommendation is below for informational purposes:

```
1124 http://docs.oasis-open.org/ws-rx/wsrn/200702698/fault
```

1125 The faults defined in this section are generated if the condition stated in the preamble is met. Fault
1126 handling rules are defined in section 6 of WS-Addressing SOAP Binding.

1127 The definitions of faults use the following properties:

1128 [Code] The fault code.

1129 [Subcode] The fault subcode.

1130 [Reason] The English language reason element.

1131 [Detail] The detail element(s). If absent, no detail element is defined for the fault. If more than one detail
1132 element is defined for a fault, implementations MUST include the elements in the order that they are
1133 specified.

1134 Entities that generate WS-ReliableMessaging faults MUST set the [Code] property to either "Sender" or
1135 "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

1136 The properties above bind to a SOAP 1.2 fault as follows:

```
1137 <S:Envelope>  
1138   <S:Header>  
1139     <wsa:Action>  
1140       http://docs.oasis-open.org/ws-rx/wsrn/200702698/fault  
1141     </wsa:Action>  
1142     <!-- Headers elided for brevity. -->  
1143   </S:Header>  
1144   <S:Body>  
1145     <S:Fault>  
1146       <S:Code>  
1147         <S:Value> [Code] </S:Value>  
1148         <S:Subcode>  
1149           <S:Value> [Subcode] </S:Value>  
1150         </S:Subcode>  
1151       </S:Code>
```

```

1152     <S:Reason>
1153         <S:Text xml:lang="en"> [Reason] </S:Text>
1154     </S:Reason>
1155     <S:Detail>
1156         [Detail]
1157         ...
1158     </S:Detail>
1159 </S:Fault>
1160 </S:Body>
1161 </S:Envelope>

```

1162 The properties above bind to a SOAP 1.1 fault as follows when the fault is triggered by processing an RM
1163 header block:

```

1164 <S11:Envelope>
1165 <S11:Header>
1166     <wsrm:SequenceFault>
1167         <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
1168         <wsrm:Detail> [Detail] </wsrm:Detail>
1169         ...
1170     </wsrm:SequenceFault>
1171     <!-- Headers elided for brevity. -->
1172 </S11:Header>
1173 <S11:Body>
1174     <S11:Fault>
1175         <faultcode> [Code] </faultcode>
1176         <faultstring> [Reason] </faultstring>
1177     </S11:Fault>
1178 </S11:Body>
1179 </S11:Envelope>

```

1180 The properties bind to a SOAP 1.1 fault as follows when the fault is generated as a result of processing a
1181 CreateSequence request message:

```

1182 <S11:Envelope>
1183 <S11:Body>
1184     <S11:Fault>
1185         <faultcode> [Subcode] </faultcode>
1186         <faultstring> [Reason] </faultstring>
1187     </S11:Fault>
1188 </S11:Body>
1189 </S11:Envelope>

```

1190 4.1 SequenceFault Element

1191 The purpose of the `SequenceFault` element is to carry the specific details of a fault generated during
1192 the reliable messaging specific processing of a message belonging to a Sequence. WS-
1193 ReliableMessaging nodes MUST use the `SequenceFault` container only in conjunction with the SOAP
1194 1.1 fault mechanism. WS-ReliableMessaging nodes MUST NOT use the `SequenceFault` container in
1195 conjunction with the SOAP 1.2 binding.

1196 The following exemplar defines its syntax:

```

1197 <wsrm:SequenceFault ...>
1198     <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
1199     <wsrm:Detail> ... </wsrm:Detail> ?
1200     ...
1201 </wsrm:SequenceFault>

```

1202 The following describes the content model of the `SequenceFault` element.

1203 /wsm:SequenceFault
 1204 This is the element containing Sequence information for WS-ReliableMessaging
 1205 /wsm:SequenceFault/wsm:FaultCode
 1206 WS-ReliableMessaging nodes that generate a `SequenceFault` MUST set the value of this element to a
 1207 qualified name from the set of fault [Subcodes] defined below.
 1208 /wsm:SequenceFault/wsm:Detail
 1209 This element, if present, carries application specific error information related to the fault being described.
 1210 /wsm:SequenceFault/wsm:Detail/{any}
 1211 The application specific error information related to the fault being described.
 1212 /wsm:SequenceFault/wsm:Detail/@{any}
 1213 The application specific error information related to the fault being described.
 1214 /wsm:SequenceFault/{any}
 1215 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
 1216 to be passed.
 1217 /wsm:SequenceFault/@{any}
 1218 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
 1219 element.

1220 4.2 Sequence Terminated

1221 The Endpoint that generates this fault SHOULD make every reasonable effort to notify the corresponding
 1222 Endpoint of this decision.
 1223 Properties:
 1224 [Code] Sender or Receiver
 1225 [Subcode] wsm:SequenceTerminated
 1226 [Reason] The Sequence has been terminated due to an unrecoverable error.
 1227 [Detail]

1228 `<wsm:Identifier ...> xs:anyURI </wsm:Identifier>`

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

1229 4.3 Unknown Sequence

1230 Properties:

- 1231 [Code] Sender
 1232 [Subcode] wsrn:UnknownSequence
 1233 [Reason] The value of wsrn:Identifier is not a known Sequence identifier.
 1234 [Detail]

1235 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`

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

1236 4.4 Invalid Acknowledgement

1237 An example of when this fault is generated is when a message is Received by the RM Source containing
 1238 a SequenceAcknowledgement covering messages that have not been sent.

- 1239 [Code] Sender
 1240 [Subcode] wsrn:InvalidAcknowledgement
 1241 [Reason] The SequenceAcknowledgement violates the cumulative Acknowledgement invariant.
 1242 [Detail]

1243 `<wsrm:SequenceAcknowledgement ...> ... </wsrm:SequenceAcknowledgement>`

Generated by	Condition	Action Upon Generation	Action Upon Receipt
RM Source.	In response to a SequenceAcknowledgement that violate the invariants stated in 2.3 or any of the requirements in 3.9 about valid combinations of AckRange, Nack and None in a single SequenceAcknowledgement element or with respect to already Received such elements.	Unspecified.	Unspecified.

1244 4.5 Message Number Rollover

1245 If the condition listed below is reached, the RM Destination MUST generate this fault.

- 1246 Properties:
 1247 [Code] Sender

- 1248 [Subcode] wsrn:MessageNumberRollover
- 1249 [Reason] The maximum value for wsrn:MessageNumber has been exceeded.
- 1250 [Detail]

```
1251 <wsrn:Identifier ...> xs:anyURI </wsrn:Identifier>
1252 <wsrn:MaxMessageNumber> wsrn:MessageNumberType </wsrn:MaxMessageNumber>
```

Generated by	Condition	Action Upon Generation	Action Upon Receipt
RM Destination.	Message number in /wsrn:Sequence/wsrn:MessageNumber of a Received message exceeds the internal limitations of an RM Destination or reaches the maximum value of 9,223,372,036,854,775,807.	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.

1253 4.6 Create Sequence Refused

- 1254 Properties:
- 1255 [Code] Sender or Receiver
- 1256 [Subcode] wsrn:CreateSequenceRefused
- 1257 [Reason] The Create Sequence request has been refused by the RM Destination.
- 1258 [Detail]

```
1259 xs:any
```

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

1260 4.7 Sequence Closed

- 1261 This fault is generated by an RM Destination to indicate that the specified Sequence has been closed.
- 1262 This fault MUST be generated when an RM Destination is asked to accept a message for a Sequence that
- 1263 is closed or when an RM Destination is asked to close a Sequence that is already closed.
- 1264 Properties:
- 1265 [Code] Sender

1266 [Subcode] wsm:SequenceClosed

1267 [Reason] The Sequence is closed and can-not accept new messages.

1268 [Detail]

1269 `<wsm:Identifier...> xs:anyURI </wsm:Identifier>`

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

1270 4.8 WSRM Required

1271 If an RM Destination requires the use of WS-RM, this fault is generated when it Receives an incoming
1272 message that did not use this protocol.

1273 Properties:

1274 [Code] Sender

1275 [Subcode] wsm:WSRMRequired

1276 [Reason] The RM Destination requires the use of WSRM.

1277 [Detail]

1278 `xs:any`

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

1279 1.4 Unsupported Selection

1280 ~~The QName of the unsupported element(s) are included in the detail.~~

1281 ~~Properties:-~~

1282 ~~[Code] Receiver~~

1283 ~~[Subcode] wsm:UnsupportedSelection~~

1284 ~~[Reason] The extension element used in the message selection is not supported by the RM Source~~

1285 ~~[Detail]~~

1286 ~~`<wsm:UnsupportedElement> xs:QName </wsm:UnsupportedElement>`~~

Generated-by	Condition	Action-Upon-Generation	Action-Upon-Receipt
RM-Source-or-RM-Destination:	In response to a MakeConnection message containing a selection criteria in the extensibility section of the message that is not support.ed	Unspecified:	Unspecified:

1287 **5 Security Threats and Countermeasures**

1288 This specification considers two sets of security requirements, those of the applications that use the WS-
1289 RM protocol and those of the protocol itself.

1290 This specification makes no assumptions about the security requirements of the applications that use WS-
1291 RM. However, once those requirements have been satisfied within a given operational context, the
1292 addition of WS-RM to this operational context should not undermine the fulfillment of those requirements;
1293 the use of WS-RM should not create additional attack vectors within an otherwise secure system.

1294 There are many other security concerns that one may need to consider when implementing or using this
1295 protocol. The material below should not be considered as a "check list". Implementers and users of this
1296 protocol are urged to perform a security analysis to determine their particular threat profile and the
1297 appropriate responses to those threats.

1298 Implementers are also advised that there is a core tension between security and reliable messaging that
1299 can be problematic if not addressed by implementations; one aspect of security is to prevent message
1300 replay but one of the invariants of this protocol is to resend messages until they are acknowledged.
1301 Consequently, if the security sub-system processes a message but a failure occurs before the reliable
1302 messaging sub-system Receives that message, then it is possible (and likely) that the security sub-system
1303 will treat subsequent copies as replays and discard them. At the same time, the reliable messaging sub-
1304 system will likely continue to expect and even solicit the missing message(s). Care should be taken to
1305 avoid and prevent this condition.

1306 **5.1 Threats and Countermeasures**

1307 The primary security requirement of this protocol is to protect the specified semantics and protocol
1308 invariants against various threats. The following sections describe several threats to the integrity and
1309 operation of this protocol and provide some general outlines of countermeasures to those threats.
1310 Implementers and users of this protocol should keep in mind that all threats are not necessarily applicable
1311 to all operational contexts.

1312 **5.1.1 Integrity Threats**

1313 In general, any mechanism which allows an attacker to alter the information in a Sequence Traffic
1314 Message, Sequence Lifecycle Message, Acknowledgement Messages, Acknowledgement Request, or
1315 Sequence-related fault, or which allows an attacker to alter the correlation of a RM Protocol Header Block
1316 to its intended message represents a threat to the WS-RM protocol.

1317 For example, if an attacker is able to swap `Sequence` headers on messages in transit between the RM
1318 Source and RM Destination then they have undermined the implementation's ability to guarantee the first
1319 invariant described in Section 2.3. The result is that there is no way of guaranteeing that messages will be
1320 Delivered to the Application Destination in the same order that they were sent by the Application Source.

1321 **5.1.1.1 Countermeasures**

1322 Integrity threats are generally countered via the use of digital signatures some level of the communication
1323 protocol stack. Note that, in order to counter header swapping attacks, the signature SHOULD include
1324 both the SOAP body and any relevant SOAP headers (e.g. `Sequence` header). Because some headers
1325 (`AckRequested`, `SequenceAcknowledgement`) are independent of the body of the SOAP message in which
1326 they occur, implementations MUST allow for signatures that cover only these headers.

1327 **5.1.2 Resource Consumption Threats**

1328 The creation of a Sequence with an RM Destination consumes various resources on the systems used to
1329 implement that RM Destination. These resources can include network connections, database tables,
1330 message queues, etc. This behavior can be exploited to conduct denial of service attacks against an RM
1331 Destination. For example, a simple attack is to repeatedly send `CreateSequence` messages to an RM
1332 Destination. Another attack is to create a Sequence for a service that is known to require in-order
1333 message Delivery and use this Sequence to send a stream of very large messages to that service,
1334 making sure to omit message number “1” from that stream.

1335 **5.1.2.1 Countermeasures**

1336 There are a number of countermeasures against the described resource consumption threats. The
1337 technique advocated by this specification is for the RM Destination to restrict the ability to create a
1338 Sequence to a specific set of entities/principals. This reduces the number of potential attackers and, in
1339 some cases, allows the identity of any attackers to be determined.

1340 The ability to restrict Sequence creation depends, in turn, upon the RM Destination's ability identify and
1341 authenticate the RM Source that issued the `CreateSequence` message.

1342 **5.1.3 Sequence Spoofing Threats**

1343 Sequence spoofing is a class of threats in which the attacker uses knowledge of the `Identifier` for a
1344 particular Sequence to forge Sequence Lifecycle or Traffic Messages. For example the attacker creates a
1345 fake `TerminateSequence` message that references the target Sequence and sends this message to the
1346 appropriate RM Destination. Some sequence spoofing attacks also require up-to-date knowledge of the
1347 current `MessageNumber` for their target Sequence.

1348 In general any Sequence Lifecycle Message, RM Protocol Header Block, or sequence-correlated SOAP
1349 fault (e.g. `InvalidAcknowledgement`) can be used by someone with knowledge of the Sequence identifier
1350 to attack the Sequence. These attacks are “two-way” in that an attacker may choose to target the RM
1351 Source by, for example, inserting a fake `SequenceAcknowledgement` header into a message that it sends
1352 to the `AcksTo` EPR of an RM Source.

1353 **5.1.3.1 Sequence Hijacking**

1354 Sequence hijacking is a specific case of a sequence spoofing attack. The attacker attempts to inject
1355 Sequence Traffic Messages into an existing Sequence by inserting fake `Sequence` headers into those
1356 messages.

1357 Note that “sequence hijacking” should not be equated with “security session hijacking”. Although a
1358 Sequence may be bound to some form of a security session in order to counter the threats described in
1359 this section, applications MUST NOT rely on WS-RM-related information to make determinations about
1360 the identity of the entity that created a message; applications SHOULD rely only upon information that is
1361 established by the security infrastructure to make such determinations. Failure to observe this rule
1362 creates, among other problems, a situation in which the absence of WS-RM may deprive an application of
1363 the ability to authenticate its peers even though the necessary security processing has taken place.

1364 **5.1.3.2 Countermeasures**

1365 There are a number of countermeasures against sequence spoofing threats. The technique advocated by
1366 this specification is to consider the Sequence to be a shared resource that is jointly owned by the RM

1367 Source that initiated its creation (i.e. that sent the `CreateSequence` message) and the RM Destination that
1368 serves as its terminus (i.e. that sent the `CreateSequenceResponse` message). To counter sequence
1369 spoofing attempts the RM Destination SHOULD ensure that every message or fault that it Receives that
1370 refers to a particular Sequence originated from the RM Source that jointly owns the referenced Sequence.
1371 For its part the RM Source SHOULD ensure that every message or fault that it Receives that refers to a
1372 particular Sequence originated from the RM Destination that jointly owns the referenced Sequence.

1373 For the RM Destination to be able to identify its sequence peer it MUST be able to identify and
1374 authenticate the entity that sent the `CreateSequence` message. Similarly for the RM Source to identify its
1375 sequence peer it MUST be able to identify and authenticate the entity that sent the
1376 `CreateSequenceResponse` message. For either the RM Destination or the RM Source to determine if a
1377 message was sent by its sequence peer it MUST be able to identify and authenticate the initiator of that
1378 message and, if necessary, correlate this identity with the sequence peer identity established at sequence
1379 creation time.

1380 **5.2 Security Solutions and Technologies**

1381 The security threats described in the previous sections are neither new nor unique. The solutions that
1382 have been developed to secure other SOAP-based protocols can be used to secure WS-RM as well. This
1383 section maps the facilities provided by common web services security solutions against countermeasures
1384 described in the previous sections.

1385 Before continuing this discussion, however, some examination of the underlying requirements of the
1386 previously described countermeasures is necessary. Specifically it should be noted that the technique
1387 described in Section 5.1.2.1 has two components. Firstly, the RM Destination identifies and authenticates
1388 the issuer of a `CreateSequence` message. Secondly, the RM Destination ~~to~~ performs an authorization
1389 check against this authenticated identity and determines if the RM Source is permitted to create
1390 Sequences with the RM Destination. Since the facilities for performing this authorization check (runtime
1391 infrastructure, policy frameworks, etc.) lie completely within the domain of individual implementations, any
1392 discussion of such facilities is considered to be beyond the scope of this specification.

1393 **5.2.1 Transport Layer Security**

1394 This section describes how the ~~the~~ facilities provided by SSL/TLS [RFC 4346] can be used to implement
1395 the countermeasures described in the previous sections. The use of SSL/TLS is subject to the constraints
1396 defined in Section 4 of the Basic Security Profile 1.0 [BSP 1.0].

1397 The description provided here is general in nature and is not intended to serve as a complete definition on
1398 the use of SSL/TLS to protect WS-RM. In order to interoperate implementations need to agree on the
1399 choice of features as well as the manner in which they will be used. The mechanisms described in the
1400 Web Services Security Policy Language [SecurityPolicy] MAY be used by services to describe the
1401 requirements and constraints of the use of SSL/TLS.

1402 **5.2.1.1 Model**

1403 The basic model for using SSL/TLS is as follows:

- 1404 1. The RM Source establishes an SSL/TLS session with the RM Destination.
- 1405 2. The RM Source uses this SSL/TLS session to send a `CreateSequence` message to the RM
1406 Destination.

- 1407 3. The RM Destination establishes an SSL/TLS session with the RM Source and sends an
1408 asynchronous `CreateSequenceResponse` using this session. Alternately it may respond with a
1409 synchronous `CreateSequenceResponse` using the session established in (1).
- 1410 4. For the lifetime of the Sequence the RM Source uses the SSL/TLS session from (1) to Transmit
1411 any and all messages or faults that refer to that Sequence.
- 1412 5. For the lifetime of the Sequence the RM Destination either uses the SSL/TLS session established
1413 in (3) to Transmit any and all messages or faults that refer to that Sequence or, for synchronous
1414 exchanges, the RM Destination uses the SSL/TLS session established in (1).

1415 5.2.1.2 Countermeasure Implementation

1416 Used in its simplest fashion (without relying upon any authentication mechanisms), SSL/TLS provides the
1417 necessary integrity qualities to counter the threats described in Section 5.1.1. Note, however, that the
1418 nature of SSL/TLS limits the scope of this integrity protection to a single transport level session. If
1419 SSL/TLS is the only mechanism used to provide integrity, any intermediaries between the RM Source and
1420 the RM Destination MUST be trusted to preserve the integrity of the messages that flow through them.

1421 As noted, the technique described in Sections 5.1.2.1 involves the use of authentication. This specification
1422 advocates either of two mechanisms for authenticating entities using SSL/TLS. In both of these methods
1423 the SSL/TLS server (the party accepting the SSL/TLS connection) authenticates itself to the SSL/TLS
1424 client using an X.509 certificate that is exchanged during the SSL/TLS handshake.

- 1425 • **HTTP Basic Authentication:** This method of authentication presupposes that a SOAP/HTTP
1426 binding is being used as part of the protocol stack beneath WS-RM. Subsequent to the
1427 establishment of the ~~the~~ SSL/TLS session, the sending party authenticates itself to the receiving
1428 party using HTTP Basic Authentication [RFC 2617]. For example, a RM Source might
1429 authenticate itself to a RM Destination (e.g. when transmitting a Sequence Traffic Message) using
1430 BasicAuth. Similarly the RM Destination might authenticate itself to the RM Source (e.g. when
1431 sending an Acknowledgement) using BasicAuth.
- 1432 • **SSL/TLS Client Authentication:** In this method of authentication, the party initiating the
1433 connection authenticates itself to the party accepting the connection using an X.509 certificate
1434 that is exchanged during the SSL/TLS handshake.

1435 To implement the countermeasures described in section 5.1.2.1 the RM Source must authenticate itself
1436 using one the above mechanisms. The authenticated identity can then be used to determine if the RM
1437 Source is authorized to create a Sequence with the RM Destination.

1438 This specification advocates implementing the countermeasures described in section 5.1.3.2 by requiring
1439 an RM node's Sequence peer to be equivalent to their SSL/TLS session peer. This allows the
1440 authorization decisions described in section 5.1.3.2 to be based on SSL/TLS session identity rather than
1441 on authentication information. For example, an RM Destination can determine that a Sequence Traffic
1442 Message rightfully belongs to its referenced Sequence if that message arrived over the same SSL/TLS
1443 session that was used to carry the `CreateSequence` message for that Sequence. Note that requiring a
1444 one-to-one relationship between SSL/TLS session peer and Sequence peer constrains the lifetime of a
1445 SSL/TLS-protected Sequence to be less than or equal to the lifetime of the SSL/TLS session that is used
1446 to protect that Sequence.

1447 This specification does not preclude the use of other methods of using SSL/TLS to implement the
1448 countermeasures (such as associating specific authentication information with a Sequence) although such
1449 methods are not covered by this document.

1450 Issues specific to the life-cycle management of SSL/TLS sessions (such as the resumption of a SSL/TLS
1451 session) are outside the scope of this specification.

1452 **5.2.2 SOAP Message Security**

1453 The mechanisms described in WS-Security may be used in various ways to implement the
1454 countermeasures described in the previous sections. This specification advocates using the protocol
1455 described by WS-SecureConversation [SecureConversation] (optionally in conjunction with WS-Trust
1456 [Trust]) as a mechanism for protecting Sequences. The use of WS-Security (as an underlying component
1457 of WS-SecureConversation) is subject to the constraints defined in the Basic Security Profile 1.0.

1458 The description provided here is general in nature and is not intended to serve as a complete definition on
1459 the use of WS-SecureConversation/WS-Trust to protect WS-RM. In order to interoperate implementations
1460 need to agree on the choice of features as well as the manner in which they will be used. The
1461 mechanisms described in the Web Services Security Policy Language MAY be used by services to
1462 describe the requirements and constraints of the use of WS-SecureConversation.

1463 **5.2.2.1 Model**

1464 The basic model for using WS-SecureConversation is as follows:

- 1465 1. The RM Source and the RM Destination create a WS-SecureConversation security context. This
1466 may involve the participation of third parties such as a security token service. The tokens
1467 exchanged may contain authentication claims (e.g. X.509 certificates or Kerberos service tickets).
- 1468 2. During the `CreateSequence` exchange, the RM Source SHOULD explicitly identify the security
1469 context that will be used to protect the Sequence. This is done so that, in cases where the
1470 `CreateSequence` message is signed by more than one security context, the RM Source can
1471 indicate which security context should be used to protect the newly created Sequence.
- 1472 3. For the lifetime of the Sequence the RM Source and the RM Destination use the session key(s)
1473 associated with the security context to sign (as defined by WS-Security) at least the body and any
1474 relevant WS-RM-defined headers of any and all messages or faults that refer to that Sequence.

1475 **5.2.2.2 Countermeasure Implementation**

1476 Without relying upon any authentication information, the per-message signatures provide the necessary
1477 integrity qualities to counter the threats described in Section 5.1.1.

1478 To implement the countermeasures described in section 5.1.2.1 some mutually agreed upon form of
1479 authentication claims must be provided by the RM Source to the RM Destination during the establishment
1480 of the Security Context. These claims can then be used to determine if the RM Source is authorized to
1481 create a Sequence with the RM Destination.

1482 This specification advocates implementing the countermeasures described in section 5.1.3.2 by requiring
1483 an RM node's Sequence peer to be equivalent to their security context session peer. This allows the
1484 authorization decisions described in section 5.1.3.2 to be based on the identity of the message's security
1485 context rather than on any authentication claims that may have been established during security context
1486 initiation. Note that other methods of using WS-SecurityConversation to implement the countermeasures
1487 (such as associating specific authentication claims to a Sequence) are possible but not covered by this
1488 document.

1489 As with transport security, the requisite equivalence of a security context peer and with a Sequence peer
1490 limits the lifetime of a Sequence to the lifetime of the protecting security context. Unlike transport security,

1491 the association between a Sequence and its protecting security context cannot always be established
1492 implicitly at Sequence creation time. This is due to the fact that the `CreateSequence` and
1493 `CreateSequenceResponse` messages may be signed by more than one security context.

1494 Issues specific to the life-cycle management of WS-SecurityConversation security contexts (such as
1495 amending or renewing contexts) are outside the scope of this specification.

1496 6 Securing Sequences

1497 As noted in Section 5, the RM Source and RM Destination should be able to protect their shared
1498 Sequences against the threat of Sequence Spoofing attacks. There are a number of OPTIONAL means of
1499 achieving this objective depending upon the underlying security infrastructure.

1500 6.1 Securing Sequences Using WS-Security

1501 One mechanism for protecting a Sequence is to include a security token using a
1502 `wsse:SecurityTokenReference` element from WS-Security (see section 9 in WS-
1503 SecureConversation) in the `CreateSequence` element. This establishes an association between the
1504 created (and, if present, offered) Sequence(s) and the referenced security token, such that the RM Source
1505 and Destination MUST use the security token as the basis for authorization of all subsequent interactions
1506 related to the Sequence(s). The `wsse:SecurityTokenReference` explicitly identifies the token as
1507 there may be more than one token on a `CreateSequence` message or inferred from the communication
1508 context (e.g. transport protection).

1509 It is RECOMMENDED that a message independent referencing mechanism be used to identify the token,
1510 if the token being referenced supports such mechanism.

1511 The following exemplar defines the `CreateSequence` syntax when extended to include a
1512 `wsse:SecurityTokenReference`:

```
1513 <wsrm:CreateSequence ...>  
1514   <wsrm:AcksTo> wsa:EndpointReferenceType </wsrm:AcksTo>  
1515   <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
1516   <wsrm:Offer ...>  
1517     <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
1518     <wsrm:Endpoint> wsa:EndpointReferenceType </wsrm:Endpoint>  
1519     <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
1520     <wsrm:IncompleteSequenceBehavior>  
1521       wsrml:IncompleteSequenceBehaviorType  
1522     </wsrm:IncompleteSequenceBehavior> ?  
1523     ...  
1524   </wsrm:Offer> ?  
1525   ...  
1526   <wsse:SecurityTokenReference>  
1527     ...  
1528   </wsse:SecurityTokenReference> ?  
1529   ...  
1530 </wsrm:CreateSequence>
```

1531 The following describes the content model of the additional `CreateSequence` elements.

1532 `/wsrm:CreateSequence/wsse:SecurityTokenReference`

1533 This element uses the extensibility mechanism defined for the `CreateSequence` element (defined in
1534 section 3.41) to communicate an explicit reference to the security token, using a
1535 `wsse:SecurityTokenReference` as documented in WS-Security, that the RM Source and Destination
1536 MUST use to authorize messages for the created (and, if present, the offered) Sequence(s). All
1537 subsequent messages related to the created (and, if present, the offered) Sequence(s) MUST
1538 demonstrate proof-of-possession of the secret associated with the token (e.g., by using or deriving from a
1539 private or secret key).

1540 When a RM Source transmits a `CreateSequence` that has been extended to include a
1541 `wsse:SecurityTokenReference` it SHOULD ensure that the RM Destination both understands and
1542 will conform to transmits a `CreateSequence` that has been extended to include a

1543 ~~wsse:SecurityTokenReference~~ it SHOULD ensure that the RM Destination both understands and
1544 will conform with the requirements listed above. In order to achieve this, the RM Source SHOULD include
1545 the UsesSequenceSTR element as a SOAP header block within the CreateSequence message. This
1546 element MUST include a soap:mustUnderstand attribute with a value of 'true'. Thus the RM Source
1547 can be assured that a RM Destination that responds with a CreateSequenceResponse understands
1548 and conforms with the requirements listed above. Note that an RM Destination understanding this header
1549 does not mean that it has processed and understood any WS-Security headers, the fault behavior defined
1550 in WS-Security still applies.

1551 The following exemplar defines the UsesSequenceSTR syntax:

```
1552 <wsrm:UsesSequenceSTR ... />
```

1553 [The following describes the content model of the UsesSequenceSTR header block.](#)

1554 /wsrm:UsesSequenceSTR

1555 This element SHOULD be included as a SOAP header block in CreateSequence messages that use the
1556 extensibility mechanism described above in this section. The soap:mustUnderstand attribute value
1557 MUST be 'true'. The receiving RM Destination MUST understand and correctly implement the extension
1558 described above or else generate a soap:MustUnderstand fault, thus aborting the requested
1559 Sequence creation.

1560 The following is an example of a CreateSequence message using the

1561 wsse:SecurityTokenReference extension and the UsesSequenceSTR header block:

```
1562 <soap:Envelope ...>  
1563   <soap:Header>  
1564     ...  
1565     <wsrm:UsesSequenceSTR soap:mustUnderstand='true' />  
1566     ...  
1567   </soap:Header>  
1568   <soap:Body>  
1569     <wsrm:CreateSequence>  
1570       <wsrm:AcksTo>  
1571         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>  
1572       </wsrm:AcksTo>  
1573       <wsse:SecurityTokenReference>  
1574         ...  
1575       </wsse:SecurityTokenReference>  
1576     </wsrm:CreateSequence>  
1577   </soap:Body>  
1578 </soap:Envelope>
```

1579 6.2 Securing Sequences Using SSL/TLS

1580 One mechanism for protecting a Sequence is to bind the Sequence to the underlying SSL/TLS session(s).
1581 The RM Source indicates to the RM Destination that a Sequence is to be bound to the underlying
1582 SSL/TLS session(s) via the UsesSequenceSSL header block. If the RM Source wishes to bind a
1583 Sequence to the underlying SSL/TLS sessions(s) it MUST include the UsesSequenceSSL element as a
1584 SOAP header block within the CreateSequence message.

1585 The following exemplar defines the UsesSequenceSSL syntax:

```
1586 <wsrm:UsesSequenceSSL soap:mustUnderstand="true" ... />
```

1587 [The following describes the content model of the UsesSequenceSSL header block.](#)

1588 /wsrm:UsesSequenceSSL

1589 The RM Source MAY include this element as a SOAP header block of a `CreateSequence` message to
1590 indicate to the RM Destination that the resulting Sequence is to be bound to the SSL/TLS session that was
1591 used to carry the `CreateSequence` message. If included, the RM Source MUST mark this header with a
1592 `soap:mustUnderstand` attribute with a value of 'true'. The receiving RM Destination MUST understand
1593 and correctly implement the functionality described in Section 5.2.1 or else generate a
1594 `soap:MustUnderstand` fault, thus aborting the requested Sequence creation.

1595 Note that the use inclusion of the above header by the RM Source implies that all Sequence-related
1596 information (Sequence Lifecycle or Acknowledgment messages or Sequence-related faults) flowing from
1597 the RM Destination to the RM Source will be bound to the SSL/TLS session that is used to carry the
1598 `CreateSequenceResponse` message.

1599 **7 References**

1600 **7.1 Normative**

1601 **[KEYWORDS]**

1602 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, Harvard University,
1603 March 1997-

1604 <http://www.ietf.org/rfc/rfc2119.txt>

1605 **[WS-RM Policy]**

1606 [OASIS WS-RX Technical Committee Draft, "Web Services ReliableMessaging Policy Assertion\(WS-RM
1607 Policy\)" February 2007](#)

1608 <http://docs.oasis-open.org/ws-rx/wsrmp/200702/wsrmp-1.1-spec-cd-05.pdf>

1609 **[SOAP 1.1]**

1610 W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.-

1611 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

1612 **[SOAP 1.2]**

1613 W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging Framework" June 2003.-

1614 <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>

1615 **[URI]**

1616 T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 3986,
1617 MIT/LCS, U.C. Irvine, Xerox Corporation, January 2005.-

1618 <http://ietf.org/rfc/rfc3986>

1619 **[UUID]**

1620 P. Leach, M. Mealling, R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace," RFC 4122,
1621 Microsoft, Refactored Networks - LLC, DataPower Technology Inc, July 2005_

1622 <http://www.ietf.org/rfc/rfc4122.txt>

1623 **[XML]**

1624 W3C Recommendation, "Extensible Markup Language (XML) 1.0 ([Fourth Edition](#)). [September](#)
1625 [2006](#). [Second Edition](#)); [October 2000](#).-

1626 <http://www.w3.org/TR/REC-xml/>

1627 **[XML-ns]**

1628 W3C Recommendation, "Namespaces in XML," 14 January 1999.-

1629 <http://www.w3.org/TR/1999/REC-xml-names-19990114/>

1630 **[XML-Schema Part1]**

1631 W3C Recommendation, "XML Schema Part 1: Structures," [October 2004](#)~~2~~[May 2001](#).-

- 1632 <http://www.w3.org/TR/xmlschema-1/>
- 1633 **[XML-Schema Part2]**
- 1634 W3C Recommendation, "XML Schema Part 2: Datatypes," ~~October 2004~~ ~~May 2004~~.
- 1635 <http://www.w3.org/TR/xmlschema-2/>
- 1636 **[XPath 1.0]**
- 1637 W3C Recommendation, "XML Path Language (XPath) Version 1.0," 16 November 1999.
- 1638 <http://www.w3.org/TR/xpath>
- 1639 **[WSDL 1.1]**
- 1640 W3C Note, "Web Services Description Language (WSDL 1.1)," 15 March 2001.
- 1641 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- 1642 **[WS-Addressing]**
- 1643 W3C Recommendation, "Web Services Addressing 1.0 - Core", May 2006.
- 1644 <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>
- 1645 W3C Recommendation, "Web Services Addressing 1.0 – SOAP Binding", May 2006.
- 1646 <http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/>
- 1647 **7.2 Non-Normative**
- 1648 **[BSP 1.0]**
- 1649 WS-I Working Group Draft. "Basic Security Profile Version 1.0," ~~August~~ ~~March~~ 2006
- 1650 <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>
- 1651 **[RDDL 2.0]**
- 1652 Jonathan Borden, Tim Bray, eds. "Resource Directory Description Language (RDDL) 2.0," January 2004
- 1653 <http://www.openhealth.org/RDDL/20040118/rddl-20040118.html>
- 1654 **[RFC 2617]**
- 1655 J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Loutonen, L. Stewart, "HTTP
- 1656 Authentication: Basic and Digest Access Authentication," June 1999.
- 1657 <http://www.ietf.org/rfc/rfc2617.txt>
- 1658 **[RFC 4346]**
- 1659 T. Dierks, E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," April 2006.
- 1660 <http://www.ietf.org/rfc/rfc4346.txt>
- 1661 **[WS-Policy]**
- 1662 W3C Member Submission, "Web Services Policy Framework (WS-Policy)," April 2006.
- 1663 <http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/>
- 1664 **[WS-PolicyAttachment]**
- 1665 W3C Member Submission, "Web Services Policy Attachment (WS-PolicyAttachment)," April 2006.

1666 <http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/>
1667

1668 **[WS-Security]**
1669 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "OASIS Web Services Security:
1670 SOAP Message Security 1.0 (WS-Security 2004)", OASIS Standard 200401, March 2004.
1671 <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>
1672 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "OASIS Web Services Security:
1673 SOAP Message Security 1.1 (WS-Security 2004)", OASIS Standard 200602, February 2006.
1674 <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>

1675 **[RTTM]**
1676 V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", RFC 1323, May
1677 1992.
1678 <http://www.rfc-editor.org/rfc/rfc1323.txt>

1679 **[SecurityPolicy]**
1680 G. Della-Libra, et. al. "Web Services Security Policy Language (WS-SecurityPolicy)", July 2005
1681 <http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf>

1682 **[SecureConversation]**
1683 S. Anderson, et al, "Web Services Secure Conversation Language (WS-SecureConversation)," February
1684 2005.
1685 <http://schemas.xmlsoap.org/ws/2004/04/sc/>

1686 **[Trust]**
1687 S. Anderson, et al, "Web Services Trust Language (WS-Trust)," February 2005.
1688 <http://schemas.xmlsoap.org/ws/2005/02/trust>

1689 Appendix A. Schema

1690 The normative schema that is defined for WS-ReliableMessaging using [XML-Schema Part1] and [XML-
1691 Schema Part2] is located at:

1692 [http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-schema-200702608/wsrn-1.1-schema-
200608.xsd](http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-schema-200702608/wsrn-1.1-schema-
1693 200608.xsd)

1694 The following copy is provided for reference.

```
1695 <?xml version="1.0" encoding="UTF-8"?>
1696 <!-- Copyright (C) OASIS (R) 1993-2007. All Rights Reserved.
1697 OASIS takes no position regarding the validity or scope of any intellectual-
1698 property or other rights that might be claimed to pertain to the-
1699 implementation or use of the technology described in this document or the-
1700 extent to which any license under such rights might or might not be available;
1701 neither does it represent that it has made any effort to identify any such-
1702 rights. Information on OASIS's procedures with respect to rights in OASIS-
1703 specifications can be found at the OASIS website. Copies of claims of rights-
1704 made available for publication and any assurances of licenses to be made-
1705 available, or the result of an attempt made to obtain a general license or-
1706 permission for the use of such proprietary rights by implementors or users of-
1707 this specification, can be obtained from the OASIS Executive Director.
1708 OASIS invites any interested party to bring to its attention any copyrights,-
1709 patents or patent applications, or other proprietary rights which may cover-
1710 technology that may be required to implement this specification. Please-
1711 address the information to the OASIS Executive Director.
1712 Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
1713 This document and translations of it may be copied and furnished to others,-
1714 and derivative works that comment on or otherwise explain it or assist in its-
1715 implementation may be prepared, copied, published and distributed, in whole or-
1716 in part, without restriction of any kind, provided that the above copyright-
1717 notice and this paragraph are included on all such copies and derivative-
1718 works. However, this document itself does not be modified in any way, such as-
1719 by removing the copyright notice or references to OASIS, except as needed for-
1720 the purpose of developing OASIS specifications, in which case the procedures-
1721 for copyrights defined in the OASIS Intellectual Property Rights document must-
1722 be followed, or as required to translate it into languages other than English.
1723 The limited permissions granted above are perpetual and will not be revoked by-
1724 OASIS or its successors or assigns.
1725 This document and the information contained herein is provided on an "AS IS"-
1726 basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT-
1727 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT-
1728 INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS-
1729 FOR A PARTICULAR PURPOSE.
1730 ->
1731 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"-
1732 xmlns:wsa="http://www.w3.org/2005/08/addressing"-
1733 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200608"-
1734 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200608"-
1735 elementFormDefault="qualified" attributeFormDefault="unqualified">
1736 <xs:import namespace="http://www.w3.org/2005/08/addressing"-
1737 schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
1738 <!-- Protocol Elements -->
1739 <xs:complexType name="SequenceType">
1740 <xs:sequence>
1741 <xs:element ref="wsrm:Identifier"/>
1742 <xs:element name="MessageNumber" type="wsrm:MessageNumberType"/>
1743 <xs:any namespace="##other" processContents="lax" minOccurs="0"-
1744 maxOccurs="unbounded"/>
```

```

1745 </xs:sequence>
1746 <xs:anyAttribute namespace="##other" processContents="lax"/>
1747 </xs:complexType>
1748 <xs:element name="Sequence" type="wsrm:SequenceType"/>
1749 <xs:element name="SequenceAcknowledgement">
1750 <xs:complexType>
1751 <xs:sequence>
1752 <xs:element ref="wsrm:Identifier"/>
1753 <xs:choice>
1754 <xs:sequence>
1755 <xs:choice>
1756 <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
1757 <xs:complexType>
1758 <xs:sequence/>
1759 <xs:attribute name="Upper" type="xs:unsignedLong"
1760 use="required"/>
1761 <xs:attribute name="Lower" type="xs:unsignedLong"
1762 use="required"/>
1763 <xs:anyAttribute namespace="##other" processContents="lax"/>
1764 </xs:complexType>
1765 </xs:element>
1766 <xs:element name="None">
1767 <xs:complexType>
1768 <xs:sequence/>
1769 </xs:complexType>
1770 </xs:element>
1771 </xs:choice>
1772 <xs:element name="Final" minOccurs="0">
1773 <xs:complexType>
1774 <xs:sequence/>
1775 </xs:complexType>
1776 </xs:element>
1777 </xs:sequence>
1778 <xs:element name="Nack" type="xs:unsignedLong"
1779 maxOccurs="unbounded"/>
1780 </xs:choice>
1781 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1782 maxOccurs="unbounded"/>
1783 </xs:sequence>
1784 <xs:anyAttribute namespace="##other" processContents="lax"/>
1785 </xs:complexType>
1786 </xs:element>
1787 <xs:complexType name="AckRequestedType">
1788 <xs:sequence>
1789 <xs:element ref="wsrm:Identifier"/>
1790 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1791 maxOccurs="unbounded"/>
1792 </xs:sequence>
1793 <xs:anyAttribute namespace="##other" processContents="lax"/>
1794 </xs:complexType>
1795 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1796 <xs:complexType name="MessagePendingType">
1797 <xs:sequence>
1798 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1799 maxOccurs="unbounded"/>
1800 </xs:sequence>
1801 <xs:attribute name="pending" type="xs:boolean"/>
1802 <xs:anyAttribute namespace="##other" processContents="lax"/>
1803 </xs:complexType>
1804 <xs:element name="MessagePending" type="wsrm:MessagePendingType"/>
1805 <xs:element name="Identifier">
1806 <xs:complexType>
1807 <xs:annotation>

```

```

1808 <del><xs:documentation>
1809 <del>  This type is for elements whose [children] is an anyURI and can have
1810 <del> arbitrary attributes.
1811 <del></xs:documentation>
1812 <del></xs:annotation>
1813 <del><xs:simpleContent>
1814 <del>  <xs:extension base="xs:anyURI">
1815 <del>    <xs:anyAttribute namespace="##other" processContents="lax"/>
1816 <del>  </xs:extension>
1817 <del></xs:simpleContent>
1818 <del></xs:complexType>
1819 <del></xs:element>
1820 <del><xs:element name="Address">
1821 <del>  <xs:complexType>
1822 <del>    <xs:simpleContent>
1823 <del>      <xs:extension base="xs:anyURI">
1824 <del>        <xs:anyAttribute namespace="##other" processContents="lax"/>
1825 <del>      </xs:extension>
1826 <del>    </xs:simpleContent>
1827 <del>  </xs:complexType>
1828 <del></xs:element>
1829 <del><xs:complexType name="MakeConnectionType">
1830 <del>  <xs:sequence>
1831 <del>    <xs:element ref="wsrm:Identifier" minOccurs="0" maxOccurs="1"/>
1832 <del>    <xs:element ref="wsrm:Address" minOccurs="0" maxOccurs="1"/>
1833 <del>    <xs:any namespace="##other" processContents="lax" minOccurs="0"
1834 <del>maxOccurs="unbounded"/>
1835 <del>  </xs:sequence>
1836 <del>  <xs:anyAttribute namespace="##other" processContents="lax"/>
1837 <del></xs:complexType>
1838 <del><xs:element name="MakeConnection" type="wsrm:MakeConnectionType"/>
1839 <del><xs:simpleType name="MessageNumberType">
1840 <del>  <xs:restriction base="xs:unsignedLong">
1841 <del>    <xs:minInclusive value="1"/>
1842 <del>    <xs:maxInclusive value="9223372036854775807"/>
1843 <del>  </xs:restriction>
1844 <del></xs:simpleType>
1845 <del><!-- Fault Container and Codes -->
1846 <del><xs:simpleType name="FaultCodes">
1847 <del>  <xs:restriction base="xs:QName">
1848 <del>    <xs:enumeration value="wsrm:SequenceTerminated"/>
1849 <del>    <xs:enumeration value="wsrm:UnknownSequence"/>
1850 <del>    <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1851 <del>    <xs:enumeration value="wsrm:MessageNumberRollover"/>
1852 <del>    <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1853 <del>    <xs:enumeration value="wsrm:SequenceClosed"/>
1854 <del>    <xs:enumeration value="wsrm:WSRMRequired"/>
1855 <del>    <xs:enumeration value="wsrm:UnsupportedSelection"/>
1856 <del>  </xs:restriction>
1857 <del></xs:simpleType>
1858 <del><xs:complexType name="SequenceFaultType">
1859 <del>  <xs:sequence>
1860 <del>    <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1861 <del>    <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
1862 <del>    <xs:any namespace="##other" processContents="lax" minOccurs="0"
1863 <del>maxOccurs="unbounded"/>
1864 <del>  </xs:sequence>
1865 <del>  <xs:anyAttribute namespace="##other" processContents="lax"/>
1866 <del></xs:complexType>
1867 <del><xs:complexType name="DetailType">
1868 <del>  <xs:sequence>
1869 <del>    <xs:any namespace="##other" processContents="lax" minOccurs="0"
1870 <del>maxOccurs="unbounded"/>

```

```

1871 </xs:sequence>
1872 <xs:anyAttribute namespace="##other" processContents="lax"/>
1873 </xs:complexType>
1874 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1875 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1876 <xs:element name="CreateSequenceResponse"
1877 type="wsrm:CreateSequenceResponseType"/>
1878 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1879 <xs:element name="CloseSequenceResponse"
1880 type="wsrm:CloseSequenceResponseType"/>
1881 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1882 <xs:element name="TerminateSequenceResponse"
1883 type="wsrm:TerminateSequenceResponseType"/>
1884 <xs:complexType name="CreateSequenceType">
1885 <xs:sequence>
1886 <xs:element ref="wsrm:AcksTo"/>
1887 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1888 <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1889 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1890 maxOccurs="unbounded">
1891 <xs:annotation>
1892 <xs:documentation>
1893 It is the authors intent that this extensibility be used to
1894 transfer a Security Token Reference as defined in WS Security.
1895 </xs:documentation>
1896 </xs:annotation>
1897 </xs:any>
1898 </xs:sequence>
1899 <xs:anyAttribute namespace="##other" processContents="lax"/>
1900 </xs:complexType>
1901 <xs:complexType name="CreateSequenceResponseType">
1902 <xs:sequence>
1903 <xs:element ref="wsrm:Identifier"/>
1904 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1905 <xs:element name="IncompleteSequenceBehavior"
1906 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
1907 <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1908 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1909 maxOccurs="unbounded"/>
1910 </xs:sequence>
1911 <xs:anyAttribute namespace="##other" processContents="lax"/>
1912 </xs:complexType>
1913 <xs:complexType name="CloseSequenceType">
1914 <xs:sequence>
1915 <xs:element ref="wsrm:Identifier"/>
1916 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1917 maxOccurs="unbounded"/>
1918 </xs:sequence>
1919 <xs:anyAttribute namespace="##other" processContents="lax"/>
1920 </xs:complexType>
1921 <xs:complexType name="CloseSequenceResponseType">
1922 <xs:sequence>
1923 <xs:element ref="wsrm:Identifier"/>
1924 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1925 maxOccurs="unbounded"/>
1926 </xs:sequence>
1927 <xs:anyAttribute namespace="##other" processContents="lax"/>
1928 </xs:complexType>
1929 <xs:complexType name="TerminateSequenceType">
1930 <xs:sequence>
1931 <xs:element ref="wsrm:Identifier"/>
1932 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1933 maxOccurs="unbounded"/>

```

```

1934 </xs:sequence>
1935 <xs:anyAttribute namespace="##other" processContents="lax"/>
1936 </xs:complexType>
1937 <xs:complexType name="TerminateSequenceResponseType">
1938 <xs:sequence>
1939 <xs:element ref="wsrm:Identifier"/>
1940 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1941 maxOccurs="unbounded"/>
1942 </xs:sequence>
1943 <xs:anyAttribute namespace="##other" processContents="lax"/>
1944 </xs:complexType>
1945 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
1946 <xs:complexType name="OfferType">
1947 <xs:sequence>
1948 <xs:element ref="wsrm:Identifier"/>
1949 <xs:element name="Endpoint" type="wsa:EndpointReferenceType"/>
1950 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1951 <xs:element name="IncompleteSequenceBehavior"
1952 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
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="AcceptType">
1959 <xs:sequence>
1960 <xs:element ref="wsrm:AcksTo"/>
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="Expires">
1967 <xs:complexType>
1968 <xs:simpleContent>
1969 <xs:extension base="xs:duration">
1970 <xs:anyAttribute namespace="##other" processContents="lax"/>
1971 </xs:extension>
1972 </xs:simpleContent>
1973 </xs:complexType>
1974 </xs:element>
1975 <xs:simpleType name="IncompleteSequenceBehaviorType">
1976 <xs:restriction base="xs:string">
1977 <xs:enumeration value="DiscardEntireSequence"/>
1978 <xs:enumeration value="DiscardFollowingFirstGap"/>
1979 <xs:enumeration value="NoDiscard"/>
1980 </xs:restriction>
1981 </xs:simpleType>
1982 OASIS trademark, IPR and other policies apply. -->
1983 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1984 xmlns:wsa="http://www.w3.org/2005/08/addressing"
1985 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200702"
1986 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrm/200702"
1987 elementFormDefault="qualified" attributeFormDefault="unqualified">
1988 <xs:import namespace="http://www.w3.org/2005/08/addressing"
1989 schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
1990 <!-- Protocol Elements -->
1991 <xs:complexType name="SequenceType">
1992 <xs:sequence>
1993 <xs:element ref="wsrm:Identifier"/>
1994 <xs:element name="MessageNumber" type="wsrm:MessageNumberType"/>
1995 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1996 maxOccurs="unbounded"/>

```

```

1997 </xs:sequence>
1998 <xs:anyAttribute namespace="##other" processContents="lax"/>
1999 </xs:complexType>
2000 <xs:element name="Sequence" type="wsrm:SequenceType"/>
2001 <xs:element name="SequenceAcknowledgement">
2002 <xs:complexType>
2003 <xs:sequence>
2004 <xs:element ref="wsrm:Identifier"/>
2005 <xs:choice>
2006 <xs:sequence>
2007 <xs:choice>
2008 <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
2009 <xs:complexType>
2010 <xs:sequence/>
2011 <xs:attribute name="Upper" type="xs:unsignedLong"
2012 use="required"/>
2013 <xs:attribute name="Lower" type="xs:unsignedLong"
2014 use="required"/>
2015 <xs:anyAttribute namespace="##other" processContents="lax"/>
2016 </xs:complexType>
2017 </xs:element>
2018 <xs:element name="None">
2019 <xs:complexType>
2020 <xs:sequence/>
2021 </xs:complexType>
2022 </xs:element>
2023 </xs:choice>
2024 <xs:element name="Final" minOccurs="0">
2025 <xs:complexType>
2026 <xs:sequence/>
2027 </xs:complexType>
2028 </xs:element>
2029 </xs:sequence>
2030 <xs:element name="Nack" type="xs:unsignedLong"
2031 maxOccurs="unbounded"/>
2032 </xs:choice>
2033 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2034 maxOccurs="unbounded"/>
2035 </xs:sequence>
2036 <xs:anyAttribute namespace="##other" processContents="lax"/>
2037 </xs:complexType>
2038 </xs:element>
2039 <xs:complexType name="AckRequestedType">
2040 <xs:sequence>
2041 <xs:element ref="wsrm:Identifier"/>
2042 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2043 maxOccurs="unbounded"/>
2044 </xs:sequence>
2045 <xs:anyAttribute namespace="##other" processContents="lax"/>
2046 </xs:complexType>
2047 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
2048 <xs:element name="Identifier">
2049 <xs:complexType>
2050 <xs:annotation>
2051 <xs:documentation>
2052 This type is for elements whose [children] is an anyURI and can have
2053 arbitrary attributes.
2054 </xs:documentation>
2055 </xs:annotation>
2056 <xs:simpleContent>
2057 <xs:extension base="xs:anyURI">
2058 <xs:anyAttribute namespace="##other" processContents="lax"/>
2059 </xs:extension>

```

```

2060     </xs:simpleContent>
2061   </xs:complexType>
2062 </xs:element>
2063   <xs:element name="Address">
2064     <xs:complexType>
2065       <xs:simpleContent>
2066         <xs:extension base="xs:anyURI">
2067           <xs:anyAttribute namespace="##other" processContents="lax"/>
2068         </xs:extension>
2069       </xs:simpleContent>
2070     </xs:complexType>
2071   </xs:element>
2072   <xs:simpleType name="MessageNumberType">
2073     <xs:restriction base="xs:unsignedLong">
2074       <xs:minInclusive value="1"/>
2075       <xs:maxInclusive value="9223372036854775807"/>
2076     </xs:restriction>
2077   </xs:simpleType>
2078   <!-- Fault Container and Codes -->
2079   <xs:simpleType name="FaultCodes">
2080     <xs:restriction base="xs:QName">
2081       <xs:enumeration value="wsrm:SequenceTerminated"/>
2082       <xs:enumeration value="wsrm:UnknownSequence"/>
2083       <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
2084       <xs:enumeration value="wsrm:MessageNumberRollover"/>
2085       <xs:enumeration value="wsrm:CreateSequenceRefused"/>
2086       <xs:enumeration value="wsrm:SequenceClosed"/>
2087       <xs:enumeration value="wsrm:WSRMRequired"/>
2088       <xs:enumeration value="wsrm:UnsupportedSelection"/>
2089     </xs:restriction>
2090   </xs:simpleType>
2091   <xs:complexType name="SequenceFaultType">
2092     <xs:sequence>
2093       <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
2094       <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
2095       <xs:any namespace="##other" processContents="lax" minOccurs="0"
2096 maxOccurs="unbounded"/>
2097     </xs:sequence>
2098     <xs:anyAttribute namespace="##other" processContents="lax"/>
2099   </xs:complexType>
2100   <xs:complexType name="DetailType">
2101     <xs:sequence>
2102       <xs:any namespace="##other" processContents="lax" minOccurs="0"
2103 maxOccurs="unbounded"/>
2104     </xs:sequence>
2105     <xs:anyAttribute namespace="##other" processContents="lax"/>
2106   </xs:complexType>
2107   <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
2108   <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
2109   <xs:element name="CreateSequenceResponse"
2110 type="wsrm:CreateSequenceResponseType"/>
2111   <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
2112   <xs:element name="CloseSequenceResponse"
2113 type="wsrm:CloseSequenceResponseType"/>
2114   <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
2115   <xs:element name="TerminateSequenceResponse"
2116 type="wsrm:TerminateSequenceResponseType"/>
2117   <xs:complexType name="CreateSequenceType">
2118     <xs:sequence>
2119       <xs:element ref="wsrm:AcksTo"/>
2120       <xs:element ref="wsrm:Expires" minOccurs="0"/>
2121       <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
2122       <xs:any namespace="##other" processContents="lax" minOccurs="0"

```

```

2123 maxOccurs="unbounded">
2124   <xs:annotation>
2125     <xs:documentation>
2126       It is the authors intent that this extensibility be used to
2127       transfer a Security Token Reference as defined in WS-Security.
2128     </xs:documentation>
2129   </xs:annotation>
2130 </xs:any>
2131 </xs:sequence>
2132 <xs:anyAttribute namespace="##other" processContents="lax"/>
2133 </xs:complexType>
2134 <xs:complexType name="CreateSequenceResponseType">
2135   <xs:sequence>
2136     <xs:element ref="wsrm:Identifier"/>
2137     <xs:element ref="wsrm:Expires" minOccurs="0"/>
2138     <xs:element name="IncompleteSequenceBehavior"
2139     type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
2140     <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
2141     <xs:any namespace="##other" processContents="lax" minOccurs="0"
2142     maxOccurs="unbounded"/>
2143   </xs:sequence>
2144   <xs:anyAttribute namespace="##other" processContents="lax"/>
2145 </xs:complexType>
2146 <xs:complexType name="CloseSequenceType">
2147   <xs:sequence>
2148     <xs:element ref="wsrm:Identifier"/>
2149     <xs:element name="LastMsgNumber" type="wsrm:MessageNumberType"
2150     minOccurs="0"/>
2151     <xs:any namespace="##other" processContents="lax" minOccurs="0"
2152     maxOccurs="unbounded"/>
2153   </xs:sequence>
2154   <xs:anyAttribute namespace="##other" processContents="lax"/>
2155 </xs:complexType>
2156 <xs:complexType name="CloseSequenceResponseType">
2157   <xs:sequence>
2158     <xs:element ref="wsrm:Identifier"/>
2159     <xs:any namespace="##other" processContents="lax" minOccurs="0"
2160     maxOccurs="unbounded"/>
2161   </xs:sequence>
2162   <xs:anyAttribute namespace="##other" processContents="lax"/>
2163 </xs:complexType>
2164 <xs:complexType name="TerminateSequenceType">
2165   <xs:sequence>
2166     <xs:element ref="wsrm:Identifier"/>
2167     <xs:element name="LastMsgNumber" type="wsrm:MessageNumberType"
2168     minOccurs="0"/>
2169     <xs:any namespace="##other" processContents="lax" minOccurs="0"
2170     maxOccurs="unbounded"/>
2171   </xs:sequence>
2172   <xs:anyAttribute namespace="##other" processContents="lax"/>
2173 </xs:complexType>
2174 <xs:complexType name="TerminateSequenceResponseType">
2175   <xs:sequence>
2176     <xs:element ref="wsrm:Identifier"/>
2177     <xs:any namespace="##other" processContents="lax" minOccurs="0"
2178     maxOccurs="unbounded"/>
2179   </xs:sequence>
2180   <xs:anyAttribute namespace="##other" processContents="lax"/>
2181 </xs:complexType>
2182 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
2183 <xs:complexType name="OfferType">
2184   <xs:sequence>
2185     <xs:element ref="wsrm:Identifier"/>

```

```

2186 <xs:element name="Endpoint" type="wsa:EndpointReferenceType"/>
2187 <xs:element ref="wsrm:Expires" minOccurs="0"/>
2188 <xs:element name="IncompleteSequenceBehavior"
2189 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
2190 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2191 maxOccurs="unbounded"/>
2192 </xs:sequence>
2193 <xs:anyAttribute namespace="##other" processContents="lax"/>
2194 </xs:complexType>
2195 <xs:complexType name="AcceptType">
2196 <xs:sequence>
2197 <xs:element ref="wsrm:AcksTo"/>
2198 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2199 maxOccurs="unbounded"/>
2200 </xs:sequence>
2201 <xs:anyAttribute namespace="##other" processContents="lax"/>
2202 </xs:complexType>
2203 <xs:element name="Expires">
2204 <xs:complexType>
2205 <xs:simpleContent>
2206 <xs:extension base="xs:duration">
2207 <xs:anyAttribute namespace="##other" processContents="lax"/>
2208 </xs:extension>
2209 </xs:simpleContent>
2210 </xs:complexType>
2211 </xs:element>
2212 <xs:simpleType name="IncompleteSequenceBehaviorType">
2213 <xs:restriction base="xs:string">
2214 <xs:enumeration value="DiscardEntireSequence"/>
2215 <xs:enumeration value="DiscardFollowingFirstGap"/>
2216 <xs:enumeration value="NoDiscard"/>
2217 </xs:restriction>
2218 </xs:simpleType>
2219 <xs:element name="UsesSequenceSTR">
2220 <xs:complexType sequence/>
2221 <xs:sequence<xs:anyAttribute namespace="##other" processContents="lax"/>
2222 <xs:anyAttribute namespace="##other" processContents="lax"/>
2223 </xs:complexType>
2224 </xs:element>
2225 <xs:element name="UsesSequenceSSL">
2226 <xs:complexType sequence/>
2227 <xs:sequence<xs:anyAttribute namespace="##other" processContents="lax"/>
2228 <xs:anyAttribute namespace="##other" processContents="lax"/>
2229 </xs:complexType>
2230 </xs:element>
2231 <xs:element name="UnsupportedElement">
2232 <xs:simpleType>
2233 <xs:restriction base="xs:QName"/>
2234 </xs:simpleType>
2235 </xs:element>
2236 </xs:schema>

```

2237 Appendix B. WSDL

2238 This WSDL describes the WS-RM protocol from the point of view of an RM Destination. In the case where
2239 an endpoint acts both as an RM Destination and an RM Source, note that additional messages may be
2240 present in exchanges with that endpoint.

2241 Also note that this WSDL is intended to describe the internal structure of the WS-RM protocol, and will not
2242 generally appear in a description of a WS-RM-capable Web service. See WS-RM Policy [WS-RM Policy]
2243 for a higher-level mechanism to indicate that WS-RM is engaged.

2244 The normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

2245 [http://docs.oasis-open.org/ws-rx/wsrml/200702/wsd/wsrml-1.1-wsd-200702608/wsd/wsrml-1.1-wsd-](http://docs.oasis-open.org/ws-rx/wsrml/200702/wsd/wsrml-1.1-wsd-200702608/wsd/wsrml-1.1-wsd-200608.wsdl)
2246 [200608.wsdl](http://docs.oasis-open.org/ws-rx/wsrml/200702/wsd/wsrml-1.1-wsd-200702608/wsd/wsrml-1.1-wsd-200608.wsdl)

2247 The following non-normative copy is provided for reference.

```
2248 <?xml version="1.0" encoding="utf-8"?>
2249 <!-- Copyright (C) OASIS (R) 1993-2007. All Rights Reserved.
2250 OASIS takes no position regarding the validity or scope of any intellectual
2251 property or other rights that might be claimed to pertain to the
2252 implementation or use of the technology described in this document or the
2253 extent to which any license under such rights might or might not be available;
2254 neither does it represent that it has made any effort to identify any such
2255 rights. Information on OASIS's procedures with respect to rights in OASIS
2256 specifications can be found at the OASIS website. Copies of claims of rights
2257 made available for publication and any assurances of licenses to be made
2258 available, or the result of an attempt made to obtain a general license or
2259 permission for the use of such proprietary rights by implementors or users of
2260 this specification, can be obtained from the OASIS Executive Director.
2261 OASIS invites any interested party to bring to its attention any copyrights,
2262 patents or patent applications, or other proprietary rights which may cover
2263 technology that may be required to implement this specification. Please
2264 address the information to the OASIS Executive Director.
2265 Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
2266 This document and translations of it may be copied and furnished to others,
2267 and derivative works that comment on or otherwise explain it or assist in its
2268 implementation may be prepared, copied, published and distributed, in whole or
2269 in part, without restriction of any kind, provided that the above copyright
2270 notice and this paragraph are included on all such copies and derivative
2271 works. However, this document itself does not be modified in any way, such as
2272 by removing the copyright notice or references to OASIS, except as needed for
2273 the purpose of developing OASIS specifications, in which case the procedures
2274 for copyrights defined in the OASIS Intellectual Property Rights document must
2275 be followed, or as required to translate it into languages other than English.
2276 The limited permissions granted above are perpetual and will not be revoked by
2277 OASIS or its successors or assigns.
2278 This document and the information contained herein is provided on an "AS IS"
2279 basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
2280 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT
2281 ENFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS
2282 FOR A PARTICULAR PURPOSE.
2283 -->
2284 <wsl:definitions xmlns:wsl="http://schemas.xmlsoap.org/wsl/"
2285 xmlns:xs="http://www.w3.org/2001/XMLSchema"
2286 xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
2287 open.org/ws-rx/wsrml/200608" xmlns:tns="http://docs.oasis-open.org/ws-
2288 rx/wsrml/200608/wsl" targetNamespace="http://docs.oasis-open.org/ws-
2289 rx/wsrml/200608/wsl">
```

```

2290 <wsdl:types>
2291 <xs:schema
2292 <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsrml/200608"
2293 schemaLocation="http://docs.oasis-open.org/ws-rx/wsrml/200608/wsrml-1.1-schema-
2294 200608.xsd"/>
2295 </xs:schema>
2296 </wsdl:types>

2297 <wsdl:message name="CreateSequence">
2298 <wsdl:part name="create" element="rm:CreateSequence"/>
2299 </wsdl:message>
2300 <wsdl:message name="CreateSequenceResponse">
2301 <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
2302 </wsdl:message>
2303 <wsdl:message name="CloseSequence">
2304 <wsdl:part name="close" element="rm:CloseSequence"/>
2305 </wsdl:message>
2306 <wsdl:message name="CloseSequenceResponse">
2307 <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
2308 </wsdl:message>
2309 <wsdl:message name="TerminateSequence">
2310 <wsdl:part name="terminate" element="rm:TerminateSequence"/>
2311 </wsdl:message>
2312 <wsdl:message name="TerminateSequenceResponse">
2313 <wsdl:part name="terminateResponse"
2314 element="rm:TerminateSequenceResponse"/>
2315 </wsdl:message>
2316 <wsdl:message name="MakeConnection">
2317 <wsdl:part name="makeConnection" element="rm:MakeConnection"/>
2318 </wsdl:message>

2319 <wsdl:portType name="SequenceAbstractPortType">
2320 <wsdl:operation name="CreateSequence">
2321 <wsdl:input message="tns:CreateSequence" wsaw:Action="http://docs.oasis-
2322 open.org/ws-rx/wsrml/200608/CreateSequence"/>
2323 <wsdl:output message="tns:CreateSequenceResponse"
2324 wsaw:Action="http://docs.oasis-open.org/ws-
2325 rx/wsrml/200608/CreateSequenceResponse"/>
2326 </wsdl:operation>
2327 <wsdl:operation name="CloseSequence">
2328 <wsdl:input message="tns:CloseSequence" wsaw:Action="http://docs.oasis-
2329 open.org/ws-rx/wsrml/200608/CloseSequence"/>
2330 <wsdl:output message="tns:CloseSequenceResponse"
2331 wsaw:Action="http://docs.oasis-open.org/ws-
2332 rx/wsrml/200608/CloseSequenceResponse"/>
2333 </wsdl:operation>
2334 <wsdl:operation name="TerminateSequence">
2335 <wsdl:input message="tns:TerminateSequence"
2336 wsaw:Action="http://docs.oasis-open.org/ws-rx/wsrml/200608/TerminateSequence"/>
2337 <wsdl:output message="tns:TerminateSequenceResponse"
2338 wsaw:Action="http://docs.oasis-open.org/ws-
2339 rx/wsrml/200608/TerminateSequenceResponse"/>
2340 </wsdl:operation>
2341 <wsdl:operation name="MakeConnection">
2342 <wsdl:input message="tns:MakeConnection" wsaw:Action="http://docs.oasis-
2343 open.org/ws-rx/wsrml/200608/MakeConnection"/>
2344 </wsdl:operation>
2345 </wsdl:portType>

2346 </wsdl:definitions>
2347 OASIS trademark, IPR and other policies apply. -->
2348 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
2349 xmlns:xs="http://www.w3.org/2001/XMLSchema"

```

```

2350 xmlns:wsa="http://www.w3.org/2005/08/addressing"
2351 xmlns:wsam="http://www.w3.org/2007/02/addressing/metadata"
2352 xmlns:rm="http://docs.oasis-open.org/ws-rx/wsrn/200702"
2353 xmlns:tns="http://docs.oasis-open.org/ws-rx/wsrn/200702/wsd1"
2354 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200702/wsd1">
|
2355 <wsdl:types>
2356 <xs:schema>
2357 <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsrn/200702"
2358 schemaLocation="http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-schema-
2359 200702.xsd"/>
2360 </xs:schema>
2361 </wsdl:types>
|
2362 <wsdl:message name="CreateSequence">
2363 <wsdl:part name="create" element="rm:CreateSequence"/>
2364 </wsdl:message>
2365 <wsdl:message name="CreateSequenceResponse">
2366 <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
2367 </wsdl:message>
2368 <wsdl:message name="CloseSequence">
2369 <wsdl:part name="close" element="rm:CloseSequence"/>
2370 </wsdl:message>
2371 <wsdl:message name="CloseSequenceResponse">
2372 <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
2373 </wsdl:message>
2374 <wsdl:message name="TerminateSequence">
2375 <wsdl:part name="terminate" element="rm:TerminateSequence"/>
2376 </wsdl:message>
2377 <wsdl:message name="TerminateSequenceResponse">
2378 <wsdl:part name="terminateResponse"
2379 element="rm:TerminateSequenceResponse"/>
2380 </wsdl:message>
|
2381 <wsdl:portType name="SequenceAbstractPortType">
2382 <wsdl:operation name="CreateSequence">
2383 <wsdl:input message="tns:CreateSequence" wsam:Action="http://docs.oasis-
2384 open.org/ws-rx/wsrn/200702/CreateSequence"/>
2385 <wsdl:output message="tns:CreateSequenceResponse"
2386 wsam:Action="http://docs.oasis-open.org/ws-
2387 rx/wsrn/200702/CreateSequenceResponse"/>
2388 </wsdl:operation>
2389 <wsdl:operation name="CloseSequence">
2390 <wsdl:input message="tns:CloseSequence" wsam:Action="http://docs.oasis-
2391 open.org/ws-rx/wsrn/200702/CloseSequence"/>
2392 <wsdl:output message="tns:CloseSequenceResponse"
2393 wsam:Action="http://docs.oasis-open.org/ws-
2394 rx/wsrn/200702/CloseSequenceResponse"/>
2395 </wsdl:operation>
2396 <wsdl:operation name="TerminateSequence">
2397 <wsdl:input message="tns:TerminateSequence"
2398 wsam:Action="http://docs.oasis-open.org/ws-rx/wsrn/200702/TerminateSequence"/>
2399 <wsdl:output message="tns:TerminateSequenceResponse"
2400 wsam:Action="http://docs.oasis-open.org/ws-
2401 rx/wsrn/200702/TerminateSequenceResponse"/>
2402 </wsdl:operation>
2403 </wsdl:portType>
|
2404 </wsdl:definitions>

```

2405 Appendix C. Message Examples

2406 Appendix C.1 Create Sequence

2407 Create Sequence

```
2408 <?xml version="1.0" encoding="UTF-8"?>
2409 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2410 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702608"
2411 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2412   <S:Header>
2413     <wsa:MessageID>
2414       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
2415     </wsa:MessageID>
2416     <wsa:To>http://example.com/serviceB/123</wsa:To>
2417     <wsa:Action>http://docs.oasis-open.org/ws-
2418 rx/wsmr/200702608/CreateSequence</wsa:Action>
2419     <wsa:ReplyTo>
2420       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2421     </wsa:ReplyTo>
2422   </S:Header>
2423   <S:Body>
2424     <wsmr:CreateSequence>
2425       <wsmr:AcksTo>
2426         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2427       </wsmr:AcksTo>
2428     </wsmr:CreateSequence>
2429   </S:Body>
2430 </S:Envelope>
```

2431 Create Sequence Response

```
2432 <?xml version="1.0" encoding="UTF-8"?>
2433 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2434 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702608"
2435 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2436   <S:Header>
2437     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
2438     <wsa:RelatesTo>
2439       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
2440     </wsa:RelatesTo>
2441     <wsa:Action>
2442       http://docs.oasis-open.org/ws-rx/wsmr/200702608/CreateSequenceResponse
2443     </wsa:Action>
2444   </S:Header>
2445   <S:Body>
2446     <wsmr:CreateSequenceResponse>
2447       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2448     </wsmr:CreateSequenceResponse>
2449   </S:Body>
2450 </S:Envelope>
```

2451 Appendix C.2 Initial Transmission

2452 The following example WS-ReliableMessaging headers illustrate the message exchange in the above
2453 figure. The three messages have the following headers; the third message is identified as the last
2454 message in the Sequence:

2455 **Message 1**

```
2456 <?xml version="1.0" encoding="UTF-8"?>
2457 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2458 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702698"
2459 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2460   <S:Header>
2461     <wsa:MessageID>
2462       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
2463     </wsa:MessageID>
2464     <wsa:To>http://example.com/serviceB/123</wsa:To>
2465     <wsa:From>
2466       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2467     </wsa:From>
2468     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2469     <wsmr:Sequence>
2470       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2471       <wsmr:MessageNumber>1</wsmr:MessageNumber>
2472     </wsmr:Sequence>
2473   </S:Header>
2474   <S:Body>
2475     <!-- Some Application Data -->
2476   </S:Body>
2477 </S:Envelope>
```

2478 **Message 2**

```
2479 <?xml version="1.0" encoding="UTF-8"?>
2480 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2481 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702698"
2482 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2483   <S:Header>
2484     <wsa:MessageID>
2485       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
2486     </wsa:MessageID>
2487     <wsa:To>http://example.com/serviceB/123</wsa:To>
2488     <wsa:From>
2489       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2490     </wsa:From>
2491     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2492     <wsmr:Sequence>
2493       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2494       <wsmr:MessageNumber>2</wsmr:MessageNumber>
2495     </wsmr:Sequence>
2496   </S:Header>
2497   <S:Body>
2498     <!-- Some Application Data -->
2499   </S:Body>
2500 </S:Envelope>
```

2501 **Message 3**

```
2502 <?xml version="1.0" encoding="UTF-8"?>
2503 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2504 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702698"
2505 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2506   <S:Header>
2507     <wsa:MessageID>
2508       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
2509     </wsa:MessageID>
2510     <wsa:To>http://example.com/serviceB/123</wsa:To>
2511     <wsa:From>
2512       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
```

```

2513 </wsa:From>
2514 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2515 <wsrm:Sequence>
2516 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2517 <wsrm:MessageNumber>3</wsrm:MessageNumber>
2518 </wsrm:Sequence>
2519 <wsrm:AckRequested>
2520 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2521 </wsrm:AckRequested>
2522 </S:Header>
2523 <S:Body>
2524 <!-- Some Application Data -->
2525 </S:Body>
2526 </S:Envelope>

```

2527 Appendix C.3 First Acknowledgement

2528 Message number 2 has not been accepted by the RM Destination due to some transmission error so it
2529 responds with an Acknowledgement for messages 1 and 3:

```

2530 <?xml version="1.0" encoding="UTF-8"?>
2531 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2532 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200702608"
2533 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2534 <S:Header>
2535 <wsa:MessageID>
2536 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
2537 </wsa:MessageID>
2538 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
2539 <wsa:From>
2540 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
2541 </wsa:From>
2542 <wsa:Action>
2543 http://docs.oasis-open.org/ws-rx/wsrn/200702608/SequenceAcknowledgement
2544 </wsa:Action>
2545 <wsrm:SequenceAcknowledgement>
2546 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2547 <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
2548 <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
2549 </wsrm:SequenceAcknowledgement>
2550 </S:Header>
2551 <S:Body/>
2552 </S:Envelope>

```

2553 Appendix C.4 Retransmission

2554 The RM Sourcediscovers that message number 2 was not accepted so it resends the message and
2555 requests an Acknowledgement:

```

2556 <?xml version="1.0" encoding="UTF-8"?>
2557 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2558 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200702608"
2559 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2560 <S:Header>
2561 <wsa:MessageID>
2562 http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
2563 </wsa:MessageID>
2564 <wsa:To>http://example.com/serviceB/123</wsa:To>
2565 <wsa:From>
2566 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2567 </wsa:From>

```

```

2568 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2569 <wsrm:Sequence>
2570 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2571 <wsrm:MessageNumber>2</wsrm:MessageNumber>
2572 </wsrm:Sequence>
2573 <wsrm:AckRequested>
2574 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2575 </wsrm:AckRequested>
2576 </S:Header>
2577 <S:Body>
2578 <!-- Some Application Data -->
2579 </S:Body>
2580 </S:Envelope>

```

2581 Appendix C.5 Termination

2582 The RM Destination now responds with an Acknowledgement for the complete Sequence which can then
 2583 be terminated:

```

2584 <?xml version="1.0" encoding="UTF-8"?>
2585 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2586 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702698"
2587 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2588 <S:Header>
2589 <wsa:MessageID>
2590 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
2591 </wsa:MessageID>
2592 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
2593 <wsa:From>
2594 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
2595 </wsa:From>
2596 <wsa:Action>
2597 http://docs.oasis-open.org/ws-rx/wsmr/200702698/SequenceAcknowledgement
2598 </wsa:Action>
2599 <wsmr:SequenceAcknowledgement>
2600 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2601 <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
2602 </wsmr:SequenceAcknowledgement>
2603 </S:Header>
2604 <S:Body/>
2605 </S:Envelope>

```

2606 Terminate Sequence

```

2607 <?xml version="1.0" encoding="UTF-8"?>
2608 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2609 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702698"
2610 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2611 <S:Header>
2612 <wsa:MessageID>
2613 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
2614 </wsa:MessageID>
2615 <wsa:To>http://example.com/serviceB/123</wsa:To>
2616 <wsa:Action>
2617 http://docs.oasis-open.org/ws-rx/wsmr/200702698/TerminateSequence
2618 </wsa:Action>
2619 <wsa:From>
2620 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2621 </wsa:From>
2622 </S:Header>
2623 <S:Body>
2624 <wsmr:TerminateSequence>

```

```

2625     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2626     <wsrm>LastMsgNumber> 3 </wsrm>LastMsgNumber>
2627   </wsrm:TerminateSequence>
2628 </S:Body>
2629 </S:Envelope>

```

2630 Terminate Sequence Response

```

2631 <?xml version="1.0" encoding="UTF-8"?>
2632 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2633 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrmp/200608"
2634 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2635   <S:Header>
2636     <wsa:MessageID>
2637       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
2638     </wsa:MessageID>
2639     <wsa:To>http://example.com/serviceA/789</wsa:To>
2640     <wsa:Action>
2641       http://docs.oasis-open.org/ws-rx/wsrmp/200608/TerminateSequenceResponse
2642     </wsa:Action>
2643     <wsa:RelatesTo>
2644       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
2645     </wsa:RelatesTo>
2646     <wsa:From>
2647       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2648     </wsa:From>
2649   </S:Header>
2650   <S:Body>
2651     <wsrm:TerminateSequenceResponse>
2652       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2653     </wsrm:TerminateSequenceResponse>
2654   </S:Body>
2655 </S:Envelope>

```

2656 Appendix C.6 MakeConnection

2657 To illustrate how a `MakeConnection` message exchange can be used to deliver messages to an
 2658 Endpoint that is not addressable, consider the case of a pub/sub scenario in which the Endpoint to which
 2659 notifications are to be delivered (the "event consumer") is not addressable by the notification sending
 2660 Endpoint (the "event producer"). In this scenario the event consumer must initiate the connections in order
 2661 for the notifications to be delivered. One possible set of message exchanges (using HTTP) that
 2662 demonstrate how this can be achieved using `MakeConnection` is shown below:

2663 **Step 1**—During a "subscribe" operation, the event consumer's EPR specifies the RM anonymous URI
 2664 and the RM Policy Assertion to indicate whether or not RM is required:

```

2665 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2666 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrmp/200608"
2667 xmlns:wsrmp="http://docs.oasis-open.org/ws-rx/wsrmp/200608"
2668 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2669   <S:Header>
2670     <wsa:To> http://example.org/subscriptionService </wsa:To>
2671     <wsa:MessageID> http://client456.org/id-a6d8-a7c2eb546813</wsa:MessageID>
2672     <wsa:ReplyTo>
2673       <wsa:To> http://client456.org/response </wsa:To>
2674     </wsa:ReplyTo>
2675   </S:Header>
2676   <S:Body>
2677     <sub:Subscribe xmlns:sub="http://example.org/subscriptionService">
2678       <!-- subscription service specific data -->
2679     </targetEPR>

```

```

2680 <wsa:Address>http://docs.oasis-open.org/ws-
2681 rx/wsrmp/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:Address>
2682 <wsa:Metadata>
2683 <wsp:Policy wsu:Id="MyPolicy">
2684 <wsrmp:RMAssertion/>
2685 </wsp:Policy>
2686 </wsa:Metadata>
2687 </targetEPR>
2688 </sub:Subscribe>
2689 </S:Body>
2690 </S:Envelope>

```

2691 In this example the `subscribe` and `targetEPR` elements are simply examples of what a subscription-
2692 request message might contain. Note: the `wsa:Address` element contains the RM anonymous URI
2693 indicating that the notification producer needs to queue the messages until they are requested using the
2694 `MakeConnection` message exchange. The EPR also contains the RM Policy Assertion indicating the RM-
2695 must be used when notifications related to this subscription are sent.

2696 **Step 2**—Once the subscription is established, the event consumer checks for a pending message:

```

2697 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"-
2698 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrmp/200608"-
2699 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2700 <S:Header>
2701 <wsa:Action>http://docs.oasis-open.org/ws-
2702 rx/wsrmp/200608/MakeConnection</wsa:Action>
2703 <wsa:To>http://example.org/subscriptionService</wsa:To>
2704 </S:Header>
2705 <S:Body>
2706 <wsrm:MakeConnection>
2707 <wsrm:Address>http://docs.oasis-open.org/ws-
2708 rx/wsrmp/200608/anonymous?id=550e8400-e29b-11d4-a716-
2709 446655440000</wsrm:Address>
2710 </wsrm:MakeConnection>
2711 </S:Body>
2712 </S:Envelope>

```

2713 **Step 3**—If there are messages waiting to be delivered then a message will be returned back to the event
2714 consumer. However, because WS-RM is being used to deliver the messages, the first message returned
2715 is a `CreateSequence`:

```

2716 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"-
2717 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrmp/200608"-
2718 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2719 <S:Header>
2720 <wsa:Action>http://docs.oasis-open.org/ws-
2721 rx/wsrmp/200608/CreateSequence</wsa:Action>
2722 <wsa:To>http://docs.oasis-open.org/ws-
2723 rx/wsrmp/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:To>
2724 <wsa:ReplyTo>http://example.org/subscriptionService</wsa:ReplyTo>
2725 <wsa:MessageID>http://example.org/id-123-456</wsa:MessageID>
2726 </S:Header>
2727 <S:Body>
2728 <wsrm:CreateSequence>
2729 <wsrm:AcksTo>
2730 <wsa:Address>http://example.org/subscriptionService</wsa:Address>
2731 </wsrm:AcksTo>
2732 </wsrm:CreateSequence>
2733 </S:Body>

```

2734 `</S:Envelope>`

2735 Notice from the perspective of how the RM Source on the event producer interacts with the RM-
2736 Destination of those messages, nothing new is introduced by the use of the `MakeConnection`, the use-
2737 of RM protocol is the same as the case where the event consumer is addressable.

2738 **Step 4**—The event consumer will respond with a `CreateSequenceResponse` message per normal WS-
2739 Addressing rules:

```
2740 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"-  
2741 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200608"-  
2742 xmlns:wsa="http://www.w3.org/2005/08/addressing">  
2743   <S:Header>  
2744     <wsa:Action>http://docs.oasis-open.org/ws-  
2745 rx/wsmr/200608/CreateSequenceResponse</wsa:Action>  
2746     <wsa:To> http://example.org/subscriptionService </wsa:To>  
2747     <wsa:RelatesTo> http://example.org/id-123-456 </wsa:RelatesTo>  
2748   </S:Header>  
2749   <S:Body>  
2750     <wsmr:CreateSequenceResponse>  
2751       <wsmr:Identifier> http://example.org/rmid-456 </wsmr:Identifier>  
2752     </wsmr:CreateSequenceResponse>  
2753   </S:Body>  
2754 </S:Envelope>
```

2755 Note, this message is carried on an HTTP request directed to the `wsa:ReplyTo` EPR, and the HTTP-
2756 response will be an HTTP 202.

2757 **Step 5**—The event consumer checks for another message pending:

```
2758 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"-  
2759 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200608"-  
2760 xmlns:wsa="http://www.w3.org/2005/08/addressing">  
2761   <S:Header>  
2762     <wsa:Action>http://docs.oasis-open.org/ws-  
2763 rx/wsmr/200608/MakeConnection</wsa:Action>  
2764     <wsa:To> http://example.org/subscriptionService </wsa:To>  
2765   </S:Header>  
2766   <S:Body>  
2767     <wsmr:MakeConnection>  
2768       <wsmr:Address>http://docs.oasis-open.org/ws-  
2769 rx/wsmr/200608/anonymous?id=550e8400-e29b-11d4-a716-  
2770 446655440000</wsmr:Address>  
2771     </wsmr:MakeConnection>  
2772   </S:Body>  
2773 </S:Envelope>
```

2774 Notice this is the same message as the one sent in step 2.

2775 **Step 6**—If there is a message pending for this destination then it is returned on the HTTP response:

```
2776 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"-  
2777 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200608"-  
2778 xmlns:wsa="http://www.w3.org/2005/08/addressing">  
2779   <S:Header>  
2780     <wsa:Action> http://example.org/eventType1 </wsa:Action>  
2781     <wsa:To>http://docs.oasis-open.org/ws-  
2782 rx/wsmr/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:To>
```

```

2783 <wsm:Sequence>
2784 <wsm:Identifier> http://example.org/rmid-456 </wsm:Identifier>
2785 </wsm:Sequence>
2786 <wsm:MessagePending pending="true"/>
2787 </S:Header>
2788 <S:Body>
2789 <!-- event specific data -->
2790 </S:Body>
2791 </S:Envelope>

```

2792 As noted in step 3, the use of the RM protocol does not change when using `MakeConnection`. The
2793 format of the messages, the order of the messages sent and the timing of when to send it remains the
2794 same.

2795 **Step 7**— At some later interval, or immediately due to the `MessagePending` header's "pending"
2796 attribute being set to "true", the event consumer will poll again:

```

2797 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2798 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200608"
2799 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2800 <S:Header>
2801 <wsa:Action>http://docs.oasis-open.org/ws-
2802 rx/wsm/200608/MakeConnection</wsa:Action>
2803 <wsa:To> http://example.org/subscriptionService </wsa:To>
2804 </S:Header>
2805 <S:Body>
2806 <wsm:MakeConnection>
2807 <wsm:Address>http://docs.oasis-open.org/ws-
2808 rx/wsm/200608/anonymous?id=550e8400-e29b-11d4-a716-
2809 446655440000</wsm:Address>
2810 </wsm:MakeConnection>
2811 </S:Body>
2812 </S:Envelope>

```

2813 Notice this is the same message as the one sent in steps 2 and 5. As in steps 3 and 6, the response to
2814 the `MakeConnection` can be any message destined to the specified Endpoint. This allows the event
2815 producer to send not only application messages but RM protocol messages (e.g. `CloseSequence`,
2816 `TerminateSequence` or even additional `CreateSequences`) as needed.

2817 **Step 8**— If at any point in time there are no messages pending, in response to a `MakeConnection` the
2818 event producer returns an HTTP 202 back to the event consumer. The process then repeats (back to step
2819 7) until the subscription ends.

2820 Appendix D. State Tables

2821 This appendix specifies the non-normative state transition tables for RM Source and RM Destination.

2822 The state tables describe the lifetime of a sequence in both the RM Source and the RM Destination

2823 Legend:

2824 The first column of these tables contains the motivating event and has the following format:

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

2825 Where:

2826 ● Event Name: indicates the name of the event. Event Names surrounded by "<>" are optional as
2827 described by the specification.

2828 ● [source]: indicates the source of the event; one of:

2829 ● [msg] a Received message

2830 ● [int]: an internal event such as the firing of a timer

2831 ● [app]: the application

2832 ● [unspec]: the source is unspecified

2833 Each event / state combination cell in the tables in this appendix has the following format:

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

2834 Where:

2835 ● action to take: indicates that the state machine performs the following action. Actions surrounded
2836 by "<>" are optional as described by the specification. "Xmit" is used as a short form for the word
2837 "Transmit"

2838 ● [next state]: indicates the state to which the state machine will advance upon the performance of
2839 the action. For ease of reading the next state "same" indicates that the state does not change.

2840 ● {ref} is a reference to the document section describing the behavior in this cell

2841 "N/A" in a cell indicates a state / event combination self-inconsistent with the state machine; should these
2842 conditions occur, it would indicate an implementation error. A blank cell indicates that the behavior is not
2843 described in this specification and does not indicate normal protocol operation. Implementations MAY
2844 generate a Sequence Terminated fault (see section 4.2) in these circumstances. Robust implementations
2845 MUST be able to operate in a stable manner despite the occurrence of unspecified event / state
2846 combinations.

2847 Table 1 RM Source Sequence State Transition Table

Events	Sequence States					
	None	Creating	Created	Closing	Closed	Terminating
Create Sequence [unspec] {3.4}	Xmit Create Sequence [Creating] {3.4}	N/A	N/A	N/A	N/A	N/A
Create Sequence Response [msg] {3.4}		Process Create Sequence Response [Created] {3.4}				
Create Sequence Refused Fault [msg] {3.4}		No action [None] {4.6}				
Send message [app] {2.1}	N/A	N/A	Xmit message [Same] {2}	No action [Same] {2}	N/A	N/A
Retransmit of un-ack'd message [int]	N/A	N/A	Xmit message [Same] {2.3}	Xmit message [Same] {2.3}	N/A	N/A
SeqAck (non-final) [msg] {3.9}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Process Ack ranges [Same] {3.9}	Process Ack ranges [Same] {3.9}	Process Ack ranges [Same] {3.9}	Process Ack ranges [Same] {3.9}
Nack [msg] {3.9}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	<Xmit message(s)> [Same] {3.9}	<Xmit message(s)> [Same] {3.9}	No action [Same]	No action [Same]
Message Number Rollover Fault [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]
CloseSequence [msg] {3.5}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Xmit CloseSequence Response [Closed] {3.5}	Xmit CloseSequence Response [Closed] {3.5}	Xmit CloseSequence Response [Closed] {3.5}	Generate Unknown Sequence Fault [Same] {4.3}
<Close Sequence> [int] {3.5}	N/A		Xmit Close Sequence [Closing] {3.5}	N/A	N/A	N/A
Close Sequence Response [msg] {3.5}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}		No action [Closed] {3.5}	No action [Same] {3.5}	No action [Same] {3.5}

Events	Sequence States					
	None	Creating	Created	Closing	Closed	Terminating
SeqAck (final) [msg] {3.9}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Process Ack ranges [Closed] {3.9}	Process Ack ranges [Closed] {3.9}	Process Ack ranges [Same]	Process Ack ranges [Same]
Sequence Closed Fault [msg] {4.7}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	No action [Closed] {4.7}	No action [Closed] {4.7}	No action [Same]	No action [Same]
Unknown Sequence Fault [msg] {4.3}			Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}
Sequence Terminated Fault [msg] {4.2}	N/A		Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.2}
TerminateSequence [msg] {3.6}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Xmit Terminate Sequence Response [None] {3.6}	Xmit Terminate Sequence Response [None] {3.6}	Xmit Terminate Sequence Response [None] {3.6}	Generate Unknown Sequence Fault [Same] {4.3}
Terminate Sequence [int]	N/A	No action [None] {unspec}	Xmit Terminate Sequence [Terminating]	Xmit Terminate Sequence [Terminating]	Xmit Terminate Sequence [Terminating]	N/A
Terminate Sequence Response [msg]	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}				Terminate Sequence [None] {3.6}
Expires exceeded [int]	N/A	Terminate Sequence [None] {3.7}	Terminate Sequence [None] {3.7}	Terminate Sequence [None] {3.7}	Terminate Sequence [None] {3.7}	Terminate Sequence [None] {3.7}
Invalid Acknowledgement [msg] {4.4}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Invalid Acknowledgemen t Fault [Same] {4.4}	Generate Invalid Acknowledgemen t Fault [Same] {4.4}	Generate Invalid Acknowledgemen t Fault [Same] {4.4}	Generate Invalid Acknowledgemen t Fault [Same] {4.4}
Events	Sequence States					
	None	Creating	Created	Closing	Closed	Terminating
Create- Sequence [unspec] {3.1}	Xmit Create- Sequence [Creating] {3.1}	N/A	N/A	N/A	N/A	N/A
Create- Sequence- Response [msg]	-	Process Create- Sequence- Response [Created]	-	-	-	-

Events	Sequence States					
	None	Creating	Created	Closing	Closed	Terminating
{3-1}		{3-1}				
Create-Sequence-Refused-Fault [msg] {3-1}	-	No action [None] {4.6}	-	-	-	-
Send-message [app] {2.1}	N/A	N/A	Xmit message [Same] {2}	No action [Same] {2}	N/A	N/A
Retransmit-of-un-ack'd-message [int]	N/A	N/A	Xmit message [Same] {2.4}	Xmit message [Same] {2.4}	N/A	N/A
SeqAck (non-final) [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}
Nack [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]
Message-Number-Rollover-Fault [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]
<Close-Sequence> [int] {3.2}	N/A	-	Xmit Close-Sequence [Closing] {3.2}	N/A	N/A	N/A
Close-Sequence-Response [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}
SeqAck (final) [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]
Sequence-Closed-Fault [msg] {4.7}	Generate-Unknown-Sequence-Fault [Same] {4.3}	Generate-Unknown-Sequence-Fault [Same] {4.3}	No action [Closed] {4.7}	No action [Closed] {4.7}	No action [Same]	No action [Same]
Unknown-Sequence-Fault [msg] {4.3}	-	-	Terminate-Sequence [None] {4.3}	Terminate-Sequence [None] {4.3}	Terminate-Sequence [None] {4.3}	Terminate-Sequence [None] {4.3}
Sequence-Terminated	N/A	-	Terminate-Sequence	Terminate-Sequence	Terminate-Sequence	Terminate-Sequence

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

2848 Table 2 RM Destination Sequence State Transition Table

Events	Sequence States			
	None	Created	Closed	Terminating
CreateSequence (successful) [msg/int] {3.4}	Xmit Create Sequence Response [Created] {3.4}	N/A	N/A	
CreateSequence (unsuccessful) [msg/int] {3.4}	Generate Create Sequence Refused Fault [None] {3.4}	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] {3.5}	Generate Sequence Terminated Fault [Same] {4.2}
Message (with message number outside of range) [msg]	Generate Unknown Sequence Fault [Same] {4.3}	Xmit Message Number Rollover Fault [Same] {3.7}{4.5}	Generate Sequence Closed Fault (with SeqAck+Final) [Same] {3.5}	Generate Sequence Terminated Fault [Same] {4.2}
<AckRequested> [msg] {3.8}	Generate Unknown Seq. Fault [Same] {4.3}	Xmit SeqAck [Same] {3.8}	Xmit SeqAck+Final [Same] {3.9}	Generate Sequence Terminated Fault [Same] {4.2}
CloseSequence [msg] {3.5}	Generate Unknown Sequence Fault [Same] {4.3}	Xmit CloseSequence Response with SeqAck+Final [Closed] {3.5}	Xmit CloseSequence Response with SeqAck+Final [Closed] {3.5}	Generate Sequence Terminated Fault [Same] {4.2}
<CloseSequence		Xmit CloseSequence	Xmit CloseSequence	

Events	Sequence States			
	None	Created	Closed	Terminating
autonomously> [int]		with SeqAck+Final [Closed] {3.5}	with SeqAck+Final [Same] {3.5}	
CloseSequenceResponse [msg] {3.5}	Generate Unknown Sequence Fault [Same] {4.3}		No Action [Closed] {3.5}	Generate Sequence Terminated Fault [Same] {4.2}
TerminateSequence [msg] {3.6}	Generate Unknown Sequence Fault [Same] {4.3}	Xmit Terminate Sequence Response [None] {3.6}	Xmit Terminate Sequence Response [None] {3.6}	Xmit Terminate Sequence Response [None] {3.6}
<TerminateSequence autonomously> [int]		Xmit TerminateSequence with SeqAck+Final [Terminating] {3.6}	Xmit TerminateSequence with SeqAck+Final [Terminating] {3.6}	Xmit TerminateSequence with SeqAck+Final [Terminating] {3.6}
TerminateSequenceResponse [msg]	Generate Unknown Sequence Fault [Same] {4.3}			Terminate Sequence [None]
UnknownSequence Fault [msg] {4.3}		Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}
SequenceTerminated Fault [msg] {4.2}		Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.3}
Invalid Acknowledgement Fault [msg] {4.4}	N/A			
Expires exceeded [int]	N/A	Terminate Sequence [None] {3.4}	Terminate Sequence [None] {3.4}	
<Seq Acknowledgement autonomously> [int] {3.9}	N/A	Xmit SeqAck [Same] {3.9}	Xmit SeqAck+Final [Same] {3.9}	
Non WSRM message when WSRM required [msg] {4.8}	Generate WSRMRequired Fault [Same] {4.8}	Generate WSRMRequired Fault [Same] {4.8}	Generate WSRMRequired Fault [Same] {4.8}	
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-	Generate Unknown Sequence-	Accept Message;	Generate Sequence Closed-	

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

2849 The following two tables apply only if the `MakeConnection` mechanism is utilized.

2850 Table 3 Sending Endpoint Message Transfer Engine

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

2851 Table 4 Receiving Endpoint Message Transfer Engine

Event	None	Polling
Expectation of unreceived message [int, unspecified]	No Action [Polling]	No Action [Same]
Polling trigger [int, unspecified]	-	Xmit MakeConnection [Polling] {3.7}

2852 **Appendix E. Acknowledgments**

2853 This document is based on initial contribution to OASIS WS-RX Technical Committee by the following
2854 authors:

2855 Ruslan Bilorusets(BEA), Don Box(Microsoft), Luis Felipe Cabrera(Microsoft), Doug Davis(IBM),
2856 Donald Ferguson(IBM), Christopher Ferris(~~l-Editor~~(BM), Tom Freund(IBM), Mary Ann
2857 Hondo(IBM), John Ibbotson(IBM), Lei Jin(BEA), Chris Kaler(Microsoft), David Langworthy-
2858 Editor(Microsoft), Amelia Lewis(TIBCO Software), Rodney Limprecht(Microsoft), Steve
2859 Lucco(Microsoft), Don Mullen(TIBCO Software), Anthony Nadalin(IBM), Mark Nottingham(BEA),
2860 David Orchard(BEA), Jamie Roots(IBM), Shivajee Samdarshi(TIBCO Software), John
2861 Shewchuk(Microsoft), Tony Storey(IBM).

2862 The following individuals have provided invaluable input into the initial contribution:

2863 Keith Ballinger(Microsoft), Stefan Batres(Microsoft), Rebecca Bergersen(Iona), Allen
2864 Brown(Microsoft), Michael Conner(IBM), George Copeland(Microsoft), Francisco Curbera(IBM),
2865 Paul Fremantle(IBM), Steve Graham(IBM), Pat Helland(Microsoft), Rick Hill(Microsoft), Scott
2866 Hinkelman(IBM), Tim Holloway(IBM), Efim Hudis(Microsoft), David Ingham(Microsoft), Gopal
2867 Kakivaya(Microsoft), Johannes Klein(Microsoft), Frank Leymann(IBM), Martin Nally(IBM), Peter
2868 Niblett(IBM), Jeffrey Schlimmer(Microsoft), James Snell(IBM), Keith Stobie(Microsoft), Satish
2869 Thatte(Microsoft), Stephen Todd(IBM), Sanjiva Weerawarana(IBM), Roger Wolter(Microsoft).

2870 The following individuals were members of the committee during the development of this specification:

2871 Abbie Barbir(Nortel), Charlton Barreto(Adobe), Stefan Batres(Microsoft), Hamid Ben
2872 Malek(Fujitsu), Andreas Bjarlestam(Ericsson), Toufic Boubez(Layer 7), Doug Bunting(Sun), Lloyd
2873 Burch(Novell), Steve Carter(Novell), Martin Chapman(Oracle), Dave Chappell(Sonic), Paul
2874 Cotton(Microsoft), Glen Daniels(Sonic), Doug Davis(IBM), Blake Dournaee(Intel), Jacques
2875 Durand(Fujitsu), Colleen Evans(Microsoft), Christopher Ferris(IBM), Paul Fremantle(WSO2),
2876 Robert Freund(Hitachi), Peter Furniss(Erebor), Marc Goodner(Microsoft), Alastair
2877 Green(Choreology), Mike Grogan(Sun), Ondrej Hrebicek(Microsoft), Kazunori Iwasa(Fujitsu),
2878 Chamikara Jayalath(WSO2), Lei Jin(BEA), Ian Jones(BTplc), Anish Karmarkar(Oracle), Paul
2879 Knight(Nortel), Dan Leshchiner(Tibco), Mark Little(JBoss), Lily Liu(webMethods), Matt
2880 Lovett(IBM), Ashok Malhotra(Oracle), Jonathan Marsh(Microsoft), Daniel Millwood(IBM), Jeff
2881 Mischkinsky(Oracle), Nilo Mitra(Ericsson), Peter Niblett(IBM), Duane Nickull(Adobe), Eisaku
2882 Nishiyama(Hitachi), Dave Orchard(BEA), Chouthri Palanisamy(NEC), Sanjay Patil(SAP), Gilbert
2883 Pilz(BEA), Martin Raeppele(SAP), Eric Rajkovic(Oracle), Stefan Rossmannith(SAP), Tom
2884 Rutt(Fujitsu), Rich Salz(IBM), Shivajee Samdarshi(Tibco), Vladimir Videlov(SAP), Claus von
2885 Riegen(SAP), Pete Wenzel(Sun), Steve Winkler(SAP), Ümit Yalçinalp(SAP), Nobuyuki
2886 Yamamoto(Hitachi).

Appendix F. Revision History

Rev	Date	By Whom	What
wd-01	2005-07-07	Christopher Ferris	Initial version created based on submission by the authors.
ws-02	2005-07-21	Doug Davis	I011 (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 'optiona'/'required' to 'OPTIONAL'/'REQUIRED') added
wd-05	2005-09-30	Anish Karmarkar	i017 (Change NS to http://docs.oasis-open.org/wsrn/200510/)
wd-05	2005-09-30	Anish Karmarkar	i045 (Include SecureConversation as a reference and move it to non-normative citation)
wd-05	2005-09-30	Anish Karmarkar	i046 (change the type of wsrn:FaultCode element)
wd-06	2005-11-02	Gilbert Pilz	Start wd-06 by changing title page from cd-01.
wd-06	2005-11-03	Gilbert Pilz	i047 (Reorder spec sections)
wd-07	2005-11-17	Gilbert Pilz	Start wd-07
wd-07	2005-11-28	Doug Davis	i071 – except for period in Appendix headings
wd-07	2005-11-28	Doug Davis	i10
wd-07	2005-11-28	Doug Davis	i030
wd-07	2005-11-28	Doug Davis	i037
wd-07	2005-11-28	Doug Davis	i038
wd-07	2005-11-28	Doug Davis	i041
wd-07	2005-11-28	Doug Davis	i043
wd-07	2005-11-28	Doug Davis	i044

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

<u>Rev</u>	<u>Date</u>	<u>By Whom</u>	<u>What</u>
			<u>OpenDocument Text format. Changed line numbers to be a single style.</u>
<u>wd-08</u>	<u>2005-12-28</u>	<u>Anish Karmarkar</u>	<u>Included a section link to c:\temp\wsrm-1.1-schema-200510.xsd and to c:\temp\wsrm-1.1-wsdl-200510.wsdl</u>
<u>wd-08</u>	<u>2006-01-04</u>	<u>Gilbert Pilz</u>	<u>Fixed formatting for included sections.</u>
<u>wd-08</u>	<u>2006-01-05</u>	<u>Gilbert Pilz</u>	<u>Created links for unused references. Fixed exemplars for CloseSequence and CloseSequenceResponse.</u>
<u>wd-09</u>	<u>2006-01-11</u>	<u>Doug Davis</u>	<u>Minor tweaks to text/typos.</u>
<u>wd-10</u>	<u>2006-01-23</u>	<u>Doug Davis</u>	<u>Accept all changes from wd-09</u> <u>Make some minor editorial tweaks from Marc's comments.</u>
<u>wd-10</u>	<u>2006-02-14</u>	<u>Doug Davis</u>	<u>Issue 082 resolution</u>
<u>wd-10</u>	<u>2006-02-14</u>	<u>Doug Davis</u>	<u>Issue 083 resolution</u>
<u>wd-10</u>	<u>2006-02-14</u>	<u>Doug Davis</u>	<u>Issue 085 resolution</u>
<u>wd-10</u>	<u>2006-02-14</u>	<u>Doug Davis</u>	<u>Issues 086, 087 resolutions</u> <u>Defined MessageNumberType</u>
<u>wd-10</u>	<u>2006-02-15</u>	<u>Doug Davis</u>	<u>Issue 078 resolution</u>
<u>wd-10</u>	<u>2006-02-15</u>	<u>Doug Davis</u>	<u>Issue 094 resolution</u>
<u>wd-10</u>	<u>2006-02-15</u>	<u>Doug Davis</u>	<u>Issue 095 resolution</u>
<u>wd-10</u>	<u>2006-02-15</u>	<u>Gilbert Pilz</u>	<u>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</u>
<u>wd-10</u>	<u>2006-02-17</u>	<u>Anish Karmarkar</u>	<u>Namespace changed to 200602 for both WSDL and XSD docs.</u>
<u>wd-10</u>	<u>2006-02-17</u>	<u>Anish Karmarkar</u>	<u>Issue i087 as it applies to WSRM spec.</u>
<u>wd-10</u>	<u>2006-02-17</u>	<u>Anish Karmarkar</u>	<u>Added titles and minor text for state table (issue i058).</u>
<u>wd-11</u>	<u>2006-02-22</u>	<u>Doug Davis</u>	<u>Accept all changes for new WD</u> <u>Minor typos fixed</u>
<u>wd-11</u>	<u>2006-02-23</u>	<u>Doug Davis</u>	<u>s'/close'/close/g – per Marc Goodner</u> <u>Added first ref to [URI] – per Marc G again</u>
<u>wd-11</u>	<u>2006-02-27</u>	<u>Doug Davis</u>	<u>Issue i061 applied</u>
<u>wd-11</u>	<u>2006-02-28</u>	<u>Doug Davis</u>	<u>Fixed typo around the use of "above" and "below"</u>
<u>wd-11</u>	<u>2006-03-01</u>	<u>Doug Davis</u>	<u>Minor typos found by Marc Goodner</u>
<u>wd-11</u>	<u>2006-03-02</u>	<u>Doug Davis</u>	<u>Minor typos found by Matt Lovett</u>
<u>wd-11</u>	<u>2006-03-08</u>	<u>Doug Davis</u>	<u>Issue 091 applied</u>
<u>wd-11</u>	<u>2006-03-08</u>	<u>Doug Davis</u>	<u>Issue 092 applied</u>
<u>wd-11</u>	<u>2006-03-08</u>	<u>Doug Davis</u>	<u>Issue 100 applied</u>

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

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.
wd-16	2006-10-25	Doug Davis	Accept all changes, update to wd16
wd-16	2006-10-26	Doug Davis	PR002 applied
wd-16	2006-10-26	Doug Davis	PR003 applied
wd-16	2006-10-26	Doug Davis	PR004 applied
wd-16	2006-10-27	Doug Davis	PR005 applied
wd-16	2006-10-27	Doug Davis	PR006 applied
wd-16	2006-10-27	Doug Davis	PR024 applied
wd-16	2006-11-13	Doug Davis	PR010 applied
wd-16	2006-11-13	Doug Davis	PR011 applied (technically as part of PR004)
wd-16	2006-11-13	Doug Davis	PR016 applied
wd-16	2006-11-13	Doug Davis	PR032 applied
wd-16	2006-11-20	Doug Davis	PR025 applied
wd-16	2006-11-20	Doug Davis	PR023 applied
wd-16	2006-12-03	Doug Davis	PR036 applied
wd-16	2006-12-03	Doug Davis	PR017 applied
wd-16	2006-12-11	Doug Davis	PR012 applied (and PR013)
wd-16	2006-12-14	Doug Davis	PR033 applied – changed a 'return' to 'generate' when talking about a fault
wd-16	2007-01-04	Doug Davis	PR018 applied
wd-16	2007-01-05	Doug Davis	Moved MakeConnection to new spec
wd-16	2007-01-17	Doug Davis	PR026 applied
wd-16	2007-01-18	Doug Davis	PR021 applied
wd-16	2007-01-18	Doug Davis	PR022 applied
wd-16	2007-01-18	Doug Davis	Fixed a few typos (Doug.Gil)
wd-16	2007-01-18	Gilbert Pilz	PR007 applied
wd-16	2007-01-25	Doug Davis	PR039 applied
wd-17	2007-01-31	Doug Davis	Lots of typos from MarcG Updated WD number and date
wd-17	2007-02-01	Doug Davis	PR038 applied
wd-17	2007-02-01	Doug Davis	PR035 (009,020 dups) applied Fixed typo in state table
Rev	Date	By Whom	What
wd-01	2005-07-07	Christopher Ferris	Initial version created based on submission by the authors.-
ws-02	2005-07-21	Doug Davis	I011 (PT0S) added
wd-02	2005-08-16	Anish Karmarkar	Trivial editorial changes
ws-03	2005-09-15	Doug Davis	I019 and i028 (CloseSeq) added

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

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

Rev	Date	By Whom	What
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 Piltz	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).
wd-11	2006-02-22	Doug Davis	Accept all changes for new WD Minor types fixed
wd-11	2006-02-23	Doug Davis	s/'close'/close/g — per Marc Goodner Added first ref to [URI] — per Marc G again
wd-11	2006-02-27	Doug Davis	Issue i061 applied
wd-11	2006-02-28	Doug Davis	Fixed typo around the use of "above" and "below"
wd-11	2006-03-01	Doug Davis	Minor types found by Marc Goodner
wd-11	2006-03-02	Doug Davis	Minor types found by Matt Lovett
wd-11	2006-03-08	Doug Davis	Issue 091 applied
wd-11	2006-03-08	Doug Davis	Issue 092 applied
wd-11	2006-03-08	Doug Davis	Issue 100 applied
wd-12	2006-03-20	Doug Davis	Added space in "SOAP1.x" — Paul Cotton
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

Rev	Date	By Whom	What
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
wd-15	2006-06-14	Doug Davis	Remove blank rows/columns from state table. Fix italics in state table
wd-15	2006-06-15	Doug Davis	Type — section D was empty
wd-15	2006-06-16	Doug Davis	Issue 125 applied
wd-15	2006-06-16	Doug Davis	Issue 126 applied
wd-15	2006-06-16	Doug Davis	Issue 127 applied
wd-15	2006-06-16	Doug Davis	Issue 133 applied
wd-15	2006-06-16	Doug Davis	Issue 136 applied

Rev	Date	By-Whom	What
wd-15	2006-06-16	Doug-Davis	Issue-138 applied
wd-15	2006-06-16	Doug-Davis	Issue-135 applied
wd-15	2006-06-20	Doug-Davis	Added all-TC-members to the ack list
wd-15	2006-06-22	Doug-Davis	Issue-129 applied
wd-15	2006-06-22	Doug-Davis	Issue-130 applied
wd-15	2006-06-22	Doug-Davis	Issue-137 applied
wd-15	2006-06-26	Doug-Davis	Issue-111 applied
wd-15	2006-06-26	Doug-Davis	Missed a part of issue-129
wd-15	2006-06-30	Doug-Davis	Fixed a typo in schema
wd-15	2006-06-30	Doug-Davis	Issue-141 applied
wd-15	2006-06-30	Doug-Davis	Issue-142 applied
wd-15	2006-06-30	Doug-Davis	Issue-148 applied
wd-15	2006-06-30	Doug-Davis	Issue-149 applied
wd-15	2006-06-30	Doug-Davis	Issue-150 applied
wd-15	2006-07-06	Doug-Davis	Issue-121 applied
wd-15	2006-07-21	Doug-Davis	Issue-139 applied
wd-15	2006-07-21	Doug-Davis	Issue-144 applied
wd-15	2006-07-21	Doug-Davis	Issue-147 applied
wd-15	2006-07-21	Doug-Davis	Issues-122-124 applied
wd-15	2006-07-27	Doug-Davis	Updated list of oasis-TC-members (i134)
wd-15	2006-07-27	Doug-Davis	Issue-140 applied
wd-15	2006-07-27	Doug-Davis	Issue-145 applied
wd-15	2006-07-27	Doug-Davis	Issue-143 applied
wd-15	2006-07-28	Doug-Davis	Lots of minor typos found by Matt L.
wd-15	2006-07-28	Doug-Davis	Issue-113 applied
wd-15	2006-08-04	Doug-Davis	Update old namespaces — found by PaulG
wd-15	2006-08-04	Doug-Davis	Issue-150 applied
wd-15	2006-08-04	Doug-Davis	Minor typos — found by PeterN
wd-15	2006-08-04	Doug-Davis	Verify all [refs]
wd-15	2006-08-04	Doug-Davis	Change namespace to 2006/08
wd-15	2006-08-04	Doug-Davis	Issue-148 applied
wd-15	2006-08-07	Doug-Davis	Add some new glossary terms — per GilP
cd-04	2006-08-10	Gilbert Pitz	Formatting changes for better HTML-rendering.
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 Pitz	More formatting changes for better HTML-rendering.

2888 **Appendix G. Notices**

2889 ~~OASIS takes no position regarding the validity or scope of any intellectual property or other rights that~~
2890 ~~might be claimed to pertain to the implementation or use of the technology described in this document or~~
2891 ~~the extent to which any license under such rights might or might not be available; neither does it represent~~
2892 ~~that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to~~
2893 ~~rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made~~
2894 ~~available for publication and any assurances of licenses to be made available, or the result of an attempt~~
2895 ~~made to obtain a general license or permission for the use of such proprietary rights by implementors or~~
2896 ~~users of this specification, can be obtained from the OASIS Executive Director.~~

2897 ~~OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or~~
2898 ~~other proprietary rights which may cover technology that may be required to implement this specification.~~
2899 ~~Please address the information to the OASIS Executive Director.~~

2900 ~~Copyright (C) OASIS Open (2006). All Rights Reserved.~~

2901 ~~This document and translations of it may be copied and furnished to others, and derivative works that~~
2902 ~~comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and~~
2903 ~~distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and~~
2904 ~~this paragraph are included on all such copies and derivative works. However, this document itself may~~
2905 ~~not be modified in any way, such as by removing the copyright notice or references to OASIS, except as~~
2906 ~~needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights~~
2907 ~~defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it~~
2908 ~~into languages other than English.~~

2909 ~~The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors~~
2910 ~~or assigns.~~

2911 ~~This document and the information contained herein is provided on an "AS-IS" basis and OASIS~~
2912 ~~DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY~~
2913 ~~WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR~~
2914 ~~ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.~~