



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/wsrn-1.1-spec-cd-05.pdf>

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

21 **Latest Approved Version:**

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

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

24 **Technical Committee:**

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

26 **Chairs:**

27 [Paul Fremantle <paul@wso2.com>](#)

28 [Sanjay Patil <sanjay.patil@sap.com>](#)

29 **Editors:**

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

31 [Anish Karmarkar, Oracle <Anish.Karmarkar@oracle.com>](#)

32 [Gilbert Pilz, BEA <gpilz@bea.com>](#)

33 [Steve Winkler, SAP <steve.winkler@sap.com>](#)

35 **Contributors:**

36 See the Acknowledgments (Appendix E).

37 **Abstract:**

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

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

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

53 **Status:**

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

66 Technical Committee members should send comments on this specification to the Technical
67 Committee's email list. Others should send comments to the Technical Committee by using the
68 "Send A Comment" button on the Technical Committee's web page at [http://www.oasis-](http://www.oasis-open.org/committees/ws-rx)
69 [open.org/committees/ws-rx](http://www.oasis-open.org/committees/ws-rx).

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

74 The non-normative errata page for this specification is located at [http://www.oasis-](http://www.oasis-open.org/committees/ws-rx)
75 [open.org/committees/ws-rx](http://www.oasis-open.org/committees/ws-rx).

Notices

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

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

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

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

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

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

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

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

The names "OASIS", [insert specific trademarked names, abbreviations, etc. here] are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications,

119 while reserving the right to enforce its marks against misleading uses. Please see [open.org/who/trademark.php](http://www.oasis-</u>
120 <u><a href=) for above guidance.

Table of Contents

| | | |
|-----|-----|---|
| 121 | | |
| 122 | 1 | Introduction..... 7 |
| 123 | 1.1 | Terminology..... 7 |
| 124 | 1.2 | Namespace..... 8 |
| 125 | 1.3 | Conformance..... 8 |
| 126 | 2 | Reliable Messaging Model..... 9 |
| 127 | 2.1 | Glossary..... 9 |
| 128 | 2.2 | Protocol Preconditions..... 10 |
| 129 | 2.3 | Protocol Invariants..... 11 |
| 130 | 2.4 | Delivery Assurances..... 11 |
| 131 | 2.5 | Example Message Exchange..... 12 |
| 132 | 3 | RM Protocol Elements..... 14 |
| 133 | 3.1 | Considerations on the Use of Extensibility Points..... 14 |
| 134 | 3.2 | Considerations on the Use of "Piggy-Backing"..... 14 |
| 135 | 3.3 | Composition with WS-Addressing..... 14 |
| 136 | 3.4 | Sequence Creation..... 15 |
| 137 | 3.5 | Closing A Sequence..... 19 |
| 138 | 3.6 | Sequence Termination..... 21 |
| 139 | 3.7 | Sequences..... 23 |
| 140 | 3.8 | Request Acknowledgement..... 24 |
| 141 | 3.9 | Sequence Acknowledgement..... 25 |
| 142 | 4 | Faults..... 28 |
| 143 | 4.1 | SequenceFault Element..... 29 |
| 144 | 4.2 | Sequence Terminated..... 30 |
| 145 | 4.3 | Unknown Sequence..... 30 |
| 146 | 4.4 | Invalid Acknowledgement..... 31 |
| 147 | 4.5 | Message Number Rollover..... 31 |
| 148 | 4.6 | Create Sequence Refused..... 32 |
| 149 | 4.7 | Sequence Closed..... 32 |
| 150 | 4.8 | WSRM Required..... 33 |
| 151 | 5 | Security Threats and Countermeasures..... 34 |
| 152 | 5.1 | Threats and Countermeasures..... 34 |
| 153 | 5.2 | Security Solutions and Technologies..... 36 |
| 154 | 6 | Securing Sequences..... 40 |
| 155 | 6.1 | Securing Sequences Using WS-Security..... 40 |
| 156 | 6.2 | Securing Sequences Using SSL/TLS..... 41 |
| 157 | 7 | References..... 43 |
| 158 | 7.1 | Normative..... 43 |

| | | |
|-----|---|----|
| 159 | 7.2 Non-Normative | 44 |
| 160 | Appendix A. Schema..... | 46 |
| 161 | Appendix B. WSDL..... | 51 |
| 162 | Appendix C. Message Examples..... | 53 |
| 163 | Appendix C.1 Create Sequence..... | 53 |
| 164 | Appendix C.2 Initial Transmission..... | 53 |
| 165 | Appendix C.3 First Acknowledgement..... | 55 |
| 166 | Appendix C.4 Retransmission..... | 55 |
| 167 | Appendix C.5 Termination..... | 56 |
| 168 | Appendix D. State Tables..... | 58 |
| 169 | Appendix E. Acknowledgments..... | 63 |
| 170 | Appendix F. Revision History..... | 64 |

1 Introduction

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

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

1.1 Terminology

1.1 Notational Conventions

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

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

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

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

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

2 Reliable Messaging Model

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

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

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

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

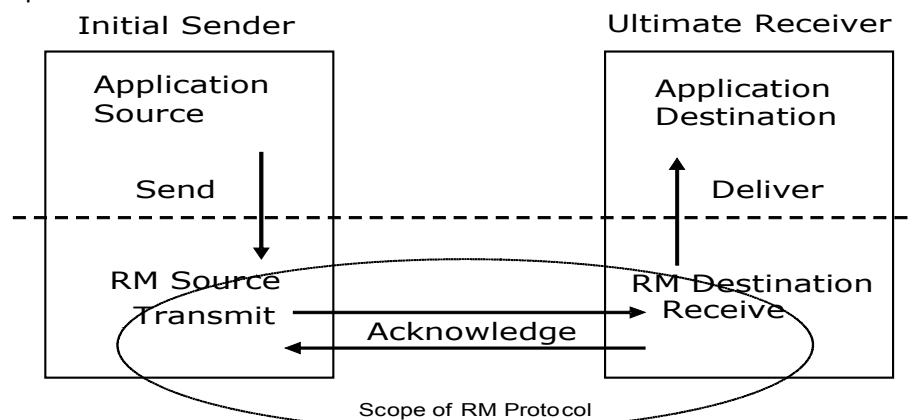


Figure 1: Reliable Messaging Model

2.1 Glossary

The following definitions are used throughout this specification:

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

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

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

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

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

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

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

273 **Deliver:** The act of transferring responsibility for a message from the RM Destination to the Application
 274 Destination.

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

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

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

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

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

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

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

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

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

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

292 2.2 Protocol Preconditions

- 293 The correct operation of the protocol requires that a number of preconditions MUST be established prior
 294 to the processing of the initial sequenced message:
- 295 • For any single message exchange the RM Source MUST have an endpoint reference that uniquely
 296 identifies the RM Destination Endpoint.
 - 297 • The RM Source MUST have successfully created a Sequence with the RM Destination.
 - 298 • The RM Source MUST be capable of formulating messages that adhere to the RM Destination's
 299 policies.

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

302 2.3 Protocol Invariants

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

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

316 2.4 Delivery Assurances

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

320 AtLeastOnce

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

327 AtMostOnce

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

332 ExactlyOnce

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

340 InOrder

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

351 2.5 Example Message Exchange

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



Figure 2: The WS-ReliableMessaging Protocol

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

357 4. The RM Source begins Transmitting messages in the Sequence beginning with MessageNumber 1.
358 In the figure above, the RM Source sends 3 messages in the Sequence.

359 5. The 2nd message in the Sequence is lost in transit.

360 6. The 3rd message is the last in this Sequence and the RM Source includes an AckRequested
361 header to ensure that it gets a timely SequenceAcknowledgement for the Sequence.

362 7. The RM Destination acknowledges receipt of message numbers 1 and 3 as a result of receiving the
363 RM Source's AckRequested header.

364 8. The RM Source retransmits the unacknowledged message with MessageNumber 2. This is a new
365 message from the perspective of the underlying transport, but it has the same Sequence Identifier
366 and MessageNumber so the RM Destination can recognize it as a duplicate of the earlier message,
367 in case the original and retransmitted messages are both Received. The RM Source includes an
368 AckRequested header in the retransmitted message so the RM Destination will expedite an
369 acknowledgement.

370 9. The RM Destination Receives the second transmission of the message with MessageNumber 2
371 and acknowledges receipt of message numbers 1, 2, and 3.

372 10. The RM Source Receives this Acknowledgement and sends a TerminateSequence message to the
373 RM Destination indicating that the Sequence is completed. The TerminateSequence message
374 indicates that message number 3 was the last message in the Sequence. The RM Destination then
375 and reclaims any resources associated with the Sequence.

376 11. The RM Destination Receives the TerminateSequence message indicating that the RM Source will
377 not be sending any more messages. The RM Destination sends a TerminateSequenceResponse
378 message to the RM Source and and reclaims any resources associated with the Sequence.

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

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

3 RM Protocol Elements

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

3.1 Considerations on the Use of Extensibility Points

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

3.2 Considerations on the Use of "Piggy-Backing"

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

3.3 Composition with WS-Addressing

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

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

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

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

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


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

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

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

3.4 Sequence Creation

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

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

The following exemplar defines the `CreateSequence` syntax:

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

The following describes the content model of the `CreateSequence` element.

`/wsrm:CreateSequence`

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

`/wsrm:CreateSequence/wsrm:AcksTo`

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

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

476 /wsrm:CreateSequence/wsrm:Expires

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

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

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

484 /wsrm:CreateSequence/wsrm:Offer

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

487 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier

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

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

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

493 /wsrm:CreateSequence/wsrm:Offer/wsrm:Endpoint

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

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

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

515 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires

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

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

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

522 `/wsrm:CreateSequence/wsrm:Offer/wsrm:IncompleteSequenceBehavior`

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

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

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

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

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

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

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

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

539 `/wsrm:CreateSequence/{any}`

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

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

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

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

549 The following exemplar defines the `CreateSequenceResponse` syntax:

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

```

559     </wsrm:Accept> ?
560     ...
561 </wsrm:CreateSequenceResponse>

```

562 The following describes the content model of the `CreateSequenceResponse` element.

563 `/wsrm:CreateSequenceResponse`

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

567 `/wsrm:CreateSequenceResponse/wsrm:Identifier`

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

571 `/wsrm:CreateSequenceResponse/wsrm:Identifier/@{any}`

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

574 `/wsrm:CreateSequenceResponse/wsrm:Expires`

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

583 `/wsrm:CreateSequenceResponse/wsrm:Expires/@{any}`

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

586 `/wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior`

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

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

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

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

597 `/wsrm:CreateSequenceResponse/wsrm:Accept`

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

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

603 `/wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo`

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

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

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

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

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

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

618 `/wsrm:CreateSequenceResponse/{any}`

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

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

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

624 3.5 Closing A Sequence

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

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

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

641 RM Destination can use this information, for example, to implement the behavior indicated by
642 /wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior. The value of the
643 LastMsgNumber element MUST be the same in all the CloseSequence messages for the closing
644 Sequence.
645 If the RM Destination decides to close a Sequence of its own volition, it MAY inform the RM Source of this
646 event by sending a CloseSequence element, in the body of a message, to the AcksTo EPR of that
647 Sequence. The RM Destination MUST include a final SequenceAcknowledgement (within which the RM
648 Destination MUST include the Final element) header block in this message and any subsequent
649 messages associated with the Sequence destined to the RM Source.

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

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

658 The following exemplar defines the CloseSequence syntax:

```
659 <wsrm:CloseSequence ...>  
660   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
661   <wsrm>LastMsgNumber> wsrm:MessageNumberType </wsrm>LastMsgNumber> ?  
662   ...  
663 </wsrm:CloseSequence>
```

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

665 /wsrm:CloseSequence

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

671 /wsrm:CloseSequence/wsrm:Identifier

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

677 /wsrm:CloseSequence/wsrm>LastMessageNumber

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

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

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

684 /wsrm:CloseSequence/{any}

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

687 /wsrm:CloseSequence@{any}

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

690 A CloseSequenceResponse is sent in the body of a message in response to receipt of a
691 CloseSequence request message. It indicates that the responderresponse message by an RM-
692 Destination in response to receipt of a CloseSequence request message. It indicates that the RM-
693 Destination has closed the Sequence.

694 The following exemplar defines the CloseSequenceResponse syntax:

```
695 <wsrm:CloseSequenceResponse ...>  
696   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
697   ...  
698 </wsrm:CloseSequenceResponse>
```

699 The following describes the content model of the CloseSequenceResponse element.

700 /wsrm:CloseSequenceResponse

701 This element is sent in the body of a message in response to receipt of a CloseSequence request
702 message. It indicates that the responderresponse message by an RM-Destination in response to receipt-
703 of a CloseSequence request message. It indicates that the RM-Destination has closed the Sequence.

704 /wsrm:CloseSequenceResponse/wsrm:Identifier

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

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

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

713 /wsrm:CloseSequenceResponse/{any}

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

716 /wsrm:CloseSequenceResponse@{any}

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

719 3.6 Sequence Termination

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

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

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

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

The following exemplar defines the `TerminateSequence` syntax:

```
<wsrm:TerminateSequence ...>
  <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
  <wsrm>LastMsgNumber> wsrm:MessageNumberType </wsrm>LastMsgNumber> ?
  ...
</wsrm:TerminateSequence>
```

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

`/wsrm:TerminateSequence`

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

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

`/wsrm:TerminateSequence/wsrm:Identifier`

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

`/wsrm:TerminateSequence/wsrm>LastMsgNumber`

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

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

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

769 /wsrm:TerminateSequence/{any}

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

772 /wsrm:TerminateSequence/@{any}

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

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

779 The following exemplar defines the TerminateSequenceResponse syntax:

```
780 <wsrm:TerminateSequenceResponse ...>  
781   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
782   ...  
783 </wsrm:TerminateSequenceResponse>
```

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

785 /wsrm:TerminateSequenceResponse

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

791 /wsrm:TerminateSequenceResponse/wsrm:Identifier

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

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

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

801 /wsrm:TerminateSequenceResponse/{any}

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

804 /wsrm:TerminateSequenceResponse/@{any}

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

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

3.7 Sequences

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

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

A following exemplar defines its syntax:

```
<wsrm:Sequence ...>
  <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
  <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>
  ...
</wsrm:Sequence>
```

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

/wsrm:Sequence

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

/wsrm:Sequence/wsrm:Identifier

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

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

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

/wsrm:Sequence/wsrm:MessageNumber

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

/wsrm:Sequence/{any}

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

/wsrm:Sequence/@{any}

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

The following example illustrates a Sequence header block.


```

851 <wsrm:Sequence>
852   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
853   <wsrm:MessageNumber>10</wsrm:MessageNumber>
854 </wsrm:Sequence>

```

3.8 Request Acknowledgement

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

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

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

The following exemplar defines its syntax:

```

878 <wsrm:AckRequested ...>
879   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
880   ...
881 </wsrm:AckRequested>

```

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

`/wsrm:AckRequested`

This element requests an Acknowledgement for the identified Sequence.

`/wsrm:AckRequested/wsrm:Identifier`

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

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

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

`/wsrm:AckRequested/{any}`

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

895 /wsrm:AckRequested/@{any}

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

898 3.9 Sequence Acknowledgement

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

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

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

929 The following exemplar defines its syntax:

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

939 ...
940 </wsrm:SequenceAcknowledgement>

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

942 `/wsrm:SequenceAcknowledgement`

943 This element contains the Sequence Acknowledgement information.

944 `/wsrm:SequenceAcknowledgement/wsrm:Identifier`

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

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

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

953 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

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

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

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

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

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

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

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

968 `/wsrm:SequenceAcknowledgement/wsrm:None`

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

973 `/wsrm:SequenceAcknowledgement/wsrm:Final`

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

980 /wsrm:SequenceAcknowledgement/wsrm:Nack

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

992 /wsrm:SequenceAcknowledgement/{any}

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

995 /wsrm:SequenceAcknowledgement/@{any}

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

998 The following examples illustrate SequenceAcknowledgement elements:

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

```
1000 <wsrm:SequenceAcknowledgement>
1001   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
1002   <wsrm:AcknowledgementRange Upper="10" Lower="1"/>
1003 </wsrm:SequenceAcknowledgement>
```

- 1004 • Message numbers 1..2, 4..6, and 8..10 inclusive in a Sequence have been accepted by the RM
1005 Destination, messages 3 and 7 have not been accepted.

```
1006 <wsrm:SequenceAcknowledgement>
1007   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
1008   <wsrm:AcknowledgementRange Upper="2" Lower="1"/>
1009   <wsrm:AcknowledgementRange Upper="6" Lower="4"/>
1010   <wsrm:AcknowledgementRange Upper="10" Lower="8"/>
1011 </wsrm:SequenceAcknowledgement>
```

- 1012 • Message number 3 in a Sequence has not been accepted by the RM Destination.

```
1013 <wsrm:SequenceAcknowledgement>
1014   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
1015   <wsrm:Nack>3</wsrm:Nack>
1016 </wsrm:SequenceAcknowledgement>
```

1017 1.2 MakeConnection

1018 When an Endpoint is not directly addressable (e.g. behind a firewall or not able to allow incoming-
1019 connections), an anonymous URI in the EPR address property can indicate such an Endpoint. The WS-
1020 Addressing anonymous URI is one such anonymous URI. This specification defines a URI template (the
1021 WS-RM anonymous URI) which may be used to uniquely identify anonymous Endpoints.

```
1022 http://docs.oasis-open.org/ws-rx/wsrm/200608/anonymous?id={uuid}
```

1023 This URI template in an EPR indicates a protocol-specific back-channel will be established through a
1024 mechanism such as `MakeConnection`, defined below. When using this URI template, "{uuid}" MUST be
1025 replaced by a UUID value as defined by RFC4122[UUID]. This UUID value uniquely distinguishes the
1026 Endpoint. A sending Endpoint SHOULD Transmit messages at Endpoints identified with the URI template
1027 using a protocol-specific back-channel, including but not limited to those established with a
1028 `MakeConnection` message. Note, this URI is semantically similar to the WS-Addressing anonymous-
1029 URI if a protocol-specific back-channel is available.

1030 The `MakeConnection` is a one-way operation that establishes a contextualized back-channel for the
1031 transmission of messages according to matching criteria (defined below). In the non-faulting case, if no
1032 matching message is available then no SOAP envelopes will be returned on the back-channel. A common
1033 usage will be a client RM Destination sending `MakeConnection` to a server RM Source for the purpose
1034 of receiving asynchronous response messages.

1035 The following exemplar defines the `MakeConnection` syntax:

```
1036 <wsrm:MakeConnection ...>  
1037   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier> ?  
1038   <wsrm:Address ...> xs:anyURI </wsrm:Address> ?  
1039   ...  
1040 </wsrm:MakeConnection>
```

1041 `/wsrm:MakeConnection`

1042 This element allows the sender to create a transport-specific back-channel that can be used to return a
1043 message that matches the selection criteria. Endpoints MUST NOT send this element as a header block.

1044 `/wsrm:MakeConnection/wsrm:Identifier`

1045 This element specifies the WS-RM Sequence Identifier that establishes the context for the transport-
1046 specific back-channel. The Sequence Identifier should be compared with the Sequence Identifiers
1047 associated with the messages held by the sending Endpoint, and if there is a matching message it will be
1048 returned. If this element is omitted from the message then the `Address` MUST be included in the
1049 message.

1050 `/wsrm:MakeConnection/wsrm:Identifier/@{any}`

1051 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
1052 element.

1053 `/wsrm:MakeConnection/wsrm:Address`

1054 This element specifies the URI (`wsa:Address`) of the initiating Endpoint. Endpoints MUST NOT return
1055 messages on the transport-specific back-channel unless they have been addressed to this URI. This
1056 `Address` property and a message's WS-Addressing destination property are considered identical when
1057 they are exactly the same character-for-character. Note that URIs which are not identical in this sense
1058 may in fact be functionally equivalent. Examples include URI references which differ only in case, or
1059 which are in external entities which have different effective base URIs. If this element is omitted from the
1060 message then the `Identifier` MUST be included in the message.

1061 `/wsrm:MakeConnection/wsrm:Address/@{any}`

1062 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
1063 element.

1064 `/wsrm:MakeConnection/{any}`

1065 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
1066 to be passed. This allows fine-tuning of the messages to be returned, additional selection criteria included

1067 ~~here are logically ANDed with the Address and/or Identifier. If an extension is not supported by the~~
1068 ~~Endpoint then it should return a UnsupportedSelection fault.~~

1069 ~~/wsrm:MakeConnection/@{any}~~

1070 ~~This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the~~
1071 ~~element.~~

1072 ~~If both Identifier and Address are present, then the Endpoint processing the MakeConnection~~
1073 ~~message MUST insure that any SOAP Envelope flowing on the backchannel MUST be associated with~~
1074 ~~the given Sequence and MUST be addressed to the given URI.~~

1075 ~~The management of messages that are awaiting the establishment of a back-channel to their receiving~~
1076 ~~Endpoint is an implementation detail that is outside the scope of this specification. Note, however, that~~
1077 ~~these messages form a class of asynchronous messages that is not dissimilar from "ordinary"~~
1078 ~~asynchronous messages that are waiting for the establishment of a connection to their destination~~
1079 ~~Endpoints.~~

1080 ~~This specification places no constraint on the types of messages that can be returned on the transport~~
1081 ~~specific back-channel. As in an asynchronous environment, it is up to the recipient of the~~
1082 ~~MakeConnection message to decide which messages are appropriate for transmission to any particular~~
1083 ~~Endpoint. However, the Endpoint processing the MakeConnection message MUST insure that the~~
1084 ~~messages match the selection criteria as specified by the child elements of the MakeConnection~~
1085 ~~element.~~

1086 **1.3 MessagePending**

1087 ~~When MakeConnection is used, and a message is returned on the transport specific back-channel, the~~
1088 ~~MessagePending header SHOULD be included on the returned message as an indicator whether there~~
1089 ~~are additional messages waiting to be retrieved using the same selection criteria that was specified in the~~
1090 ~~MakeConnection element.~~

1091 ~~The following exemplar defines the MessagePending syntax:~~

```
1092 <wsrm:MessagePending pending="xs:boolean" ...>  
1093   ...  
1094 </wsrm:MessagePending>
```

1095 ~~/wsrm:MessagePending~~

1096 ~~This element indicates whether additional messages are waiting to be retrieved.~~

1097 ~~/wsrm:MessagePending@pending~~

1098 ~~This attribute, when set to "true", indicates that there is at least one message waiting to be retrieved.~~
1099 ~~When this attribute is set to "false" it indicates there are currently no messages waiting to be retrieved.~~

1100 ~~/wsrm:MessagePending/{any}~~

1101 ~~This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,~~
1102 ~~to be passed.~~

1103 ~~/wsrm:MessagePending/@{any}~~

1104 ~~This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the~~
1105 ~~element.~~

1106 ~~The absence of the MessagePending header has no implication as to whether there are additional~~
1107 ~~messages waiting to be retrieved.~~

4 Faults

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

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

```
http://docs.oasis-open.org/ws-rx/wsrn/20070208/fault
```

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

The definitions of faults use the following properties:

[Code] The fault code.

[Subcode] The fault subcode.

[Reason] The English language reason element.

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

Entities that generate WS-ReliableMessaging faults MUST set the [Code] property to either "Sender" or "Receiver". These properties are serialized into text XML as follows:

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

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

```
<S:Envelope>
  <S:Header>
    <wsa:Action>
      http://docs.oasis-open.org/ws-rx/wsrn/20070208/fault
    </wsa:Action>
    <!-- Headers elided for brevity. -->
  </S:Header>
  <S:Body>
    <S:Fault>
      <S:Code>
        <S:Value> [Code] </S:Value>
        <S:Subcode>
          <S:Value> [Subcode] </S:Value>
        </S:Subcode>
      </S:Code>
```

```

1150     <S:Reason>
1151         <S:Text xml:lang="en"> [Reason] </S:Text>
1152     </S:Reason>
1153     <S:Detail>
1154         [Detail]
1155         ...
1156     </S:Detail>
1157 </S:Fault>
1158 </S:Body>
1159 </S:Envelope>

```

1160 The properties above bind to a SOAP 1.1 fault as follows when the fault is triggered by processing an RM
 1161 header block:

```

1162 <S11:Envelope>
1163   <S11:Header>
1164     <wsrm:SequenceFault>
1165       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
1166       <wsrm:Detail> [Detail] </wsrm:Detail>
1167       ...
1168     </wsrm:SequenceFault>
1169     <!-- Headers elided for brevity. -->
1170   </S11:Header>
1171   <S11:Body>
1172     <S11:Fault>
1173       <faultcode> [Code] </faultcode>
1174       <faultstring> [Reason] </faultstring>
1175     </S11:Fault>
1176   </S11:Body>
1177 </S11:Envelope>

```

1178 The properties bind to a SOAP 1.1 fault as follows when the fault is generated as a result of processing a
 1179 CreateSequence request message:

```

1180 <S11:Envelope>
1181   <S11:Body>
1182     <S11:Fault>
1183       <faultcode> [Subcode] </faultcode>
1184       <faultstring> [Reason] </faultstring>
1185     </S11:Fault>
1186   </S11:Body>
1187 </S11:Envelope>

```

1188 4.1 SequenceFault Element

1189 The purpose of the `SequenceFault` element is to carry the specific details of a fault generated during
 1190 the reliable messaging specific processing of a message belonging to a Sequence. WS-
 1191 ReliableMessaging nodes MUST use the `SequenceFault` container only in conjunction with the SOAP
 1192 1.1 fault mechanism. WS-ReliableMessaging nodes MUST NOT use the `SequenceFault` container in
 1193 conjunction with the SOAP 1.2 binding.

1194 The following exemplar defines its syntax:

```

1195 <wsrm:SequenceFault ...>
1196   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
1197   <wsrm:Detail> ... </wsrm:Detail> ?
1198   ...
1199 </wsrm:SequenceFault>

```

1200 The following describes the content model of the `SequenceFault` element.

1201 /wsrm:SequenceFault
1202 This is the element containing Sequence information for WS-ReliableMessaging
1203 /wsrm:SequenceFault/wsrm:FaultCode
1204 WS-ReliableMessaging nodes that generate a `SequenceFault` MUST set the value of this element to a
1205 qualified name from the set of fault [Subcodes] defined below.
1206 /wsrm:SequenceFault/wsrm:Detail
1207 This element, if present, carries application specific error information related to the fault being described.
1208 /wsrm:SequenceFault/wsrm:Detail/{any}
1209 The application specific error information related to the fault being described.
1210 /wsrm:SequenceFault/wsrm:Detail/@{any}
1211 The application specific error information related to the fault being described.
1212 /wsrm:SequenceFault/{any}
1213 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
1214 to be passed.
1215 /wsrm:SequenceFault/@{any}
1216 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
1217 element.

1218 **4.2 Sequence Terminated**

1219 The Endpoint that generates this fault SHOULD make every reasonable effort to notify the corresponding
1220 Endpoint of this decision.
1221 Properties:
1222 [Code] Sender or Receiver
1223 [Subcode] wsrm:SequenceTerminated
1224 [Reason] The Sequence has been terminated due to an unrecoverable error.
1225 [Detail]

1226 `<wsrm:Identifier ...> xs:anyURI </wsrm: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. |

1227 **4.3 Unknown Sequence**

1228 Properties:

1229 [Code] Sender
 1230 [Subcode] wsmr:UnknownSequence
 1231 [Reason] The value of wsmr:Identifier is not a known Sequence identifier.
 1232 [Detail]

1233 `<wsmr:Identifier ...> xs:anyURI </wsmr: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. |

1234 4.4 Invalid Acknowledgement

1235 An example of when this fault is generated is when a message is Received by the RM Source containing
 1236 a SequenceAcknowledgement covering messages that have not been sent.

1237 [Code] Sender
 1238 [Subcode] wsmr:InvalidAcknowledgement
 1239 [Reason] The SequenceAcknowledgement violates the cumulative Acknowledgement invariant.
 1240 [Detail]

1241 `<wsmr:SequenceAcknowledgement ...> ... </wsmr: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. |

1242 4.5 Message Number Rollover

1243 If the condition listed below is reached, the RM Destination MUST generate this fault.
 1244 Properties:
 1245 [Code] Sender

1246 [Subcode] wsrn:MessageNumberRollover

1247 [Reason] The maximum value for wsrn:MessageNumber has been exceeded.

1248 [Detail]

```
1249 <wsrn:Identifier ...> xs:anyURI </wsrn:Identifier>
1250 <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. |

1251 4.6 Create Sequence Refused

1252 Properties:

1253 [Code] Sender or Receiver

1254 [Subcode] wsrn:CreateSequenceRefused

1255 [Reason] The Create Sequence request has been refused by the RM Destination.

1256 [Detail]

```
1257 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. |

1258 4.7 Sequence Closed

1259 This fault is generated by an RM Destination to indicate that the specified Sequence has been closed.

1260 This fault MUST be generated when an RM Destination is asked to accept a message for a Sequence that
1261 is closed ~~or when an RM Destination is asked to close a Sequence that is already closed.~~

1262 Properties:

1263 [Code] Sender

1264 [Subcode] wsrn:SequenceClosed
1265 [Reason] The Sequence is closed and can-not accept new messages.
1266 [Detail]

1267 `<wsrm:Identifier...> xs:anyURI </wsrm: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. |

1268 **4.8 WSRM Required**

1269 If an RM Destination requires the use of WS-RM, this fault is generated when it Receives an incoming
1270 message that did not use this protocol.
1271 Properties:
1272 [Code] Sender
1273 [Subcode] wsrn:WSRMRequired
1274 [Reason] The RM Destination requires the use of WSRM.
1275 [Detail]

1276 `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. |

1277 **~~1.4 Unsupported Selection~~**

1278 ~~The QName of the unsupported element(s) are included in the detail.~~
1279 ~~Properties:-~~
1280 ~~[Code] Receiver~~
1281 ~~[Subcode] wsrn:UnsupportedSelection~~
1282 ~~[Reason] The extension element used in the message selection is not supported by the RM Source~~
1283 ~~[Detail]~~

1284 ~~`<wsrm:UnsupportedElement> xs:QName </wsrm: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: |

5 Security Threats and Countermeasures

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

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

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

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

5.1 Threats and Countermeasures

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

5.1.1 Integrity Threats

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

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

5.1.1.1 Countermeasures

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

5.1.2 Resource Consumption Threats

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

5.1.2.1 Countermeasures

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

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

5.1.3 Sequence Spoofing Threats

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

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

5.1.3.1 Sequence Hijacking

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

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

5.1.3.2 Countermeasures

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

1365 Source that initiated its creation (i.e. that sent the `CreateSequence` message) and the RM Destination that
1366 serves as its terminus (i.e. that sent the `CreateSequenceResponse` message). To counter sequence
1367 spoofing attempts the RM Destination SHOULD ensure that every message or fault that it Receives that
1368 refers to a particular Sequence originated from the RM Source that jointly owns the referenced Sequence.
1369 For its part the RM Source SHOULD ensure that every message or fault that it Receives that refers to a
1370 particular Sequence originated from the RM Destination that jointly owns the referenced Sequence.

1371 For the RM Destination to be able to identify its sequence peer it MUST be able to identify and
1372 authenticate the entity that sent the `CreateSequence` message. Similarly for the RM Source to identify its
1373 sequence peer it MUST be able to identify and authenticate the entity that sent the
1374 `CreateSequenceResponse` message. For either the RM Destination or the RM Source to determine if a
1375 message was sent by its sequence peer it MUST be able to identify and authenticate the initiator of that
1376 message and, if necessary, correlate this identity with the sequence peer identity established at sequence
1377 creation time.

1378 5.2 Security Solutions and Technologies

1379 The security threats described in the previous sections are neither new nor unique. The solutions that
1380 have been developed to secure other SOAP-based protocols can be used to secure WS-RM as well. This
1381 section maps the facilities provided by common web services security solutions against countermeasures
1382 described in the previous sections.

1383 Before continuing this discussion, however, some examination of the underlying requirements of the
1384 previously described countermeasures is necessary. Specifically it should be noted that the technique
1385 described in Section 5.1.2.1 has two components. Firstly, the RM Destination identifies and authenticates
1386 the issuer of a `CreateSequence` message. Secondly, the RM Destination ~~to~~ performs an authorization
1387 check against this authenticated identity and determines if the RM Source is permitted to create
1388 Sequences with the RM Destination. Since the facilities for performing this authorization check (runtime
1389 infrastructure, policy frameworks, etc.) lie completely within the domain of individual implementations, any
1390 discussion of such facilities is considered to be beyond the scope of this specification.

1391 5.2.1 Transport Layer Security

1392 This section describes how the ~~the~~ facilities provided by SSL/TLS [RFC 4346] can be used to implement
1393 the countermeasures described in the previous sections. The use of SSL/TLS is subject to the constraints
1394 defined in Section 4 of the Basic Security Profile 1.0 [BSP 1.0].

1395 The description provided here is general in nature and is not intended to serve as a complete definition on
1396 the use of SSL/TLS to protect WS-RM. In order to interoperate implementations need to agree on the
1397 choice of features as well as the manner in which they will be used. The mechanisms described in the
1398 Web Services Security Policy Language [SecurityPolicy] MAY be used by services to describe the
1399 requirements and constraints of the use of SSL/TLS.

1400 5.2.1.1 Model

1401 The basic model for using SSL/TLS is as follows:

- 1402 1. The RM Source establishes an SSL/TLS session with the RM Destination.
- 1403 2. The RM Source uses this SSL/TLS session to send a `CreateSequence` message to the RM
1404 Destination.

- 1405 3. The RM Destination establishes an SSL/TLS session with the RM Source and sends an
1406 asynchronous `CreateSequenceResponse` using this session. Alternately it may respond with a
1407 synchronous `CreateSequenceResponse` using the session established in (1).
- 1408 4. For the lifetime of the Sequence the RM Source uses the SSL/TLS session from (1) to Transmit
1409 any and all messages or faults that refer to that Sequence.
- 1410 5. For the lifetime of the Sequence the RM Destination either uses the SSL/TLS session established
1411 in (3) to Transmit any and all messages or faults that refer to that Sequence or, for synchronous
1412 exchanges, the RM Destination uses the SSL/TLS session established in (1).

1413 5.2.1.2 Countermeasure Implementation

1414 Used in its simplest fashion (without relying upon any authentication mechanisms), SSL/TLS provides the
1415 necessary integrity qualities to counter the threats described in Section 5.1.1. Note, however, that the
1416 nature of SSL/TLS limits the scope of this integrity protection to a single transport level session. If
1417 SSL/TLS is the only mechanism used to provide integrity, any intermediaries between the RM Source and
1418 the RM Destination MUST be trusted to preserve the integrity of the messages that flow through them.

1419 As noted, the technique described in Sections 5.1.2.1 involves the use of authentication. This specification
1420 advocates either of two mechanisms for authenticating entities using SSL/TLS. In both of these methods
1421 the SSL/TLS server (the party accepting the SSL/TLS connection) authenticates itself to the SSL/TLS
1422 client using an X.509 certificate that is exchanged during the SSL/TLS handshake.

- 1423 • **HTTP Basic Authentication:** This method of authentication presupposes that a SOAP/HTTP
1424 binding is being used as part of the protocol stack beneath WS-RM. Subsequent to the
1425 establishment of the ~~the~~ SSL/TLS session, the sending party authenticates itself to the receiving
1426 party using HTTP Basic Authentication [RFC 2617]. For example, a RM Source might
1427 authenticate itself to a RM Destination (e.g. when transmitting a Sequence Traffic Message) using
1428 BasicAuth. Similarly the RM Destination might authenticate itself to the RM Source (e.g. when
1429 sending an Acknowledgement) using BasicAuth.
- 1430 • **SSL/TLS Client Authentication:** In this method of authentication, the party initiating the
1431 connection authenticates itself to the party accepting the connection using an X.509 certificate
1432 that is exchanged during the SSL/TLS handshake.

1433 To implement the countermeasures described in section 5.1.2.1 the RM Source must authenticate itself
1434 using one the above mechanisms. The authenticated identity can then be used to determine if the RM
1435 Source is authorized to create a Sequence with the RM Destination.

1436 This specification advocates implementing the countermeasures described in section 5.1.3.2 by requiring
1437 an RM node's Sequence peer to be equivalent to their SSL/TLS session peer. This allows the
1438 authorization decisions described in section 5.1.3.2 to be based on SSL/TLS session identity rather than
1439 on authentication information. For example, an RM Destination can determine that a Sequence Traffic
1440 Message rightfully belongs to its referenced Sequence if that message arrived over the same SSL/TLS
1441 session that was used to carry the `CreateSequence` message for that Sequence. Note that requiring a
1442 one-to-one relationship between SSL/TLS session peer and Sequence peer constrains the lifetime of a
1443 SSL/TLS-protected Sequence to be less than or equal to the lifetime of the SSL/TLS session that is used
1444 to protect that Sequence.

1445 This specification does not preclude the use of other methods of using SSL/TLS to implement the
1446 countermeasures (such as associating specific authentication information with a Sequence) although such
1447 methods are not covered by this document.

1448 Issues specific to the life-cycle management of SSL/TLS sessions (such as the resumption of a SSL/TLS
1449 session) are outside the scope of this specification.

1450 **5.2.2 SOAP Message Security**

1451 The mechanisms described in WS-Security may be used in various ways to implement the
1452 countermeasures described in the previous sections. This specification advocates using the protocol
1453 described by WS-SecureConversation [SecureConversation] (optionally in conjunction with WS-Trust
1454 [Trust]) as a mechanism for protecting Sequences. The use of WS-Security (as an underlying component
1455 of WS-SecureConversation) is subject to the constraints defined in the Basic Security Profile 1.0.

1456 The description provided here is general in nature and is not intended to serve as a complete definition on
1457 the use of WS-SecureConversation/WS-Trust to protect WS-RM. In order to interoperate implementations
1458 need to agree on the choice of features as well as the manner in which they will be used. The
1459 mechanisms described in the Web Services Security Policy Language MAY be used by services to
1460 describe the requirements and constraints of the use of WS-SecureConversation.

1461 **5.2.2.1 Model**

1462 The basic model for using WS-SecureConversation is as follows:

- 1463 1. The RM Source and the RM Destination create a WS-SecureConversation security context. This
1464 may involve the participation of third parties such as a security token service. The tokens
1465 exchanged may contain authentication claims (e.g. X.509 certificates or Kerberos service tickets).
- 1466 2. During the `CreateSequence` exchange, the RM Source SHOULD explicitly identify the security
1467 context that will be used to protect the Sequence. This is done so that, in cases where the
1468 `CreateSequence` message is signed by more than one security context, the RM Source can
1469 indicate which security context should be used to protect the newly created Sequence.
- 1470 3. For the lifetime of the Sequence the RM Source and the RM Destination use the session key(s)
1471 associated with the security context to sign (as defined by WS-Security) at least the body and any
1472 relevant WS-RM-defined headers of any and all messages or faults that refer to that Sequence.

1473 **5.2.2.2 Countermeasure Implementation**

1474 Without relying upon any authentication information, the per-message signatures provide the necessary
1475 integrity qualities to counter the threats described in Section 5.1.1.

1476 To implement the countermeasures described in section 5.1.2.1 some mutually agreed upon form of
1477 authentication claims must be provided by the RM Source to the RM Destination during the establishment
1478 of the Security Context. These claims can then be used to determine if the RM Source is authorized to
1479 create a Sequence with the RM Destination.

1480 This specification advocates implementing the countermeasures described in section 5.1.3.2 by requiring
1481 an RM node's Sequence peer to be equivalent to their security context session peer. This allows the
1482 authorization decisions described in section 5.1.3.2 to be based on the identity of the message's security
1483 context rather than on any authentication claims that may have been established during security context
1484 initiation. Note that other methods of using WS-SecurityConversation to implement the countermeasures
1485 (such as associating specific authentication claims to a Sequence) are possible but not covered by this
1486 document.

1487 As with transport security, the requisite equivalence of a security context peer and with a Sequence peer
1488 limits the lifetime of a Sequence to the lifetime of the protecting security context. Unlike transport security,

1489 the association between a Sequence and its protecting security context cannot always be established
1490 implicitly at Sequence creation time. This is due to the fact that the `CreateSequence` and
1491 `CreateSequenceResponse` messages may be signed by more than one security context.

1492 Issues specific to the life-cycle management of WS-SecurityConversation security contexts (such as
1493 amending or renewing contexts) are outside the scope of this specification.

6 Securing Sequences

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

6.1 Securing Sequences Using WS-Security

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

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

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

```
<wsrm:CreateSequence ...>
  <wsrm:AcksTo> wsa:EndpointReferenceType </wsrm:AcksTo>
  <wsrm:Expires ...> xs:duration </wsrm:Expires> ?
  <wsrm:Offer ...>
    <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
    <wsrm:Endpoint> wsa:EndpointReferenceType </wsrm:Endpoint>
    <wsrm:Expires ...> xs:duration </wsrm:Expires> ?
    <wsrm:IncompleteSequenceBehavior>
      wsrml:IncompleteSequenceBehaviorType
    </wsrm:IncompleteSequenceBehavior> ?
    ...
  </wsrm:Offer> ?
  ...
  <wsse:SecurityTokenReference>
    ...
  </wsse:SecurityTokenReference> ?
  ...
</wsrm:CreateSequence>
```

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

`/wsrm:CreateSequence/wsse:SecurityTokenReference`

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

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

1549 The following exemplar defines the UsesSequenceSTR syntax:

```
1550 <wsrm:UsesSequenceSTR ... />
```

1551 The following describes the content model of the UsesSequenceSTR header block.

1552 /wsrm:UsesSequenceSTR

1553 This element SHOULD be included as a SOAP header block in CreateSequence messages that use the
1554 extensibility mechanism described above in this section. The soap:mustUnderstand attribute value
1555 MUST be 'true'. The receiving RM Destination MUST understand and correctly implement the extension
1556 described above or else generate a soap:MustUnderstand fault, thus aborting the requested
1557 Sequence creation.

1558 The following is an example of a CreateSequence message using the

1559 wsse:SecurityTokenReference extension and the UsesSequenceSTR header block:

```
1560 <soap:Envelope ...>
1561   <soap:Header>
1562     ...
1563     <wsrm:UsesSequenceSTR soap:mustUnderstand='true' />
1564     ...
1565   </soap:Header>
1566   <soap:Body>
1567     <wsrm:CreateSequence>
1568       <wsrm:AcksTo>
1569         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1570       </wsrm:AcksTo>
1571       <wsse:SecurityTokenReference>
1572         ...
1573       </wsse:SecurityTokenReference>
1574     </wsrm:CreateSequence>
1575   </soap:Body>
1576 </soap:Envelope>
```

1577 6.2 Securing Sequences Using SSL/TLS

1578 One mechanism for protecting a Sequence is to bind the Sequence to the underlying SSL/TLS session(s).
1579 The RM Source indicates to the RM Destination that a Sequence is to be bound to the underlying
1580 SSL/TLS session(s) via the UsesSequenceSSL header block. If the RM Source wishes to bind a
1581 Sequence to the underlying SSL/TLS sessions(s) it MUST include the UsesSequenceSSL element as a
1582 SOAP header block within the CreateSequence message.

1583 The following exemplar defines the UsesSequenceSSL syntax:

```
1584 <wsrm:UsesSequenceSSL soap:mustUnderstand="true" ... />
```

1585 The following describes the content model of the UsesSequenceSSL header block.

1586 /wsrm:UsesSequenceSSL

1587 The RM Source MAY include this element as a SOAP header block of a `CreateSequence` message to
1588 indicate to the RM Destination that the resulting Sequence is to be bound to the SSL/TLS session that was
1589 used to carry the `CreateSequence` message. If included, the RM Source MUST mark this header with a
1590 `soap:mustUnderstand` attribute with a value of 'true'. The receiving RM Destination MUST understand
1591 and correctly implement the functionality described in Section 5.2.1 or else generate a
1592 `soap:MustUnderstand` fault, thus aborting the requested Sequence creation.

1593 Note that the use inclusion of the above header by the RM Source implies that all Sequence-related
1594 information (Sequence Lifecycle or Acknowledgment messages or Sequence-related faults) flowing from
1595 the RM Destination to the RM Source will be bound to the SSL/TLS session that is used to carry the
1596 `CreateSequenceResponse` message.

7 References

7.1 Normative

[KEYWORDS]

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

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

[WS-RM Policy]

OASIS WS-RX Technical Committee Draft, "Web Services ReliableMessaging Policy Assertion(WS-RM Policy)" February 2007

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

[SOAP 1.1]

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

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

[SOAP 1.2]

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

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

[URI]

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

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

[UUID]

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

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

[XML]

W3C Recommendation, "Extensible Markup Language (XML) 1.0 (Fourth Edition)". ~~September 2006. Second Edition)~~, ~~October 2000.-~~

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

[XML-ns]

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

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

[XML-Schema Part1]

W3C Recommendation, "XML Schema Part 1: Structures," ~~October 2004~~ ~~May 2001~~.

1630 <http://www.w3.org/TR/xmlschema-1/>

1631 **[XML-Schema Part2]**

1632 W3C Recommendation, "XML Schema Part 2: Datatypes," ~~October 2004~~ [May 2004](#).

1633 <http://www.w3.org/TR/xmlschema-2/>

1634 **[XPath 1.0]**

1635 W3C Recommendation, "XML Path Language (XPath) Version 1.0," 16 November 1999.

1636 <http://www.w3.org/TR/xpath>

1637 **[WSDL 1.1]**

1638 W3C Note, "Web Services Description Language (WSDL 1.1)," 15 March 2001.

1639 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

1640 **[WS-Addressing]**

1641 W3C Recommendation, "Web Services Addressing 1.0 - Core", May 2006.

1642 <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>

1643 W3C Recommendation, "Web Services Addressing 1.0 – SOAP Binding", May 2006.

1644 <http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/>

1645 **7.2 Non-Normative**

1646 **[BSP 1.0]**

1647 WS-I Working Group Draft. "Basic Security Profile Version 1.0," ~~August~~ [March](#) 2006

1648 <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>

1649 **[RDDL 2.0]**

1650 Jonathan Borden, Tim Bray, eds. "Resource Directory Description Language (RDDL) 2.0," January 2004

1651 <http://www.openhealth.org/RDDL/20040118/rddl-20040118.html>

1652 **[RFC 2617]**

1653 J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Loutonen, L. Stewart, "HTTP

1654 Authentication: Basic and Digest Access Authentication," June 1999.

1655 <http://www.ietf.org/rfc/rfc2617.txt>

1656 **[RFC 4346]**

1657 T. Dierks, E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," April 2006.

1658 <http://www.ietf.org/rfc/rfc4346.txt>

1659 **[WS-Policy]**

1660 W3C Member Submission, "Web Services Policy Framework (WS-Policy)," April 2006.

1661 <http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/>

1662 **[WS-PolicyAttachment]**

1663 W3C Member Submission, "Web Services Policy Attachment (WS-PolicyAttachment)," April 2006.

1664 [http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-](http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/)
1665 [20060425/](http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/)

1666 **[WS-Security]**

1667 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "OASIS Web Services Security:
1668 SOAP Message Security 1.0 (WS-Security 2004)", OASIS Standard 200401, March 2004.

1669 <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

1670 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "OASIS Web Services Security:
1671 SOAP Message Security 1.1 (WS-Security 2004)", OASIS Standard 200602, February 2006.

1672 <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>

1673 **[RTTM]**

1674 V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", RFC 1323, May
1675 1992.

1676 <http://www.rfc-editor.org/rfc/rfc1323.txt>

1677 **[SecurityPolicy]**

1678 G. Della-Libra, et. al. "Web Services Security Policy Language (WS-SecurityPolicy)", July 2005

1679 <http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf>

1680 **[SecureConversation]**

1681 S. Anderson, et al, "Web Services Secure Conversation Language (WS-SecureConversation)," February
1682 2005.

1683 <http://schemas.xmlsoap.org/ws/2004/04/sc/>

1684 **[Trust]**

1685 S. Anderson, et al, "Web Services Trust Language (WS-Trust)," February 2005.

1686 <http://schemas.xmlsoap.org/ws/2005/02/trust>

Appendix A. Schema

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

<http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-schema-200702608/wsrn-1.1-schema-200608.xsd>

The following copy is provided for reference.

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

```

1743 </xs:sequence>
1744 <xs:anyAttribute namespace="##other" processContents="lax"/>
1745 </xs:complexType>
1746 <xs:element name="Sequence" type="wsrm:SequenceType"/>
1747 <xs:element name="SequenceAcknowledgement">
1748 <xs:complexType>
1749 <xs:sequence>
1750 <xs:element ref="wsrm:Identifier"/>
1751 <xs:choice>
1752 <xs:sequence>
1753 <xs:choice>
1754 <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
1755 <xs:complexType>
1756 <xs:sequence/>
1757 <xs:attribute name="Upper" type="xs:unsignedLong"
1758 use="required"/>
1759 <xs:attribute name="Lower" type="xs:unsignedLong"
1760 use="required"/>
1761 <xs:anyAttribute namespace="##other" processContents="lax"/>
1762 </xs:complexType>
1763 </xs:element>
1764 <xs:element name="None">
1765 <xs:complexType>
1766 <xs:sequence/>
1767 </xs:complexType>
1768 </xs:element>
1769 </xs:choice>
1770 <xs:element name="Final" minOccurs="0">
1771 <xs:complexType>
1772 <xs:sequence/>
1773 </xs:complexType>
1774 </xs:element>
1775 </xs:sequence>
1776 <xs:element name="Nack" type="xs:unsignedLong"
1777 maxOccurs="unbounded"/>
1778 </xs:choice>
1779 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1780 maxOccurs="unbounded"/>
1781 </xs:sequence>
1782 <xs:anyAttribute namespace="##other" processContents="lax"/>
1783 </xs:complexType>
1784 </xs:element>
1785 <xs:complexType name="AckRequestedType">
1786 <xs:sequence>
1787 <xs:element ref="wsrm:Identifier"/>
1788 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1789 maxOccurs="unbounded"/>
1790 </xs:sequence>
1791 <xs:anyAttribute namespace="##other" processContents="lax"/>
1792 </xs:complexType>
1793 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1794 <xs:complexType name="MessagePendingType">
1795 <xs:sequence>
1796 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1797 maxOccurs="unbounded"/>
1798 </xs:sequence>
1799 <xs:attribute name="pending" type="xs:boolean"/>
1800 <xs:anyAttribute namespace="##other" processContents="lax"/>
1801 </xs:complexType>
1802 <xs:element name="MessagePending" type="wsrm:MessagePendingType"/>
1803 <xs:element name="Identifier">
1804 <xs:complexType>
1805 <xs:annotation>

```

```

1806 <xs:documentation>
1807   This type is for elements whose [children] is an anyURI and can have
1808   arbitrary attributes.
1809 </xs:documentation>
1810 </xs:annotation>
1811 <xs:simpleContent>
1812   <xs:extension base="xs:anyURI">
1813     <xs:anyAttribute namespace="##other" processContents="lax"/>
1814   </xs:extension>
1815 </xs:simpleContent>
1816 </xs:complexType>
1817 </xs:element>
1818 <xs:element name="Address">
1819   <xs:complexType>
1820     <xs:simpleContent>
1821       <xs:extension base="xs:anyURI">
1822         <xs:anyAttribute namespace="##other" processContents="lax"/>
1823       </xs:extension>
1824     </xs:simpleContent>
1825   </xs:complexType>
1826 </xs:element>
1827 <xs:complexType name="MakeConnectionType">
1828   <xs:sequence>
1829     <xs:element ref="wsrm:Identifier" minOccurs="0" maxOccurs="1"/>
1830     <xs:element ref="wsrm:Address" minOccurs="0" maxOccurs="1"/>
1831     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1832     maxOccurs="unbounded"/>
1833   </xs:sequence>
1834   <xs:anyAttribute namespace="##other" processContents="lax"/>
1835 </xs:complexType>
1836 <xs:element name="MakeConnection" type="wsrm:MakeConnectionType"/>
1837 <xs:simpleType name="MessageNumberType">
1838   <xs:restriction base="xs:unsignedLong">
1839     <xs:minInclusive value="1"/>
1840     <xs:maxInclusive value="9223372036854775807"/>
1841   </xs:restriction>
1842 </xs:simpleType>
1843 <!-- Fault Container and Codes -->
1844 <xs:simpleType name="FaultCodes">
1845   <xs:restriction base="xs:QName">
1846     <xs:enumeration value="wsrm:SequenceTerminated"/>
1847     <xs:enumeration value="wsrm:UnknownSequence"/>
1848     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1849     <xs:enumeration value="wsrm:MessageNumberRollover"/>
1850     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1851     <xs:enumeration value="wsrm:SequenceClosed"/>
1852     <xs:enumeration value="wsrm:WSRMRequired"/>
1853     <xs:enumeration value="wsrm:UnsupportedSelection"/>
1854   </xs:restriction>
1855 </xs:simpleType>
1856 <xs:complexType name="SequenceFaultType">
1857   <xs:sequence>
1858     <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1859     <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
1860     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1861     maxOccurs="unbounded"/>
1862   </xs:sequence>
1863   <xs:anyAttribute namespace="##other" processContents="lax"/>
1864 </xs:complexType>
1865 <xs:complexType name="DetailType">
1866   <xs:sequence>
1867     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1868     maxOccurs="unbounded"/>

```

```

1869 </xs:sequence>
1870 <xs:anyAttribute namespace="##other" processContents="lax"/>
1871 </xs:complexType>
1872 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1873 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1874 <xs:element name="CreateSequenceResponse"
1875 type="wsrm:CreateSequenceResponseType"/>
1876 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1877 <xs:element name="CloseSequenceResponse"
1878 type="wsrm:CloseSequenceResponseType"/>
1879 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1880 <xs:element name="TerminateSequenceResponse"
1881 type="wsrm:TerminateSequenceResponseType"/>
1882 <xs:complexType name="CreateSequenceType">
1883 <xs:sequence>
1884 <xs:element ref="wsrm:AcksTo"/>
1885 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1886 <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1887 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1888 maxOccurs="unbounded"/>
1889 <xs:annotation>
1890 <xs:documentation>
1891 It is the authors intent that this extensibility be used to
1892 transfer a Security Token Reference as defined in WS-Security.
1893 </xs:documentation>
1894 </xs:annotation>
1895 </xs:any>
1896 </xs:sequence>
1897 <xs:anyAttribute namespace="##other" processContents="lax"/>
1898 </xs:complexType>
1899 <xs:complexType name="CreateSequenceResponseType">
1900 <xs:sequence>
1901 <xs:element ref="wsrm:Identifier"/>
1902 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1903 <xs:element name="IncompleteSequenceBehavior"
1904 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
1905 <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1906 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1907 maxOccurs="unbounded"/>
1908 </xs:sequence>
1909 <xs:anyAttribute namespace="##other" processContents="lax"/>
1910 </xs:complexType>
1911 <xs:complexType name="CloseSequenceType">
1912 <xs:sequence>
1913 <xs:element ref="wsrm:Identifier"/>
1914 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1915 maxOccurs="unbounded"/>
1916 </xs:sequence>
1917 <xs:anyAttribute namespace="##other" processContents="lax"/>
1918 </xs:complexType>
1919 <xs:complexType name="CloseSequenceResponseType">
1920 <xs:sequence>
1921 <xs:element ref="wsrm:Identifier"/>
1922 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1923 maxOccurs="unbounded"/>
1924 </xs:sequence>
1925 <xs:anyAttribute namespace="##other" processContents="lax"/>
1926 </xs:complexType>
1927 <xs:complexType name="TerminateSequenceType">
1928 <xs:sequence>
1929 <xs:element ref="wsrm:Identifier"/>
1930 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1931 maxOccurs="unbounded"/>

```



```

1932 </xs:sequence>
1933 <xs:anyAttribute namespace="##other" processContents="lax"/>
1934 </xs:complexType>
1935 <xs:complexType name="TerminateSequenceResponseType">
1936 <xs:sequence>
1937 <xs:element ref="wsrm:Identifier"/>
1938 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1939 maxOccurs="unbounded"/>
1940 </xs:sequence>
1941 <xs:anyAttribute namespace="##other" processContents="lax"/>
1942 </xs:complexType>
1943 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
1944 <xs:complexType name="OfferType">
1945 <xs:sequence>
1946 <xs:element ref="wsrm:Identifier"/>
1947 <xs:element name="Endpoint" type="wsa:EndpointReferenceType"/>
1948 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1949 <xs:element name="IncompleteSequenceBehavior"
1950 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
1951 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1952 maxOccurs="unbounded"/>
1953 </xs:sequence>
1954 <xs:anyAttribute namespace="##other" processContents="lax"/>
1955 </xs:complexType>
1956 <xs:complexType name="AcceptType">
1957 <xs:sequence>
1958 <xs:element ref="wsrm:AcksTo"/>
1959 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1960 maxOccurs="unbounded"/>
1961 </xs:sequence>
1962 <xs:anyAttribute namespace="##other" processContents="lax"/>
1963 </xs:complexType>
1964 <xs:element name="Expires">
1965 <xs:complexType>
1966 <xs:simpleContent>
1967 <xs:extension base="xs:duration">
1968 <xs:anyAttribute namespace="##other" processContents="lax"/>
1969 </xs:extension>
1970 </xs:simpleContent>
1971 </xs:complexType>
1972 </xs:element>
1973 <xs:simpleType name="IncompleteSequenceBehaviorType">
1974 <xs:restriction base="xs:string">
1975 <xs:enumeration value="DiscardEntireSequence"/>
1976 <xs:enumeration value="DiscardFollowingFirstGap"/>
1977 <xs:enumeration value="NoDiscard"/>
1978 </xs:restriction>
1979 </xs:simpleType>
1980 OASIS trademark, IPR and other policies apply. -->
1981 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1982 xmlns:wsa="http://www.w3.org/2005/08/addressing"
1983 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200702"
1984 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrm/200702"
1985 elementFormDefault="qualified" attributeFormDefault="unqualified">
1986 <xs:import namespace="http://www.w3.org/2005/08/addressing"
1987 schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
1988 <!-- Protocol Elements -->
1989 <xs:complexType name="SequenceType">
1990 <xs:sequence>
1991 <xs:element ref="wsrm:Identifier"/>
1992 <xs:element name="MessageNumber" type="wsrm:MessageNumberType"/>
1993 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1994 maxOccurs="unbounded"/>

```



```

1995     </xs:sequence>
1996     <xs:anyAttribute namespace="##other" processContents="lax"/>
1997 </xs:complexType>
1998 <xs:element name="Sequence" type="wsrm:SequenceType"/>
1999 <xs:element name="SequenceAcknowledgement">
2000   <xs:complexType>
2001     <xs:sequence>
2002       <xs:element ref="wsrm:Identifier"/>
2003       <xs:choice>
2004         <xs:sequence>
2005           <xs:choice>
2006             <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
2007               <xs:complexType>
2008                 <xs:sequence/>
2009                 <xs:attribute name="Upper" type="xs:unsignedLong"
2010 use="required"/>
2011                 <xs:attribute name="Lower" type="xs:unsignedLong"
2012 use="required"/>
2013             <xs:anyAttribute namespace="##other" processContents="lax"/>
2014           </xs:choice>
2015         </xs:sequence>
2016       <xs:element name="None">
2017         <xs:complexType>
2018           <xs:sequence/>
2019         </xs:complexType>
2020       </xs:element>
2021     </xs:choice>
2022     <xs:element name="Final" minOccurs="0">
2023       <xs:complexType>
2024         <xs:sequence/>
2025       </xs:complexType>
2026     </xs:element>
2027   </xs:sequence>
2028   <xs:element name="Nack" type="xs:unsignedLong"
2029 maxOccurs="unbounded"/>
2030 </xs:choice>
2031 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2032 maxOccurs="unbounded"/>
2033 </xs:sequence>
2034 <xs:anyAttribute namespace="##other" processContents="lax"/>
2035 </xs:complexType>
2036 </xs:element>
2037 <xs:complexType name="AckRequestedType">
2038   <xs:sequence>
2039     <xs:element ref="wsrm:Identifier"/>
2040     <xs:any namespace="##other" processContents="lax" minOccurs="0"
2041 maxOccurs="unbounded"/>
2042   </xs:sequence>
2043   <xs:anyAttribute namespace="##other" processContents="lax"/>
2044 </xs:complexType>
2045 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
2046 <xs:element name="Identifier">
2047   <xs:complexType>
2048     <xs:annotation>
2049       <xs:documentation>
2050         This type is for elements whose [children] is an anyURI and can have
2051 arbitrary attributes.
2052       </xs:documentation>
2053     </xs:annotation>
2054     <xs:simpleContent>
2055       <xs:extension base="xs:anyURI">
2056         <xs:anyAttribute namespace="##other" processContents="lax"/>
2057       </xs:extension>

```

```

2058     </xs:simpleContent>
2059   </xs:complexType>
2060 </xs:element>
2061   <xs:element name="Address">
2062     <xs:complexType>
2063       <xs:simpleContent>
2064         <xs:extension base="xs:anyURI">
2065           <xs:anyAttribute namespace="##other" processContents="lax"/>
2066         </xs:extension>
2067       </xs:simpleContent>
2068     </xs:complexType>
2069   </xs:element>
2070   <xs:simpleType name="MessageNumberType">
2071     <xs:restriction base="xs:unsignedLong">
2072       <xs:minInclusive value="1"/>
2073       <xs:maxInclusive value="9223372036854775807"/>
2074     </xs:restriction>
2075   </xs:simpleType>
2076   <!-- Fault Container and Codes -->
2077   <xs:simpleType name="FaultCodes">
2078     <xs:restriction base="xs:QName">
2079       <xs:enumeration value="wsrm:SequenceTerminated"/>
2080       <xs:enumeration value="wsrm:UnknownSequence"/>
2081       <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
2082       <xs:enumeration value="wsrm:MessageNumberRollover"/>
2083       <xs:enumeration value="wsrm:CreateSequenceRefused"/>
2084       <xs:enumeration value="wsrm:SequenceClosed"/>
2085       <xs:enumeration value="wsrm:WSRMRequired"/>
2086       <xs:enumeration value="wsrm:UnsupportedSelection"/>
2087     </xs:restriction>
2088   </xs:simpleType>
2089   <xs:complexType name="SequenceFaultType">
2090     <xs:sequence>
2091       <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
2092       <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
2093       <xs:any namespace="##other" processContents="lax" minOccurs="0"
2094 maxOccurs="unbounded"/>
2095     </xs:sequence>
2096     <xs:anyAttribute namespace="##other" processContents="lax"/>
2097   </xs:complexType>
2098   <xs:complexType name="DetailType">
2099     <xs:sequence>
2100       <xs:any namespace="##other" processContents="lax" minOccurs="0"
2101 maxOccurs="unbounded"/>
2102     </xs:sequence>
2103     <xs:anyAttribute namespace="##other" processContents="lax"/>
2104   </xs:complexType>
2105   <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
2106   <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
2107   <xs:element name="CreateSequenceResponse"
2108 type="wsrm:CreateSequenceResponseType"/>
2109   <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
2110   <xs:element name="CloseSequenceResponse"
2111 type="wsrm:CloseSequenceResponseType"/>
2112   <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
2113   <xs:element name="TerminateSequenceResponse"
2114 type="wsrm:TerminateSequenceResponseType"/>
2115   <xs:complexType name="CreateSequenceType">
2116     <xs:sequence>
2117       <xs:element ref="wsrm:AcksTo"/>
2118       <xs:element ref="wsrm:Expires" minOccurs="0"/>
2119       <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
2120       <xs:any namespace="##other" processContents="lax" minOccurs="0"

```

```

2121 maxOccurs="unbounded">
2122     <xs:annotation>
2123         <xs:documentation>
2124             It is the authors intent that this extensibility be used to
2125             transfer a Security Token Reference as defined in WS-Security.
2126         </xs:documentation>
2127     </xs:annotation>
2128 </xs:any>
2129 </xs:sequence>
2130 <xs:anyAttribute namespace="##other" processContents="lax"/>
2131 </xs:complexType>
2132 <xs:complexType name="CreateSequenceResponseType">
2133     <xs:sequence>
2134         <xs:element ref="wsrm:Identifier"/>
2135         <xs:element ref="wsrm:Expires" minOccurs="0"/>
2136         <xs:element name="IncompleteSequenceBehavior"
2137             type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
2138         <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
2139         <xs:any namespace="##other" processContents="lax" minOccurs="0"
2140             maxOccurs="unbounded"/>
2141     </xs:sequence>
2142     <xs:anyAttribute namespace="##other" processContents="lax"/>
2143 </xs:complexType>
2144 <xs:complexType name="CloseSequenceType">
2145     <xs:sequence>
2146         <xs:element ref="wsrm:Identifier"/>
2147         <xs:element name="LastMsgNumber" type="wsrm:MessageNumberType"
2148             minOccurs="0"/>
2149         <xs:any namespace="##other" processContents="lax" minOccurs="0"
2150             maxOccurs="unbounded"/>
2151     </xs:sequence>
2152     <xs:anyAttribute namespace="##other" processContents="lax"/>
2153 </xs:complexType>
2154 <xs:complexType name="CloseSequenceResponseType">
2155     <xs:sequence>
2156         <xs:element ref="wsrm:Identifier"/>
2157         <xs:any namespace="##other" processContents="lax" minOccurs="0"
2158             maxOccurs="unbounded"/>
2159     </xs:sequence>
2160     <xs:anyAttribute namespace="##other" processContents="lax"/>
2161 </xs:complexType>
2162 <xs:complexType name="TerminateSequenceType">
2163     <xs:sequence>
2164         <xs:element ref="wsrm:Identifier"/>
2165         <xs:element name="LastMsgNumber" type="wsrm:MessageNumberType"
2166             minOccurs="0"/>
2167         <xs:any namespace="##other" processContents="lax" minOccurs="0"
2168             maxOccurs="unbounded"/>
2169     </xs:sequence>
2170     <xs:anyAttribute namespace="##other" processContents="lax"/>
2171 </xs:complexType>
2172 <xs:complexType name="TerminateSequenceResponseType">
2173     <xs:sequence>
2174         <xs:element ref="wsrm:Identifier"/>
2175         <xs:any namespace="##other" processContents="lax" minOccurs="0"
2176             maxOccurs="unbounded"/>
2177     </xs:sequence>
2178     <xs:anyAttribute namespace="##other" processContents="lax"/>
2179 </xs:complexType>
2180 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
2181 <xs:complexType name="OfferType">
2182     <xs:sequence>
2183         <xs:element ref="wsrm:Identifier"/>

```

```

2184     <xs:element name="Endpoint" type="wsa:EndpointReferenceType"/>
2185     <xs:element ref="wsrm:Expires" minOccurs="0"/>
2186     <xs:element name="IncompleteSequenceBehavior"
2187 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
2188     <xs:any namespace="##other" processContents="lax" minOccurs="0"
2189 maxOccurs="unbounded"/>
2190   </xs:sequence>
2191   <xs:anyAttribute namespace="##other" processContents="lax"/>
2192 </xs:complexType>
2193 <xs:complexType name="AcceptType">
2194   <xs:sequence>
2195     <xs:element ref="wsrm:AcksTo"/>
2196     <xs:any namespace="##other" processContents="lax" minOccurs="0"
2197 maxOccurs="unbounded"/>
2198   </xs:sequence>
2199   <xs:anyAttribute namespace="##other" processContents="lax"/>
2200 </xs:complexType>
2201 <xs:element name="Expires">
2202   <xs:complexType>
2203     <xs:simpleContent>
2204       <xs:extension base="xs:duration">
2205         <xs:anyAttribute namespace="##other" processContents="lax"/>
2206       </xs:extension>
2207     </xs:simpleContent>
2208   </xs:complexType>
2209 </xs:element>
2210 <xs:simpleType name="IncompleteSequenceBehaviorType">
2211   <xs:restriction base="xs:string">
2212     <xs:enumeration value="DiscardEntireSequence"/>
2213     <xs:enumeration value="DiscardFollowingFirstGap"/>
2214     <xs:enumeration value="NoDiscard"/>
2215   </xs:restriction>
2216 </xs:simpleType>
2217 <xs:element name="UsesSequenceSTR">
2218   <xs:complexType>
2219     <xs:sequence>
2220       <xs:anyAttribute namespace="##other" processContents="lax"/>
2221     </xs:sequence>
2222   </xs:complexType>
2223 </xs:element>
2224 <xs:element name="UsesSequenceSSL">
2225   <xs:complexType>
2226     <xs:sequence>
2227       <xs:anyAttribute namespace="##other" processContents="lax"/>
2228     </xs:sequence>
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>

```

Appendix B. WSDL

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

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

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

<http://docs.oasis-open.org/ws-rx/wsrn/200702/wsd/wsrn-1.1-wsd-200702608/wsd/wsrn-1.1-wsd-200608.wsd>

The following non-normative copy is provided for reference.

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

```

2288 <wsdl:types>
2289 <xs:schema
2290 <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsrn/200608"
2291 schemaLocation="http://docs.oasis-open.org/ws-rx/wsrn/200608/wsrn-1.1-schema-
2292 200608.xsd"/>
2293 </xs:schema>
2294 </wsdl:types>

2295 <wsdl:message name="CreateSequence">
2296 <wsdl:part name="create" element="rm:CreateSequence"/>
2297 </wsdl:message>
2298 <wsdl:message name="CreateSequenceResponse">
2299 <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
2300 </wsdl:message>
2301 <wsdl:message name="CloseSequence">
2302 <wsdl:part name="close" element="rm:CloseSequence"/>
2303 </wsdl:message>
2304 <wsdl:message name="CloseSequenceResponse">
2305 <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
2306 </wsdl:message>
2307 <wsdl:message name="TerminateSequence">
2308 <wsdl:part name="terminate" element="rm:TerminateSequence"/>
2309 </wsdl:message>
2310 <wsdl:message name="TerminateSequenceResponse">
2311 <wsdl:part name="terminateResponse"
2312 element="rm:TerminateSequenceResponse"/>
2313 </wsdl:message>
2314 <wsdl:message name="MakeConnection">
2315 <wsdl:part name="makeConnection" element="rm:MakeConnection"/>
2316 </wsdl:message>

2317 <wsdl:portType name="SequenceAbstractPortType">
2318 <wsdl:operation name="CreateSequence">
2319 <wsdl:input message="tns:CreateSequence" wsaw:Action="http://docs.oasis-
2320 open.org/ws-rx/wsrn/200608/CreateSequence"/>
2321 <wsdl:output message="tns:CreateSequenceResponse"
2322 wsaw:Action="http://docs.oasis-open.org/ws-
2323 rx/wsrn/200608/CreateSequenceResponse"/>
2324 </wsdl:operation>
2325 <wsdl:operation name="CloseSequence">
2326 <wsdl:input message="tns:CloseSequence" wsaw:Action="http://docs.oasis-
2327 open.org/ws-rx/wsrn/200608/CloseSequence"/>
2328 <wsdl:output message="tns:CloseSequenceResponse"
2329 wsaw:Action="http://docs.oasis-open.org/ws-
2330 rx/wsrn/200608/CloseSequenceResponse"/>
2331 </wsdl:operation>
2332 <wsdl:operation name="TerminateSequence">
2333 <wsdl:input message="tns:TerminateSequence"
2334 wsaw:Action="http://docs.oasis-open.org/ws-rx/wsrn/200608/TerminateSequence"/>
2335 <wsdl:output message="tns:TerminateSequenceResponse"
2336 wsaw:Action="http://docs.oasis-open.org/ws-
2337 rx/wsrn/200608/TerminateSequenceResponse"/>
2338 </wsdl:operation>
2339 <wsdl:operation name="MakeConnection">
2340 <wsdl:input message="tns:MakeConnection" wsaw:Action="http://docs.oasis-
2341 open.org/ws-rx/wsrn/200608/MakeConnection"/>
2342 </wsdl:operation>
2343 </wsdl:portType>

2344 </wsdl:definitions>
2345 OASIS trademark, IPR and other policies apply. -->
2346 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
2347 xmlns:xs="http://www.w3.org/2001/XMLSchema"

```



```

2348 xmlns:wsa="http://www.w3.org/2005/08/addressing"
2349 xmlns:wsam="http://www.w3.org/2007/02/addressing/metadata"
2350 xmlns:rm="http://docs.oasis-open.org/ws-rx/wsr/200702"
2351 xmlns:tns="http://docs.oasis-open.org/ws-rx/wsr/200702/wsd1"
2352 targetNamespace="http://docs.oasis-open.org/ws-rx/wsr/200702/wsd1">
2353
2354   <wsdl:types>
2355     <xs:schema>
2356       <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsr/200702"
2357       schemaLocation="http://docs.oasis-open.org/ws-rx/wsr/200702/wsr-1.1-schema-
2358       200702.xsd"/>
2359     </xs:schema>
2360   </wsdl:types>
2361
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
2382   <wsdl:portType name="SequenceAbstractPortType">
2383     <wsdl:operation name="CreateSequence">
2384       <wsdl:input message="tns:CreateSequence" wsam:Action="http://docs.oasis-
2385       open.org/ws-rx/wsr/200702/CreateSequence"/>
2386       <wsdl:output message="tns:CreateSequenceResponse"
2387       wsam:Action="http://docs.oasis-open.org/ws-
2388       rx/wsr/200702/CreateSequenceResponse"/>
2389     </wsdl:operation>
2390     <wsdl:operation name="CloseSequence">
2391       <wsdl:input message="tns:CloseSequence" wsam:Action="http://docs.oasis-
2392       open.org/ws-rx/wsr/200702/CloseSequence"/>
2393       <wsdl:output message="tns:CloseSequenceResponse"
2394       wsam:Action="http://docs.oasis-open.org/ws-
2395       rx/wsr/200702/CloseSequenceResponse"/>
2396     </wsdl:operation>
2397     <wsdl:operation name="TerminateSequence">
2398       <wsdl:input message="tns:TerminateSequence"
2399       wsam:Action="http://docs.oasis-open.org/ws-rx/wsr/200702/TerminateSequence"/>
2400       <wsdl:output message="tns:TerminateSequenceResponse"
2401       wsam:Action="http://docs.oasis-open.org/ws-
2402       rx/wsr/200702/TerminateSequenceResponse"/>
2403     </wsdl:operation>
2404   </wsdl:portType>
2405 </wsdl:definitions>

```


Appendix C. Message Examples

Appendix C.1 Create Sequence

Create Sequence

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/20070208"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:MessageID>
      http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
    </wsa:MessageID>
    <wsa:To>http://example.com/serviceB/123</wsa:To>
    <wsa:Action>http://docs.oasis-open.org/ws-
rx/wsmr/20070208/CreateSequence</wsa:Action>
    <wsa:ReplyTo>
      <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
    </wsa:ReplyTo>
  </S:Header>
  <S:Body>
    <wsmr:CreateSequence>
      <wsmr:AcksTo>
        <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
      </wsmr:AcksTo>
    </wsmr:CreateSequence>
  </S:Body>
</S:Envelope>
```

Create Sequence Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/20070208"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:To>http://Business456.com/serviceA/789</wsa:To>
    <wsa:RelatesTo>
      http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
    </wsa:RelatesTo>
    <wsa:Action>
      http://docs.oasis-open.org/ws-rx/wsmr/20070208/CreateSequenceResponse
    </wsa:Action>
  </S:Header>
  <S:Body>
    <wsmr:CreateSequenceResponse>
      <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
    </wsmr:CreateSequenceResponse>
  </S:Body>
</S:Envelope>
```

Appendix C.2 Initial Transmission

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

2453 Message 1

```
2454 <?xml version="1.0" encoding="UTF-8"?>
2455 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2456 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702698"
2457 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2458   <S:Header>
2459     <wsa:MessageID>
2460       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
2461     </wsa:MessageID>
2462     <wsa:To>http://example.com/serviceB/123</wsa:To>
2463     <wsa:From>
2464       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2465     </wsa:From>
2466     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2467     <wsmr:Sequence>
2468       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2469       <wsmr:MessageNumber>1</wsmr:MessageNumber>
2470     </wsmr:Sequence>
2471   </S:Header>
2472   <S:Body>
2473     <!-- Some Application Data -->
2474   </S:Body>
2475 </S:Envelope>
```

2476 Message 2

```
2477 <?xml version="1.0" encoding="UTF-8"?>
2478 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2479 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702698"
2480 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2481   <S:Header>
2482     <wsa:MessageID>
2483       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
2484     </wsa:MessageID>
2485     <wsa:To>http://example.com/serviceB/123</wsa:To>
2486     <wsa:From>
2487       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2488     </wsa:From>
2489     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2490     <wsmr:Sequence>
2491       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2492       <wsmr:MessageNumber>2</wsmr:MessageNumber>
2493     </wsmr:Sequence>
2494   </S:Header>
2495   <S:Body>
2496     <!-- Some Application Data -->
2497   </S:Body>
2498 </S:Envelope>
```

2499 Message 3

```
2500 <?xml version="1.0" encoding="UTF-8"?>
2501 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2502 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702698"
2503 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2504   <S:Header>
2505     <wsa:MessageID>
2506       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
2507     </wsa:MessageID>
2508     <wsa:To>http://example.com/serviceB/123</wsa:To>
2509     <wsa:From>
2510       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
```

```

2511 </wsa:From>
2512 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2513 <wsrm:Sequence>
2514 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2515 <wsrm:MessageNumber>3</wsrm:MessageNumber>
2516 </wsrm:Sequence>
2517 <wsrm:AckRequested>
2518 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2519 </wsrm:AckRequested>
2520 </S:Header>
2521 <S:Body>
2522 <!-- Some Application Data -->
2523 </S:Body>
2524 </S:Envelope>

```

2525 Appendix C.3 First Acknowledgement

2526 Message number 2 has not been accepted by the RM Destination due to some transmission error so it
2527 responds with an Acknowledgement for messages 1 and 3:

```

2528 <?xml version="1.0" encoding="UTF-8"?>
2529 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2530 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200702698"
2531 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2532 <S:Header>
2533 <wsa:MessageID>
2534 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
2535 </wsa:MessageID>
2536 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
2537 <wsa:From>
2538 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
2539 </wsa:From>
2540 <wsa:Action>
2541 http://docs.oasis-open.org/ws-rx/wsr/200702698/SequenceAcknowledgement
2542 </wsa:Action>
2543 <wsrm:SequenceAcknowledgement>
2544 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2545 <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
2546 <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
2547 </wsrm:SequenceAcknowledgement>
2548 </S:Header>
2549 <S:Body/>
2550 </S:Envelope>

```

2551 Appendix C.4 Retransmission

2552 The RM Sourcediscovers that message number 2 was not accepted so it resends the message and
2553 requests an Acknowledgement:

```

2554 <?xml version="1.0" encoding="UTF-8"?>
2555 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2556 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200702698"
2557 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2558 <S:Header>
2559 <wsa:MessageID>
2560 http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
2561 </wsa:MessageID>
2562 <wsa:To>http://example.com/serviceB/123</wsa:To>
2563 <wsa:From>
2564 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2565 </wsa:From>

```

```

2566 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2567 <wsrm:Sequence>
2568 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2569 <wsrm:MessageNumber>2</wsrm:MessageNumber>
2570 </wsrm:Sequence>
2571 <wsrm:AckRequested>
2572 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2573 </wsrm:AckRequested>
2574 </S:Header>
2575 <S:Body>
2576 <!-- Some Application Data -->
2577 </S:Body>
2578 </S:Envelope>

```

2579 Appendix C.5 Termination

2580 The RM Destination now responds with an Acknowledgement for the complete Sequence which can then
 2581 be terminated:

```

2582 <?xml version="1.0" encoding="UTF-8"?>
2583 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2584 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200702608"
2585 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2586 <S:Header>
2587 <wsa:MessageID>
2588 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
2589 </wsa:MessageID>
2590 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
2591 <wsa:From>
2592 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
2593 </wsa:From>
2594 <wsa:Action>
2595 http://docs.oasis-open.org/ws-rx/wsrn/200702608/SequenceAcknowledgement
2596 </wsa:Action>
2597 <wsrm:SequenceAcknowledgement>
2598 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2599 <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
2600 </wsrm:SequenceAcknowledgement>
2601 </S:Header>
2602 <S:Body/>
2603 </S:Envelope>

```

2604 Terminate Sequence

```

2605 <?xml version="1.0" encoding="UTF-8"?>
2606 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2607 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200702608"
2608 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2609 <S:Header>
2610 <wsa:MessageID>
2611 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
2612 </wsa:MessageID>
2613 <wsa:To>http://example.com/serviceB/123</wsa:To>
2614 <wsa:Action>
2615 http://docs.oasis-open.org/ws-rx/wsrn/200702608/TerminateSequence
2616 </wsa:Action>
2617 <wsa:From>
2618 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2619 </wsa:From>
2620 </S:Header>
2621 <S:Body>
2622 <wsrm:TerminateSequence>

```

```

2623     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2624     <wsrm:LastMsgNumber> 3 </wsrm:LastMsgNumber>
2625   </wsrm:TerminateSequence>
2626 </S:Body>
2627 </S:Envelope>

```

2628 Terminate Sequence Response

```

2629 <?xml version="1.0" encoding="UTF-8"?>
2630 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2631 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200702608"
2632 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2633   <S:Header>
2634     <wsa:MessageID>
2635       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
2636     </wsa:MessageID>
2637     <wsa:To>http://example.com/serviceA/789</wsa:To>
2638     <wsa:Action>
2639       http://docs.oasis-open.org/ws-rx/wsrm/200702608/TerminateSequenceResponse
2640     </wsa:Action>
2641     <wsa:RelatesTo>
2642       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
2643     </wsa:RelatesTo>
2644     <wsa:From>
2645       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2646     </wsa:From>
2647   </S:Header>
2648   <S:Body>
2649     <wsrm:TerminateSequenceResponse>
2650       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2651     </wsrm:TerminateSequenceResponse>
2652   </S:Body>
2653 </S:Envelope>

```

2654 Appendix C.6 MakeConnection

2655 To illustrate how a MakeConnection message exchange can be used to deliver messages to an
 2656 Endpoint that is not addressable, consider the case of a pub/sub scenario in which the Endpoint to which
 2657 notifications are to be delivered (the "event consumer") is not addressable by the notification sending
 2658 Endpoint (the "event producer"). In this scenario the event consumer must initiate the connections in order
 2659 for the notifications to be delivered. One possible set of message exchanges (using HTTP) that
 2660 demonstrate how this can be achieved using MakeConnection is shown below:

2661 **Step 1** — During a "subscribe" operation, the event consumer's EPR specifies the RM anonymous URI
 2662 and the RM Policy Assertion to indicate whether or not RM is required:

```

2663 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2664 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200608"
2665 xmlns:wsrmp="http://docs.oasis-open.org/ws-rx/wsrmp/200608"
2666 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2667   <S:Header>
2668     <wsa:To> http://example.org/subscriptionService </wsa:To>
2669     <wsa:MessageID> http://client456.org/id-a6d8-a7c2eb546813</wsa:MessageID>
2670     <wsa:ReplyTo>
2671       <wsa:To> http://client456.org/response </wsa:To>
2672     </wsa:ReplyTo>
2673   </S:Header>
2674   <S:Body>
2675     <sub:Subscribe xmlns:sub="http://example.org/subscriptionService">
2676       <!-- subscription service specific data -->
2677     </sub:Subscribe>

```

```

2678 <wsa:Address>http://docs.oasis-open.org/ws-
2679 rx/wsrn/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:Address>
2680 <wsa:Metadata>
2681 <wsp:Policy wsu:Id="MyPolicy">
2682 <wsrmp:RMAssertion/>
2683 </wsp:Policy>
2684 </wsa:Metadata>
2685 </targetEPR>
2686 </sub:Subscribe>
2687 </S:Body>
2688 </S:Envelope>

```

2689 In this example the `subscribe` and `targetEPR` elements are simply examples of what a subscription-
2690 request message might contain. Note: the `wsa:Address` element contains the RM anonymous URI
2691 indicating that the notification producer needs to queue the messages until they are requested using the
2692 `MakeConnection` message exchange. The EPR also contains the RM Policy Assertion indicating the RM-
2693 must be used when notifications related to this subscription are sent.

2694 **Step 2**—Once the subscription is established, the event consumer checks for a pending message:

```

2695 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2696 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200608"
2697 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2698 <S:Header>
2699 <wsa:Action>http://docs.oasis-open.org/ws-
2700 rx/wsrn/200608/MakeConnection</wsa:Action>
2701 <wsa:To>http://example.org/subscriptionService</wsa:To>
2702 </S:Header>
2703 <S:Body>
2704 <wsrm:MakeConnection>
2705 <wsrm:Address>http://docs.oasis-open.org/ws-
2706 rx/wsrn/200608/anonymous?id=550e8400-e29b-11d4-a716-
2707 446655440000</wsrm:Address>
2708 </wsrm:MakeConnection>
2709 </S:Body>
2710 </S:Envelope>

```

2711 **Step 3**—If there are messages waiting to be delivered then a message will be returned back to the event-
2712 consumer. However, because WS-RM is being used to deliver the messages, the first message returned
2713 is a `CreateSequence`:

```

2714 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2715 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200608"
2716 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2717 <S:Header>
2718 <wsa:Action>http://docs.oasis-open.org/ws-
2719 rx/wsrn/200608/CreateSequence</wsa:Action>
2720 <wsa:To>http://docs.oasis-open.org/ws-
2721 rx/wsrn/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:To>
2722 <wsa:ReplyTo>http://example.org/subscriptionService</wsa:ReplyTo>
2723 <wsa:MessageID>http://example.org/id-123-456</wsa:MessageID>
2724 </S:Header>
2725 <S:Body>
2726 <wsrm:CreateSequence>
2727 <wsrm:AcksTo>
2728 <wsa:Address>http://example.org/subscriptionService</wsa:Address>
2729 </wsrm:AcksTo>
2730 </wsrm:CreateSequence>
2731 </S:Body>

```


2732 `</S:Envelope>`

2733 Notice from the perspective of how the RM Source on the event producer interacts with the RM-
2734 Destination of those messages, nothing new is introduced by the use of the `MakeConnection`, the use-
2735 of RM protocol is the same as the case where the event consumer is addressable.

2736 **Step 4**—The event consumer will respond with a `CreateSequenceResponse` message per normal WS-
2737 Addressing rules:

```
2738 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2739 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200608"
2740 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2741   <S:Header>
2742     <wsa:Action>http://docs.oasis-open.org/ws-
2743 rx/wsm/200608/CreateSequenceResponse</wsa:Action>
2744     <wsa:To>http://example.org/subscriptionService</wsa:To>
2745     <wsa:RelatesTo>http://example.org/id-123-456</wsa:RelatesTo>
2746   </S:Header>
2747   <S:Body>
2748     <wsm:CreateSequenceResponse>
2749       <wsm:Identifier>http://example.org/rmid-456</wsm:Identifier>
2750     </wsm:CreateSequenceResponse>
2751   </S:Body>
2752 </S:Envelope>
```

2753 Note, this message is carried on an HTTP request directed to the `wsa:ReplyTo` EPR, and the HTTP-
2754 response will be an HTTP 202.

2755 **Step 5**—The event consumer checks for another message pending:

```
2756 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2757 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200608"
2758 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2759   <S:Header>
2760     <wsa:Action>http://docs.oasis-open.org/ws-
2761 rx/wsm/200608/MakeConnection</wsa:Action>
2762     <wsa:To>http://example.org/subscriptionService</wsa:To>
2763   </S:Header>
2764   <S:Body>
2765     <wsm:MakeConnection>
2766       <wsm:Address>http://docs.oasis-open.org/ws-
2767 rx/wsm/200608/anonymous?id=550e8400-e29b-11d4-a716-
2768 446655440000</wsm:Address>
2769     </wsm:MakeConnection>
2770   </S:Body>
2771 </S:Envelope>
```

2772 Notice this is the same message as the one sent in step 2.

2773 **Step 6**—If there is a message pending for this destination then it is returned on the HTTP response:

```
2774 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2775 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200608"
2776 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2777   <S:Header>
2778     <wsa:Action>http://example.org/eventType1</wsa:Action>
2779     <wsa:To>http://docs.oasis-open.org/ws-
2780 rx/wsm/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:To>
```



```

2781 <wsrm:Sequence>
2782 <wsrm:Identifier> http://example.org/rmid-456 </wsrm:Identifier>
2783 </wsrm:Sequence>
2784 <wsrm:MessagePending pending="true"/>
2785 </S:Header>
2786 <S:Body>
2787 <!-- event specific data -->
2788 </S:Body>
2789 </S:Envelope>

```

2790 As noted in step 3, the use of the RM protocol does not change when using `MakeConnection`. The
 2791 format of the messages, the order of the messages sent and the timing of when to send it remains the
 2792 same.

2793 **Step 7**— At some later interval, or immediately due to the `MessagePending` header's "pending"
 2794 attribute being set to "true", the event consumer will poll again:

```

2795 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2796 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200608"
2797 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2798 <S:Header>
2799 <wsa:Action>http://docs.oasis-open.org/ws-
2800 rx/wsrm/200608/MakeConnection</wsa:Action>
2801 <wsa:To> http://example.org/subscriptionService </wsa:To>
2802 </S:Header>
2803 <S:Body>
2804 <wsrm:MakeConnection>
2805 <wsrm:Address>http://docs.oasis-open.org/ws-
2806 rx/wsrm/200608/anonymous?id=550e8400-e29b-11d4-a716-
2807 446655440000</wsrm:Address>
2808 </wsrm:MakeConnection>
2809 </S:Body>
2810 </S:Envelope>

```

2811 Notice this is the same message as the one sent in steps 2 and 5. As in steps 3 and 6, the response to
 2812 the `MakeConnection` can be any message destined to the specified Endpoint. This allows the event
 2813 producer to send not only application messages but RM protocol messages (e.g. `CloseSequence`,
 2814 `TerminateSequence` or even additional `CreateSequences`) as needed.

2815 **Step 8**— If at any point in time there are no messages pending, in response to a `MakeConnection` the
 2816 event producer returns an HTTP 202 back to the event consumer. The process then repeats (back to step
 2817 7) until the subscription ends.

2818 **Appendix D. State Tables**

2819 This appendix specifies the non-normative state transition tables for RM Source and RM Destination.

2820 The state tables describe the lifetime of a sequence in both the RM Source and the RM Destination

2821 Legend:

2822 The first column of these tables contains the motivating event and has the following format:

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

2823 Where:

- 2824 ● Event Name: indicates the name of the event. Event Names surrounded by "<>" are optional as
2825 described by the specification.
- 2826 ● [source]: indicates the source of the event; one of:
 - 2827 ● [msg] a Received message
 - 2828 ● [int]: an internal event such as the firing of a timer
 - 2829 ● [app]: the application
 - 2830 ● [unspec]: the source is unspecified

2831 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} |

2832 Where:

- 2833 ● action to take: indicates that the state machine performs the following action. Actions surrounded
2834 by "<>" are optional as described by the specification. "Xmit" is used as a short form for the word
2835 "Transmit"
- 2836 ● [next state]: indicates the state to which the state machine will advance upon the performance of
2837 the action. For ease of reading the next state "same" indicates that the state does not change.
- 2838 ● {ref} is a reference to the document section describing the behavior in this cell

2839 "N/A" in a cell indicates a state / event combination self-inconsistent with the state machine; should these
2840 conditions occur, it would indicate an implementation error. A blank cell indicates that the behavior is not
2841 described in this specification and does not indicate normal protocol operation. Implementations MAY
2842 generate a Sequence Terminated fault (see section 4.2) in these circumstances. Robust implementations
2843 MUST be able to operate in a stable manner despite the occurrence of unspecified event / state
2844 combinations.

2845 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] {2.3} | 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] {4.3} | 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 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} |
| 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} |

2846 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} | 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} |

2847 ~~The following two tables apply only if the MakeConnection mechanism is utilized.~~

2848 ~~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)} |

2849 ~~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} |

Appendix E. Acknowledgments

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

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

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

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

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

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

| Rev | Date | By Whom | What |
|-----------------------|----------------------------|---------------------------------|---|
| | | | OpenDocument Text format. Changed line numbers to be a single style. |
| wd-08 | 2005-12-28 | Anish Karmarkar | Included a section link to c:\temp\wsrm-1.1-schema-200510.xsd and to c:\temp\wsrm-1.1-wsdl-200510.wsdl |
| wd-08 | 2006-01-04 | Gilbert Pilz | Fixed formatting for included sections. |
| wd-08 | 2006-01-05 | Gilbert Pilz | Created links for unused references. Fixed exemplars for CloseSequence and CloseSequenceResponse . |
| wd-09 | 2006-01-11 | Doug Davis | Minor tweaks to text/typos. |
| wd-10 | 2006-01-23 | Doug Davis | Accept all changes from wd-09 Make some minor editorial tweaks from Marc's comments. |
| wd-10 | 2006-02-14 | Doug Davis | Issue 082 resolution |
| wd-10 | 2006-02-14 | Doug Davis | Issue 083 resolution |
| wd-10 | 2006-02-14 | Doug Davis | Issue 085 resolution |
| wd-10 | 2006-02-14 | Doug Davis | Issues 086, 087 resolutions Defined MessageNumberType |
| wd-10 | 2006-02-15 | Doug Davis | Issue 078 resolution |
| wd-10 | 2006-02-15 | Doug Davis | Issue 094 resolution |
| wd-10 | 2006-02-15 | Doug Davis | Issue 095 resolution |
| wd-10 | 2006-02-15 | Gilbert Pilz | Issue 088 – added namespace URI link to namespace URI; added text explaining that this URI could be dereferenced to produce the RDDDL doc; added non-normative reference to RDDDL 2.0 |
| wd-10 | 2006-02-17 | Anish Karmarkar | Namespace changed to 200602 for both WSDL and XSD docs. |
| wd-10 | 2006-02-17 | Anish Karmarkar | Issue i087 as it applies to WSRM spec. |
| wd-10 | 2006-02-17 | Anish Karmarkar | Added titles and minor text for state table (issue i058). |
| wd-11 | 2006-02-22 | Doug Davis | Accept all changes for new WD Minor typos fixed |
| wd-11 | 2006-02-23 | Doug Davis | s'/close'/close/g – per Marc Goodner Added first ref to [URI] – per Marc G again |
| wd-11 | 2006-02-27 | Doug Davis | Issue i061 applied |
| wd-11 | 2006-02-28 | Doug Davis | Fixed typo around the use of "above" and "below" |
| wd-11 | 2006-03-01 | Doug Davis | Minor typos found by Marc Goodner |
| wd-11 | 2006-03-02 | Doug Davis | Minor typos found by Matt Lovett |
| wd-11 | 2006-03-08 | Doug Davis | Issue 091 applied |
| wd-11 | 2006-03-08 | Doug Davis | Issue 092 applied |
| wd-11 | 2006-03-08 | Doug Davis | Issue 100 applied |

| Rev | Date | By Whom | What |
|-----------------------|----------------------------|------------------------------|---|
| wd-12 | 2006-03-20 | Doug Davis | Added space in "SOAP1.x" – PaulCotton |
| wd-12 | 2006-04-11 | Doug Davis | Issue 007 applied |
| wd-12 | 2006-04-11 | Doug Davis | Issue 090 applied |
| wd-12 | 2006-04-11 | Doug Davis | Issue 098 applied |
| wd-12 | 2006-04-11 | Doug Davis | Issue 099 applied |
| wd-12 | 2006-04-11 | Doug Davis | Issue 101 applied |
| wd-12 | 2006-04-11 | Doug Davis | Issue 103 applied |
| wd-12 | 2006-04-11 | Doug Davis | Issue 104 applied |
| wd-12 | 2006-04-11 | Doug Davis | Issue 105 applied |
| wd-12 | 2006-04-11 | Doug Davis | Issue 107 applied |
| wd-12 | 2006-04-11 | Doug Davis | Issue 109 applied |
| wd-12 | 2006-04-11 | Doug Davis | Issue 110 applied |
| wd-12 | 2006-04-12 | Doug Davis | Used "generated" instead of "issue" or "send" when talking about faults. |
| wd-12 | 2006-04-24 | Gilbert Pilz | Update references to WS-Addressing to the Proposed Recommendations; update WS-RM namespace to "200604". |
| 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 'optional'/'required' to 'OPTIONAL'/'REQUIRED') added |
| wd-05 | 2005-09-30 | Anish Karmarkar | i017 (Change NS to http://docs.oasis-open.org/wsrn/200510/) |
| wd-05 | 2005-09-30 | Anish Karmarkar | i045 (Include SecureConversation as a reference and move it to non-normative citation) |
| wd-05 | 2005-09-30 | Anish Karmarkar | i046 (change the type of wsrn:FaultCode element) |
| wd-06 | 2005-11-02 | Gilbert Pilz | Start wd-06 by changing title page from cd-01. |
| wd-06 | 2005-11-03 | Gilbert Pilz | i047 (Reorder spec sections) |
| wd-07 | 2005-11-17 | Gilbert Pilz | Start wd-07 |
| wd-07 | 2005-11-28 | Doug Davis | i071 — except for period in Appendix headings |
| wd-07 | 2005-11-28 | Doug Davis | i10 |
| wd-07 | 2005-11-28 | Doug Davis | i030 |
| wd-07 | 2005-11-28 | Doug Davis | i037 |
| wd-07 | 2005-11-28 | Doug Davis | i038 |
| wd-07 | 2005-11-28 | Doug Davis | i041 |
| 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 Pilz | Issue 088 — added namespace URI link to namespace URI; added text explaining that this URI could be dereferenced to produce the RDDL doc; added non-normative reference to RDDL 2.0 |
| wd-10 | 2006-02-17 | Anish Karmarkar | Namespace changed to 200602 for both WSDL and XSD docs. |
| wd-10 | 2006-02-17 | Anish Karmarkar | Issue i087 as it applies to WSRM spec. |
| wd-10 | 2006-02-17 | Anish Karmarkar | Added titles and minor text for state table (issue i058). |
| 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; bump 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 Pilz | 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 Pilz | More formatting changes for better HTML rendering. |

Appendix G. Notices

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

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

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

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

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

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