



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

3 **Committee Draft 01, September 26 Working Draft**
4 **01, July 7th 2005**

5 **Document identifier:**

6 [wsm-1.1-spec-cd-01](#)
7 [WS-ReliableMessaging-1.0draft-01.doc](#)

8 **Location:**

9 <http://docs.oasis-open.org/ws-rx/wsm/200510/wsm-1.1-spec-cd-01.pdf>
10 [2005/07/WS-ReliableMessaging-1.0-draft-01.doc](#)

11 **Editors:**

12 [Gilbert Pilz, BEA <gilbert.pilz@bea.com>](mailto:gilbert.pilz@bea.com)
13 [Doug Davis, IBM <dug@us.ibm.com>](mailto:dug@us.ibm.com)
14 [Anish Karmarkar, Oracle <Anish.Karmarkar@oracle.com>](mailto:Anish.Karmarkar@oracle.com)
15 [TBD](#)

16 **Contributors:**

17 [TBD](#)

File name

Date

Copyright © OASIS Open 2005. All Rights Reserved.
numbers of Statistics

Page Page

18 **Copyright:**

19 ~~(c) 2002-2005 BEA Systems, IBM, Microsoft Corporation, Inc., and TIBCO Software Inc.~~
20 ~~All rights reserved.~~

21 ~~Permission to copy and display the Web Services Reliable Messaging Policy Assertion~~
22 ~~Specification (the "Specification", which includes schema documents), in any medium~~
23 ~~without fee or royalty is hereby granted, provided that you include the following on ALL~~
24 ~~copies of the Specification that you make:~~

25 ~~1. A link or URL to the Specification at one of the Authors' websites~~

26 ~~2. The copyright notice as shown in the Specification.~~

27 ~~BEA Systems, IBM, Microsoft and TIBCO Software (collectively, the "Authors") each~~
28 ~~agree to grant you a license, under royalty free and otherwise reasonable, non-~~
29 ~~discriminatory terms and conditions, to their respective essential patent claims that they~~
30 ~~deem necessary to implement the Specification.~~

31 ~~THE SPECIFICATION IS PROVIDED "AS IS," AND THE AUTHORS MAKE NO~~
32 ~~REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT~~
33 ~~NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A~~
34 ~~PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS~~
35 ~~OF THE SPECIFICATION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE~~
36 ~~IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY~~
37 ~~PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.~~

38 ~~THE AUTHORS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL,~~
39 ~~INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO~~
40 ~~ANY USE OR DISTRIBUTION OF THE SPECIFICATION.~~

41 ~~The name and trademarks of the Authors may NOT be used in any manner, including~~
42 ~~advertising or publicity pertaining to the Specification or its contents without specific,~~
43 ~~written prior permission. Title to copyright in the Specification will at all times remain with~~
44 ~~the Authors.~~

45 ~~No other rights are granted by implication, estoppel or otherwise.~~

46 **Abstract:**

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

File name

Date

Copyright © OASIS Open 2005. All Rights Reserved.
numbers of Statistics

Page Page

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

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

64 **Composable Architecture:-**

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

72 **Status:**

73 This document is a Committee Draft.

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

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

81 Table of Contents

82	1INTRODUCTION.....	5
83	1.1GOALS AND REQUIREMENTS.....	5
84	1.1.1Requirements.....	5
85	1.2NOTATIONAL CONVENTIONS.....	5
86	1.3NAMESPACE.....	6
87	1.4COMPLIANCE.....	7
88	2RELIABLE MESSAGING MODEL.....	8
89	2.1GLOSSARY.....	9
90	2.2PROTOCOL PRECONDITIONS.....	10
91	2.3PROTOCOL INVARIANTS.....	10
92	2.4EXAMPLE MESSAGE EXCHANGE.....	11
93	3RM PROTOCOL ELEMENTS.....	13
94	3.1SEQUENCES.....	13
95	3.2SEQUENCE ACKNOWLEDGEMENT.....	15
96	3.3REQUEST ACKNOWLEDGEMENT.....	17
97	3.4SEQUENCE CREATION.....	19
98	3.5SEQUENCE TERMINATION.....	22
99	3.6CLOSING A SEQUENCE.....	23
100	4FAULTS.....	26
101	4.1SEQUENCEFAULT ELEMENT.....	28
102	4.2SEQUENCE TERMINATED.....	29
103	4.3UNKNOWN SEQUENCE.....	29
104	4.4INVALID ACKNOWLEDGEMENT.....	29
105	4.5MESSAGE NUMBER ROLLOVER.....	30
106	4.6LAST MESSAGE NUMBER EXCEEDED.....	30
107	4.7CREATE SEQUENCE REFUSED.....	31
108	4.8SEQUENCE CLOSED.....	31
109	5SECURITY CONSIDERATIONS.....	32
110	6REFERENCES.....	34
111	6.1NORMATIVE.....	34

File name

Date

Copyright © OASIS Open 2005. All Rights Reserved.
numbers of Statistics

Page Page

112	6.2NON-NORMATIVE.....	34
113	APPENDIX A.SCHEMA	36
114	APPENDIX B.MESSAGE EXAMPLES.....	41
115	B.1.CREATE SEQUENCE.....	42
116	B.2. INITIAL TRANSMISSION.....	44
117	B.3.FIRST ACKNOWLEDGEMENT.....	46
118	B.4.RETRANSMISSION.....	47
119	B.5.TERMINATION.....	48
120	APPENDIX C.WSDL.....	50
121	APPENDIX D.ACKNOWLEDGMENTS.....	52
122	APPENDIX E.REVISION HISTORY.....	53
123	APPENDIX F.NOTICES.....	54

124 **1 Introduction**

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

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

136 **1.1 Goals and Requirements**

137 **1.1.1 Requirements**

138 **1.2 Notational Conventions**

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

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

- 144 • The syntax appears as an XML instance, but values in italics indicate data types instead
145 of values.
- 146 • Characters are appended to elements and attributes to indicate cardinality:
 - 147 ○ "?" (0 or 1)
 - 148 ○ "*" (0 or more)
 - 149 ○ "+" (1 or more)
- 150 • The character "|" is used to indicate a choice between alternatives.
- 151 • The characters "[" and "]" are used to indicate that contained items are to be treated as a
152 group with respect to cardinality or choice.

- 153 • An ellipsis (i.e. "...") indicates a point of extensibility that allows other child, or attribute,
154 content. Additional children **elements** and/or attributes MAY be added at the indicated
155 extension points but MUST NOT contradict the semantics of the parent and/or owner,
156 respectively. If an extension is not recognized it SHOULD be ignored.
- 157 • XML namespace prefixes (See Section **Namespace**) are used to indicate the namespace
158 of the element being defined.

159 1.3 Namespace

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

162 [http://docs.oasis-](http://docs.oasis-open.org/wsrn/200510/schemas.xmlsoap.org/ws/2005/02/rm/policy)
163 [open.org/wsrn/200510/schemas.xmlsoap.org/ws/2005/02/rm/policy](http://docs.oasis-open.org/wsrn/200510/schemas.xmlsoap.org/ws/2005/02/rm/policy)

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

166 The following namespaces are used in this document:

167 *Table Number range Table*

Prefix	Namespace
<u>S</u>	<u>http://www.w3.org/2003/05/soap-envelope</u>
<u>S11</u>	<u>http://schemas.xmlsoap.org/soap/envelope/</u>
<u>wsrn</u>	<u>http://docs.oasis-open.org/wsrn/200510/</u>
<u>wsa</u>	<u>http://schemas.xmlsoap.org/ws/2004/08/addressing</u>
<u>wsse</u>	<u>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</u>
<u>xs</u>	<u>http://www.w3.org/2001/XMLSchema</u>
Prefix	Namespace
S	http://www.w3.org/2003/05/soap-envelope
S11	http://schemas.xmlsoap.org/soap/envelope/
wsrn	http://schemas.xmlsoap.org/ws/2005/02/rm/
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd

File name

Date

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

169 [http://docs.oasis-
open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/wsrn.xsd](http://docs.oasis-
170 open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/wsrn.xsd)

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

173 If an action URI is used, and one is not already defined per the rules of the WS-
174 Addressing specification [WS-Addressing], then the action URI MUST consist of the
175 reliable messaging namespace URI concatenated with the **"/" character and the**
176 element name. For example:

177 [http://docs.oasis-
open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/SequenceAcknowl
edgement](http://docs.oasis-
178 open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/SequenceAcknowl
179 edgement)

180 1.4 Compliance

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

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

189 **2 Reliable Messaging Model**

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

192 ~~WS-ReliableMessaging provides an interoperable protocol that a Reliable Messaging~~
193 ~~(RM) Source and Reliable Messaging (RM) Destination use to provide Application~~
194 ~~Source and Destination a guarantee that a message that is sent will be delivered.~~
195 ~~The guarantee is specified as a delivery assurance. The protocol supports the~~
196 ~~endpoints in providing these delivery assurances. It is the responsibility of the RM~~
197 ~~Source and RM Destination to fulfill the delivery assurances, or raise an error. The~~
198 ~~protocol defined here allows endpoints to meet this guarantee for the delivery~~
199 ~~assurances defined below.~~

200 The WS-ReliableMessaging specification defines an interoperable protocol that
201 requires a Reliable Messaging (RM) Source and Reliable Messaging (RM) Destination
202 to ensure that each message transmitted by the RM Source is successfully received
203 by an RM Destination, or barring successful receipt, that an RM Source can, except in
204 the most extreme circumstances, accurately determine the disposition of each
205 message transmitted as perceived by the RM Destination, so as to resolve any in-
206 doubt status.

207 In addition, The protocol allows the RM Source and RM Destination to provide their
208 respective Application Source and Application Destination a guarantee that a
209 message that is sent by an Application Source will be delivered to the Application
210 Destination.

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

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

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

223 There are four basic delivery assurances that endpoints can provide:

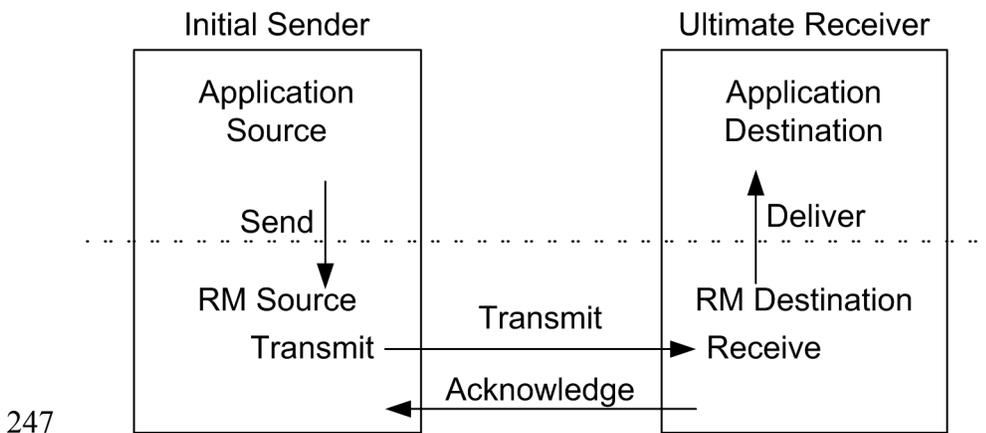
224 **AtMostOnce** Messages will be delivered at most once without duplication or an error
225 will be raised on at least one endpoint. It is possible that some messages in a
226 sequence may not be delivered.

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

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

232 **InOrder** Messages will be delivered in the order that they were sent. This delivery
233 assurance may be combined with any of the above delivery assurances. It requires
234 that the messages within a Sequence will be delivered in an order so that the
235 message numbers are monotonically increasing. Note that this assurance says
236 nothing about duplications or omissions. Note also that it is only applicable to
237 messages in the same Sequence. Cross Sequence ordering of messages is not in the
238 scope of this specification~~sequence observed by the ultimate receiver be non-~~
239 ~~decreasing. It says nothing about duplications or omissions.~~

240 ~~Figure 1~~The diagram below illustrates the entities and events in a simple reliable
241 message exchange. First, the Application Source Sends a message for reliable
242 delivery. The Reliable Messaging (RM) Source accepts the message and Transmits it
243 one or more times. After receiving the message, the RM Destination Acknowledges
244 it. Finally, the RM Destination delivers the message to the Application Destination.
245 The exact roles the entities play and the complete meaning of the events will be
246 defined throughout this specification.



248 Figure 1: Reliable Messaging Model

249 2.1 Glossary

250 The following definitions are used throughout this specification:

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

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

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

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

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

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

260 **RM Destination:** The endpoint that receives the message.

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

263 **Deliver:** The act of transferring a message from the RM Destination to the
264 Application Destination. The reliability guarantee is fulfilled at this point.

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

266 **Receive:** The act of reading a message from a network connection.

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

269 2.2 Protocol Preconditions

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

272 • The RM Source MUST have an endpoint reference that uniquely identifies the RM Destination
273 endpoint; correlations across messages addressed to the unique endpoint MUST be
274 meaningful.

275 • The RM Source MUST have knowledge of the destination's policies, if any, and the RM
276 Source MUST be capable of formulating messages that adhere to this policy.

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

279 **2.3 Protocol Invariants**

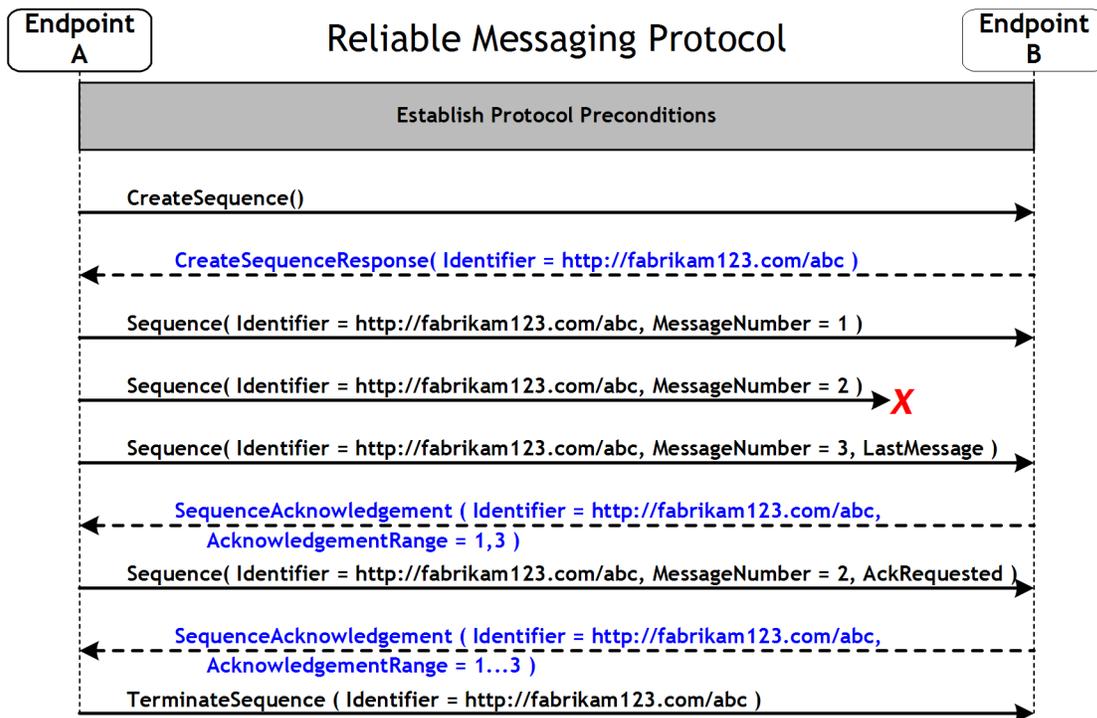
280 During the lifetime of the protocol, two invariants are REQUIRED for correctness:

- 281 • The RM Source MUST assign each reliable message a sequence number (defined below)
282 beginning at 1 and increasing by exactly 1 for each subsequent reliable message.

283 Every acknowledgement issued by the RM Destination MUST include within an
284 acknowledgement range or ranges the sequence number of every message
285 successfully received by the RM Destination and MUST exclude sequence numbers of
286 any messages not yet received.

287 **2.4 Example Message Exchange**

288 Figure 2 illustrates a possible message exchange between two reliable messaging
289 endpoints A and B~~The following figure illustrates a possible message exchange~~
290 ~~between two reliable messaging endpoints.~~



291 Figure 2: The WS-ReliableMessaging Protocol

- 292 1. The protocol preconditions are established. These include policy exchange,
293 endpoint resolution, establishing trust.
- 294 2. The RM Source requests creation of a new Sequence.
- 295 3. The RM Destination creates a Sequence by returning a globally unique identifier.
- 296 4. The RM Source begins sending messages beginning with MessageNumber 1. In
297 the figure the RM Source sends 3 messages.
- 298 5. Since the 3rd message is the last in this exchange, the RM Source includes a
299 `<wsrm:LastMessage>` token.
- 300 6. The 2nd message is lost in transit.
- 301 7. The RM Destination acknowledges receipt of message numbers 1 and 3 in
302 response to the RM Source's `<wsrm:LastMessage>` token.
- 303 8. The RM Source retransmits the 2nd message. This is a new message on the
304 underlying transport, but since it has the same sequence identifier and message
305 number so the RM Destination can recognize it as equivalent to the earlier
306 message, in case both are received.
- 307 9. The RM Source includes an `<wsrm:AckRequested>` element so the RM Destination
308 will expedite an acknowledgement.
- 309 10. The RM Destination receives the second transmission of the message with
310 MessageNumber 2 and acknowledges receipt of message numbers 1, 2, and 3
311 which carried the `<wsrm:LastMessage>` token.
- 312 11. The RM Source receives this acknowledgement and sends a TerminateSequence
313 message to the RM Destination indicating that the sequence is completed and
314 reclaims any resources associated with the Sequence.
- 315 12. The RM Destination receives the TerminateSequence message indicating that the
316 RM Source will not be sending any more messages, and reclaims any resources
317 associated with the Sequence.
- 318 Now that the basic model has been outlined, the details of the elements used in this
319 protocol are now provided [in Section 3](#).

320 3 RM Protocol Elements

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

325 3.1 Sequences

326 The RM protocol uses a `<wsrm:Sequence>` header block to track and manage the
327 [reliable delivery of messages](#). Messages for which the delivery assurance applies
328 [MUST contain a `<wsrm:Sequence>` header block](#). Each Sequence MUST have a
329 [unique `<wsrm:Identifier>` element and each message within a Sequence MUST](#)
330 [have a `<wsrm:MessageNumber>` element that increments by 1 from an initial value of](#)
331 [1. These values are contained within a `<wsrm:Sequence>` header block accompanying](#)
332 [each message being delivered in the context of a Sequence. In addition to mandatory](#)
333 [`<wsrm:Identifier>` and `<wsrm:MessageNumber>` elements, the header MAY include a](#)
334 ~~`<wsrm:Sequence>` header block to track and manage the reliable delivery~~
335 ~~of messages. Messages for which the delivery assurance applies MUST~~
336 ~~contain a `<Sequence>` header block. Each Sequence MUST have a unique~~
337 ~~`<Identifier>` element and each message within a Sequence MUST have a~~
338 ~~`<MessageNumber>` element that increments by 1 from an initial value of~~
339 ~~1. These values are contained within a `<Sequence>` header block~~
340 ~~accompanying each message being delivered in the context of a Sequence.~~
341 ~~In addition to mandatory `<Identifier>` and `<MessageNumber>` elements, the~~
342 ~~header MAY include a `<LastMessage>` element.~~

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

344 The purpose of the `<wsrm:LastMessage>` element is to signal to the RM Destination
345 that the message represents the last message in the Sequence.

346 A following exemplar defines its syntax:

```
347 <wsrm:Sequence ...>  
348   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
349   <wsrm:MessageNumber> xs:unsignedLong </wsrm:MessageNumber>  
350   <wsrm:LastMessage/>?  
351   ...  
352 </wsrm:Sequence>
```

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

354 `/wsrm:Sequence`

File name Date

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

360 /wsrm:Sequence/wsrm:Identifier

361 This ~~REQUIRED~~required element MUST contain an absolute URI conformant with RFC2396 that
362 uniquely identifies the Sequence.

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

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

366 /wsrm:Sequence/wsrm:MessageNumber

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

376 /wsrm:Sequence/wsrm:LastMessage

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

379 /wsrm:Sequence/{any}

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

382 /wsrm:Sequence/@{any}

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

385 A RM Source endpoint MUST include a <wsrm:LastMessage> element in the
386 <wsrm:Sequence> element for the last message in a Sequence. An RM Destination
387 endpoint MUST respond with a <wsrm:SequenceAcknowledgement> upon receipt of a
388 <wsrm:LastMessage> element. A Sequence MUST NOT use a <wsrm:MessageNumber>
389 value greater than that which accompanies a <wsrm:LastMessage> element. An RM

390 Destination MUST generate a LastMessageNumberExceeded (See Section Last
391 Message Number Exceeded) fault upon receipt of such a message. In the event that
392 an RM Source needs to close a Sequence and there is no application message, the
393 RM Source MAY send a message with an empty body containing <wsrm:Sequence>
394 header with the <wsrm:LastMessage> element in the <Sequence> element for
395 the last message in a Sequence. An RM Destination endpoint MUST respond
396 with a <SequenceAcknowledgement> upon receipt of a <LastMessage>
397 element. A Sequence MUST NOT use a <MessageNumber> value greater than
398 that which accompanies a <LastMessage> element. An RM Destination MUST
399 generate a LastMessageNumberExceeded fault upon receipt of such a
400 message. In the event that an RM Source needs to close a Sequence and
401 there is no application message, the RM Source MAY send a message with
402 an empty body containing <Sequence> header with the <LastMessage>
403 element. In this usage, the action URI MUST be:

```
404 http://docs.oasis-  
405 open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/LastMessage
```

406 in preference to the pattern defined in Section 1.2.

407 The following example illustrates a Sequence header block.

```
408 <wsrm:Sequence>  
409 <wsrm:Identifier>http://examplefabrikam123.com/abc</wsrm:Identifier>  
410 <wsrm:MessageNumber>10</wsrm:MessageNumber>  
411 <wsrm:LastMessage/>  
412 </wsrm:Sequence>
```

413 3.2 Sequence Acknowledgement

414 The RM Destination informs the RM Source of successful message receipt using a
415 <wsrm:SequenceAcknowledgement> header block. The
416 <wsrm:SequenceAcknowledgement> header block MAY be transmitted independently
417 or included on return messages. The RM Destination MAY send a
418 <wsrm:SequenceAcknowledgement> header block at any point during which the
419 sequence is valid. The timing of acknowledgements can be advertised using policy
420 and acknowledgements can be explicitly requested using the <wsrm:AckRequested>
421 directive (see Section RequestAcknowledgement). If a non-mustUnderstand fault
422 occurs when processing an RM Header that was piggy-backed on another message,
423 a fault MUST be generated, but the processing of the original message MUST NOT be
424 affected~~<wsrm:SequenceAcknowledgement> header block. The~~
425 ~~<SequenceAcknowledgement> header block MAY be transmitted independently or~~

426 ~~included on return messages. The RM Destination MAY send a~~
427 ~~<SequenceAcknowledgement> header block at any point. The timing of~~
428 ~~acknowledgements can be advertised using policy and acknowledgements can be~~
429 ~~explicitly requested using the <AckRequested> directive.~~

430 The following exemplar defines its syntax:

```
431 <wsrm:SequenceAcknowledgement ...>  
432   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
433   [  | <wsrm:AcknowledgementRange ...  
434        Upper="xs:unsignedLong"  
435        Lower="xs:unsignedLong"/> +  
436        <wsrm:Final/> ?| <wsrm:Nack> xs:unsignedLong </wsrm:Nack> + ]  
437     | <wsrm:Nack> xs:unsignedLong </wsrm:Nack> +  
438     | <wsrm:None/> ]  
439   ...  
440 </wsrm:SequenceAcknowledgement>
```

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

443 `/wsrm:SequenceAcknowledgement`

444 This element contains the Sequence acknowledgement information.

445 `/wsrm:SequenceAcknowledgement/wsrm:Identifier`

446 This ~~REQUIRED~~required element MUST contain an absolute URI conformant with RFC2396 that
447 uniquely identifies the Sequence.

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

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

451 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

452 This OPTIONAL element, if present, can occur 1 or more times. It contains a range of message
453 Sequence MessageNumbers successfully received by the receiving endpoint manager. The
454 ranges SHOULD NOT overlap. This element MUST NOT be present if either the <wsrm:Nack>
455 or <wsrm:None> elements are also present as a child of <wsrm:optional-element, if
456 present, can occur 1 or more times. It contains a range of message
457 Sequence MessageNumbers successfully received by the receiving endpoint
458 manager. The ranges SHOULD NOT overlap. This element MUST NOT be
459 present if <Nack> is also present as a child of
460 `<SequenceAcknowledgement>`.

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

File name

Date

462 This REQUIRED attribute contains an xs:unsignedLong representing the
463 <wsrm:MessageNumber> of the highest contiguous message in a Sequence range received by
464 the RM Destination~~required attribute contains an unsignedLong representing the~~
465 ~~<MessageNumber> of the highest contiguous message in a Sequence range.~~

466 /wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Lower

467 This REQUIRED attribute contains an xs:unsignedLong representing the
468 <wsrm:MessageNumber> of the lowest contiguous message in a Sequence range received by
469 the RM Destination~~required attribute contains an unsignedLong representing the~~
470 ~~<MessageNumber> of the lowest contiguous message in a Sequence range.~~

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

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

474 /wsrm:SequenceAcknowledgement/wsrm:Final

475 This OPTIONAL element, if present, indicates that the RM Destination is not receiving new
476 messages for the specified Sequence. The RM Source can be assured that the ranges of
477 messages acknowledged by this SequenceAcknowledgement header block will not change in the
478 future. This element MUST be present when the Sequence is no longer receiving new message
479 for the specified sequence. Note: this element MUST NOT be used when sending a Nack, it can
480 only be used when sending AcknowledgementRanges.

481 /wsrm:SequenceAcknowledgement/wsrm:Nack

482 This OPTIONAL element, if present, MUST contain an xs:unsignedLong representing the
483 <wsrm:MessageNumber> of an unreceived message in a Sequence. This element permits the
484 gap analysis of the <wsrm:AcknowledgementRange> elements to be performed at the RM
485 Destination rather than at the RM Source which may yield performance benefits in certain
486 environments. The <wsrm:Nack> element MUST NOT be present if either the
487 <wsrm:AcknowledgementRange> or <wsrm:None> elements are also present as a child of
488 <wsrm:SequenceAcknowledgement>. Upon the receipt of a Nack, an RM Source SHOULD
489 retransmit the message identified by the Nack. The RM Destination MUST NOT issue a
490 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that it has
491 previously acknowledged within a <wsrm:AcknowledgementRange>. The RM Source SHOULD
492 ignore a <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message
493 that has previously been acknowledged within a <wsrm:AcknowledgementRange>~~optional~~
494 ~~element, if present, MUST contain an unsignedLong representing the~~
495 ~~<MessageNumber> of an unreceived message in a Sequence. This element~~
496 ~~MUST NOT be present if the <AcknowledgementRange> is also present as a~~
497 ~~child of <SequenceAcknowledgement>. The <Nack> element permits the gap~~
498 ~~analysis of the <AcknowledgementRange> elements to be performed at the~~

499 ~~RM Destination rather than at the RM Source which may yield performance~~
500 ~~benefits in certain environments.~~

501 [/wsrm:SequenceAcknowledgement/wsrm:None](#)

502 This OPTIONAL element, if present, MUST be used when the RM Destination has not received
503 any messages for the specified sequence. The <wsrm:None> element MUST NOT be present if
504 either the <wsrm:AcknowledgementRange> or <wsrm:Nack> elements are also present as a
505 child of the <wsrm:SequenceAcknowledgement>.

506 /wsrm:SequenceAcknowledgement/{any}

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

509 /wsrm:SequenceAcknowledgement/@{any}

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

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

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

```
514 <wsrm:SequenceAcknowledgement>  
515 <wsrm:Identifier>http://examplefabrikam123.com/abc</wsrm:Identifier>  
516 <wsrm:AcknowledgementRange Upper="10" Lower="1"/>  
517 </wsrm:SequenceAcknowledgement>
```

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

```
520 <wsrm:SequenceAcknowledgement>  
521 <wsrm:Identifier>http://examplefabrikam123.com/abc</wsrm:Identifier>  
522 <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
523 <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
524 <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
525 </wsrm:SequenceAcknowledgement>
```

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

```
527 <wsrm:SequenceAcknowledgement>  
528 <wsrm:Identifier>http://examplefabrikam123.com/abc</wsrm:Identifier>  
529 <wsrm:Nack>3</wsrm:Nack>  
530 </wsrm:SequenceAcknowledgement>
```

531 3.3 Request Acknowledgement

532 The purpose of the ~~<wsmr:AckRequested> header block is to signal to the RM~~
533 ~~Destination that the RM Source is requesting that a <wsmr:AckRequested> header~~
534 ~~block is to signal to the RM Destination that the RM Source is~~
535 ~~requesting that a <SequenceAcknowledgement> be returned.~~

536 At any time, the RM Source may request an acknowledgement message from the RM
537 Destination ~~endpoint using an <wsmr:point using an <AckRequested> header~~
538 ~~block.~~

539 The RM Source endpoint requests this acknowledgement by including an
540 ~~<wsmr:AckRequested> header block in the message. An RM Destination that receives~~
541 ~~a message that contains an <wsmr:AckRequested> header block MUST respond with~~
542 ~~a message containing a <wsmr:SequenceAcknowledgement> header block. If a non-~~
543 ~~mustUnderstand fault occurs when processing an RM Header that was piggy-backed~~
544 ~~on another message, a fault MUST be generated, but the processing of the original~~
545 ~~message MUST NOT be affected.~~~~<AckRequested> header block in the message. An RM~~
546 ~~Destination that receives a message that contains an <AckRequested> header block~~
547 ~~MUST respond with a message containing a <SequenceAcknowledgement> header~~
548 ~~block.~~

549 The following exemplar defines its syntax:

```
550 <wsmr:AckRequested ...>  
551   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
552   <wsmr:MessageNumber> xs:unsignedLong </wsmr:MessageNumber> ?  
553   ...  
554 </wsmr:AckRequested>
```

555 /wsmr:AckRequested

556 This element requests an acknowledgement for the identified sequence.

557 /wsmr:AckRequested/wsmr:Identifier

558 This ~~REQUIRED~~~~required~~ element MUST contain an absolute URI, conformant with RFC2396,
559 that uniquely identifies the Sequence to which the request applies.

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

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

563 /wsmr:AckRequested/wsmr:MessageNumber

564 This ~~OPTIONAL~~ element, if present, MUST contain an xs:unsignedLong representing the highest
565 ~~<wsmr:MessageNumber> sent by the RM Source within the Sequence. If present, it MAY be~~

File name

Date

566 ~~treated as a hint to the RM Destination as an optimization to the process of preparing to transmit a~~
567 ~~<wsrm:optional element, if present, MUST contain an xs:unsignedLong~~
568 ~~representing the highest <MessageNumber> sent by the RM Source within a~~
569 ~~Sequence. If present, it MAY be treated as a hint to the RM Destination~~
570 ~~as an optimization to the process of preparing to transmit a~~
571 ~~<SequenceAcknowledgement>.~~

572 /wsrm:AckRequested/{any}

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

575 /wsrm:AckRequested/@{any}

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

578 3.4 Sequence Creation

579 The RM Source MUST request creation of an outbound Sequence by sending a
580 ~~<wsrm:CreateSequence> element in the body of a message to the RM Destination~~
581 ~~which in turn responds either with a <wsrm:CreateSequenceResponse> or a~~
582 ~~CreateSequenceRefused fault in the body of the response message.~~
583 ~~<wsrm:CreateSequence> MAY carry an offer to create an inbound sequence which is~~
584 ~~either accepted or rejected in the <wsrm:CreateSequence> element in the body~~
585 ~~of a message to the RM Destination which in turn responds either with a~~
586 ~~<CreateSequenceResponse> or a CreateSequenceRefused fault in the body~~
587 ~~of the response message. <CreateSequence> MAY carry an offer to create~~
588 ~~an inbound sequence which is either accepted or rejected in the~~
589 ~~<CreateSequenceResponse>.~~

590 The RM Destination of the outbound sequence is the WS-Addressing
591 EndpointReference [WS-Addressing] to which <wsrm:CreateSequence> is sent. The
592 ~~RM Destination of the inbound sequence is the WS-Addressing <wsa:ReplyTo> of the~~
593 ~~<wsrm:to which <CreateSequence> is sent. The RM Destination of the~~
594 ~~inbound sequence is the WS Addressing <wsa:ReplyTo> of the~~
595 ~~<CreateSequence>.~~

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

```
597 <wsrm:CreateSequence ...>  
598   <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>  
599   <wsrm:Expires ...> xs:duration </wsrm:Expires> ?  
600   <wsrm:Offer ...>  
601     <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
```

File name

Date

```

602     <wsm:Expires ...> xs:duration </wsm:Expires> ?
603     ...
604     </wsm:Offer> ?
605     ...
606     <wsse:SecurityTokenReference>
607     ...
608     </wsse:SecurityTokenReference> ?
609     ...
610 </wsm:CreateSequence>

```

611 /wsm:CreateSequence

612 This element requests creation of a new Sequence between the RM Source that sends it, and the
613 RM Destination to which it is sent. This element **MUST NOT** be sent as a header block. The RM
614 Destination **MUST** respond either with a [<wsm:CreateSequenceResponse>](#) response
615 message or a `CreateSequenceRefused` fault.

616 /wsm:CreateSequence/wsm:AcksTo

617 This **REQUIRED** element, of type `wsa:EndpointReferenceType` as specified by [WS-Addressing](#)
618 [\[WS-Addressing\]](#) specifies the endpoint reference to which ~~<wsm:required element, of~~
619 ~~type wsa:EndpointReferenceType as specified by WS-Addressing [WS-ADDR]~~
620 ~~specifies the endpoint reference to which <SequenceAcknowledgement>~~
621 messages and faults related to the created Sequence are to be sent.

622 /wsm:CreateSequence/wsm:Expires

623 This element, if present, of type `xs:duration` specifies the RM Source's requested duration for
624 the Sequence. The RM Destination **MAY** either accept the requested duration or assign a lesser
625 value of its choosing. A value of 'PT0S' **indicates that the Sequence will never expire. Absence of**
626 **the element indicates an implied value of 'PT0S' indicates that the Sequence will never expire.**
627 **Absence of the element indicates an implied value of 'P0S'.**

628 /wsm:CreateSequence/wsm:Expires/@{any}

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

631 /wsm:CreateSequence/wsm:Offer

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

634 /wsm:CreateSequence/wsm:Offer/wsm:Identifier

635 This **REQUIRED**~~required~~ element **MUST** contain an absolute URI conformant with RFC2396 that
636 uniquely identifies the offered Sequence.

File name

Date

637 /wsmr:CreateSequence/wsmr:Offer/wsmr:Identifier/@{any}
638 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
639 to the element.

640 /wsmr:CreateSequence/wsmr:Offer/wsmr:Expires
641 This element, if present, of type `xs:duration` specifies the duration for the Sequence. A value
642 of 'PT0S' indicates that the Sequence will never expire. Absence of the element indicates an
643 implied value of 'PT0S'.

644 /wsmr:CreateSequence/wsmr:Offer/wsmr:Expires/@{any}
645 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
646 to the element.

647 /wsmr:CreateSequence/wsmr:Offer/{any}
648 This is an extensibility mechanism to allow different (extensible) types of information, based on a
649 schema, to be passed.

650 /wsmr:CreateSequence/wsmr:Offer/@{any}
651 This is an extensibility mechanism to allow different (extensible) types of information, based on a
652 schema, to be passed.

653 ~~OPTIONAL/wsmr:CreateSequence/{any}~~
654 ~~/wsmr:CreateSequence/wsse:SecurityTokenReference~~
655 ~~This optional element uses the extensibility mechanism defined next to communicate an explicit~~
656 ~~reference to the security token to be used to authorize messages for the created outbound~~
657 ~~Sequence and if offered the inbound Sequence, using a <wsse:SecurityTokenReference>~~
658 ~~as documented in WS-Security [WSecurity]. All subsequent messages in the outbound~~
659 ~~Sequence and if offered the inbound Sequence MUST demonstrate proof-of-possession of the~~
660 ~~referenced key.~~

661 ~~/wsmr:CreateSequence/{any}~~
662 This is an extensibility mechanism to allow different (extensible) types of information, based on a
663 schema, to be passed.

664 /wsmr:CreateSequence/@{any}
665 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
666 to the element.

667 ~~A <wsmr:CreateSequenceResponse> is sent in the body of a response message by an~~
668 ~~RM Destination in response to receipt of a <wsmr:CreateSequence> request~~
669 ~~message. It carries the <wsmr:CreateSequenceResponse> is sent in the body of~~

670 ~~a response message by an RM Destination in response to receipt of a~~
671 ~~<CreateSequence> request message. It carries the <Identifier> of the~~
672 created Sequence and indicates that the RM Source may begin sending messages in
673 the context of the identified Sequence.

674 The following exemplar defines the `<wsmr:CreateSequenceResponse>` syntax:

```
675 <wsmr:CreateSequenceResponse ...>  
676   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
677   <wsmr:Expires> xs:duration </wsmr:Expires> ?  
678   <wsmr:Accept ...>  
679     <wsmr:AcksTo ...> wsa:EndpointReferenceType </wsmr:AcksTo>  
680     ...  
681   </wsmr:Accept> ?  
682   ...  
683 </wsmr:CreateSequenceResponse>
```

684 `/wsmr:CreateSequenceResponse`

685 This element is sent in the body of the response message in response to a
686 `<wsmr:CreateSequence>` request message. It indicates that the RM Destination has created
687 a new Sequence at the request of the RM Source. This element MUST NOT be sent as a header
688 block.

689 `/wsmr:CreateSequenceResponse/wsmr:Identifier`

690 This ~~REQUIRED~~required element MUST contain an absolute URI conformant with RFC2396 of
691 the Sequence that has been created by the RM Destination.

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

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

695 `/wsmr:CreateSequenceResponse/wsmr:Expires`

696 This element, if present, of type `xs:duration` accepts or refines the RM Source's requested
697 duration for the Sequence. A value of 'PT0S' ~~indicates that the Sequence will never expire.~~
698 Absence of the element indicates an implied value of 'PT0S'. This value MUST be equal or lesser
699 than the value requested by the RM Source in the corresponding <wsmr:OS' indicates that
700 ~~the Sequence will never expire. Absence of the element indicates an~~
701 ~~implied value of 'POS'. This value MUST be equal or lesser than the~~
702 ~~value requested by the RM Source in the corresponding <CreateSequence>~~
703 message.

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

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

707 /wsrm:CreateSequenceResponse/wsrm:Accept

708 This element, if present, enables an RM Destination to accept the offer of a corresponding
709 Sequence for the reliable exchange of messages transmitted from RM Destination to RM Source.
710 This element MUST be present if the corresponding <wsrm:CreateSequence> message
711 contained an <wsrm:CreateSequence> message contained an <Offer> element.

712 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo

713 This REQUIRED element, of type wsa:EndpointReferenceType as specified by WS-Addressing
714 [WS-Addressing], specifies the endpoint reference to which <wsrm:required element, of
715 type wsa:EndpointReferenceType as specified by WS-Addressing [WS-
716 Addressing], specifies the endpoint reference to which
717 <SequenceAcknowledgement> messages related to the accepted Sequence are to be sent.

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

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

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

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

724 /wsrm:CreateSequenceResponse/{any}

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

727 /wsrm:CreateSequenceResponse/@{any}

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

730 **3.5 Sequence Termination**

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

738 been acknowledged. However, the RM Source is free to Terminate or Close a
739 Sequence at any time regardless of the acknowledgement state of the messagesAfter
740 an RM Source receives the <SequenceAcknowledgement> acknowledging the
741 complete range of messages in a Sequence, it sends a <TerminateSequence>
742 element, in the body of a message to the RM Destination to indicate that the
743 Sequence is complete, and that it will not be sending any further messages related to
744 the Sequence. The RM Destination can safely reclaim any resources associated with
745 the Sequence upon receipt of the <TerminateSequence> message.

746 The following exemplar defines the TerminateSequence syntax:

```
747 <wsrm:TerminateSequence ...>  
748   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
749   ...  
750 </wsrm:TerminateSequence>
```

751 /wsrm:TerminateSequence

752 This element is sent by an RM Source to indicate it has completed its use of the Sequence, i.e. it
753 MUST NOT send any additional message to the RM Destination referencing this safter it has
754 received the final <SequenceAcknowledgement> covering the full range of a Sequence. It
755 indicates that the RM Destination can safely reclaim any resources related to the identified
756 Sequence. This element MUST NOT be sent as a header block.

757 /wsrm:TerminateSequence/wsrm:Identifier

758 This ~~REQUIRED~~required element MUST contain an absolute URI conformant with RFC2396 of
759 the Sequence that is being terminated.

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

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

763 /wsrm:TerminateSequence/{any}

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

766 /wsrm:TerminateSequence/@{any}

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

769 **3.6 Closing A Sequence**

770 There may be times during the use of an RM Sequence that the RM Source or RM
771 Destination will wish to discontinue using a Sequence even if some of the messages
772 have not been successfully delivered to the RM Destination.

773 In the case where the RM Source wishes to discontinue use of a sequence, while it
774 can send a TerminateSequence to the RM Destination, since this is a one-way
775 message and due to the possibility of late arriving (or lost) messages and
776 Acknowledgements, this would leave the RM Source unsure of the final ranges of
777 messages that were successfully delivered to the RM Destination.

778 To alleviate this, the RM Source can send a <wsrm:CloseSequence> element, in the
779 body of a message, to the RM Destination to indicate that RM Destination MUST NOT
780 receive any new messages for the specified sequence, other than those already
781 received at the time the <wsrm:CloseSequence> element is interpreted by the RMD.
782 Upon receipt of this message the RM Destination MUST send a
783 SequenceAcknowledgement to the RM Source. Note, this
784 SequenceAcknowledgement MUST include the <wsrm:Final> element.

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

789 In the case where the RM Destination wishes to discontinue use of a sequence it may
790 'close' the sequence itself. Please see wsrm:Final above and the SequenceClosed
791 fault below. Note, the SequenceClosed Fault SHOULD be used in place of the
792 SequenceTerminated Fault, whenever possible, to allow the RM Source to still receive
793 Acknowledgements.

794 The following exemplar defines the CloseSequence syntax:

```
795 <wsrm:CloseSequence wsrm:Identifier="xs:anyURI"/>
```

796 ./wsrm:CloseSequence

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

800 ./wsrm:CloseSequence@Identifier

801 This REQUIRED attribute contains an absolute URI conformant with RFC2396 that uniquely
802 identifies the sequence.

803 ./wsrm:CloseSequence/{any}

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

806 /wsrm:CloseSequence@{any}

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

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

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

813 `/wsrm:CloseSequenceResponse`

814 /wsrm:CloseSequenceResponse

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

818 /wsrm:CloseSequenceResponse/{any}

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

821 /wsrm:CloseSequenceResponse@{any}

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

824 4 Faults

825 The fault definitions defined in this section reference certain abstract properties, such
826 as [fault endpoint], that are defined in section 3 of the WS-Addressing [WS-
827 Addressing] specification. Endpoints compliant with this specification MUST include
828 required ~~Message Addressing Property~~~~message information header~~s on all fault
829 messages.

830 Sequence creation uses a CreateSequence, CreateSequenceResponse request-
831 ~~response pattern. Faults for this operation are treated as defined in WS-Addressing.~~
832 ~~CreateSequenceRefused is a possible fault reply for this operation.~~
833 ~~UnknownSequence is a fault generated by endpoints when messages carrying RM~~
834 ~~header blocks targeted at unrecognized sequences are detected, these faults are also~~
835 ~~treated as defined in WS-Addressing. All other faults in this section relate to the~~
836 ~~processing of RM header blocks targeted at known sequences and are collectively~~
837 ~~referred to as sequence faults. Sequence faults SHOULD be sent to the same~~
838 ~~[destination] as <wsrm:SequenceAcknowledgement> messages. These faults are~~
839 ~~correlated using the S-reply. Faults for this operation are treated as defined in WS-~~
840 ~~Addressing. CreateSequenceRefused is a possible fault reply for this operation.~~
841 ~~UnknownSequence is a fault generated by endpoints when messages carrying RM~~
842 ~~header blocks targeted at unrecognized sequences are detected, these faults are also~~
843 ~~treated as defined in WS-Addressing. All other faults in this section relate to the~~
844 ~~processing of RM header blocks targeted at known sequences and are collectively~~
845 ~~referred to as sequence faults. Sequence faults SHOULD be sent to the same~~
846 ~~[destination] as <SequenceAcknowledgement> messages. These faults are correlated~~
847 ~~using the~~ sequence identifier carried in the detail.

848 WS-ReliableMessaging faults MUST include as the [action] property the default fault
849 action URI defined in the version of WS-Addressing used in the message. The value
850 from the current version is below for informational purposes:

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

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

854 The definitions of faults use the following properties:

855 [Code] The fault code.

856 [Subcode] The fault subcode.

857 [Reason] The English language reason element.

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

File name

Date

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

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

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

```

862 <S:Envelope>
863   <S:Header>
864     <wsa:Action>
865       http://schemas.xmlsoap.org/ws/2004/08/addressing/fault
866     </wsa:Action>
867     <!-- Headers elided for clarity. -->
868   </S:Header>
869   <S:Body>
870     <S:Fault>
871       <S:Code>
872         <S:Value> [Code] </S:Value>
873         <S:Subcode>
874           <S:Value> [Subcode] </S:Value>
875         </S:Subcode>
876       </S:Code>
877       <S:Reason>
878         <S:Text xml:lang="en"> [Reason] </S:Text>
879       </S:Reason>
880       <S:Detail>
881         [Detail]
882         ...
883       </S:Detail>
884     </S:Fault>
885   </S:Body>
886 </S:Envelope>
  
```

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

```

889 <S11:Envelope>
890   <S11:Header>
891     <wsrm:SequenceFault>
  
```

```

892     <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
893     ...
894 </wsrm:SequenceFault>
895 <!-- Headers elided for clarity. -->
896 </S11:Header>
897 <S11:Body>
898 <S11:Fault>
899 <faultcode> [Code] </faultcode>
900 <faultstring> [Reason] </faultstring>
901 </S11:Fault>
902 </S11:Body>
903 </S11:Envelope>

```

904 The properties bind to a SOAP 1.1 fault as follows when the fault is generated as a
905 result of processing a [<wsrm:CreateSequence>](#) request message:

```

906 <S11:Envelope>
907 <S11:Body>
908 <S11:Fault>
909 <faultcode> [Subcode] </faultcode>
910 <faultstring xml:lang="en"> [Reason] </faultstring>
911 </S11:Fault>
912 </S11:Body>
913 </S11:Envelope>

```

914 4.1 SequenceFault Element

915 The purpose of the [<wsrm:SequenceFault>](#) element is to carry the specific details of
916 a fault generated during the reliable messaging specific processing of a message
917 belonging to a Sequence. ~~The <wsrm:SequenceFault> element is to carry the~~
918 ~~specific details of a fault generated during the reliable messaging~~
919 ~~specific processing of a message belonging to a Sequence. The~~
920 <SequenceFault> container MUST only be used in conjunction with the SOAP1.1
921 fault mechanism. It MUST NOT be used in conjunction with the SOAP1.2 binding.

922 The following exemplar defines its syntax:

```

923 <wsrm:SequenceFault ...>
924 <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
925 ...
926 </wsrm:SequenceFault>

```

927 The following describes the content model of the `SequenceFault` element.
928 `/wsrm:SequenceFault`
929 This is the element containing Sequence information for WS-ReliableMessaging
930 `/wsrm:SequenceFault/wsrm:FaultCode`
931 This element, if present, MUST contain a qualified name from the set of fault codes defined
932 below.
933 `/wsrm:SequenceFault/{any}`
934 This is an extensibility mechanism to allow different (extensible) types of information, based on a
935 schema, to be passed.
936 `/wsrm:SequenceFault/@{any}`
937 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added
938 to the element.

939 **4.2 Sequence Terminated**

940 This fault is sent by either the RM Source or the RM Destination to indicate that the
941 endpoint that generated^{ds} the fault has either encountered an unrecoverable
942 condition, or has detected a violation of the protocol and as a consequence, has
943 chosen to terminate the sequence. The endpoint that generates this fault should
944 make every reasonable effort to notify the corresponding endpoint of this decision.

945 Properties:

946 [Code] Sender or Receiver

947 [Subcode] `wsrm:SequenceTerminated`

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

949 [Detail]

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

951 **4.3 Unknown Sequence**

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

954 Properties:

955 [Code] Sender

956 [Subcode] wsrn:UnknownSequence

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

958 [Detail]

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

960 4.4 Invalid Acknowledgement

961 This fault is sent by the RM Source in response to a

962 <wsrn:SequenceAcknowledgement> that violates the cumulative acknowledgement

963 invariant. An example of such a violation would be a SequenceAcknowledgement

964 covering messages that have not been sent.

965 [Code] Sender

966 [Subcode] wsrn:InvalidAcknowledgement

967 [Reason] The SequenceAcknowledgement violates the cumulative acknowledgement

968 invariant.

969 [Detail]

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

971 4.5 Message Number Rollover

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

973 exhausted. ~~It is an unrecoverable error and terminates the Sequence.~~

974 Properties:

975 [Code] Sender

976 [Subcode] wsrn:MessageNumberRollover

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

978 [Detail]

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

980 4.6 Last Message Number Exceeded

981 This fault is sent by an RM Destination to indicate that it has received a message that

982 has a <wsrn:MessageNumber> within a Sequence that exceeds the value of the

File name

Date

Copyright © OASIS Open 2005. All Rights Reserved.
numbers of Statistics

Page Page

983 ~~<wsrm:MessageNumber> element that accompanied a <wsrm:LastMessage> element~~
984 ~~for <MessageNumber> within a Sequence that exceeds the value of the~~
985 ~~<MessageNumber> element that accompanied a <LastMessage> element for the~~
986 ~~Sequence. This is an unrecoverable error and terminates the Sequence.~~

987 Properties:

988 [Code] Sender

989 [Subcode] wsrm:LastMessageNumberExceeded

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

992 [Detail]

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

994 **4.7 Create Sequence Refused**

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

996 Properties:

997 [Code] Sender

998 [Subcode] wsrm:CreateSequenceRefused

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

1000 [Detail] empty

1001 **4.8 Sequence Closed**

1002 This fault is sent by an RM Destination to indicate that the specified sequence has
1003 been closed. This fault MUST be generated when an RM Destination is asked to
1004 receive a message for a sequence that is closed.

1005 Properties:

1006 [Code] Sender

1007 [Subcode] wsrm:SequenceClosed

1008 [Reason] The sequence is closed and can not receive new messages.

1009 [Detail] <wsrm:Identifier...> xs:anyURI </wsrm:Identifier>

1010 **5 Security Considerations**

1011 It is strongly recommended that the communication between services be secured
1012 using the mechanisms described in WS-Security. In order to properly secure
1013 messages, the body and all relevant headers need to be included in the signature.
1014 Specifically, the `<wsrm:Sequence>` header needs to be signed with the body in order
1015 to "bind" the two together. The `<wsrm:SequenceAcknowledgement>` header may be
1016 signed independently because a reply independent of the message is not a security
1017 concern.

1018 Because Sequences are expected to exchange a number of messages, it is
1019 recommended that a security context be established using the mechanisms described
1020 in WS-Trust and WS-SecureConversation [\[SecureConversation\]](#). If a Sequence is
1021 bound to a specific endpoint, then the security context needs to be established or
1022 shared with the endpoint servicing the Sequence. While the context can be
1023 established at any time, it is critical that the messages establishing the Sequence be
1024 secured even if they precede security context establishment. However, it is
1025 recommended that the security context be established first. Security contexts are
1026 independent of reliable messaging Sequences. Consequently, security contexts can
1027 come and go independent of the lifetime of the Sequence. In fact, it is
1028 recommended that the lifetime of a security context be less than the lifetime of the
1029 Sequence unless the Sequence is very short-lived.

1030 It is common for message Sequences to exchange a number of messages (or a large
1031 amount of data). As a result, the usage profile of a Sequence is such that it is
1032 susceptible to key attacks. For this reason it is strongly recommended that the keys
1033 be changed frequently. This "re-keying" can be effected a number of ways. The
1034 following list outlines four common techniques:

- 1035 • Closing and re-establishing a security context
- 1036 • Exchanging new secrets between the parties
- 1037 • Using a derived key sequence and switch "generations"
- 1038 • Attaching a nonce to each message and using it in a derived key function with the shared
1039 secret

1040 The security context may be re-established using the mechanisms described in WS-
1041 Trust and WS-SecureConversation. Similarly, secrets can be exchanged using the
1042 mechanisms described in WS-Trust. Note, however, that the current shared secret
1043 should not be used to encrypt the new shared secret. Derived keys, the preferred
1044 solution from this list, can be specified using the mechanisms described in WS-
1045 SecureConversation.

1046 There is a core tension between security and reliable messaging that can be
1047 problematic if not considered in implementations. That is, one aspect of security is
1048 to prevent message replay and the core tenet of reliable messaging is to replay
1049 messages until they are acknowledged. Consequently, if the security sub-system
1050 processes a message but a failure occurs before the reliable messaging sub-system
1051 records the message (or the message is considered "processed"), then it is possible
1052 (and likely) that the security sub-system will treat subsequent copies as replays and
1053 discard them. At the same time, the reliable messaging sub-system will likely
1054 continue to expect and even solicit the missing message(s). Care should be taken to
1055 avoid and prevent this rare condition.

1056 The following list summarizes common classes of attacks that apply to this protocol
1057 and identifies the mechanism to prevent/mitigate the attacks:

- 1058 • **Message alteration** – Alteration is prevented by including signatures of the message
1059 information using WS-Security.
- 1060 • **Message disclosure** – Confidentiality is preserved by encrypting sensitive data using WS-
1061 Security.
- 1062 • **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by
1063 comparing secured policies – see WS-Policy and WS-SecurityPolicy).
- 1064 • **Authentication** – Authentication is established using the mechanisms described in WS-
1065 Security and WS-Trust. Each message is authenticated using the mechanisms described in
1066 WS-Security.
- 1067 • **Accountability** – Accountability is a function of the type of and string of the key and
1068 algorithms being used. In many cases, a strong symmetric key provides sufficient
1069 accountability. However, in some environments, strong PKI signatures are required.
- 1070 • **Availability** – All reliable messaging services are subject to a variety of availability attacks.
1071 Replay detection is a common attack and it is recommended that this be addressed by the
1072 mechanisms described in WS-Security. (Note that because of legitimate message replays,
1073 detection should include a differentiator besides message id such as a timestamp). Other
1074 attacks, such as network-level denial of service attacks are harder to avoid and are outside
1075 the scope of this specification. That said, care should be taken to ensure that minimal state is
1076 saved prior to any authenticating sequences.

1077 **6 References**

1078 **6.1 Normative**

1079 **[KEYWORDS]**

1080 S. Bradner, "[Key words for use in RFCs to Indicate Requirement Levels](#)," RFC 2119, Harvard
1081 University, March 1997

1082 **[SOAP]**

1083 W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May 2000.

1084 **[URI]**

1085 T. Berners-Lee, R. Fielding, L. Masinter, "[Uniform Resource Identifiers \(URI\): Generic Syntax](#),"
1086 RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.

1087 **[XML-ns]**

1088 W3C Recommendation, "[Namespaces in XML](#)," 14 January 1999.

1089 **[XML-Schema1]**

1090 W3C Recommendation, "[XML Schema Part 1: Structures](#)," 2 May 2001.

1091 **[XML-Schema2]**

1092 W3C Recommendation, "[XML Schema Part 2: Datatypes](#)," 2 May 2001.

1093 **[WSSecurity]**

1094 "[OASIS Web Services Security: SOAP Message Security 1.0 \(WS-Security 2004\)](#)",
1095 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds, OASIS
1096 Standard 200401, March 2004.

1097 **[SecureConversation]**

1098 ~~S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#),"~~
1099 ~~May 2004.~~

1100 **[Tanenbaum]**

1101 "Computer Networks," Andrew S. Tanenbaum, Prentice Hall PTR, 2003.

1102 **[WSDL]**

1103 W3C Note, "[Web Services Description Language \(WSDL 1.1\)](#)," 15 March 2001.

1104 **[WS-Addressing]**

1105 D. Box, et al, "[Web Services Addressing \(WS-Addressing\)](#)," August 2004.

File name

Date

Copyright © OASIS Open 2005. All Rights Reserved.
numbers of Statistics

Page Page

1106 **6.2 Non-Normative**

1107 **[WS-Policy]**

1108 D. Box, et al, "[Web Services Policy Framework \(WS-Policy\)](#)," September 2004.

1109 **[WS-PolicyAttachment]**

1110 D. Box, et al, "[Web Services Policy Attachment \(WS-PolicyAttachment\)](#)," September 2004.

1111 **[SecurityPolicy]**

1112 G. Della-Libra, "[Web Services Security Policy Language \(WS-SecurityPolicy\)](#)," December 2002.

1113 **[SecureConversation]**

1114 [S. Anderson, et al, "Web Services Secure Conversation Language \(WS-SecureConversation\),"](#)
1115 [May 2004.](#)

1116

1117 Appendix A.Schema

1118 The normative schema for WS-ReliableMessaging is located at:

```
1119 http://docs.oasis-  
1120 open.org/wsrn/200510/schemas.xmlsoap.org/ws/2005/02/rm/wsrn.xsd
```

1121 The following copy is provided for reference.

```
1122 <xs:schema targetNamespace="http://docs.oasis-open.org/wsrn/200510/"  
1123 xmlns:xs="http://www.w3.org/2001/XMLSchema"  
1124 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"  
1125 xmlns:wsrm="http://docs.oasis-  
1126 open.org/wsrn/200510/schemas.xmlsoap.org/ws/2005/02/rm"  
1127 xmlns:xs="http://www.w3.org/2001/XMLSchema"  
1128 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"  
1129 xmlns:wsrm="http://schemas.xmlsoap.org/ws/2005/02/rm"  
1130 elementFormDefault="qualified" attributeFormDefault="unqualified">  
1131   <xs:import  
1132     namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"  
1133     schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>  
1134   <!-- Protocol Elements -->  
1135   <xs:complexType name="SequenceType">  
1136     <xs:sequence>  
1137       <xs:element ref="wsrm:Identifier"/>  
1138       <xs:element name="MessageNumber" type="xs:unsignedLong"/>  
1139       <xs:element name="LastMessage" minOccurs="0">  
1140         <xs:complexType>  
1141           <xs:sequence/>  
1142         </xs:complexType>  
1143       </xs:element>  
1144       <xs:any namespace="##other" processContents="lax" minOccurs="0"  
1145       maxOccurs="unbounded"/>  
1146     </xs:sequence>  
1147     <xs:anyAttribute namespace="##other" processContents="lax"/>  
1148   </xs:complexType>  
1149   <xs:element name="Sequence" type="wsrm:SequenceType"/>  
1150   <xs:element name="SequenceAcknowledgement">  
1151     <xs:complexType>  
1152       <xs:sequence>  
1153         <xs:element ref="wsrm:Identifier"/>
```

File name

Date

1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194

```
<xs:choice>
  <del>ws:sequence</del><del>xs:element name="AcknowledgementRange"
maxOccurs="unbounded">
  <del>xs:element name="AcknowledgementRange"
maxOccurs="unbounded">
    <del>xs:complexType>
      <del>xs:sequence/>
      <del>xs:attribute name="Upper" type="xs:unsignedLong"
use="required"/>
      <del>xs:attribute name="Lower" type="xs:unsignedLong"
use="required"/>
      <del>xs:anyAttribute namespace="##other"
processContents="lax"/>
    </del>xs:complexType>
  </del>xs:element>
  <del>ws:element name="Final" minOccurs="0">
    <del>xs:complexType>
      <del>xs:sequence/>
    </del>xs:complexType>
  </del>ws:element>
</del>ws:sequence>
  <del>xs:element name="Nack" type="xs:unsignedLong"
maxOccurs="unbounded"/>
  <del>xs:element name="None" minOccurs="0">
    <del>xs:complexType>
      <del>xs:sequence/>
      <del>xs:attribute name="Upper" type="xs:unsignedLong"
use="required"/>
      <del>xs:attribute name="Lower" type="xs:unsignedLong"
use="required"/>
      <del>xs:anyAttribute namespace="##other"
processContents="lax"/>
    </del>xs:complexType>
  </del>xs:element>
  <del>xs:element name="Nack" type="xs:unsignedLong"
maxOccurs="unbounded"/>
</del>xs:choice>
  <del>xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
</del>xs:sequence>
<del>xs:anyAttribute namespace="##other" processContents="lax"/>
```

File name

Date

```

1195     </xs:complexType>
1196 </xs:element>
1197 <xs:complexType name="AckRequestedType">
1198   <xs:sequence>
1199     <xs:element ref="wsrm:Identifier"/>
1200     <xs:element name="MessageNumber<del>axMessageNumberUsed"
1201 type="xs:unsignedLong" minOccurs="0"/>
1202     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1203 maxOccurs="unbounded"/>
1204   </xs:sequence>
1205   <xs:anyAttribute namespace="##other" processContents="lax"/>
1206 </xs:complexType>
1207 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1208 <xs:element name="Identifier">
1209   <xs:complexType>
1210     <xs:annotation>
1211       <xs:documentation>
1212 This type is for elements whose [children] is an anyURI and can have
1213 arbitrary attributes.
1214       </xs:documentation>
1215     </xs:annotation>
1216     <xs:simpleContent>
1217       <xs:extension base="xs:anyURI">
1218         <xs:anyAttribute namespace="##other" processContents="lax"/>
1219       </xs:extension>
1220     </xs:simpleContent>
1221   </xs:complexType>
1222 </xs:element>
1223 <!-- Fault Container and Codes -->
1224 <xs:simpleType name="FaultCodes">
1225   <xs:restriction base="xs:QName">
1226     <xs:enumeration value="wsrm:UnknownSequence"/>
1227     <xs:enumeration value="wsrm:SequenceTerminated"/>
1228     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1229     <xs:enumeration value="wsrm:MessageNumberRollover"/>
1230     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1231     <del><xs:enumeration value="wsrm:LastMessageNumberExceeded"/>
1232   </xs:restriction>
1233 </xs:simpleType>
1234 <xs:complexType name="SequenceFaultType">
1235   <xs:sequence>

```

```

1236     <xs:element name="FaultCode" type="wsmr:FaultCodesxs-QName"/>
1237     <xs:any namespace="##any" processContents="lax" minOccurs="0"
1238 maxOccurs="unbounded"/>
1239     </xs:sequence>
1240     <xs:anyAttribute namespace="##any" processContents="lax"/>
1241 </xs:complexType>
1242 <xs:element name="SequenceFault" type="wsmr:SequenceFaultType"/>
1243 <xs:element name="CreateSequence" type="wsmr:CreateSequenceType"/>
1244 <xs:element name="CreateSequenceResponse"
1245 type="wsmr:CreateSequenceResponseType"/>
1246 <xs:element name="CloseSequence" type="wsmr:CloseSequenceType"/>
1247 <xs:element name="CloseSequenceResponse"
1248 type="wsmr:CloseSequenceResponseType"/>
1249 <xs:element name="TerminateSequence"
1250 type="wsmr:TerminateSequenceType"/>
1251 <xs:complexType name="CreateSequenceType">
1252 <xs:sequence>
1253 <xs:element ref="wsmr:AcksTo"/>
1254 <xs:element ref="wsmr:Expires" minOccurs="0"/>
1255 <xs:element name="Offer" type="wsmr:OfferType" minOccurs="0"/>
1256 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1257 maxOccurs="unbounded">
1258 <xs:annotation>
1259 <xs:documentation>
1260 It is the authors intent that this extensibility be used to transfer a
1261 Security Token Reference as defined in WS-Security.
1262 </xs:documentation>
1263 </xs:annotation>
1264 </xs:any>
1265 </xs:sequence>
1266 <xs:anyAttribute namespace="##other" processContents="lax"/>
1267 </xs:complexType>
1268 <xs:complexType name="CreateSequenceResponseType">
1269 <xs:sequence>
1270 <xs:element ref="wsmr:Identifier"/>
1271 <xs:element ref="wsmr:Expires" minOccurs="0"/>
1272 <xs:element name="Accept" type="wsmr:AcceptType" minOccurs="0"/>
1273 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1274 maxOccurs="unbounded"/>
1275 </xs:sequence>
1276 <xs:anyAttribute namespace="##other" processContents="lax"/>

```

File name

Date

1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317

```
</xs:complexType>
<xs:complexType name="CloseSequenceType">
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="Identifier" type="xs:anyURI" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="CloseSequenceResponseType">
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="TerminateSequenceType">
  <xs:sequence>
    <xs:element ref="wsrm:Identifier"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
<xs:complexType name="OfferType">
  <xs:sequence>
    <xs:element ref="wsrm:Identifier"/>
    <xs:element ref="wsrm:Expires" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="AcceptType">
  <xs:sequence>
    <xs:element ref="wsrm:AcksTo"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

File name

Date

```
1318 </xs:complexType>
1319 <xs:element name="Expires">
1320   <xs:complexType>
1321     <xs:simpleContent>
1322       <xs:extension base="xs:duration">
1323         <xs:anyAttribute namespace="##other" processContents="lax"/>
1324       </xs:extension>
1325     </xs:simpleContent>
1326   </xs:complexType>
1327 </xs:element>
1328 </xs:schema>
```

1329 **Appendix B.Message Examples**

File name

Date

Copyright © OASIS Open 2005. All Rights Reserved.
numbers of Statistics

Page Page

1330 B.1.Create Sequence

1331 Create Sequence

```
1332 <?xml version="1.0" encoding="UTF-8"?>
1333 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1334 xmlns:wsm="http://docs.oasis-
1335 open.org/wsm/200510/schemas.xmlsoap.org/ws/2005/02/rm"
1336 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1337   <S:Header>
1338     <wsa:MessageID>
1339       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
1340     </wsa:MessageID>
1341     <wsa:To>http://examplefabrikam123.com/serviceB/123</wsa:To>
1342     <wsa:Action>http://docs.oasis-
1343 open.org/wsm/200510/schemas.xmlsoap.org/ws/2005/02/rm/CreateSequence</w
1344 sa:Action>
1345     <wsa:ReplyTo>
1346       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1347     </wsa:ReplyTo>
1348   </S:Header>
1349   <S:Body>
1350     <wsm:CreateSequence>
1351       <wsm:AcksTo>
1352         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1353       </wsm:AcksTo>
1354     </wsm:CreateSequence>
1355   </S:Body>
1356 </S:Envelope>
```

1357 Create Sequence Response

```
1358 <?xml version="1.0" encoding="UTF-8"?>
1359 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1360 xmlns:wsm="http://docs.oasis-
1361 open.org/wsm/200510/schemas.xmlsoap.org/ws/2005/02/rm"
1362 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1363   <S:Header>
1364     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1365     <wsa:RelatesTo>
```

File name

Date

```
1366     http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
1367     </wsa:RelatesTo>
1368     <wsa:Action>
1369         http://docs.oasis-
1370 open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/CreateSequenceRes
1371 ponse
1372     </wsa:Action>
1373     </S:Header>
1374     <S:Body>
1375         <wsrm:CreateSequenceResponse>
1376             <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1377         </wsrm:CreateSequenceResponse>
1378     </S:Body>
1379 </S:Envelope>
```

1380 B.2. Initial Transmission

1381 The following example WS-ReliableMessaging headers illustrate the message
1382 exchange in the above figure. The three messages have the following headers; the
1383 third message is identified as the last message in the sequence:

1384 Message 1

```
1385 <?xml version="1.0" encoding="UTF-8"?>
1386 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1387 xmlns:wsm="http://docs.oasis-
1388 open.org/wsm/200510/schemas.xmlsoap.org/ws/2005/02/rm"
1389 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1390   <S:Header>
1391     <wsa:MessageID>
1392       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
1393     </wsa:MessageID>
1394     <wsa:To>http://examplefabrikam123.com/serviceB/123</wsa:To>
1395     <wsa:From>
1396       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1397     </wsa:From>
1398     <wsa:Action>http://examplefabrikam123.com/serviceB/123/request</wsa
1399 :Action>
1400     <wsm:Sequence>
1401       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1402       <wsm:MessageNumber>1</wsm:MessageNumber>
1403     </wsm:Sequence>
1404   </S:Header>
1405   <S:Body>
1406     <!-- Some Application Data -->
1407   </S:Body>
1408 </S:Envelope>
```

1409 Message 2

```
1410 <?xml version="1.0" encoding="UTF-8"?>
1411 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1412 xmlns:wsm="http://docs.oasis-
1413 open.org/wsm/200510/schemas.xmlsoap.org/ws/2005/02/rm"
1414 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1415   <S:Header>
```

File name

Date

```

1416 <wsa:MessageID>
1417     http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1418 </wsa:MessageID>
1419 <wsa:To>http://examplefabrikam123.com/serviceB/123</wsa:To>
1420 <wsa:From>
1421     <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1422 </wsa:From>
1423 <wsa:Action>http://examplefabrikam123.com/serviceB/123/request</wsa
1424 :Action>
1425 <wsrm:Sequence>
1426     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1427     <wsrm:MessageNumber>2</wsrm:MessageNumber>
1428 </wsrm:Sequence>
1429 </S:Header>
1430 <S:Body>
1431     <!-- Some Application Data -->
1432 </S:Body>
1433 </S:Envelope>

```

1434 Message 3

```

1435 <?xml version="1.0" encoding="UTF-8"?>
1436 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1437 xmlns:wsmr="http://docs.oasis-
1438 open.org/wsmr/200510/schemas.xmlsoap.org/ws/2005/02/rm"
1439 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1440 <S:Header>
1441     <wsa:MessageID>
1442         http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1443     </wsa:MessageID>
1444     <wsa:To>http://examplefabrikam123.com/serviceB/123</wsa:To>
1445     <wsa:From>
1446         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1447     </wsa:From>
1448     <wsa:Action>http://examplefabrikam123.com/serviceB/123/request</wsa:A
1449 ction>
1450     <wsrm:Sequence>
1451         <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1452         <wsrm:MessageNumber>3</wsrm:MessageNumber>
1453         <wsrm:LastMessage/>
1454     </wsrm:Sequence>
1455 </S:Header>

```

File name

Date

Copyright © OASIS Open 2005. All Rights Reserved.
numbers of Statistics

Page Page

```
1456 <S:Body>
1457   <!-- Some Application Data -->
1458 </S:Body>
1459 </S:Envelope>
```

1460 B.3.First Acknowledgement

1461 Message number 2 has not been received by the RM Destination due to some
1462 transmission error so it responds with an acknowledgement for messages 1 and 3:

```
1463 <?xml version="1.0" encoding="UTF-8"?>
1464 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1465 xmlns:wsm="http://docs.oasis-
1466 open.org/wsm/200510/schemas-xmlsoap.org/ws/2005/02/wsm"
1467 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1468   <S:Header>
1469     <wsa:MessageID>
1470       http://examplefabrikami23.com/guid/0baaf88d-483b-4ecf-a6d8-
1471 a7c2eb546810
1472     </wsa:MessageID>
1473     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1474     <wsa:From>
1475       <wsa:Address>http://examplefabrikami23.com/serviceB/123</wsa:Address>
1476     </wsa:From>
1477     <wsa:Action>
1478       http://docs.oasis-
1479 open.org/wsm/200510/schemas-xmlsoap.org/ws/2005/02/wsm/SequenceAcknowl-
1480 edgement
1481     </wsa:Action>
1482     <wsm:SequenceAcknowledgement>
1483       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1484       <wsm:AcknowledgementRange Upper="1" Lower="1"/>
1485       <wsm:AcknowledgementRange Upper="3" Lower="3"/>
1486     </wsm:SequenceAcknowledgement>
1487   </S:Header>
1488   <S:Body/>
1489 </S:Envelope>
```

1490 B.4.Retransmission

1491 The sending endpoint discovers that message number 2 was not received so it
1492 resends the message and requests an acknowledgement:

```
1493 <?xml version="1.0" encoding="UTF-8"?>
1494 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1495 xmlns:wsm="http://docs.oasis-
1496 open.org/wsm/200510/schemas.xmlsoap.org/ws/2005/02/rm"
1497 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1498   <S:Header>
1499     <wsa:MessageID>
1500       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1501     </wsa:MessageID>
1502     <wsa:To>http://examplefabrikami23.com/serviceB/123</wsa:To>
1503     <wsa:From>
1504       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1505     </wsa:From>
1506     <wsa:Action>http://examplefabrikami23.com/serviceB/123/request</wsa:A
1507 ction>
1508     <wsm:Sequence>
1509       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1510       <wsm:MessageNumber>2</wsm:MessageNumber>
1511     </wsm:Sequence>
1512     <wsm:AckRequested>
1513       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1514     </wsm:AckRequested>
1515   </S:Header>
1516   <S:Body>
1517     <!-- Some Application Data -->
1518   </S:Body>
1519 </S:Envelope>
```

1520 B.5.Termination

1521 The RM Destination now responds with an acknowledgement for the complete
1522 sequence which can then be terminated:

```
1523 <?xml version="1.0" encoding="UTF-8"?>
1524 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1525 xmlns:wsm="http://docs.oasis-
1526 open.org/wsm/200510/schemas.xmlsoap.org/ws/2005/02/wsm"
1527 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1528   <S:Header>
1529     <wsa:MessageID>
1530       http://examplefabrikami23.com/guid/0baaf88d-483b-4ecf-a6d8-
1531 a7c2eb546811
1532     </wsa:MessageID>
1533     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1534     <wsa:From>
1535       <wsa:Address>http://examplefabrikami23.com/serviceB/123</wsa:Address>
1536     </wsa:From>
1537     <wsa:Action>
1538       http://docs.oasis-
1539 open.org/wsm/200510/schemas.xmlsoap.org/ws/2005/02/wsm/SequenceAcknowl-
1540 edgement
1541     </wsa:Action>
1542     <wsm:SequenceAcknowledgement>
1543       <wsm:Identifier>http://Business456.com/RM/ABC</wsm:Identifier>
1544       <wsm:AcknowledgementRange Upper="3" Lower="1"/>
1545     </wsm:SequenceAcknowledgement>
1546   </S:Header>
1547   <S:Body/>
1548 </S:Envelope>
```

1549 Terminate Sequence

```
1550 <?xml version="1.0" encoding="UTF-8"?>
1551 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1552 xmlns:wsm="http://docs.oasis-
1553 open.org/wsm/200510/schemas.xmlsoap.org/ws/2005/02/wsm"
1554 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1555   <S:Header>
1556     <wsa:MessageID>
```

File name

Date

```
1557     http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1558 </wsa:MessageID>
1559 <wsa:To>http://examplefabrikam123.com/serviceB/123</wsa:To>
1560 <wsa:Action>
1561     http://docs.oasis-
1562 open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/TerminateSequence
1563 </wsa:Action>
1564 <wsa:From>
1565     <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1566 </wsa:From>
1567 </S:Header>
1568 <S:Body>
1569     <wsrm:TerminateSequence>
1570         <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1571     </wsrm:TerminateSequence>
1572 </S:Body>
1573 </S:Envelope>
```

1574 Appendix C.WSDL

1575 The non-normative WSDL [1.1](#) definition for WS-ReliableMessaging is located at:

```
1576 http://docs.oasis-  
1577 open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/wsd1/wsrn.wsd1
```

1578 The following non-normative copy is provided for reference.

```
1579 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"  
1580 xmlns:xs="http://www.w3.org/2001/XMLSchema"  
1581 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"  
1582 xmlns:rm="http://docs.oasis-open.org/wsrn/200510/"  
1583 xmlns:tns="http://docs.oasis-open.org/wsrn/200510/wsd1"  
1584 targetNamespace="http://docs.oasis-  
1585 open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm"  
1586 xmlns:tns="http://schemas.xmlsoap.org/ws/2005/02/rm/wsd1"  
1587 targetNamespace="http://schemas.xmlsoap.org/ws/2005/02/rm/wsd1">  
1588 <wsdl:types>  
1589   <xs:schema>  
1590     <xs:import namespace="http://docs.oasis-open.org/wsrn/200510/"  
1591     schemaLocation="http://docs.oasis-  
1592     open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm"  
1593     schemaLocation="http://schemas.xmlsoap.org/ws/2005/02/rm/wsrn.xsd"/>  
1594     <xs:import  
1595     namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"  
1596     schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>  
1597   </xs:schema>  
1598 </wsdl:types>  
1599 <wsdl:message name="CreateSequence">  
1600   <wsdl:part name="create" element="rm:CreateSequence"/>  
1601 </wsdl:message>  
1602 <wsdl:message name="CreateSequenceResponse">  
1603   <wsdl:part name="createResponse"  
1604   element="rm:CreateSequenceResponse"/>  
1605 </wsdl:message>  
1606 <wsdl:message name="CloseSequence">  
1607   <wsdl:part name="close" element="rm:CloseSequence"/>  
1608 </wsdl:message>  
1609 <wsdl:message name="CloseSequenceResponse">  
1610   <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
```

File name

Date

```

1611 </wsdl:message>
1612 <wsdl:message name="TerminateSequence">
1613   <wsdl:part name="terminate" element="rm:TerminateSequence"/>
1614 </wsdl:message>
1615 <wsdl:portType name="SequenceAbstractPortType">
1616   <wsdl:operation name="CreateSequence">
1617     <wsdl:input message="tns:CreateSequence"
1618     wsa:Action="http://docs.oasis-
1619     open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/CreateSequence"/>
1620     <wsdl:output message="tns:CreateSequenceResponse"
1621     wsa:Action="http://docs.oasis-
1622     open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/CreateSequenceRes
1623     ponse"/>
1624   </wsdl:operation>
1625   <wsdl:operation name="CloseSequence">
1626     <wsdl:input name="tns:CloseSequence"
1627     wsa:Action="http://docs.oasis-open.org/wsrn/200510/CloseSequence"/>
1628     <wsdl:output name="tns:CloseSequenceResponse"
1629     wsa:Action="http://docs.oasis-
1630     open.org/wsrn/200510/CloseSequenceResponse"/>
1631   </wsdl:operation>
1632   <wsdl:operation name="TerminateSequence">
1633     <wsdl:input message="tns:TerminateSequence"
1634     wsa:Action="http://docs.oasis-
1635     open.org/wsrn/200510schemas.xmlsoap.org/ws/2005/02/rm/CreateSequenceRes
1636     ponse"/>
1637   </wsdl:operation>
1638 </wsdl:portType>
1639 </wsdl:definitions>

```

1640 **Appendix D.Acknowledgments**

1641 This document is based on initial contribution to OASIS WS-RX Technical Committee by the
1642 following authors: Ruslan Bilorusets, BEA, Don Box, Microsoft, Luis Felipe Cabrera, Microsoft,
1643 Doug Davis, IBM, Donald Ferguson, IBM, Christopher Ferris, IBM (Editor), Tom Freund, IBM,
1644 Mary Ann Hondo, IBM, John Ibbotson, IBM, Lei Jin, BEA, Chris Kaler, Microsoft, David
1645 Langworthy, Microsoft (Editor), Amelia Lewis, TIBCO Software, Rodney Limprecht, Microsoft,
1646 Steve Lucco, Microsoft, Don Mullen, TIBCO Software, Anthony Nadalin, IBM, Mark Nottingham,
1647 BEA, David Orchard, BEA, Jamie Roots, IBM, Shivajee Samdarshi, TIBCO Software, John
1648 Shewchuk, Microsoft, Tony Storey, IBM

1649 The following individuals have provided invaluable input into the initial contribution:

1650 Keith Ballinger, Microsoft, Stefan Batres, Microsoft, Allen Brown, Microsoft, Michael Conner, IBM,
1651 George Copeland, Microsoft, Francisco Curbera, IBM, Paul Fremantle, IBM, Steve Graham, IBM,
1652 Pat Helland, Microsoft, Rick Hill, Microsoft, Scott Hinkelman, IBM, Tim Holloway, IBM, Efim Hudis,
1653 Microsoft, Gopal Kakivaya, Microsoft, Johannes Klein, Microsoft, Frank Leymann, IBM, Martin
1654 Nally, IBM, Peter Niblett, IBM, Jeffrey Schlimmer, Microsoft, James Snell, IBM, Keith Stobie,
1655 Microsoft, Satish Thatte, Microsoft, Stephen Todd, IBM, Sanjiva Weerawarana, IBM, Roger
1656 Wolter, Microsoft

1657 The following individuals were members of the committee during the development of this
1658 specification:

1659 TBD

1660 **Appendix E.Revision History**

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

File name

Date

Rev	Date	By Whom	What
wd-01	2005-07-07	Christopher Ferris	Initial version created based on submission by the authors.-

File name

Date

Copyright © OASIS Open 2005. All Rights Reserved.
numbers of Statistics

Page Page

1661 **Appendix F.Notices**

1662 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
1663 that might be claimed to pertain to the implementation or use of the technology described in this
1664 document or the extent to which any license under such rights might or might not be available;
1665 neither does it represent that it has made any effort to identify any such rights. Information on
1666 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
1667 website. Copies of claims of rights made available for publication and any assurances of licenses
1668 to be made available, or the result of an attempt made to obtain a general license or permission
1669 for the use of such proprietary rights by implementors or users of this specification, can be
1670 obtained from the OASIS Executive Director.

1671 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
1672 applications, or other proprietary rights which may cover technology that may be required to
1673 implement this specification. Please address the information to the OASIS Executive Director.

1674 Copyright (C) OASIS Open (2005). All Rights Reserved.

1675 This document and translations of it may be copied and furnished to others, and derivative works
1676 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
1677 published and distributed, in whole or in part, without restriction of any kind, provided that the
1678 above copyright notice and this paragraph are included on all such copies and derivative works.
1679 However, this document itself may not be modified in any way, such as by removing the copyright
1680 notice or references to OASIS, except as needed for the purpose of developing OASIS
1681 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
1682 Property Rights document must be followed, or as required to translate it into languages other
1683 than English.

1684 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
1685 successors or assigns.

1686 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1687 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
1688 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE

File name

Date

Copyright © OASIS Open 2005. All Rights Reserved.
numbers of Statistics

Page Page

1689 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
1690 PARTICULAR PURPOSE.