



Web Services Reliable Messaging Policy Assertion

(WS-RM Policy Messaging)

(WS-Reliable Messaging)

Committee Draft ~~02, January 13, 2006~~, ~~September 26th 2005~~

Document identifier:

[wsrmp-1.1-spec-cd-02-1.1-spec-cd-01](#)

Location:

<http://docs.oasis-open.org/ws-rx/wsrmp/200510/wsrmp-1.1-spec-cd-02/200510/wsrmp-1.1-spec-cd-01.pdf>

Editors:

Gilbert Pilz, BEA <gilbert.pilz@bea.com>

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

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

[Ümit Yalçinalp, SAP](#) <umit.yalcinalp@sap.com>

TBD

Contributors:

TBD

Abstract:

This specification describes a domain-specific policy assertion for WS-ReliableMessaging [WS-RM] that that can be specified within a policy alternative as defined in WS-Policy Framework [WS-Policy](WS-ReliableMessaging) describes a protocol that allows messages to be delivered reliably between distributed applications in the presence of software component, system, or network failures. The protocol is described in this specification in a transport-independent manner allowing it to be implemented using different network technologies. To support interoperable Web services, a SOAP binding is defined within this specification.

~~By using the XML [XML], SOAP [SOAP 1.1], [SOAP 1.2] and WSDL [WSDL 1.1] extensibility models, the WS* specifications are designed to be composed with each other to provide a rich Web services environment. This by itself does not provide a negotiation solution for Web services. This is a building block that is used in conjunction with other Web service and application-specific protocols to accommodate a wide variety of policy exchange models. The protocol defined in this specification depends upon other Web services specifications for the identification of service endpoint addresses and policies. How these are identified and retrieved are detailed within those specifications and are out of scope for this document.~~

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

43 **Status:**

44 This document is a Committee Draft.

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

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

52 **Table of Contents**

53	1 Introduction.....	3
54	1.1 Goals and Requirements.....	3
55	1.1.1 Requirements.....	3
56	1.2 Notational Conventions.....	3
57	1.3 Namespace.....	3
58	1.4 Compliance.....	4
59	2 RM Policy Assertions.....	5
60	2.1 Assertion Model	5
61	2.2 Normative Outline.....	5
62	2.3 Assertion Attachment.....	6
63	2.4 Assertion Example.....	6
64	3 Security Considerations.....	8
65	4 References.....	9
66	4.1 Normative.....	9
67	4.2 Non Normative.....	9
68	A. Acknowledgments.....	10
69	B. XML Schema.....	11
70	C. Revision History.....	13
71	D. Notices.....	15

72 1 Introduction

73 ~~This specification defines a domain-specific policy assertion for reliable messaging for use with WS-Policy~~
74 ~~[WS-Policy] and WS-ReliableMessaging [WS-RM]. It is often a requirement for two Web services that wish~~
75 ~~to communicate to do so reliably in the presence of software component, system, or network failures. The~~
76 ~~primary goal of this specification is to create a modular mechanism for reliable message delivery. It~~
77 ~~defines a messaging protocol to identify, track, and manage the reliable delivery of messages between~~
78 ~~exactly two parties, a source and a destination. It also defines a SOAP binding that is required for~~
79 ~~interoperability. Additional bindings may be defined.~~

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

84 1.1 Goals and Requirements

85 1.1.1 Requirements

86 1.2 Notational Conventions

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

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

- 91 • The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- 92 • Characters are appended to elements and attributes to indicate cardinality:
 - 93 ○ "?" (0 or 1)
 - 94 ○ "*" (0 or more)
 - 95 ○ "+" (1 or more)
- 96 • The character "|" is used to indicate a choice between alternatives.
- 97 • The characters "[" and "]" are used to indicate that contained items are to be treated as a group
98 with respect to cardinality or choice.
- 99 • An ellipsis (i.e. "...") indicates a point of extensibility that allows other child, or attribute, content.
100 Additional children [elements](#) and/or attributes MAY be added at the indicated extension points but
101 MUST NOT contradict the semantics of the parent and/or owner, respectively. If an extension is
102 not recognized it SHOULD be ignored.
- 103 • XML namespace prefixes (See Section [Namespace](#)) are used to indicate the namespace of the
104 element being defined.

-

105 **1.3 Namespace**

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

107 `http://docs.oasis-open.org/ws-rx/wsrmp/200510-rm/200510/`

108 Table 1 lists [the](#) XML namespaces that are used in this specification. The choice of any namespace prefix
109 is arbitrary and not semantically significant.

110 The following namespaces are used in this document:

111 *Table 1*

Prefix	Namespace	Specification
wsp	http://schemas.xmlsoap.org/ws/2004/09/policy	[WS-Policy]
wsrmp	http://docs.oasis-open.org/ws-rx/wsrmp/200510	This specification.
Prefix	Namespace	
s	http://www.w3.org/2003/05/soap-envelope	
S11	http://schemas.xmlsoap.org/soap/envelope/	
wstrm	http://docs.oasis-open.org/wstrm/200510/	
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	
xs	http://www.w3.org/2001/XMLSchema	

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

113 `http://docs.oasis-open.org/wstrm/200510/wstrm.xsd`

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

116 ~~If an action URI is used, and one is not already defined per the rules of the WS-Addressing~~
117 ~~specification [WS-Addressing], then the action URI MUST consist of the reliable messaging~~
118 ~~namespace URI concatenated with the element name. For example:~~

119 `http://docs.oasis-open.org/wstrm/200510/SequenceAcknowledgement`

120 **1.4 Compliance**

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

125 ~~Normative text within this specification takes precedence over normative outlines, which in turn take~~
126 ~~precedence over the XML Schema [XML-Schema Part1, XML-Schema Part2] descriptionsAn~~
127 ~~implementation is not compliant with this specification if it fails to satisfy one or more of the MUST or~~
128 ~~REQUIRED level requirements defined herein. A SOAP Node MUST NOT use the XML namespace~~
129 ~~identifier for this specification (listed in SectionNamespace) within SOAP Envelopes unless it is compliant~~
130 ~~with this specification.~~

2 RM Policy Assertions

WS-Policy Framework [WS-Policy] and WS-Policy Attachment [WS-PolicyAttachment] collectively define a framework, model and grammar for expressing the requirements, and general characteristics of entities in an XML Web services-based system. To enable an RM Destination and an RM Source to describe their requirements for a given Sequence, this specification defines a single RM policy assertion that leverages the WS-Policy framework.

2.1 Assertion Model

The RM policy assertion indicates that the RM Source and RM Destination MUST use WS-ReliableMessaging [WS-RM] to ensure reliable delivery of messages. Specifically, the WS-ReliableMessaging protocol determines invariants maintained by the reliable messaging endpoints and the directives used to track and manage the delivery of a Sequence of messages.

The assertion defines a maximum message number parameter that the RM Destination MAY include to indicate the maximum message number the RM Destination will accept. This is useful for RM Destinations that may be running in constrained environments that can not accept values as large as the default value of a maximum unsigned long.

Finally, this assertion defines an acknowledgement interval parameter that the RM Destination MAY include. Per WS-ReliableMessaging [WS-RM], acknowledgements are sent on return messages or sent stand-alone. If a return message is not available to send an acknowledgement, an RM Destination MAY wait for up to the acknowledgement interval before sending a stand-alone acknowledgement. If there are no unacknowledged messages, the RM Destination MAY choose not to send an acknowledgement. This parameter does not alter the formulation of messages or acknowledgements as transmitted; it does not alter the meaning of the wsrmp:AckRequested directive. Its purpose is to communicate the timing of acknowledgements so that the RM Source may tune appropriately.

The RM assertion parameters do not affect the messages which are sent on the wire.

2.2 Normative Outline

The normative outline for the RM version assertion is:

```
<wsrmp:RMAssertion [wsp:Optional="true"]? ... >
  <wsrmp:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?
  <wsrmp:MaxMessageNumber Number="xs:unsignedLong" ... /> ?
  ...
</wsrmp:RMAssertion>
```

The following describes additional, normative constraints on the outline listed above:

/wsrmp:RMAssertion

A policy assertion that specifies that WS-ReliableMessaging [WS-RM] protocol MUST be used for a Sequence.

/wsrmp:RMAssertion/@wsp:Optional="true"

Per WS-Policy [WS-Policy], this is compact notation for two policy alternatives, one with and one without the assertion. The intuition is that the behavior indicated by the assertion is optional, or in this case, that WS-ReliableMessaging MAY be used.

/wsrmp:RMAssertion/wsrmp:AcknowledgementInterval

171 [A parameter that specifies the duration after which the RM Destination will transmit an](#)
172 [acknowledgement. If omitted, there is no implied value.](#)

173 [/wsrmp:RMAssertion/wsrmp:AcknowledgementInterval/@Milliseconds](#)
174 [The acknowledgement interval, specified in milliseconds.](#)

175 [/wsrmp:RMAssertion/wsrmp:MaxMessageNumber](#)
176 [A parameter that specifies the maximum message number that the RM Destination will accept. If](#)
177 [omitted, the default value of the maximum unsigned long will be assumed.](#)

178 [/wsrmp:RMAssertion/wsrmp:MaxMessageNumber/@Number](#)
179 [The maximum message number.](#)

180 **2.3 Assertion Attachment**

181 [Because the RM policy assertion indicates endpoint behavior over an RM Sequence, the assertion has](#)
182 [Endpoint Policy Subject \[WS-PolicyAttachment\].](#)

183 [WS-PolicyAttachment defines three WSDL \[WSDL 1.1\] policy attachment points with Endpoint Policy](#)
184 [Subject:](#)

- 185 • [wsdl:portType](#) – [A policy expression containing the RM policy assertion MUST NOT be attached to](#)
186 [a wsdl:portType; the RM policy assertion specifies a concrete behavior whereas the wsdl:portType is an](#)
187 [abstract construct.](#)
- 188 • [wsdl:binding](#) – [A policy expression containing the RM policy assertion SHOULD be attached to a](#)
189 [wsdl:binding.](#)
- 190 • [wsdl:port](#) – [A policy expression containing the RM policy assertion MAY be attached to a wsdl:port.](#)

191 [If the RM policy assertion appears in a policy expression attached to both a wsdl:port and its](#)
192 [corresponding wsdl:binding, the parameters in the former MUST be used and the latter ignored.](#)

193 **2.4 Assertion Example**

194 [Table 2 lists an example use of the RM policy assertion.](#)

195 [Table 2: Example policy with RM policy assertion](#)

```
196 (01)<wsdl:definitions  
197 (02)   targetNamespace="example.com"  
198 (03)   xmlns:tns="example.com"  
199 (04)   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"  
200 (05)   xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"  
201 (06)   xmlns:wsrmp="http://docs.oasis-open.org/ws-rx/wsrmp/200510"  
202 (07)   xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-  
203 wss-wssecurity-utility-1.0.xsd">  
204 (08)  
205 (09) <wsp:UsingPolicy wsdl:required="true" />  
206 (10)  
207 (11) <wsp:Policy wsu:Id="MyPolicy" >  
208 (12)   <wsrmp:RMAssertion>  
209 (13)     <wsrmp:AcknowledgementInterval Milliseconds="200" />  
210 (14)   </wsrmp:RMAssertion>  
211 (15)   <!-- omitted assertions -->  
212 (16) </wsp:Policy>  
213 (17)  
214 (18) <!-- omitted elements -->
```

```
215 (19)  
216 (20) <wsdl:binding name="MyBinding" type="tns:MyPortType" >  
217 (21) <wsp:PolicyReference URI="#MyPolicy" />  
218 (22) <!-- omitted elements -->  
219 (23) </wsdl:binding>  
220 (24)  
221 (25)</wsdl:definitions>
```

222 Line (09) in Table 2 indicates that WS-Policy [WS-Policy] is in use as a required extension.

223 Lines (11-16) are a policy expression that includes a RM policy assertion (Lines 12-14) to indicate that
224 WS-ReliableMessaging [WS-RM] must be used. Line (13) indicates the RM Destination may buffer
225 acknowledgements for up to two-tenths of a second.

226 Lines (20-23) are a WSDL [WSDL 1.1] binding. Line (21) indicates that the policy in Lines (11-16) applies
227 to this binding, specifically indicating that WS-ReliableMessaging must be used over all the messages in
228 the binding.

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

231 **3 Reliable Messaging Model**

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

234 —

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

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

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

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

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

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

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

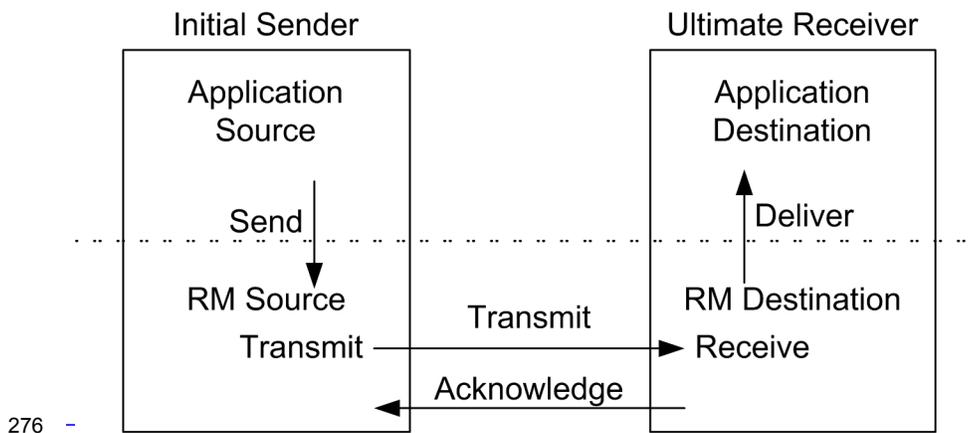
259 **AtLeastOnce** Every message sent will be delivered or an error will be raised on at least one
260 endpoint. Some messages may be delivered more than once.

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

264 **InOrder** Messages will be delivered in the order that they were sent. This delivery
265 assurance may be combined with any of the above delivery assurances. It requires that the
266 messages within a Sequence will be delivered in an order so that the message numbers are
267 monotonically increasing. Note that this assurance says nothing about duplications or
268 omissions. Note also that it is only applicable to messages in the same Sequence. Cross-
269 Sequence ordering of messages is not in the scope of this specification.

270 Figure 1 below illustrates the entities and events in a simple reliable message exchange.
271 First, the Application Source Sends a message for reliable delivery. The Reliable Messaging-
272 (RM) Source accepts the message and Transmits it one or more times. After receiving the

273 message, the RM Destination Acknowledges it. Finally, the RM Destination delivers the
 274 message to the Application Destination. The exact roles the entities play and the complete
 275 meaning of the events will be defined throughout this specification.



276 -

277 Figure 1: Reliable Messaging Model

278 3.1 Glossary

279 The following definitions are used throughout this specification:

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

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

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

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

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

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

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

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

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

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

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

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

298 **3.2 Protocol Preconditions**

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

- 301 • The RM Source **MUST** have an endpoint reference that uniquely identifies the RM Destination-
302 endpoint; correlations across messages addressed to the unique endpoint **MUST** be meaningful.
- 303 • The RM Source **MUST** have knowledge of the destination's policies, if any, and the RM Source **MUST**
304 be capable of formulating messages that adhere to this policy.

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

307 **3.3 Protocol Invariants**

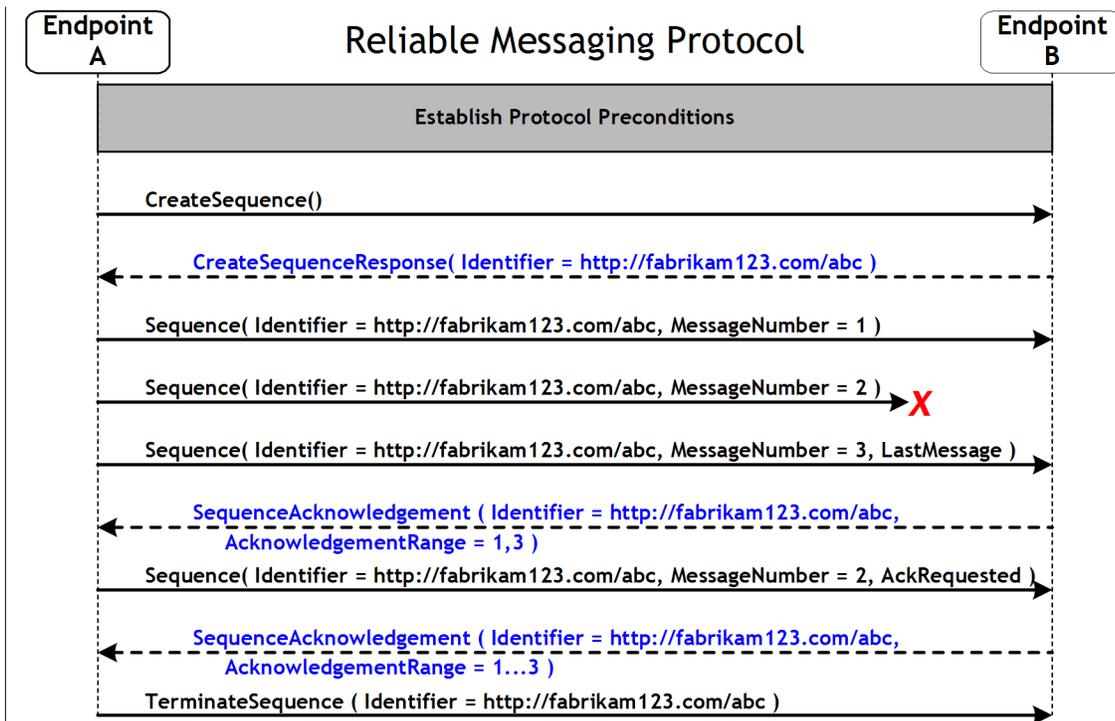
308 During the lifetime of the protocol, two invariants are **REQUIRED** for correctness:

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

311 Every acknowledgement issued by the RM Destination **MUST** include within an
312 acknowledgement range or ranges the sequence number of every message successfully
313 received by the RM Destination and **MUST** exclude sequence numbers of any messages not
314 yet received.

315 **3.4 Example Message Exchange**

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



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

- 319 1. ~~The protocol preconditions are established. These include policy exchange, endpoint-~~
320 ~~resolution, establishing trust.~~
 - 321 2. ~~The RM Source requests creation of a new Sequence.~~
 - 322 3. ~~The RM Destination creates a Sequence by returning a globally unique identifier.~~
 - 323 4. ~~The RM Source begins sending messages beginning with MessageNumber 1. In the~~
324 ~~figure the RM Source sends 3 messages.~~
 - 325 5. ~~Since the 3rd message is the last in this exchange, the RM Source includes a~~
326 ~~<wsrm:LastMessage> token.~~
 - 327 6. ~~The 2nd message is lost in transit.~~
 - 328 7. ~~The RM Destination acknowledges receipt of message numbers 1 and 3 in response to~~
329 ~~the RM Source's <wsrm:LastMessage> token.~~
 - 330 8. ~~The RM Source retransmits the 2nd message. This is a new message on the underlying-~~
331 ~~transport, but since it has the same sequence identifier and message number so the RM-~~
332 ~~Destination can recognize it as equivalent to the earlier message, in case both are~~
333 ~~received.~~
 - 334 9. ~~The RM Source includes an <wsrm:AckRequested> element so the RM Destination will~~
335 ~~expedite an acknowledgement.~~
 - 336 10. ~~The RM Destination receives the second transmission of the message with~~
337 ~~MessageNumber 2 and acknowledges receipt of message numbers 1, 2, and 3 which~~
338 ~~carried the <wsrm:LastMessage> token.~~
 - 339 11. ~~The RM Source receives this acknowledgement and sends a TerminateSequence message-~~
340 ~~to the RM Destination indicating that the sequence is completed and reclaims any~~
341 ~~resources associated with the Sequence.~~
 - 342 12. ~~The RM Destination receives the TerminateSequence message indicating that the RM-~~
343 ~~Source will not be sending any more messages, and reclaims any resources associated~~
344 ~~with the Sequence.~~
- 345 ~~Now that the basic model has been outlined, the details of the elements used in this~~
346 ~~protocol are now provided in Section 3.~~

4 RM Protocol Elements

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

4.1 Sequences

The RM protocol uses a `<wsrm:Sequence>` header block to track and manage the reliable delivery of messages. Messages for which the delivery assurance applies MUST contain a `<wsrm:Sequence>` header block. Each Sequence MUST have a unique `<wsrm:Identifier>` element and each message within a Sequence MUST have a `<wsrm:MessageNumber>` element that increments by 1 from an initial value of 1. These values are contained within a `<wsrm:Sequence>` header block accompanying each message being delivered in the context of a Sequence. In addition to mandatory `<wsrm:Identifier>` and `<wsrm:MessageNumber>` elements, the header MAY include a `<wsrm:LastMessage>` element.

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

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

A following exemplar defines its syntax:

```
<wsrm:Sequence ...>  
  <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
  <wsrm:MessageNumber> xs:unsignedLong </wsrm:MessageNumber>  
  <wsrm:LastMessage/?>  
  ...  
</wsrm:Sequence>
```

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

`/wsrm:Sequence`

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

`/wsrm:Sequence/wsrm:Identifier`

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

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

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

`/wsrm:Sequence/wsrm:MessageNumber`

This REQUIRED element MUST contain an `xs:unsignedLong` representing the ordinal position of the message within a Sequence. Sequence MessageNumbers start at 1 and monotonically increase throughout the Sequence. If the message number exceeds the internal limitations of an RM Source or

387 RM Destination or reaches the maximum value of an xs:unsignedLong (18,446,744,073,709,551,615), the
388 RM Source or Destination MUST issue a MessageNumberRollover fault.

389 /wsmr:Sequence/wsmr:LastMessage

390 This element MAY be included by the RM Source endpoint. The <wsmr:LastMessage> element has no
391 content.

392 /wsmr:Sequence/{any}

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

395 /wsmr:Sequence/@{any}

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

398 A RM Source endpoint MUST include a <wsmr:LastMessage> element in the
399 <wsmr:Sequence> element for the last message in a Sequence. An RM Destination endpoint
400 MUST respond with a <wsmr:SequenceAcknowledgement> upon receipt of a
401 <wsmr:LastMessage> element. A Sequence MUST NOT use a <wsmr:MessageNumber> value
402 greater than that which accompanies a <wsmr:LastMessage> element. An RM Destination
403 MUST generate a LastMessageNumberExceeded (See Section 5.6) fault upon receipt of such
404 a message. In the event that an RM Source needs to close a Sequence and there is no
405 application message, the RM Source MAY send a message with an empty body containing
406 <wsmr:Sequence> header with the <wsmr:LastMessage> element. In this usage, the action-
407 URI MUST be:

```
http://docs.oasis-open.org/wsmr/200510/LastMessage
```

409 in preference to the pattern defined in Section 1.2.

410 The following example illustrates a Sequence header block.

```
411 <wsmr:Sequence>  
412   <wsmr:Identifier>http://example.com/abc</wsmr:Identifier>  
413   <wsmr:MessageNumber>10</wsmr:MessageNumber>  
414   <wsmr:LastMessage/>  
415 </wsmr:Sequence>
```

416 4.2 Sequence Acknowledgement

417 The RM Destination informs the RM Source of successful message receipt using a
418 <wsmr:SequenceAcknowledgement> header block. The <wsmr:SequenceAcknowledgement>
419 header block MAY be transmitted independently or included on return messages. The RM
420 Destination MAY send a <wsmr:SequenceAcknowledgement> header block at any point
421 during which the sequence is valid. The timing of acknowledgements can be advertised
422 using policy and acknowledgements can be explicitly requested using the
423 <wsmr:AckRequested> directive (see Section 4.3). If a non-mustUnderstand fault occurs
424 when processing an RM Header that was piggy-backed on another message, a fault MUST
425 be generated, but the processing of the original message MUST NOT be affected.

426 The following exemplar defines its syntax:

```
427 <wsmr:SequenceAcknowledgement ...>  
428   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
429   [ [ <wsmr:AcknowledgementRange ...>
```

```

430         Upper="xs:unsignedLong"
431         Lower="xs:unsignedLong"/>+
432     <wsrm:Final/> ?- }
433     | <wsrm:Nack> xs:unsignedLong </wsrm:Nack> +
434     | <wsrm:None/> }
435     ...
436 </wsrm:SequenceAcknowledgement>

```

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

438
439 `/wsrm:SequenceAcknowledgement`

440 This element contains the Sequence acknowledgement information.

441 `/wsrm:SequenceAcknowledgement/wsrm:Identifier`

442 This REQUIRED element MUST contain an absolute URI conformant with RFC2396 that uniquely identifies the Sequence.

443
444 `/wsrm:SequenceAcknowledgement/wsrm:Identifier/@{any}`

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

446
447 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

448 This OPTIONAL element, if present, can occur 1 or more times. It contains a range of message Sequence MessageNumbers successfully received by the receiving endpoint manager. The ranges SHOULD NOT overlap. This element MUST NOT be present if either the `<wsrm:Nack>` or `<wsrm:None>` elements are also present as a child of `<wsrm:SequenceAcknowledgement>`.

449
450
451
452 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Upper`

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

454
455 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@Lower`

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

457
458 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange/@{any}`

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

460
461 `/wsrm:SequenceAcknowledgement/wsrm:Final` —

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

463
464
465
466
467
468 `/wsrm:SequenceAcknowledgement/wsrm:Nack`

469 This OPTIONAL element, if present, MUST contain an `xs:unsignedLong` representing the `<wsrm:MessageNumber>` of an unreceived message in a Sequence. This element permits the gap analysis of the `<wsrm:AcknowledgementRange>` elements to be performed at the RM Destination.

472 rather than at the RM Source which may yield performance benefits in certain environments. The
473 ~~<wsrm:Nack>~~ element MUST NOT be present if either the ~~<wsrm:AcknowledgementRange>~~ or
474 ~~<wsrm:None>~~ elements are also present as a child of ~~<wsrm:SequenceAcknowledgement>~~. Upon the
475 receipt of a Nack, an RM Source SHOULD retransmit the message identified by the Nack. The RM
476 Destination MUST NOT issue a ~~<wsrm:SequenceAcknowledgement>~~ containing a ~~<wsrm:Nack>~~ for
477 a message that it has previously acknowledged within a ~~<wsrm:AcknowledgementRange>~~. The RM
478 Source SHOULD ignore a ~~<wsrm:SequenceAcknowledgement>~~ containing a ~~<wsrm:Nack>~~ for a
479 message that has previously been acknowledged within a ~~<wsrm:AcknowledgementRange>~~.

480 ~~/wsrm:SequenceAcknowledgement/wsrm:None~~

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

485 ~~/wsrm:SequenceAcknowledgement/{any}~~

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

488 ~~/wsrm:SequenceAcknowledgement/@{any}~~

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

491 The following examples illustrate ~~<wsrm:SequenceAcknowledgement>~~ elements:

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

```
493 <wsrm:SequenceAcknowledgement>  
494 <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
495 <wsrm:AcknowledgementRange Upper="10" Lower="1"/>  
496 </wsrm:SequenceAcknowledgement>
```

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

```
499 <wsrm:SequenceAcknowledgement>  
500 <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
501 <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
502 <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
503 <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
504 </wsrm:SequenceAcknowledgement>
```

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

```
506 <wsrm:SequenceAcknowledgement>  
507 <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
508 <wsrm:Nack>3</wsrm:Nack>  
509 </wsrm:SequenceAcknowledgement>
```

510 4.3 Request Acknowledgement

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

513 At any time, the RM Source may request an acknowledgement message from the RM
514 Destination endpoint using an ~~<wsrm:AckRequested>~~ header block.

515 The RM Source endpoint requests this acknowledgement by including an
516 `<wsrm:AckRequested>` header block in the message. An RM Destination that receives a
517 message that contains an `<wsrm:AckRequested>` header block MUST respond with a
518 message containing a `<wsrm:SequenceAcknowledgement>` header block. If a non-
519 mustUnderstand fault occurs when processing an RM Header that was piggy-backed on
520 another message, a fault MUST be generated, but the processing of the original message
521 MUST NOT be affected.

522 The following exemplar defines its syntax:

```
523 <wsrm:AckRequested ...>  
524   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
525   <wsrm:MessageNumber> xs:unsignedLong </wsrm:MessageNumber> ?  
526   ...  
527 </wsrm:AckRequested>
```

528 `/wsrm:AckRequested`

529 This element requests an acknowledgement for the identified sequence.

530 `/wsrm:AckRequested/wsrm:Identifier`

531 This REQUIRED element MUST contain an absolute URI, conformant with RFC2396, that uniquely
532 identifies the Sequence to which the request applies.

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

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

536 `/wsrm:AckRequested/wsrm:MessageNumber`

537 This OPTIONAL element, if present, MUST contain an `xs:unsignedLong` representing the highest
538 `<wsrm:MessageNumber>` sent by the RM Source within the Sequence. If present, it MAY be treated as a
539 hint to the RM Destination as an optimization to the process of preparing to transmit a

540 `<wsrm:SequenceAcknowledgement>`.

541 `/wsrm:AckRequested/{any}`

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

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

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

547 4.4 Sequence Creation

548 The RM Source MUST request creation of an outbound Sequence by sending a
549 `<wsrm:CreateSequence>` element in the body of a message to the RM Destination which in-
550 turn responds either with a `<wsrm:CreateSequenceResponse>` or a `CreateSequenceRefused`
551 fault in the body of the response message. `<wsrm:CreateSequence>` MAY carry an offer to
552 create an inbound sequence which is either accepted or rejected in the
553 `<wsrm:CreateSequenceResponse>`.

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

557 The following exemplar defines the `<wsmr:CreateSequence>` syntax:

```
558 <wsmr:CreateSequence ...>  
559   <wsmr:AcksTo ...> wsa:EndpointReferenceType </wsmr:AcksTo>  
560   <wsmr:Expires ...> xs:duration </wsmr:Expires> ?  
561   <wsmr:Offer ...>  
562     <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
563     <wsmr:Expires ...> xs:duration </wsmr:Expires> ?  
564     ...  
565   </wsmr:Offer> ?  
566   ...  
567 </wsmr:CreateSequence>
```

568 `/wsmr:CreateSequence`

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

573 `/wsmr:CreateSequence/wsmr:AcksTo`

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

577 `/wsmr:CreateSequence/wsmr:Expires`

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

582 `/wsmr:CreateSequence/wsmr:Expires/@{any}`

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

585 `/wsmr:CreateSequence/wsmr:Offer`

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

588 `/wsmr:CreateSequence/wsmr:Offer/wsmr:Identifier`

589 This REQUIRED element MUST contain an absolute URI conformant with RFC2396 that uniquely
590 identifies the offered Sequence.

591 `/wsmr:CreateSequence/wsmr:Offer/wsmr:Identifier/@{any}`

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

594 `/wsmr:CreateSequence/wsmr:Offer/wsmr:Expires`

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

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

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

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

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

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

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

607 `OPTIONAL/wsrm:CreateSequence/{any}`

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

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

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

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

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

```
618 <wsrm:CreateSequenceResponse ...>  
619   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
620   <wsrm:Expires> xs:duration </wsrm:Expires> ?  
621   <wsrm:Accept ...>  
622     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>  
623     ...  
624   </wsrm:Accept> ?  
625   ...  
626 </wsrm:CreateSequenceResponse>
```

627 `/wsrm:CreateSequenceResponse`

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

631 `/wsrm:CreateSequenceResponse/wsrm:Identifier`

632 This REQUIRED element MUST contain an absolute URI conformant with RFC2396 of the Sequence that
633 has been created by the RM Destination.

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

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

637 ~~/wsmr:CreateSequenceResponse/wsmr:Expires~~

638 ~~This element, if present, of type *xs:duration* accepts or refines the RM Source's requested duration for~~
639 ~~the Sequence. A value of 'PT0S' indicates that the Sequence will never expire. Absence of the element~~
640 ~~indicates an implied value of 'PT0S'. This value MUST be equal or lesser than the value requested by the~~
641 ~~RM Source in the corresponding *<wsmr:CreateSequence>* message.~~

642 ~~/wsmr:CreateSequenceResponse/wsmr:Expires/@{any}~~

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

645 ~~/wsmr:CreateSequenceResponse/wsmr:Accept~~

646 ~~This element, if present, enables an RM Destination to accept the offer of a corresponding Sequence for~~
647 ~~the reliable exchange of messages transmitted from RM Destination to RM Source. This element MUST~~
648 ~~be present if the corresponding *<wsmr:CreateSequence>* message contained an *<wsmr:Offer>*~~
649 ~~element.~~

650 ~~/wsmr:CreateSequenceResponse/wsmr:Accept/wsmr:AcksTo~~

651 ~~This REQUIRED element, of type *wsa:EndpointReferenceType* as specified by WS-Addressing [WS-~~
652 ~~Addressing], specifies the endpoint reference to which *<wsmr:SequenceAcknowledgement>*~~
653 ~~messages related to the accepted Sequence are to be sent.~~

654 ~~/wsmr:CreateSequenceResponse/wsmr:Accept/{any}~~

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

657 ~~/wsmr:CreateSequenceResponse/wsmr:Accept/@{any}~~

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

660 ~~/wsmr:CreateSequenceResponse/{any}~~

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

663 ~~/wsmr:CreateSequenceResponse/@{any}~~

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

666 **4.5 Sequence Termination**

667 ~~When the RM Source has completed its use of the Sequence, it sends a~~
668 ~~*<wsmr:TerminateSequence>* element, in the body of a message to the RM Destination to~~
669 ~~indicate that the Sequence is complete, and that it will not be sending any further messages~~
670 ~~related to the Sequence. The RM Destination can safely reclaim any resources associated~~
671 ~~with the Sequence upon receipt of the *<wsmr:TerminateSequence>* message. Note, under~~
672 ~~normal usage the RM source will complete its use of the sequence when all of the messages~~
673 ~~in the Sequence have been acknowledged. However, the RM Source is free to Terminate or~~
674 ~~Close a Sequence at any time regardless of the acknowledgement state of the messages.~~

675 ~~The following exemplar defines the *TerminateSequence* syntax:~~

```

676 <wsm:TerminateSequence ...>
677   <wsm:Identifier ...> xs:anyURI </wsm:Identifier>
678   ...
679 </wsm:TerminateSequence>

```

680 /wsm:TerminateSequence

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

685 /wsm:TerminateSequence/wsm:Identifier

686 This REQUIRED element MUST contain an absolute URI conformant with RFC2396 of the Sequence that
687 is being terminated.

688 /wsm:TerminateSequence/wsm:Identifier/@{any}

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

691 /wsm:TerminateSequence/{any}

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

694 /wsm:TerminateSequence/@{any}

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

697 4.6 Closing A Sequence

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

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

706 To alleviate this, the RM Source can send a <wsm:CloseSequence> element, in the body of a
707 message, to the RM Destination to indicate that RM Destination MUST NOT receive any
708 new messages for the specified sequence, other than those already received at the time the
709 <wsm:CloseSequence> element is interpreted by the RMD. Upon receipt of this message
710 the RM Destination MUST send a SequenceAcknowledgement to the RM Source. Note, this
711 SequenceAcknowledgement MUST include the <wsm:Final> element.

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

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

720 The following exemplar defines the CloseSequence syntax:

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

722 /wsrm:CloseSequence—

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

726 /wsrm:CloseSequence@Identifier

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

729 /wsrm:CloseSequence/{any}

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

732 /wsrm:CloseSequence@{any}

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

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

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

```
739 </wsrm:CloseSequenceResponse
```

740 /wsrm:CloseSequenceResponse

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

743 /wsrm:CloseSequenceResponse/{any}—

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

746 /wsrm:CloseSequenceResponse@{any}—

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

749 5 Faults

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

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

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

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

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

769 The definitions of faults use the following properties:-

770 [Code] The fault code.

771 [Subcode] The fault subcode.

772 [Reason] The English language reason element.

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

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

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

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

```
777 <S:Envelope>  
778 <S:Header>  
779 <wsa:Action>  
780 http://schemas.xmlsoap.org/ws/2004/08/addressing/fault  
781 </wsa:Action>  
782 <!-- Headers elided for clarity. -->  
783 </S:Header>  
784 <S:Body>  
785 <S:Fault>  
786 <S:Code>
```

```

787 </S:Value> [Code] </S:Value>
788 <S:Subcode>
789 <S:Value> [Subcode] </S:Value>
790 </S:Subcode>
791 </S:Code>
792 <S:Reason>
793 <S:Text xml:lang="en"> [Reason] </S:Text>
794 </S:Reason>
795 <S:Detail>
796 [Detail]
797 ...
798 </S:Detail>
799 </S:Fault>
800 </S:Body>
801 </S:Envelope>

```

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

```

804 <S11:Envelope>
805 <S11:Header>
806 <wsm:SequenceFault>
807 <wsm:FaultCode> wsm:FaultCodes </wsm:FaultCode>
808 ...
809 </wsm:SequenceFault>
810 <!-- Headers elided for clarity. -->
811 </S11:Header>
812 <S11:Body>
813 <S11:Fault>
814 <faultcode> [Code] </faultcode>
815 <faultstring> [Reason] </faultstring>
816 </S11:Fault>
817 </S11:Body>
818 </S11:Envelope>

```

819 The properties bind to a SOAP 1.1 fault as follows when the fault is generated as a result of
820 processing a <wsm:CreateSequence> request message:

```

821 <S11:Envelope>
822 <S11:Body>
823 <S11:Fault>
824 <faultcode> [Subcode] </faultcode>
825 <faultstring xml:lang="en"> [Reason] </faultstring>
826 </S11:Fault>
827 </S11:Body>
828 </S11:Envelope>

```

829 5.1 SequenceFault Element

830 The purpose of the <wsm:SequenceFault> element is to carry the specific details of a fault-
831 generated during the reliable messaging specific processing of a message belonging to a
832 Sequence. The <wsm:SequenceFault> container MUST only be used in conjunction with
833 the SOAP1.1 fault mechanism. It MUST NOT be used in conjunction with the SOAP1.2-
834 binding.

835 The following exemplar defines its syntax:

```

836 <wsm:SequenceFault ...>
837 <wsm:FaultCode> wsm:FaultCodes </wsm:FaultCode>

```

838
839

```
...  
</wsm:SequenceFault>
```

840 The following describes the content model of the ~~SequenceFault~~ element.

841 ~~/wsm:SequenceFault~~

842 This is the element containing ~~Sequence~~ information for ~~WS-ReliableMessaging~~

843 ~~/wsm:SequenceFault/wsm:FaultCode~~

844 This element, if present, ~~MUST~~ contain a qualified name from the set of fault codes defined below.

845 ~~/wsm:SequenceFault/{any}~~

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

848 ~~/wsm:SequenceFault/@{any}~~

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

851 **5.2 Sequence Terminated**

852 This fault is sent by either the ~~RM Source~~ or the ~~RM Destination~~ to indicate that the endpoint
853 that generated the fault has either encountered an unrecoverable condition, or has detected
854 a violation of the protocol and as a consequence, has chosen to terminate the sequence.
855 The endpoint that generates this fault should make every reasonable effort to notify the
856 corresponding endpoint of this decision.

857 Properties:

858 [Code] ~~Sender or Receiver~~

859 [Subcode] ~~wsm:SequenceTerminated~~

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

861 [Detail]

862

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

863 **5.3 Unknown Sequence**

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

866 Properties:

867 [Code] ~~Sender~~

868 [Subcode] ~~wsm:UnknownSequence~~

869 [Reason] ~~The value of wsm:Identifier is not a known Sequence identifier.~~

870 [Detail]

871

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

872 **5.4 Invalid Acknowledgement**

873 This fault is sent by the RM Source in response to a `<wsrm:SequenceAcknowledgement>` that
874 violates the cumulative acknowledgement invariant. An example of such a violation would
875 be a `SequenceAcknowledgement` covering messages that have not been sent.

876 [Code] Sender

877 [Subcode] `wsrm:InvalidAcknowledgement`

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

880 [Detail]

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

882 **5.5 Message Number Rollover**

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

884 Properties:

885 [Code] Sender

886 [Subcode] `wsrm:MessageNumberRollover`

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

888 [Detail]

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

890 **5.6 Last Message Number Exceeded**

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

895 Properties:

896 [Code] Sender

897 [Subcode] `wsrm:LastMessageNumberExceeded`

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

900 [Detail]

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

902 **5.7 Create Sequence Refused**

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

904 **Properties:**
905 **[Code]** Sender
906 **[Subcode]** wsrn:CreateSequenceRefused
907 **[Reason]** The create sequence request has been refused by the RM Destination.
908 **[Detail]** empty

909 **5.8 Sequence Closed**

910 This fault is sent by an RM Destination to indicate that the specified sequence has been
911 closed. This fault **MUST** be generated when an RM Destination is asked to receive a message
912 for a sequence that is closed.

913 **Properties:**
914 **[Code]** Sender
915 **[Subcode]** wsrn:SequenceClosed
916 **[Reason]** The sequence is closed and can not receive new messages.
917 **[Detail]** <wsrn:Identifier...> xs:anyURI </wsrn:Identifier>

6 Security Considerations

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

It is RECOMMENDED that policies SHOULD NOT be accepted unless they are signed and have an associated security token to specify the signer has proper claims for the given policy. That is, a relying party shouldn't rely on a policy unless the policy is signed and presented with sufficient claims to pass the relying parties acceptance criteria. Because Sequences are expected to exchange a number of messages, it is recommended that a security context be established using the mechanisms described in WS-Trust and WS-SecureConversation [SecureConversation]. If a Sequence is bound to a specific endpoint, then the security context needs to be established or shared with the endpoint servicing the Sequence. While the context can be established at any time, it is critical that the messages establishing the Sequence be secured even if they precede security context establishment. However, it is recommended that the security context be established first. Security contexts are independent of reliable messaging Sequences. Consequently, security contexts can come and go independent of the lifetime of the Sequence. In fact, it is recommended that the lifetime of a security context be less than the lifetime of the Sequence unless the Sequence is very short-lived.

It should be noted that the mechanisms described in this document could be secured as part of a SOAP message using WS-Security [WSS] or embedded within other objects using object-specific security mechanisms. It is common for message Sequences to exchange a number of messages (or a large amount of data). As a result, the usage profile of a Sequence is such that it is susceptible to key attacks. For this reason it is strongly recommended that the keys be changed frequently. This "re-keying" can be effected a number of ways. The following list outlines four common techniques:

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

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

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

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

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

966 • **Message disclosure**—Confidentiality is preserved by encrypting sensitive data using WS-Security.

967 • **Key integrity**—Key integrity is maintained by using the strongest algorithms possible (by comparing
968 secured policies—see WS-Policy and WS-SecurityPolicy).

969 • **Authentication**—Authentication is established using the mechanisms described in WS-Security and
970 WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.

971 • **Accountability**—Accountability is a function of the type of and string of the key and algorithms being
972 used. In many cases, a strong symmetric key provides sufficient accountability. However, in some
973 environments, strong PKI signatures are required.

974 • **Availability**—All reliable messaging services are subject to a variety of availability attacks. Replay
975 detection is a common attack and it is recommended that this be addressed by the mechanisms
976 described in WS-Security. (Note that because of legitimate message replays, detection should
977 include a differentiator besides message id such as a timestamp). Other attacks, such as network-
978 level denial of service attacks are harder to avoid and are outside the scope of this specification. That
979 said, care should be taken to ensure that minimal state is saved prior to any authenticating
980 sequences.

981 7 References

982 7.1 Normative

983 ~~[KEYWORDS]~~

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

986 ~~[SOAP 1.1]~~

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

988 ~~[SOAP 1.2]URI~~

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

992 ~~[URI]XML-ns~~

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

995 ~~[WS-RM]~~

996 ~~OASIS WS-RX Technical Committee Draft, "Web Services Reliable Messaging (WS-ReliableMessaging),"
997 September 2005.~~

998 ~~W3C Recommendation, "Namespaces in XML," 14 January 1999.~~

999 ~~[XML-Schema1]~~

1000 ~~W3C Recommendation, "XML Schema Part 1: Structures," 2 May 2001.~~

1001 ~~[XML-Schema2]~~

1002 ~~W3C Recommendation, "XML Schema Part 2: Datatypes," 2 May 2001.~~

1003 ~~[WSSecurity]~~

1004 ~~"OASIS Web Services Security: SOAP Message Security 1.0 (WS-Security 2004),"~~
1005 ~~Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds, OASIS~~
1006 ~~Standard 200401, March 2004.~~

1007 ~~[Tanenbaum]~~

1008 ~~"Computer Networks," Andrew S. Tanenbaum, Prentice Hall PTR, 2003.~~

1009 ~~[WSDL]~~

1010 ~~W3C Note, "Web Services Description Language (WSDL 1.1)," 15 March 2001.~~

1011 ~~[WS-Addressing]~~

1012 ~~D. Box, et al, "Web Services Addressing (WS-Addressing)," August 2004.~~

1013 **7.2 Non-Normative**

1014 **[WS-Policy]**

1015 D. Box, et al, "Web Services Policy Framework (WS-Policy)," September 2004.

1016 ~~**[WS-PolicyAttachment]**~~

1017 D. Box, et al, "Web Services Policy Attachment (WS-PolicyAttachment)," September 2004.

1018 ~~**[WSDL 1.1SecurityPolicy]**~~

1019 ~~[W3C Note, "Web Services Description Language \(WSDL 1.1\)," 15 March 2001](#)~~~~G. Della-Libra, "Web-~~
1020 ~~[Services Security Policy Language \(WS-SecurityPolicy\)," December 2002.](#)~~

1021 ~~**[XML-[SecureConversation](#)]**~~

1022 ~~[W3C Recommendation, "Extensible Markup Language \(XML\) 1.0 \(Second Edition\)", October 2000.](#)~~

1023 ~~**[XML-ns]**~~

1024 ~~[W3C Recommendation, "Namespaces in XML," 14 January 1999.](#)~~

1025 ~~**[XML-Schema Part1]**~~

1026 ~~[W3C Recommendation, "XML Schema Part 1: Structures," 2 May 2001.](#)~~

1027 ~~**[XML-Schema Part2]**~~

1028 ~~[W3C Recommendation, "XML Schema Part 2: Datatypes," 2 May 2001.](#)~~

1029 **7.3 Non Normative**

1030 ~~**[WSS]**~~

1031 ~~[OASIS Web Services Security: SOAP Message Security 1.0 \(WS-Security 2004\)", Chris Kaler, Phillip](#)~~
1032 ~~[Hallam-Baker, Ronald Monzillo, eds, OASIS Standard 200401, March 2004.](#)~~

1033 ~~[S. Anderson, et al, "Web Services Secure Conversation Language \(WS-SecureConversation\)," May 2004.](#)~~

1034 -

1035 Appendix A. Schema

1036 The normative schema for WS-ReliableMessaging is located at:

1037 <http://docs.oasis-open.org/wsrmp/200510/wsrmp.xsd>

1038 The following copy is provided for reference.

```
1039 <xs:schema targetNamespace="http://docs.oasis-open.org/wsrmp/200510/"
1040 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1041 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1042 xmlns:wsrm="http://docs.oasis-open.org/wsrmp/200510/"
1043 elementFormDefault="qualified" attributeFormDefault="unqualified">
1044 <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1045 schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
1046 <!-- Protocol Elements -->
1047 <xs:complexType name="SequenceType">
1048 <xs:sequence>
1049 <xs:element ref="wsrm:Identifier"/>
1050 <xs:element name="MessageNumber" type="xs:unsignedLong"/>
1051 <xs:element name="LastMessage" minOccurs="0">
1052 <xs:complexType>
1053 <xs:sequence/>
1054 </xs:complexType>
1055 </xs:element>
1056 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1057 maxOccurs="unbounded"/>
1058 </xs:sequence>
1059 <xs:anyAttribute namespace="##other" processContents="lax"/>
1060 </xs:complexType>
1061 <xs:element name="Sequence" type="wsrm:SequenceType"/>
1062 <xs:element name="SequenceAcknowledgement">
1063 <xs:complexType>
1064 <xs:sequence>
1065 <xs:element ref="wsrm:Identifier"/>
1066 <xs:choice>
1067 <ws:sequence>
1068 <xs:element name="AcknowledgementRange"
1069 maxOccurs="unbounded">
1070 <xs:complexType>
1071 <xs:sequence/>
1072 <xs:attribute name="Upper" type="xs:unsignedLong"
1073 use="required"/>
1074 <xs:attribute name="Lower" type="xs:unsignedLong"
1075 use="required"/>
1076 <xs:anyAttribute namespace="##other"
1077 processContents="lax"/>
1078 </xs:complexType>
1079 </xs:element>
1080 <ws:element name="Final" minOccurs="0">
1081 <xs:complexType>
1082 <xs:sequence/>
1083 </xs:complexType>
1084 </ws:element>
1085 </ws:sequence>
1086 <xs:element name="Nack" type="xs:unsignedLong"
1087 maxOccurs="unbounded"/>
1088 <xs:element name="None" minOccurs="0">
1089 <xs:complexType>
1090 <xs:sequence/>
```

```

1091         </xs:complexType>
1092     </xs:element>
1093 </xs:choice>
1094     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1095 maxOccurs="unbounded"/>
1096 </xs:sequence>
1097     <xs:anyAttribute namespace="##other" processContents="lax"/>
1098 </xs:complexType>
1099 </xs:element>
1100 <xs:complexType name="AckRequestedType">
1101     <xs:sequence>
1102         <xs:element ref="wsrm:Identifier"/>
1103         <xs:element name="MessageNumber" type="xs:unsignedLong"
1104 minOccurs="0"/>
1105         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1106 maxOccurs="unbounded"/>
1107     </xs:sequence>
1108     <xs:anyAttribute namespace="##other" processContents="lax"/>
1109 </xs:complexType>
1110 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1111 <xs:element name="Identifier">
1112     <xs:complexType>
1113         <xs:annotation>
1114             <xs:documentation>
1115 This type is for elements whose [children] is an anyURI and can have
1116 arbitrary attributes.
1117             </xs:documentation>
1118         </xs:annotation>
1119         <xs:simpleContent>
1120             <xs:extension base="xs:anyURI">
1121                 <xs:anyAttribute namespace="##other" processContents="lax"/>
1122             </xs:extension>
1123         </xs:simpleContent>
1124     </xs:complexType>
1125 </xs:element>
1126 <!-- Fault Container and Codes -->
1127 <xs:simpleType name="FaultCodes">
1128     <xs:restriction base="xs:QName">
1129         <xs:enumeration value="wsrm:UnknownSequence"/>
1130         <xs:enumeration value="wsrm:SequenceTerminated"/>
1131         <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1132         <xs:enumeration value="wsrm:MessageNumberRollover"/>
1133         <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1134         <xs:enumeration value="wsrm:LastMessageNumberExceeded"/>
1135     </xs:restriction>
1136 </xs:simpleType>
1137 <xs:complexType name="SequenceFaultType">
1138     <xs:sequence>
1139         <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1140         <xs:any namespace="##any" processContents="lax" minOccurs="0"
1141 maxOccurs="unbounded"/>
1142     </xs:sequence>
1143     <xs:anyAttribute namespace="##any" processContents="lax"/>
1144 </xs:complexType>
1145 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1146 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1147 <xs:element name="CreateSequenceResponse"
1148 type="wsrm:CreateSequenceResponseType"/>
1149 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1150 <xs:element name="CloseSequenceResponse"
1151 type="wsrm:CloseSequenceResponseType"/>

```

```

1152 <xs:element name="TerminateSequence"
1153 type="wsrm:TerminateSequenceType"/>
1154 <xs:complexType name="CreateSequenceType">
1155 <xs:sequence>
1156 <xs:element ref="wsrm:AcksTo"/>
1157 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1158 <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1159 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1160 maxOccurs="unbounded"/>
1161 <xs:annotation>
1162 <xs:documentation>
1163 It is the authors intent that this extensibility be used to transfer a
1164 Security Token Reference as defined in WS Security.
1165 </xs:documentation>
1166 </xs:annotation>
1167 </xs:any>
1168 </xs:sequence>
1169 <xs:anyAttribute namespace="##other" processContents="lax"/>
1170 </xs:complexType>
1171 <xs:complexType name="CreateSequenceResponseType">
1172 <xs:sequence>
1173 <xs:element ref="wsrm:Identifier"/>
1174 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1175 <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1176 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1177 maxOccurs="unbounded"/>
1178 </xs:sequence>
1179 <xs:anyAttribute namespace="##other" processContents="lax"/>
1180 </xs:complexType>
1181 <xs:complexType name="CloseSequenceType">
1182 <xs:sequence>
1183 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1184 maxOccurs="unbounded"/>
1185 </xs:sequence>
1186 <xs:attribute name="Identifier" type="xs:anyURI" use="required"/>
1187 <xs:anyAttribute namespace="##other" processContents="lax"/>
1188 </xs:complexType>
1189 <xs:complexType name="CloseSequenceResponseType">
1190 <xs:sequence>
1191 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1192 maxOccurs="unbounded"/>
1193 </xs:sequence>
1194 <xs:anyAttribute namespace="##other" processContents="lax"/>
1195 </xs:complexType>
1196 <xs:complexType name="TerminateSequenceType">
1197 <xs:sequence>
1198 <xs:element ref="wsrm:Identifier"/>
1199 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1200 maxOccurs="unbounded"/>
1201 </xs:sequence>
1202 <xs:anyAttribute namespace="##other" processContents="lax"/>
1203 </xs:complexType>
1204 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
1205 <xs:complexType name="OfferType">
1206 <xs:sequence>
1207 <xs:element ref="wsrm:Identifier"/>
1208 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1209 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1210 maxOccurs="unbounded"/>
1211 </xs:sequence>
1212 <xs:anyAttribute namespace="##other" processContents="lax"/>
1213 </xs:complexType>

```

```
1214 <xs:complexType name="AcceptType">
1215 <xs:sequence>
1216 <xs:element ref="wsrm:AcksTo"/>
1217 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1218 maxOccurs="unbounded"/>
1219 </xs:sequence>
1220 <xs:anyAttribute namespace="##other" processContents="lax"/>
1221 </xs:complexType>
1222 <xs:element name="Expires">
1223 <xs:complexType>
1224 <xs:simpleContent>
1225 <xs:extension base="xs:duration">
1226 <xs:anyAttribute namespace="##other" processContents="lax"/>
1227 </xs:extension>
1228 </xs:simpleContent>
1229 </xs:complexType>
1230 </xs:element>
1231 </xs:schema>
```

1232 **Appendix B. ~~Message Examples~~**

1233 B.1. Create Sequence

1234 Create Sequence

```
1235 <?xml version="1.0" encoding="UTF-8"?>
1236 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1237 xmlns:wsrn="http://docs.oasis-open.org/wsrn/200510/"
1238 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1239   <S:Header>
1240     <wsa:MessageID>
1241       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
1242     </wsa:MessageID>
1243     <wsa:To>http://example.com/serviceB/123</wsa:To>
1244     <wsa:Action>http://docs.oasis-
1245 open.org/wsrn/200510/CreateSequence</wsa:Action>
1246     <wsa:ReplyTo>
1247       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1248     </wsa:ReplyTo>
1249   </S:Header>
1250   <S:Body>
1251     <wsrn:CreateSequence>
1252       <wsrn:AcksTo>
1253         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1254       </wsrn:AcksTo>
1255     </wsrn:CreateSequence>
1256   </S:Body>
1257 </S:Envelope>
```

1258 Create Sequence Response

```
1259 <?xml version="1.0" encoding="UTF-8"?>
1260 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1261 xmlns:wsrn="http://docs.oasis-open.org/wsrn/200510/"
1262 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1263   <S:Header>
1264     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1265     <wsa:RelatesTo>
1266       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
1267     </wsa:RelatesTo>
1268     <wsa:Action>
1269       http://docs.oasis-open.org/wsrn/200510/CreateSequenceResponse
1270     </wsa:Action>
1271   </S:Header>
1272   <S:Body>
1273     <wsrn:CreateSequenceResponse>
1274       <wsrn:Identifier>http://Business456.com/RM/ABC</wsrn:Identifier>
1275     </wsrn:CreateSequenceResponse>
1276   </S:Body>
1277 </S:Envelope>
```

1278 B.2. Initial Transmission

1279 The following example WS-ReliableMessaging headers illustrate the message exchange in
1280 the above figure. The three messages have the following headers; the third message is
1281 identified as the last message in the sequence:

1282 Message 1

```
1283 <?xml version="1.0" encoding="UTF-8"?>
1284 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1285 xmlns:wsmr="http://docs.oasis-open.org/wsmr/200510/"
1286 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1287   <S:Header>
1288     <wsa:MessageID>
1289       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
1290     </wsa:MessageID>
1291     <wsa:To>http://example.com/serviceB/123</wsa:To>
1292     <wsa:From>
1293       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1294     </wsa:From>
1295     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1296     <wsmr:Sequence>
1297       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1298       <wsmr:MessageNumber>1</wsmr:MessageNumber>
1299     </wsmr:Sequence>
1300   </S:Header>
1301   <S:Body>
1302     <!-- Some Application Data -->
1303   </S:Body>
1304 </S:Envelope>
```

1305 Message 2

```
1306 <?xml version="1.0" encoding="UTF-8"?>
1307 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1308 xmlns:wsmr="http://docs.oasis-open.org/wsmr/200510/"
1309 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1310   <S:Header>
1311     <wsa:MessageID>
1312       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1313     </wsa:MessageID>
1314     <wsa:To>http://example.com/serviceB/123</wsa:To>
1315     <wsa:From>
1316       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1317     </wsa:From>
1318     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1319     <wsmr:Sequence>
1320       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1321       <wsmr:MessageNumber>2</wsmr:MessageNumber>
1322     </wsmr:Sequence>
1323   </S:Header>
1324   <S:Body>
1325     <!-- Some Application Data -->
1326   </S:Body>
1327 </S:Envelope>
```

1328 Message 3

```
1329 <?xml version="1.0" encoding="UTF-8"?>
1330 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1331 xmlns:wsmr="http://docs.oasis-open.org/wsmr/200510/"
```

```
1332 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1333 <S:Header>
1334 <wsa:MessageID>
1335 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1336 </wsa:MessageID>
1337 <wsa:To>http://example.com/serviceB/123</wsa:To>
1338 <wsa:From>
1339 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1340 </wsa:From>
1341 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1342 <wsrm:Sequence>
1343 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1344 <wsrm:MessageNumber>3</wsrm:MessageNumber>
1345 <wsrm:LastMessage/>
1346 </wsrm:Sequence>
1347 </S:Header>
1348 <S:Body>
1349 <!-- Some Application Data -->
1350 </S:Body>
1351 </S:Envelope>
```

1352 B.3. First Acknowledgement

1353 Message number 2 has not been received by the RM-Destination due to some transmission-
1354 error so it responds with an acknowledgement for messages 1 and 3:

```
1355 <?xml version="1.0" encoding="UTF-8"?>
1356 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1357 xmlns:wsmr="http://docs.oasis-open.org/wsmr/200510/"
1358 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1359 <S:Header>
1360 <wsa:MessageID>
1361 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1362 </wsa:MessageID>
1363 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1364 <wsa:From>
1365 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1366 </wsa:From>
1367 <wsa:Action>
1368 http://docs.oasis-open.org/wsmr/200510/SequenceAcknowledgement
1369 </wsa:Action>
1370 <wsmr:SequenceAcknowledgement>
1371 <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1372 <wsmr:AcknowledgementRange Upper="1" Lower="1"/>
1373 <wsmr:AcknowledgementRange Upper="3" Lower="3"/>
1374 </wsmr:SequenceAcknowledgement>
1375 </S:Header>
1376 <S:Body/>
1377 </S:Envelope>
```

1378 B.4. Retransmission

1379 The sending endpoint discovers that message number 2 was not received so it resends the
1380 message and requests an acknowledgement:

```
1381 <?xml version="1.0" encoding="UTF-8"?>
1382 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1383 xmlns:wsmr="http://docs.oasis-open.org/wsmr/200510/"
1384 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1385 <S:Header>
1386 <wsa:MessageID>
1387 http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1388 </wsa:MessageID>
1389 <wsa:To>http://example.com/serviceB/123</wsa:To>
1390 <wsa:From>
1391 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1392 </wsa:From>
1393 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1394 <wsmr:Sequence>
1395 <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1396 <wsmr:MessageNumber>2</wsmr:MessageNumber>
1397 </wsmr:Sequence>
1398 <wsmr:AckRequested>
1399 <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1400 </wsmr:AckRequested>
1401 </S:Header>
1402 <S:Body>
1403 <!-- Some Application Data -->
1404 </S:Body>
1405 </S:Envelope>
```

1406 B.5.Termination

1407 The RM Destination now responds with an acknowledgement for the complete sequence
1408 which can then be terminated:

```
1409 <?xml version="1.0" encoding="UTF-8"?>
1410 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1411 xmlns:wsmr="http://docs.oasis-open.org/wsmr/200510/"
1412 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1413 <S:Header>
1414 <wsa:MessageID>
1415 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1416 </wsa:MessageID>
1417 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1418 <wsa:From>
1419 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1420 </wsa:From>
1421 <wsa:Action>
1422 http://docs.oasis-open.org/wsmr/200510/SequenceAcknowledgement
1423 </wsa:Action>
1424 <wsmr:SequenceAcknowledgement>
1425 <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1426 <wsmr:AcknowledgementRange Upper="3" Lower="1"/>
1427 </wsmr:SequenceAcknowledgement>
1428 </S:Header>
1429 <S:Body/>
1430 </S:Envelope>
```

1431 Terminate Sequence

```
1432 <?xml version="1.0" encoding="UTF-8"?>
1433 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1434 xmlns:wsmr="http://docs.oasis-open.org/wsmr/200510/"
1435 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
1436 <S:Header>
1437 <wsa:MessageID>
1438 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1439 </wsa:MessageID>
1440 <wsa:To>http://example.com/serviceB/123</wsa:To>
1441 <wsa:Action>
1442 http://docs.oasis-open.org/wsmr/200510/TerminateSequence
1443 </wsa:Action>
1444 <wsa:From>
1445 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1446 </wsa:From>
1447 </S:Header>
1448 <S:Body>
1449 <wsmr:TerminateSequence>
1450 <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1451 </wsmr:TerminateSequence>
1452 </S:Body>
1453 </S:Envelope>
```

1454 Appendix C.WSDL

1455 The non-normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

1456 <http://docs.oasis-open.org/wsrmp/200510/wsd1/wsrmp.wsd1>

1457 The following non-normative copy is provided for reference.

```
1458 <wSDL:definitions xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
1459 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1460 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1461 xmlns:rm="http://docs.oasis-open.org/wsrmp/200510/"
1462 xmlns:tns="http://docs.oasis-open.org/wsrmp/200510/wsd1"
1463 targetNamespace="http://docs.oasis-open.org/wsrmp/200510/wsd1">
1464 <wSDL:types>
1465 <xs:schema
1466 <xs:import namespace="http://docs.oasis-open.org/wsrmp/200510/"
1467 schemaLocation="http://docs.oasis-open.org/wsrmp/200510/wsrmp.xsd"/>
1468 <xs:import
1469 namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1470 schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
1471 </xs:schema>
1472 </wSDL:types>
1473 <wSDL:message name="CreateSequence">
1474 <wSDL:part name="create" element="rm:CreateSequence"/>
1475 </wSDL:message>
1476 <wSDL:message name="CreateSequenceResponse">
1477 <wSDL:part name="createResponse"
1478 element="rm:CreateSequenceResponse"/>
1479 </wSDL:message>
1480 <wSDL:message name="CloseSequence">
1481 <wSDL:part name="close" element="rm:CloseSequence"/>
1482 </wSDL:message>
1483 <wSDL:message name="CloseSequenceResponse">
1484 <wSDL:part name="closeResponse" element="rm:CloseSequenceResponse"/>
1485 </wSDL:message>
1486 <wSDL:message name="TerminateSequence">
1487 <wSDL:part name="terminate" element="rm:TerminateSequence"/>
1488 </wSDL:message>
1489 <wSDL:portType name="SequenceAbstractPortType">
1490 <wSDL:operation name="CreateSequence">
1491 <wSDL:input message="tns:CreateSequence"
1492 wsa:Action="http://docs.oasis-open.org/wsrmp/200510/CreateSequence"/>
1493 <wSDL:output message="tns:CreateSequenceResponse"
1494 wsa:Action="http://docs.oasis-
1495 open.org/wsrmp/200510/CreateSequenceResponse"/>
1496 </wSDL:operation>
1497 <wSDL:operation name="CloseSequence">
1498 <wSDL:input name="tns:CloseSequence" wsa:Action="http://docs.oasis-
1499 open.org/wsrmp/200510/CloseSequence"/>
1500 <wSDL:output name="tns:CloseSequenceResponse"
1501 wsa:Action="http://docs.oasis-
1502 open.org/wsrmp/200510/CloseSequenceResponse"/>
1503 </wSDL:operation>
1504 <wSDL:operation name="TerminateSequence">
1505 <wSDL:input message="tns:TerminateSequence"
1506 wsa:Action="http://docs.oasis-
1507 open.org/wsrmp/200510/CreateSequenceResponse"/>
1508 </wSDL:operation>
1509 </wSDL:portType>
```

1510

`</wsdl:definitions>`

1511 A. Acknowledgments

1512 This document is based on initial contribution to OASIS WS-RX Technical Committee by the following
1513 authors: [Stefan Batres, Microsoft \(Editor\)](#), [Ruslan Bilorusets, BEA](#), [Don Box, Microsoft](#), [Luis Felipe](#)
1514 [Cabrera, Microsoft](#), [Derek Collison, TIBCO Software](#), [Donald Ferguson, IBM](#), [Christopher Ferris, IBM](#)
1515 [\(Editor\)](#), [Tom Freund, IBM](#), [Mary Ann Hondo, IBM](#), [John Ibbotson, IBM](#), [Lei Jin, BEA](#), [Chris Kaler,](#)
1516 [Microsoft](#), [David Langworthy, Microsoft](#), [Amelia Lewis, TIBCO Software](#), [Rodney Limprecht, Microsoft](#),
1517 [Steve Lucco, Microsoft](#), [Don Mullen, TIBCO Software](#), [Anthony Nadalin, IBM](#), [Mark Nottingham, BEA](#),
1518 [David Orchard, BEA](#), [Shivajee Samdarshi, TIBCO Software](#), [John Shewchuk, Microsoft](#), [Tony Storey,](#)
1519 [IBM](#), [Ruslan Bilorusets, BEA](#), [Don Box, Microsoft](#), [Luis Felipe Cabrera, Microsoft](#), [Doug Davis, IBM](#),
1520 [Donald Ferguson, IBM](#), [Christopher Ferris, IBM \(Editor\)](#), [Tom Freund, IBM](#), [Mary Ann Hondo, IBM](#), [John](#)
1521 [Ibbotson, IBM](#), [Lei Jin, BEA](#), [Chris Kaler, Microsoft](#), [David Langworthy, Microsoft \(Editor\)](#), [Amelia Lewis,](#)
1522 [TIBCO Software](#), [Rodney Limprecht, Microsoft](#), [Steve Lucco, Microsoft](#), [Don Mullen, TIBCO Software](#),
1523 [Anthony Nadalin, IBM](#), [Mark Nottingham, BEA](#), [David Orchard, BEA](#), [Jamie Roots, IBM](#), [Shivajee](#)
1524 [Samdarshi, TIBCO Software](#), [John Shewchuk, Microsoft](#), [Tony Storey, IBM](#)

1525 The following individuals have provided invaluable input into the initial contribution:

1526 [Keith Ballinger, Microsoft](#), [Allen Brown, Microsoft](#), [Michael Conner, IBM](#), [Francisco Curbera, IBM](#), [Steve](#)
1527 [Graham, IBM](#), [Pat Helland, Microsoft](#), [Rick Hill, Microsoft](#), [Scott Hinkelman, IBM](#), [Tim Holloway, IBM](#), [Efim](#)
1528 [Hudis, Microsoft](#), [Johannes Klein, Microsoft](#), [Frank Leymann, IBM](#), [Martin Nally, IBM](#), [Peter Niblett, IBM](#),
1529 [Jeffrey Schlimmer, Microsoft](#), [Chris Sharp, IBM](#), [James Snell, IBM](#), [Keith Stobie, Microsoft](#), [Satish Thatte,](#)
1530 [Microsoft](#), [Stephen Todd, IBM](#), [Sanjiva Weerawarana, IBM](#), [Roger Wolter, Microsoft](#), ~~[Stefan Batres,](#)~~
1531 ~~[Microsoft](#)~~, ~~[Allen Brown, Microsoft](#)~~, ~~[Michael Conner, IBM](#)~~, ~~[George Copeland, Microsoft](#)~~, ~~[Francisco Curbera,](#)~~
1532 ~~[IBM](#)~~, ~~[Paul Fremantle, IBM](#)~~, ~~[Steve Graham, IBM](#)~~, ~~[Pat Helland, Microsoft](#)~~, ~~[Rick Hill, Microsoft](#)~~, ~~[Scott](#)~~
1533 ~~[Hinkelman, IBM](#)~~, ~~[Tim Holloway, IBM](#)~~, ~~[Efim Hudis, Microsoft](#)~~, ~~[Gopal Kakivaya, Microsoft](#)~~, ~~[Johannes Klein,](#)~~
1534 ~~[Microsoft](#)~~, ~~[Frank Leymann, IBM](#)~~, ~~[Martin Nally, IBM](#)~~, ~~[Peter Niblett, IBM](#)~~, ~~[Jeffrey Schlimmer, Microsoft](#)~~,
1535 ~~[James Snell, IBM](#)~~, ~~[Keith Stobie, Microsoft](#)~~, ~~[Satish Thatte, Microsoft](#)~~, ~~[Stephen Todd, IBM](#)~~, ~~[Sanjiva](#)~~
1536 ~~[Weerawarana, IBM](#)~~, ~~[Roger Wolter, Microsoft](#)~~

1537 The following individuals were members of the committee during the development of this specification:

1538 TBD

1539 B. XML Schema

1540 [A normative copy of the XML Schema \[XML-Schema Part1, XML-Schema Part2\] description for this](#)
1541 [specification may be retrieved from the following address:](#)

1542 <http://docs.oasis-open.org/ws-rx/wsrmp/200510/wsrmp-1.1-schema-200510.xsd>

1543 [The following copy is provided for reference.](#)

```
1544 <?xml version="1.0" encoding="UTF-8"?>
1545 <!--
1546 OASIS takes no position regarding the validity or scope of any
1547 intellectual property or other rights that might be claimed to pertain to
1548 the implementation or use of the technology described in this document or
1549 the extent to which any license under such rights might or might not be
1550 available; neither does it represent that it has made any effort to
1551 identify any such rights. Information on OASIS's procedures with respect
1552 to rights in OASIS specifications can be found at the OASIS website.
1553 Copies of claims of rights made available for publication and any
1554 assurances of licenses to be made available, or the result of an attempt
1555 made to obtain a general license or permission for the use of such
1556 proprietary rights by implementors or users of this specification, can be
1557 obtained from the OASIS Executive Director.
1558 OASIS invites any interested party to bring to its attention any
1559 copyrights, patents or patent applications, or other proprietary rights
1560 which may cover technology that may be required to implement this
1561 specification. Please address the information to the OASIS Executive
1562 Director.
1563 Copyright © OASIS Open 2002-2006. All Rights Reserved.
1564 This document and translations of it may be copied and furnished to
1565 others, and derivative works that comment on or otherwise explain it or
1566 assist in its implementation may be prepared, copied, published and
1567 distributed, in whole or in part, without restriction of any kind,
1568 provided that the above copyright notice and this paragraph are included
1569 on all such copies and derivative works. However, this document itself
1570 does not be modified in any way, such as by removing the copyright notice
1571 or references to OASIS, except as needed for the purpose of developing
1572 OASIS specifications, in which case the procedures for copyrights defined
1573 in the OASIS Intellectual Property Rights document must be followed, or
1574 as required to translate it into languages other than English.
1575 The limited permissions granted above are perpetual and will not be
1576 revoked by OASIS or its successors or assigns.
1577 This document and the information contained herein is provided on an
1578 "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
1579 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
1580 HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
1581 MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
1582 -->
1583 <xs:schema xmlns:tns="http://docs.oasis-open.org/ws-rx/wsrmp/200510"
1584 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1585 targetNamespace="http://docs.oasis-open.org/ws-rx/wsrmp/200510"
1586 elementFormDefault="qualified" attributeFormDefault="unqualified">
1587   <xs:element name="RMAssertion">
1588     <xs:complexType>
1589       <xs:sequence>
1590         <xs:element
1591 name="AcknowledgementInterval" minOccurs="0">
1592           <xs:complexType>
1593             <xs:attribute
1594 name="Milliseconds" type="xs:unsignedLong" use="required"/>

```

```
1595                                     <xs:anyAttribute
1596 namespace="##any" processContents="lax"/>
1597                                     </xs:complexType>
1598                                     </xs:element>
1599                                     <xs:element name="MaxMessageNumber"
1600 minOccurs="0">
1601                                     <xs:complexType>
1602                                     <xs:attribute
1603 name="Number" type="xs:unsignedLong" use="required"/>
1604                                     <xs:anyAttribute
1605 namespace="##any" processContents="lax"/>
1606                                     </xs:complexType>
1607                                     </xs:element>
1608                                     <xs:any namespace="##other"
1609 processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
1610                                     </xs:sequence>
1611                                     <xs:anyAttribute namespace="##any"
1612 processContents="lax"/>
1613                                     </xs:complexType>
1614                                     </xs:element>
1615 </xs:schema>
```

C. Revision History

Revision	Date	By Whom	What
wd-01.doc	2005-07-06	Ümit Yalçinalp	Initial version created based on submission by the authors.
1.0-wd-01.swx	2005-09-01	Ümit Yalçinalp	Reformatted using Open Office
1.1-wd-01.swx	2005-09-18	Ümit Yalçinalp	Applied resolution i001 Applied resolution i015/16 (doc identifier) Partial application of i017, final yyyy/mm required, changed doc URI to TBD pending yyyy/mm Deleted original copyright section
1.1-wd-01.swx	2005-10-02	Anish Karmarkar	Applied resolution of i013 + minor editorial changes + fixed resolution of i017
1.1-wd-01.swx	2005-10-04	Ümit Yalçinalp	Applied actual value for yyyy/mm. Added resolution of i009
1.1-wd-01.swx	2005-10-06	Ümit Yalçinalp	Editorial fixes suggested by Anish Updated wd draft date to October 6th
1.1-wd-01.swx	2005-10-19	Ümit Yalçinalp	Editorial change to remove .swx suffix from doc id.
wd-02	2005-11-03	Gilbert Pilz	Start wd-02 by changing title page from cd-01.
wd-02	2005-11-30	Gilbert Pilz	i072 – editorial nits
wd-02	2005-11-30	Gilbert Pilz	i074 - Use of [tcShortName] in artifact locations namespaces, etc
wd-02	2005-12-01	Gilbert Pilz	Updated fix to i074 to remove trailing '/' from wsrmp namespace.
wd-02	2005-12-01	Anish Karmarkar	Applied resolution for i022
wd-02	2005-12-01	Anish Karmarkar	Applied resolution for i024
wd-02	2005-12-01	Anish Karmarkar	Applied resolution for i054
wd-02	2005-12-01	Anish Karmarkar	Applied resolution of i073
wd-2	2005-12-05	Anish Karmarkar	Applied resolution of i055
wd-2	2005-12-05	Ümit Yalçinalp	Changed fixed date in footer to current date
wd-3	2005-12-21	Doug Davis	Added i050
wd-3	2005-12-23	Ümit Yalçinalp	i057 resolution
wd-3	2005-12-23	Ümit Yalçinalp	Changed the ref to WS-RM to the WS-RX committee.

<u>Revision</u>	<u>Date</u>	<u>By Whom</u>	<u>What</u>
			draft instead of original version Fixed Dug's email address
wd-3	2005-12-23	Ümit Yalçınalp	I060 resolution
wd-03	2005-12-27	Gilbert Pilz	Remove schema example and put it in its own artifact (wsrmp-1.1-schema-200510.xsd). Convert source file to OpenDocument format. Make line numbers all the same style.
wd-03	2005-12-28	Anish Karmarkar	Included a section link to c:\temp\wsrmp-1.1-schema-200510.xsd
wd-03	2006-01-04	Gilbert Pilz	Fixed formatting of included section.
wd-03	2006-01-05	Gilbert Pilz	Fix closing tag of normative outline for RMAssertion.
wd-04	2006-11-11	Doug Davis	Minor tweaks/typos
cd-02	2006-01-13	Gilbert Pilz	Titles, boilerplate, etc. for cd-02

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

Rev	Date	By Whom	What
			http://docs.oasis-open.org/wsrn/200510/
wd-05	2005-09-30	Anish Karmarkar	i045 (Include SecureConversation as a reference and move it to non-normative citation)
wd-05	2005-09-30	Anish Karmarkar	i046 (change the type of wsrn:FaultCode element)

1617 D. Notices

1618 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
1619 might be claimed to pertain to the implementation or use of the technology described in this document or
1620 the extent to which any license under such rights might or might not be available; neither does it represent
1621 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to
1622 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made
1623 available for publication and any assurances of licenses to be made available, or the result of an attempt
1624 made to obtain a general license or permission for the use of such proprietary rights by implementors or
1625 users of this specification, can be obtained from the OASIS Executive Director.

1626 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or
1627 other proprietary rights which may cover technology that may be required to implement this specification.
1628 Please address the information to the OASIS Executive Director.

1629 [Copyright \(C\) OASIS Open \(2006\). All Rights Reserved.](#)

1630 [This document and translations of it may be copied and furnished to others, and derivative works that
1631 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
1632 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
1633 this paragraph are included on all such copies and derivative works. However, this document itself may
1634 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as
1635 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights
1636 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it
1637 into languages other than English.](#)
Copyright (C) OASIS Open (2005). All Rights Reserved.

1638 [The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
1639 or assigns.](#)

~~1640 This document and translations of it may be copied and furnished to others, and derivative works that
1641 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
1642 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
1643 this paragraph are included on all such copies and derivative works. However, this document itself may
1644 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as
1645 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights
1646 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it
1647 into languages other than English.~~

~~1648 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
1649 or assigns.~~

1650 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1651 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
1652 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR
1653 ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.