



1 Web Services Reliable Messaging 2 (WS-ReliableMessaging)

3 Working Draft 14, June 8, 2006

4 **Document identifier:**

5 wsrn-1.1-spec-wd-14

6 **Location:**

7 **Editors:**

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

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

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

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

12 Ümit Yalçınalp, SAP <umit.yalcinalp@sap.com>

13 **Contributors:**

14 TBD

15 **Abstract:**

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

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

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

30 **Status:**

31 This document is a work in progress and will be updated to reflect issues as they are resolved by the
32 Web Services Reliable Exchange (WS-RX) Technical Committee.

33 **Table of Contents**

34	1 Introduction.....	4
35	1.1 Goals and Requirements.....	4
36	1.1.1 Requirements.....	4
37	1.2 Notational Conventions.....	4
38	1.3 Namespace.....	4
39	1.4 Compliance.....	5
40	2 Reliable Messaging Model.....	6
41	2.1 Glossary.....	6
42	2.2 Protocol Preconditions.....	7
43	2.3 Protocol Invariants.....	7
44	2.4 Example Message Exchange.....	7
45	3 RM Protocol Elements.....	10
46	3.1 Sequence Creation.....	10
47	3.2 Closing A Sequence.....	13
48	3.3 Sequence Termination.....	15
49	3.4 Sequences.....	16
50	3.5 Request Acknowledgement.....	17
51	3.6 Sequence Acknowledgement.....	18
52	4 Faults.....	21
53	4.1 SequenceFault Element.....	22
54	4.2 Sequence Terminated.....	23
55	4.3 Unknown Sequence.....	23
56	4.4 Invalid Acknowledgement.....	24
57	4.5 Message Number Rollover.....	24
58	4.6 Create Sequence Refused.....	24
59	4.7 Sequence Closed.....	25
60	4.8 WSRM Required.....	25
61	5 Security Considerations.....	26
62	6 References.....	28
63	6.1 Normative.....	28
64	6.2 Non-Normative.....	28
65	A. Schema.....	30
66	B. Message Examples.....	35
67	B.1 Create Sequence.....	35
68	B.2 Initial Transmission.....	35
69	B.3 First Acknowledgement.....	37
70	B.4 Retransmission.....	37

71	B.5 Termination.....	38
72	C. WSDL.....	40
73	D. State Tables.....	42
74	E. Acknowledgments.....	46
75	F. Revision History.....	47
76	G. Notices.....	51

77 1 Introduction

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

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

87 1.1 Notational Conventions

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

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

- 88 • The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- 89 • Characters are appended to elements and attributes to indicate cardinality:
 - 90 ○ "?" (0 or 1)
 - 91 ○ "*" (0 or more)
 - 92 ○ "+" (1 or more)
- 93 • The character "|" is used to indicate a choice between alternatives.
- 94 • The characters "[" and "]" are used to indicate that contained items are to be treated as a group
95 with respect to cardinality or choice.
- 96 • An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content
97 specified in this document. Additional children elements and/or attributes MAY be added at the
98 indicated extension points but they MUST NOT contradict the semantics of the parent and/or
99 owner, respectively. If an extension is not recognized it SHOULD be ignored.
- 100 • XML namespace prefixes (See Section 1.2) are used to indicate the namespace of the element
101 being defined.

102 1.2 Namespace

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

103 <http://docs.oasis-open.org/ws-rx/wsrn/200604>

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

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

103 Table 1

Prefix	Namespace
S	(Either SOAP 1.1 or 1.2)
S11	http://schemas.xmlsoap.org/soap/envelope/
S12	http://www.w3.org/2003/05/soap-envelope
wstrm	http://docs.oasis-open.org/ws-rx/wstrm/200604
wsa	http://www.w3.org/2005/08/addressing
xs	http://www.w3.org/2001/XMLSchema

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

104 <http://docs.oasis-open.org/ws-rx/wstrm/200604/wstrm-1.1-schema-200604.xsd>

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

104 1.3 Compliance

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

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

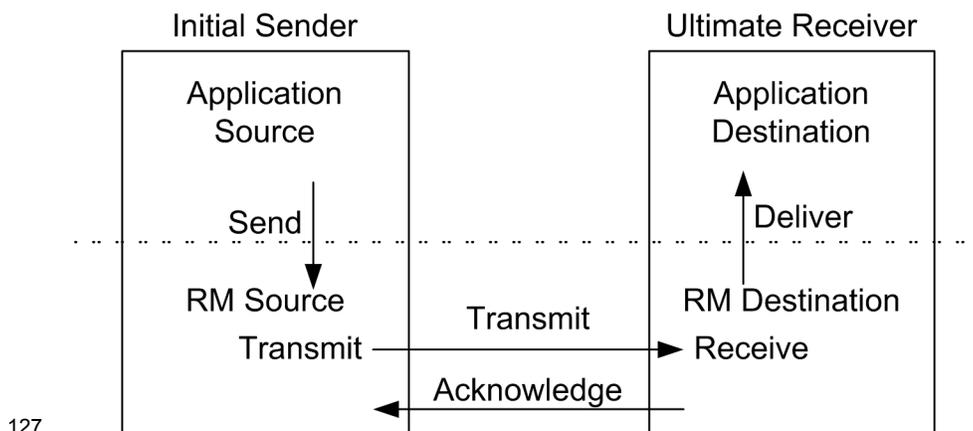
109 2 Reliable Messaging Model

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

110 The WS-ReliableMessaging specification defines an interoperable protocol that requires a Reliable
111 Messaging (RM) Source and Reliable Messaging Destination to ensure that each message transmitted by
112 the RM Source is successfully received by an RM Destination, or barring successful receipt, that an RM
113 Source can, except in the most extreme circumstances, accurately determine the disposition of each
114 message transmitted as perceived by the RM Destination, so as to resolve any in-doubt status. Note that
115 this specification makes no restriction on the scope of the RM Source or RM Destination entities. For
116 example, either may span multiple WSDL Ports or endpoints.

117 The protocol supports reliability features that enable ordered delivery, duplicate elimination, and
118 guaranteed receipt for the RM Destination. It is expected that the Application Destination and RM
119 Destination will implement as many of these or as few of these characteristics as necessary. Regardless
120 of which of the reliability features are employed, the wire protocol does not change.

121 Figure 1 below illustrates the entities and events in a simple reliable exchange of messages. First, the
122 Application Source Sends a message for reliable delivery. The Reliable Messaging Source accepts the
123 message and transmits it one or more times. After receiving the message, the RM Destination
124 Acknowledges it. Finally, the RM Destination delivers the message to the Application Destination. The
125 exact roles the entities play and the complete meaning of the events will be defined throughout this
126 specification.



127 Figure 1: Reliable Messaging Model

127 2.1 Glossary

128 The following definitions are used throughout this specification:

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

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

128 **Application Source:** The endpoint that sends a message.

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

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

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

134 **RM Destination:** For any one reliably sent message the endpoint that receives the message.

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

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

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

139 **2.2 Protocol Preconditions**

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

- 142 • For any single message exchange the RM Source **MUST** have an endpoint reference that uniquely
143 identifies the RM Destination endpoint.
- 144 • The RM Source **MUST** have knowledge of the destination's policies, if any, and the RM Source
145 **MUST** be capable of formulating messages that adhere to this policy.

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

148 **2.3 Protocol Invariants**

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

- 150 • The RM Source **MUST** assign each message within a Sequence a message number (defined
151 below) beginning at 1 and increasing by exactly 1 for each subsequent message. These numbers
152 **MUST** be assigned in the same order in which messages are sent by the Application Source.
- 153 • Within every acknowledgement it issues, the RM Destination **MUST** include one or more
154 acknowledgement ranges that contain the message number of every message successfully
155 received by the RM Destination. The RM Destination **MUST** exclude the message numbers of any
156 messages it has not received.

153 **2.4 Example Message Exchange**

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

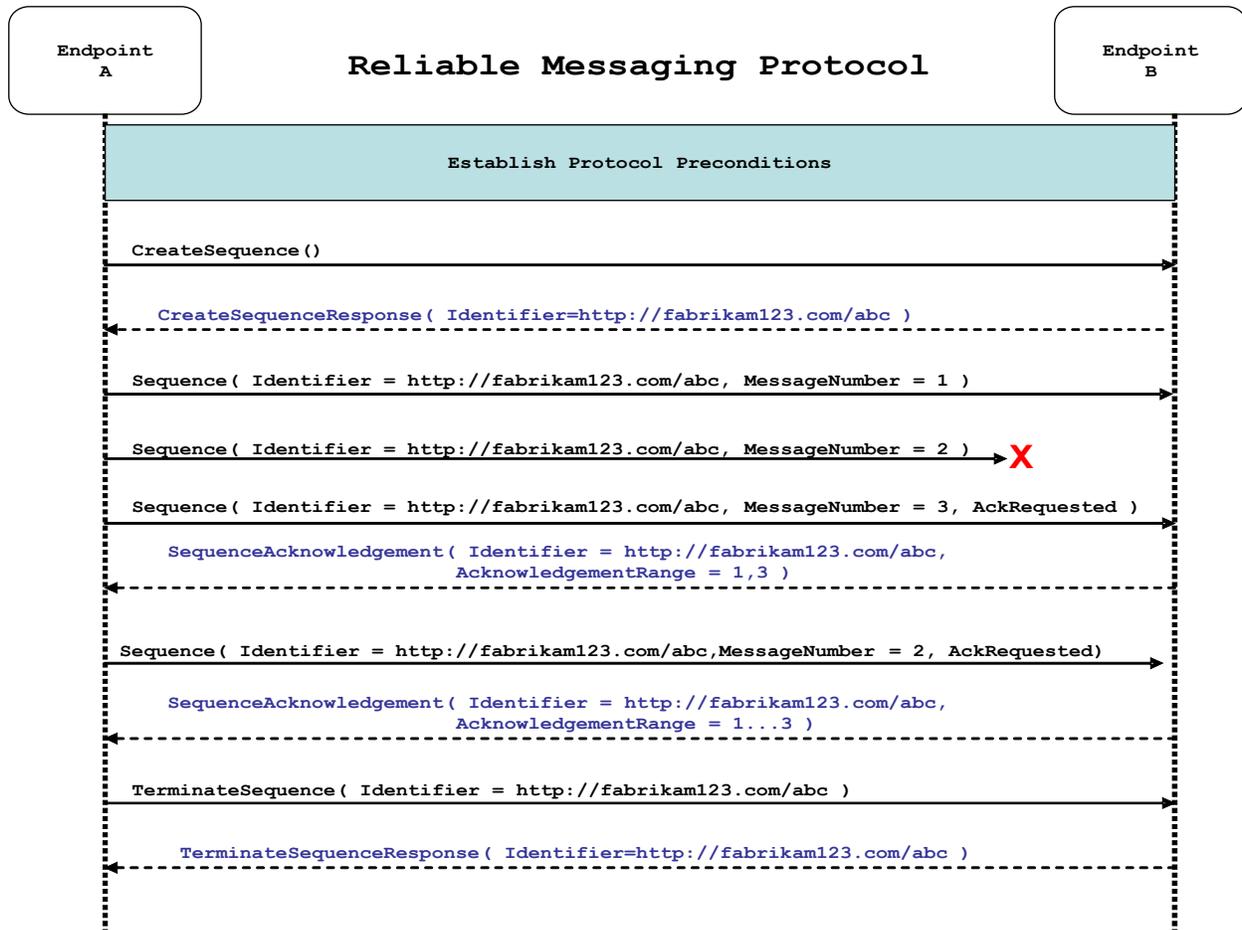


Figure 2: The WS-ReliableMessaging Protocol

- 154 1. The protocol preconditions are established. These include policy exchange, endpoint resolution,
155 and establishing trust.
- 154 2. The RM Source requests creation of a new Sequence.
- 154 3. The RM Destination creates a new Sequence and returns its globally unique identifier.
- 154 4. The RM Source begins transmitting messages in the Sequence beginning with MessageNumber 1.
155 In the figure above, the RM Source sends 3 messages in the Sequence.
- 154 5. The 2nd message in the Sequence is lost in transit.
- 154 6. The 3rd message is the last in this Sequence and the RM Source includes a
155 `<wsrm:AckRequested>` header to ensure that it gets a timely
156 `<wsrm:SequenceAcknowledgement>` for the Sequence.
- 154 7. The RM Destination acknowledges receipt of message numbers 1 and 3 as a result of receiving the
155 RM Source's `<wsrm:AckRequested>` header.
- 154 8. The RM Source retransmits the unacknowledged message with MessageNumber 2. This is a new
155 message from the perspective of the underlying transport, but it has the same Sequence Identifier
156 and MessageNumber so the RM Destination can recognize it as a duplicate of the earlier message,
157 in case the original and retransmitted messages are both received. The RM Source includes an
158 `<wsrm:AckRequested>` header in the retransmitted message so the RM Destination will expedite
159 an acknowledgement.

154 9. The RM Destination receives the second transmission of the message with MessageNumber 2 and
155 acknowledges receipt of message numbers 1, 2, and 3.

154 10. The RM Source receives this acknowledgement and sends a TerminateSequence message to the
155 RM Destination indicating that the Sequence is completed and reclaims any resources associated
156 with the Sequence.

154 11. The RM Destination receives the TerminateSequence message indicating that the RM Source will
155 not be sending any more messages. The RM Destination sends a TerminateSequenceResponse
156 message to the RM Source and reclaims any resources associated with the Sequence.

154 The RM Source will expect to receive acknowledgements from the RM Destination during the course of a
155 message exchange at occasions described in Section 3 below. Should an acknowledgement not be
156 received in a timely fashion, the RM Source MUST re-transmit the message since either the message or
157 the associated acknowledgement might have been lost. Since the nature and dynamic characteristics of
158 the underlying transport and potential intermediaries are unknown in the general case, the timing of re-
159 transmissions cannot be specified. Additionally, over-aggressive re-transmissions have been
160 demonstrated to cause transport or intermediary flooding which are counterproductive to the intention of
161 providing a reliable exchange of messages. Consequently, implementers are encouraged to utilize
162 adaptive mechanisms that dynamically adjust re-transmission time and the back-off intervals that are
163 appropriate to the nature of the transports and intermediaries envisioned. For the case of TCP/IP
164 transports, a mechanism similar to that described as RTTM in RFC 1323 [RTTM] should be considered.

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

3 RM Protocol Elements

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

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

If action IRIs are used by either the RM Source or the RM Destination, and one is not already defined as per the rules of the WS-Addressing specification, then the RM Source or the RM Destination MUST use an action IRI that consists of the WS-RM namespace URI concatenated with a '/', followed by the message element name. For example:

```
http://docs.oasis-open.org/ws-rx/wsrn/200604/SequenceAcknowledgement
```

3.1 Sequence Creation

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

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

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

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

`/wsrm:CreateSequence`

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

`/wsrm:CreateSequence/wsrn:AcksTo`

The RM Source MUST include this element in any `CreateSequence` message it sends. This element is of type `wsa:EndpointReferenceType` (as specified by WS-Addressing). It specifies the endpoint reference to which messages containing `<wsrm:SequenceAcknowledgement>` header blocks and

179 faults related to the created Sequence are to be sent, unless otherwise noted in this specification (for
180 example, see Section 3.2).

181 Implementations MUST NOT use an endpoint reference in the AcksTo element that would prevent the
182 sending of Sequence Acknowledgements back to the RM Source. For example, using the WS-Addressing
183 "none" IRI would make it impossible for the RM Destination to ever send Sequence Acknowledgements.

181 /wsmr:CreateSequence/wsmr:Expires

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

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

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

181 /wsmr:CreateSequence/wsmr:Offer

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

181 /wsmr:CreateSequence/wsmr:Offer/wsmr:Identifier

181 The RM Source MUST set the value of this element to an absolute URI (conformant with RFC3986 [URI])
182 that will uniquely identify the offered Sequence.

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

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

181 /wsmr:CreateSequence/wsmr:Offer/wsmr:Endpoint

181 An RM Source MUST include this element, of type `wsa:EndpointReferenceType` (as specified by
182 WS-Addressing [WSAddressing]) This element specifies the endpoint reference to which WS-RM protocol
183 messages related to the offered Sequence are to be sent.

181 /wsmr:CreateSequence/wsmr:Offer/wsmr:Expires

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

184 /wsmr:CreateSequence/wsmr:Offer/wsmr:Expires/@{any}

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

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

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

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

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

184 /wsmr:CreateSequence/{any}

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

184 /wsmr:CreateSequence/@{any}

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

184 A <wsmr:CreateSequenceResponse> is sent in the body of a response message by an RM
185 Destination in response to receipt of a <wsmr:CreateSequence> request message. It carries the
186 <wsmr:Identifier> of the created Sequence and indicates that the RM Source may begin sending
187 messages in the context of the identified Sequence.

184 The following exemplar defines the <wsmr:CreateSequenceResponse> syntax:

```
184 <wsmr:CreateSequenceResponse ...>  
184   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
184   <wsmr:Expires> xs:duration </wsmr:Expires> ?  
184   <wsmr:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?  
184   <wsmr:IncompleteSequenceBehavior> wsmr:IncompleteSequenceBehaviorType  
185 </wsmr:IncompleteSequenceBehavior> ?  
184   <wsmr:Accept ...>  
184     <wsmr:AcksTo ...> wsa:EndpointReferenceType </wsmr:AcksTo>  
184     ...  
184   </wsmr:Accept> ?  
184   ...  
184 </wsmr:CreateSequenceResponse>
```

184 /wsmr:CreateSequenceResponse

184 This element is sent in the body of the response message in response to a <wsmr:CreateSequence>
185 request message. It indicates that the RM Destination has created a new Sequence at the request of the
186 RM Source. The RM Destination MUST NOT send this element as a header block.

184 /wsmr:CreateSequenceResponse/wsmr:Identifier

184 The RM Destination MUST include this element within any CreateSequenceResponse message it sends.
185 The RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986
186 [URI]) of the Sequence that has been created by the RM Destination.

184 /wsmr:CreateSequenceResponse/wsmr:Identifier/@{any}

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

184 /wsmr:CreateSequenceResponse/wsmr:Expires

184 This element, if present, of type xs:duration accepts or refines the RM Source's requested duration for
185 the Sequence. A value of 'PT0S' indicates that the Sequence will never expire. Absence of the element
186 indicates an implied value of 'PT0S'. The RM Destination MUST set the value of this element to be equal
187 to or less than the value requested by the RM Source in the corresponding <wsmr:CreateSequence>
188 message.

184 /wsmr:CreateSequenceResponse/wsmr:Expires/@{any}

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

184 /wsmr:CreateSequenceResponse/wsmr:AcknowledgementInterval

184 This element, if present, specifies the duration after which the RM Destination will transmit an
185 acknowledgement. If omitted, there is no implied value.

184 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@Milliseconds`
184 The acknowledgement interval, specified in milliseconds.

184 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@{any}`
184 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the
185 element.

184 `/wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior`
184 This OPTIONAL element, if present, specifies the behavior that the RM Destination will exhibit upon the
185 closure of an incomplete sequence.

186 A value of "DiscardEntireSequence" indicates that the entire sequence will be discarded by the RM
187 Destination if the sequence is closed when there are one or more gaps in the final
188 `SequenceAcknowledgement`.

189 A value of "DiscardFollowingFirstGap" indicates that messages in the sequence beyond the first gap will
190 be discarded by the RM Destination when there are one or more gaps in the final
191 `SequenceAcknowledgement`.

192 The default value of "NoDiscard" indicates that no acknowledged messages in the sequence will be
193 discarded by the RM Destination.

192 `/wsrm:CreateSequenceResponse/wsrm:Accept`
192 This element, if present, enables an RM Destination to accept the offer of a corresponding Sequence for
193 the reliable exchange of messages transmitted from RM Destination to RM Source.

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

192 `/wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo`
192 The RM Destination MUST include this element, of type `wsa:EndpointReferenceType` (as specified
193 by WS-Addressing). The RM Source SHOULD send messages with
194 `<wsrm:SequenceAcknowledgement>` header blocks related to the accepted Sequence to the
195 referenced endpoint.

196 `/wsrm:CreateSequenceResponse/wsrm:Accept/{any}`
197 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
198 to be passed.

199 `/wsrm:CreateSequenceResponse/wsrm:Accept/@{any}`
200 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
201 to be passed.

202 `/wsrm:CreateSequenceResponse/{any}`
203 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
204 to be passed.

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

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

207 **3.2 Closing A Sequence**

208 There may be times during the use of an RM Sequence that the RM Source or RM Destination will wish to
209 discontinue using a Sequence. Simply terminating the Sequence discards the state managed by the RM
210 Destination, leaving the RM Source unaware of the final ranges of messages that were successfully
211 delivered to the RM Destination. To ensure that the Sequence ends with a known final state both the RM
212 Source and RM Destination MAY choose to close the Sequence before terminating it.

213 If the RM Source wishes to close the Sequence, then it sends a `<wsrm:CloseSequence>` element, in
214 the body of a message, to the RM Destination. This message indicates that the RM Destination MUST
215 NOT receive any new messages for the specified Sequence, other than those already received at the time
216 the `<wsrm:CloseSequence>` element is interpreted by the RM Destination. Upon receipt of this
217 message, or subsequent to the RM Destination closing the Sequence of its own volition, the RM
218 Destination MUST include a final `<wsrm:SequenceAcknowledgement>` (within which the RM
219 Destination MUST include the `<wsrm:Final>` element) header block on any messages associated with
220 the Sequence destined to the RM Source, including the `CloseSequenceResponse` message or on any
221 Sequence Fault transmitted to the RM Source.

222 While the RM Destination MUST NOT receive any new messages for the specified Sequence it MUST still
223 process RM protocol messages. For example, it MUST respond to `AckRequested`, `TerminateSequence`
224 as well as `CloseSequence` messages. Note, subsequent `CloseSequence` messages have no effect on the
225 state of the Sequence.

222 In the case where the RM Destination wishes to discontinue use of a Sequence it is RECOMMENDED
223 that it close the Sequence. Please see `<wsrm:Final>` and the `SequenceClosed` fault. Whenever
224 possible the `SequenceClosed` Fault SHOULD be used in place of the `SequenceTerminated` Fault,
225 whenever possible, to allow the RM Source to still receive Acknowledgements.

222 The following exemplar defines the `CloseSequence` syntax:

```
222 <wsrm:CloseSequence ...>  
222   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
222   ...  
222 </wsrm:CloseSequence>
```

222 `/wsrm:CloseSequence`

222 This element is sent by an RM Source to indicate that the RM Destination MUST NOT receive any new
223 messages for this Sequence. A `SequenceClosed` fault MUST be generated by the RM Destination when it
224 receives a message for a Sequence that is already closed.

225 `/wsrm:CloseSequence/wsrm:Identifier`

225 The RM Source MUST include this element in any `CloseSequence` messages it sends. The RM Source
226 MUST set the value of this element to the absolute URI (conformant with RFC3986 [URI]) of the
227 Sequence that is being closed.

225 `/wsrm:CloseSequence/wsrm:Identifier/@{any}`

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

225 `/wsrm:CloseSequence/{any}`

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

225 /wsm:CloseSequence@{any}

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

225 A <wsm:CloseSequenceResponse> is sent in the body of a response message by an RM Destination
226 in response to receipt of a <wsm:CloseSequence> request message. It indicates that the RM
227 Destination has closed the Sequence.

225 The following exemplar defines the <wsm:CloseSequenceResponse> syntax:

```
225 <wsm:CloseSequenceResponse ...>  
225   <wsm:Identifier ...> xs:anyURI </wsm:Identifier>  
225   ...  
225 </wsm:CloseSequenceResponse>
```

225 /wsm:CloseSequenceResponse

225 This element is sent in the body of a response message by an RM Destination in response to receipt of a
226 <wsm:CloseSequence> request message. It indicates that the RM Destination has closed the
227 Sequence.

225 /wsm:CloseSequenceResponse/wsm:Identifier

225 The RM Destination MUST include this element in any CloseSequenceResponse message it sends. The
226 RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986 [URI])
227 of the Sequence that is being closed.

225 /wsm:CloseSequenceResponse/wsm:Identifier/@{any}

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

225 /wsm:CloseSequenceResponse/{any}

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

225 /wsm:CloseSequenceResponse@{any}

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

225 3.3 Sequence Termination

226 When the RM Source has completed its use of the Sequence it sends a <wsm:TerminateSequence>
227 element, in the body of a message, to the RM Destination to indicate that the Sequence is complete and
228 that it will not be sending any further messages related to the Sequence. The RM Destination can safely
229 reclaim any resources associated with the Sequence upon receipt of the <wsm:TerminateSequence>
230 message. Under normal usage the RM Source will complete its use of the Sequence when all of the
231 messages in the Sequence have been acknowledged. However, the RM Source is free to Terminate or
232 Close a Sequence at any time regardless of the acknowledgement state of the messages.

226 The following exemplar defines the TerminateSequence syntax:

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

226
226

```
...  
</wsrm:TerminateSequence>
```

226 /wsrm:TerminateSequence

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

226 /wsrm:TerminateSequence/wsrm:Identifier

226 The RM Source MUST include this element in any TerminateSequence message it sends. The RM
227 Source MUST set the value of this element to the absolute URI (conformant with RFC3986 [URI]) of the
228 Sequence that is being terminated.

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

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

229 /wsrm:TerminateSequence/{any}

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

229 /wsrm:TerminateSequence/@{any}

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

229 A <wsrm:TerminateSequenceResponse> is sent in the body of a response message by an RM
230 Destination in response to receipt of a <wsrm:TerminateSequence> request message. It indicates that
231 the RM Destination has terminated the Sequence.

229 The following exemplar defines the <wsrm:TerminateSequenceResponse> syntax:

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

229 /wsrm:TerminateSequenceResponse

229 This element is sent in the body of a response message by an RM Destination in response to receipt of a
230 <wsrm:TerminateSequence> request message. It indicates that the RM Destination has terminated
231 the sequence. The RM Destination MUST NOT send this element as a header block.

229 /wsrm:TerminateSequenceResponse/wsrm:Identifier

229 The RM Destination MUST include this element in any TerminateSequenceResponse message it sends.
230 The RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986
231 [URI]) of the Sequence that is being terminated.

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

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

232 /wsrm:TerminateSequenceResponse/{any}

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

232 /wsm:TerminateSequenceResponse/{any}

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

232 On receipt of a <wsm:TerminateSequence> message an RM Destination MUST respond with a
233 corresponding <wsm:TerminateSequenceResponse> message or generate a fault.

232 3.4 Sequences

233 The RM protocol uses a <wsm:Sequence> header block to track and manage the reliable delivery of
234 messages. The RM Source MUST include a <wsm:Sequence> header block in all messages for
235 which reliable delivery is required. The RM Source MUST identify Sequences with unique
236 <wsm:Identifier> elements and the RM Source MUST assign each message within a Sequence a
237 <wsm:MessageNumber> element that increments by 1 from an initial value of 1. These values are
238 contained within a <wsm:Sequence> header block accompanying each message being delivered in the
239 context of a Sequence.

233 The RM Source MUST NOT include more than one <wsm:Sequence> header block in any message.

233 A following exemplar defines its syntax:

```
233 <wsm:Sequence ...>  
233   <wsm:Identifier ...> xs:anyURI </wsm:Identifier>  
233   <wsm:MessageNumber> wsm:MessageNumberType </wsm:MessageNumber>  
233   ...  
233 </wsm:Sequence>
```

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

233 /wsm:Sequence

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

233 /wsm:Sequence/wsm:Identifier

233 An RM Source that includes a <wsm:Sequence> header block in a SOAP envelope MUST include this
234 element in that header block. The RM Source MUST set the value of this element to the absolute URI
235 (conformant with RFC3986 [URI]) that uniquely identifies the Sequence.

236 /wsm:Sequence/wsm:Identifier/{any}

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

236 /wsm:Sequence/wsm:MessageNumber

236 The RM Source MUST include this element within any Sequence headers it creates. This element is of
237 type *wsm:MessageNumberType*. It represents the ordinal position of the message within a Sequence.
238 Sequence message numbers start at 1 and monotonically increase by 1 throughout the Sequence. If the

239 message number exceeds the internal limitations of an RM Source or RM Destination or reaches the
240 maximum value of 9,223,372,036,854,775,807 the RM Source or Destination MUST generate a
241 MessageNumberRollover fault.

239 /wsm:Sequence/{any}

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

239 /wsm:Sequence/@{any}

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

239 The following example illustrates a Sequence header block.

```
239 <wsm:Sequence>  
239   <wsm:Identifier>http://example.com/abc</wsm:Identifier>  
239   <wsm:MessageNumber>10</wsm:MessageNumber>  
239 </wsm:Sequence>
```

239 3.5 Request Acknowledgement

240 The purpose of the <wsm:AckRequested> header block is to signal to the RM Destination that the RM
241 Source is requesting that a <wsm:SequenceAcknowledgement> be sent.

240 The RM Source MAY request an acknowledgement message from the RM Destination at any time by
241 including an <wsm:AckRequested> header block in any message targeted to the RM Destination. An
242 RM Destination that receives a message that contains an <wsm:AckRequested> header block MUST
243 send a message containing a <wsm:SequenceAcknowledgement> header block to the wsm:AcksTo
244 endpoint reference (see Section 3.1). If a non-mustUnderstand fault occurs when processing an RM
245 header that was piggy-backed on another message, a fault MUST be generated, but the processing of the
246 original message MUST NOT be affected. It is RECOMMENDED that the RM Destination return a
247 <wsm:AcknowledgementRange> or <wsm:None> element instead of a <wsm:Nack> element (see
248 Section 3.6).

249 The following exemplar defines its syntax:

```
249 <wsm:AckRequested ...>  
249   <wsm:Identifier ...> xs:anyURI </wsm:Identifier>  
  
249   ...  
249 </wsm:AckRequested>
```

249 /wsm:AckRequested

249 This element requests an acknowledgement for the identified Sequence.

249 /wsm:AckRequested/wsm:Identifier

249 An RM Source that includes a <wsm:AckRequested> header block in a SOAP envelope MUST include
250 this element in that header block. The RM Source MUST set the value of this element to the absolute URI,
251 (conformant with RFC3986 [URI]), that uniquely identifies the Sequence to which the request applies.

249 /wsm:AckRequested/wsm:Identifier/@{any}

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

251 /wsm:AckRequested/{any}

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

254 /wsmr:AckRequested/@{any}

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

257 3.6 Sequence Acknowledgement

258 The RM Destination informs the RM Source of successful message receipt using a
259 <wsmr:SequenceAcknowledgement> header block. The RM Destination MAY transmit the
260 <wsmr:SequenceAcknowledgement> header block independently or it MAY include the
261 <wsmr:SequenceAcknowledgement> header block on any message targeted to the AcksTo EPR.
262 Acknowledgements can be explicitly requested using the <wsmr:AckRequested> directive (see Section
263 3.5). If a non-mustUnderstand fault occurs when processing an RM header that was piggy-backed on
264 another message, a fault MUST be generated, but the processing of the original message MUST NOT be
265 affected.

266 A RM Destination MAY include a wsmr:SequenceAcknowledgement header block on any SOAP envelope
267 targetted to the endpoint referenced by the wsmr:AcksTo EPR.

268 During creation of a Sequence the RM Source MAY specify the WS-Addressing anonymous IRI as the
269 address of the <wsmr:AcksTo> EPR for that Sequence. When the RM Source specifies the WS-
270 Addressing anonymous IRI as the address of the <wsmr:AcksTo> EPR, the RM Destination MUST
271 transmit any <wsmr:SequenceAcknowledgement> headers for the created Sequence in a SOAP
272 envelope to be transmitted on the protocol binding-specific channel. Such a channel is provided by the
273 context of a received message containing a SOAP envelope that contains a <wsmr:Sequence> header
274 block and/or a <wsmr:AckRequested> header block for that same Sequence identifier.

275 The following exemplar defines its syntax:

```
275 <wsmr:SequenceAcknowledgement ...>  
276   <wsmr:Identifier ...> xs:anyURI </wsmr:Identifier>  
277   [ [ [ <wsmr:AcknowledgementRange ...  
278         Upper="wsmr:MessageNumberType"  
279         Lower="wsmr:MessageNumberType"/> +  
  
280         | <wsmr:None/> ]  
281         <wsmr:Final/> ? ]  
282         | <wsmr:Nack> wsmr:MessageNumberType </wsmr:Nack> + ]  
283   ...  
284   ...  
285 </wsmr:SequenceAcknowledgement>
```

286 The following describes the content model of the <wsmr:SequenceAcknowledgement> header block.

287 /wsmr:SequenceAcknowledgement

288 This element contains the Sequence acknowledgement information.

289 /wsmr:SequenceAcknowledgement/wsmr:Identifier

290 An RM Destination that includes a <wsmr:SequenceAcknowledgement> header block in a SOAP
291 envelope MUST include this element in that header block. The RM Destination MUST set the value of this
292 element to the absolute URI (conformant with RFC3986 [URI]) that uniquely identifies the Sequence. The

293 RM Destination MUST NOT include multiple `<wsrm:SequenceAcknowledgement>` header blocks that
294 share the same value for `<wsrm:Identifier>` within the same SOAP envelope.

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

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

295 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

295 The RM Destination MAY include one or more instances of this element within a
296 `<wsrm:SequenceAcknowledgement>` header block. It contains a range of Sequence
297 MessageNumbers successfully received by the RM Destination. The ranges SHOULD NOT overlap. The
298 RM Destination MUST NOT include this element if a sibling `<wsrm:Nack>` or `<wsrm:None>` element is
299 also present as a child of `<wsrm:SequenceAcknowledgement>`.

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

295 The RM Destination MUST set the value of this attribute equal to the message number of the highest
296 contiguous message in a Sequence range received by the RM Destination.

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

295 The RM Destination MUST set the value of this attribute equal to the message number of the lowest
296 contiguous message in a Sequence range received by the RM Destination.

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

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

295 `/wsrm:SequenceAcknowledgement/wsrm:Final`

295 The RM Destination MAY include this element within a `<wsrm:SequenceAcknowledgement>` header
296 block. This element indicates that the RM Destination is not receiving new messages for the specified
297 Sequence. The RM Source can be assured that the ranges of messages acknowledged by this
298 SequenceAcknowledgement header block will not change in the future. The RM Destination MUST
299 include this element when the Sequence is closed. The RM Destination MUST NOT include this element
300 when sending a Nack; it can only be used when sending `<wsrm:AcknowledgementRange>`s or
301 `<wsrm:None>`.

302 `/wsrm:SequenceAcknowledgement/wsrm:Nack`

302 The RM Destination MAY include this element within a `<wsrm:SequenceAcknowledgement>` header
303 block. If used, the RM Destination MUST set the value of this element to a `wsrm:MessageNumberType`
304 representing the `<wsrm:MessageNumber>` of an unreceived message in a Sequence. The RM
305 Destination MUST NOT include a `<wsrm:Nack>` element if a sibling
306 `<wsrm:AcknowledgementRange>` or `<wsrm:None>` element is also present as a child of
307 `<wsrm:SequenceAcknowledgement>`. Upon the receipt of a Nack, an RM Source SHOULD retransmit
308 the message identified by the Nack. The RM Destination MUST NOT issue a
309 `<wsrm:SequenceAcknowledgement>` containing a `<wsrm:Nack>` for a message that it has previously
310 acknowledged within a `<wsrm:AcknowledgementRange>`. The RM Source SHOULD ignore a
311 `<wsrm:SequenceAcknowledgement>` containing a `<wsrm:Nack>` for a message that has previously
312 been acknowledged within a `<wsrm:AcknowledgementRange>`.

302 `/wsrm:SequenceAcknowledgement/wsrm:None`

302 The RM Destination MUST include this element within a `<wsrm:SequenceAcknowledgement>` header
303 block if the RM Destination has not received any messages for the specified Sequence. The RM
304 Destination MUST NOT include this element if a sibling `<wsrm:AcknowledgementRange>` or
305 `<wsrm:Nack>` element is also present as a child of the `<wsrm:SequenceAcknowledgement>`.

302 `/wsrm:SequenceAcknowledgement/{any}`

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

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

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

302 The following examples illustrate `<wsrm:SequenceAcknowledgement>` elements:

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

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

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

```
302 <wsrm:SequenceAcknowledgement>  
302   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
302   <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
302   <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
302   <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
302 </wsrm:SequenceAcknowledgement>
```

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

```
302 <wsrm:SequenceAcknowledgement>  
302   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
302   <wsrm:Nack>3</wsrm:Nack>  
302 </wsrm:SequenceAcknowledgement>
```

302 4 Faults

303 The faults defined in this section fall into one of two categories; those faults that are the result of
304 messages or operations within a specific Sequence and those faults that are not. By their nature the
305 CreateSequenceRefused, UnknownSequence, and WSRMRequired faults cannot be correlated with a
306 Sequence. All other faults defined in this section relate to the processing of WS-RM protocol messages or
307 messages containing WS-RM header blocks targeted at a specific Sequence and are collectively referred
308 to as "Sequence faults".

303 Faults for the CreateSequence message exchange are treated as defined in WS-Addressing.
304 CreateSequenceRefused is a possible fault reply for this operation. UnknownSequence is a fault
305 generated by endpoints when messages carrying RM header blocks targeted at unrecognized or
306 terminated Sequences are detected. All other faults in this section relate to the processing of RM header
307 blocks targeted at known Sequences and are collectively referred to as Sequence faults. Entities that
308 generate Sequence faults SHOULD send those faults to the same [destination] as
309 `<wsrm:SequenceAcknowledgement>` messages. These faults are correlated using the Sequence
310 identifier carried in the detail.

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

```
313 http://docs.oasis-open.org/ws-rx/wsr/200604/fault
```

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

316 The definitions of faults use the following properties:

316 [Code] The fault code.

316 [Subcode] The fault subcode.

316 [Reason] The English language reason element.

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

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

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

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

```
323 <S:Envelope>  
324   <S:Header>  
325     <wsa:Action>  
326       http://docs.oasis-open.org/ws-rx/wsr/200604/fault  
327     </wsa:Action>  
328     <!-- Headers elided for clarity. -->  
329   </S:Header>  
330   <S:Body>  
331     <S:Fault>  
332       <S:Code>  
333         <S:Value> [Code] </S:Value>
```

```

334     <S:Subcode>
335         <S:Value> [Subcode] </S:Value>
336     </S:Subcode>
337 </S:Code>
338 <S:Reason>
339     <S:Text xml:lang="en"> [Reason] </S:Text>
340 </S:Reason>
341 <S:Detail>
342     [Detail]
343     ...
344 </S:Detail>
345 </S:Fault>
346 </S:Body>
347 </S:Envelope>

```

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

```

347 <S11:Envelope>
347 <S11:Header>
347     <wsrm:SequenceFault>
347         <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
347         <wsrm:Detail> [Detail] </wsrm:Detail>
347         ...
347     </wsrm:SequenceFault>
347 <!-- Headers elided for clarity. -->
347 </S11:Header>
347 <S11:Body>
347     <S11:Fault>
347         <faultcode> [Code] </faultcode>
347         <faultstring> [Reason] </faultstring>
347     </S11:Fault>
347 </S11:Body>
347 </S11:Envelope>

```

347 The properties bind to a SOAP 1.1 fault as follows when the fault is generated as a result of processing a
348 <wsrm:CreateSequence> request message:

```

347 <S11:Envelope>
347 <S11:Body>
347     <S11:Fault>
347         <faultcode> [Subcode] </faultcode>
347         <faultstring> [Reason] </faultstring>
347     </S11:Fault>
347 </S11:Body>
347 </S11:Envelope>

```

347 4.1 SequenceFault Element

348 The purpose of the <wsrm:SequenceFault> element is to carry the specific details of a fault generated
349 during the reliable messaging specific processing of a message belonging to a Sequence. WS-
350 ReliableMessaging nodes MUST use the <wsrm:SequenceFault> container only in conjunction with
351 the SOAP 1.1 fault mechanism. WS-ReliableMessaging nodes MUST NOT use the
352 <wsrm:SequenceFault> container in conjunction with the SOAP 1.2 binding.

348 The following exemplar defines its syntax:

```

348 <wsrm:SequenceFault ...>
348     <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
348     <wsrm:Detail> ... </wsrm:Detail> ?

```

```
348     ...
348     </wsrm:SequenceFault>
```

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

348 `/wsrm:SequenceFault`

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

348 `/wsrm:SequenceFault/wsrm:FaultCode`

348 WS-ReliableMessaging nodes that generate a `<wsrm:SequenceFault>` MUST set the value of this
349 element to a qualified name from the set of fault [Subcodes] defined below.

348 `/wsrm:SequenceFault/wsrm:Detail`

348 This OPTIONAL element is intended for carrying application specific error information related to the fault
349 being described.

350 `/wsrm:SequenceFault/wsrm:Detail/{any}`

350 The application specific error information related to the fault being described.

350 `/wsrm:SequenceFault/wsrm:Detail/@{any}`

350 The application specific error information related to the fault being described.

350 `/wsrm:SequenceFault/{any}`

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

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

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

350 4.2 Sequence Terminated

351 This fault is generated by either the RM Source or the RM Destination to indicate that it has either
352 encountered an unrecoverable condition, or has detected a violation of the protocol and as a
353 consequence, has chosen to terminate the Sequence. The endpoint that generates this fault should make
354 every reasonable effort to notify the corresponding endpoint of this decision.

351 Receipt of `SequenceTerminated` by either the RM Destination or the RM Source shall terminate the
352 Sequence if it is not otherwise terminated.

353 Properties:

353 [Code] Sender or Receiver

353 [Subcode] `wsrm:SequenceTerminated`

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

353 [Detail]

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

353 **4.3 Unknown Sequence**

354 This fault is generated by either the RM Source or the RM Destination in response to a message
355 containing an unknown or terminated Sequence identifier. Receipt of UnknownSequence by either the RM
356 Destination or the RM Source shall terminate the Sequence if it is not otherwise terminated.

357 Properties:

357 [Code] Sender

357 [Subcode] wsrn:UnknownSequence

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

357 [Detail]

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

357 **4.4 Invalid Acknowledgement**

358 This fault is generated by the RM Source in response to a <wsrn:SequenceAcknowledgement> that
359 violates the cumulative acknowledgement invariant. An example of such a violation would be a
360 SequenceAcknowledgement covering messages that have not been sent.

358 [Code] Sender

358 [Subcode] wsrn:InvalidAcknowledgement

358 [Reason] The SequenceAcknowledgement violates the cumulative acknowledgement invariant.

358 [Detail]

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

358 **4.5 Message Number Rollover**

359 This fault is generated to indicate that message numbers for a Sequence have been exhausted.

359 Properties:

359 [Code] Sender

359 [Subcode] wsrn:MessageNumberRollover

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

359 [Detail]

```
359 <wsrn:Identifier ...> xs:anyURI </wsrn:Identifier>  
360 <wsrn:MaxMessageNumber> wsrn:MessageNumberType </wsrn:MaxMessageNumber>
```

359 **4.6 Create Sequence Refused**

360 This fault is generated in response to a create Sequence request that cannot be satisfied.

360 Properties:

360 [Code] Sender

360 [Subcode] wsrn:CreateSequenceRefused

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

360 [Detail]

360 `xs:any`

360 4.7 Sequence Closed

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

362 This fault MUST be generated when an RM Destination is asked to receive a message for a Sequence
363 that is closed or when an RM Destination is asked to close a Sequence that is already closed.

364 Properties:

364 [Code] Sender

364 [Subcode] wsrn:SequenceClosed

364 [Reason] The Sequence is closed and can not receive new messages.

364 [Detail]

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

364 4.8 WSRM Required

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

365 Properties:

365 [Code] Sender

365 [Subcode] wsrn:WSRMRequired

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

365 [Detail]

365 `xs:any`

365 5 Security Considerations

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

366 Because Sequences are expected to exchange a number of messages, it is recommended that a security
367 context be established using the mechanisms described in WS-Trust[Trust] and WS-SecureConversation
368 [SecureConversation]. If a Sequence is bound to a specific destination, then the security context needs to
369 be established or shared with the destination servicing the Sequence. While the context can be
370 established at any time, it is critical that the messages establishing the Sequence be secured even if they
371 precede security context establishment. However, it is recommended that the security context be
372 established first. Security contexts are independent of reliable messaging Sequences. Consequently,
373 security contexts can come and go independent of the lifetime of the Sequence. In fact, it is
374 recommended that the lifetime of a security context be less than the lifetime of the Sequence unless the
375 Sequence is very short-lived.

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

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

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

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

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

- 366 • **Message alteration** – Alteration is prevented by including signatures of the message information
367 using WS-Security.
- 366 • **Message disclosure** – Confidentiality is preserved by encrypting sensitive data using WS-Security.

- 366 • **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by comparing
367 secured policies – see WS-Policy and WS-SecurityPolicy).
- 366 • **Authentication** – Authentication is established using the mechanisms described in WS-Security
367 and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- 366 • **Accountability** – Accountability is a function of the type of and string of the key and algorithms
367 being used. In many cases, a strong symmetric key provides sufficient accountability. However, in
368 some environments, strong PKI signatures are required.
- 366 • **Availability** – All reliable messaging services are subject to a variety of availability attacks. Replay
367 detection is a common attack and it is recommended that this be addressed by the mechanisms
368 described in WS-Security. (Note that because of legitimate message replays, detection should
369 include a differentiator besides message id such as a timestamp). Other attacks, such as network-
370 level denial of service attacks are harder to avoid and are outside the scope of this specification.
371 That said, care should be taken to ensure that minimal state is saved prior to any authenticating
372 Sequences.

366 **6 References**

367 **6.1 Normative**

368 **[KEYWORDS]**

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

368 **[SOAP 1.1]**

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

368 **[SOAP 1.2]**

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

368 **[URI]**

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

368 **[XML]**

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

368 **[XML-ns]**

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

368 **[XML-Schema Part1]**

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

368 **[XML-Schema Part2]**

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

368 **[WSDL 1.1]**

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

368 **[WS-Addressing]**

368 W3C Recommendation, "[Web Services Addressing 1.0 - Core](#)", May 2006.

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

370 **6.2 Non-Normative**

371 **[RDDL 2.0]**

371 Johnathan Borden, Tim Bray, eds. "[Resource Directory Description Language \(RDDL\) 2.0](#)," January 2004

371 **[WS-Policy]**

371 W3C Member Submission, "[Web Services Policy Framework \(WS-Policy\)](#)," April 2006.

372 **[WS-PolicyAttachment]**

372 W3C Member Submission, "[Web Services Policy Attachment \(WS-PolicyAttachment\)](#)," April 2006.

373 **[WS-Security]**

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

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

373 **[RTTM]**

373 V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", RFC 1323, May
374 1992.

373 **[SecurityPolicy]**

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

373 **[SecureConversation]**

373 S. Anderson, et al, "Web Services Secure Conversation Language (WS-SecureConversation)," February
374 2005.

373 **[Trust]**

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

373 **A. Schema**

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

373 <http://docs.oasis-open.org/ws-rx/wsrn/200604/wsrn-1.1-schema-200604.xsd>

373 The following copy is provided for reference.

```

373 <?xml version="1.0" encoding="UTF-8"?>
374 <!--
375 OASIS takes no position regarding the validity or scope of any intellectual
376 property or other rights that might be claimed to pertain to the
377 implementation or use of the technology described in this document or the
378 extent to which any license under such rights might or might not be available;
379 neither does it represent that it has made any effort to identify any such
380 rights. Information on OASIS's procedures with respect to rights in OASIS
381 specifications can be found at the OASIS website. Copies of claims of rights
382 made available for publication and any assurances of licenses to be made
383 available, or the result of an attempt made to obtain a general license or
384 permission for the use of such proprietary rights by implementors or users of
385 this specification, can be obtained from the OASIS Executive Director.
386 OASIS invites any interested party to bring to its attention any copyrights,
387 patents or patent applications, or other proprietary rights which may cover
388 technology that may be required to implement this specification. Please
389 address the information to the OASIS Executive Director.
390 Copyright © OASIS Open 2002-2006. All Rights Reserved.
391 This document and translations of it may be copied and furnished to others,
392 and derivative works that comment on or otherwise explain it or assist in its
393 implementation may be prepared, copied, published and distributed, in whole or
394 in part, without restriction of any kind, provided that the above copyright
395 notice and this paragraph are included on all such copies and derivative
396 works. However, this document itself does not be modified in any way, such as
397 by removing the copyright notice or references to OASIS, except as needed for
398 the purpose of developing OASIS specifications, in which case the procedures
399 for copyrights defined in the OASIS Intellectual Property Rights document must
400 be followed, or as required to translate it into languages other than English.
401 The limited permissions granted above are perpetual and will not be revoked by
402 OASIS or its successors or assigns.
403 This document and the information contained herein is provided on an "AS
404 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
405 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
406 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
407 FITNESS FOR A PARTICULAR PURPOSE.
408 -->
409 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
410 xmlns:wsa="http://www.w3.org/2005/08/addressing"
411 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200604"
412 targetNamespace="http://docs.oasis-open.org/ws-rx/wsm/200604"
413 elementFormDefault="qualified" attributeFormDefault="unqualified">
414   <xs:import namespace="http://www.w3.org/2005/08/addressing"
415   schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
416   <!-- Protocol Elements -->
417   <xs:complexType name="SequenceType">
418     <xs:sequence>
419       <xs:element ref="wsm:Identifier"/>
420       <xs:element name="MessageNumber" type="wsm:MessageNumberType"/>
421       <xs:any namespace="##other" processContents="lax" minOccurs="0"
422 maxOccurs="unbounded"/>
423     </xs:sequence>
424     <xs:anyAttribute namespace="##other" processContents="lax"/>
425   </xs:complexType>
426   <xs:element name="Sequence" type="wsm:SequenceType"/>
427   <xs:element name="SequenceAcknowledgement">
428     <xs:complexType>
429       <xs:sequence>
430         <xs:element ref="wsm:Identifier"/>
431         <xs:choice>
432           <xs:sequence>
433             <xs:choice>
434               <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
435                 <xs:complexType>

```

```

373         <xs:sequence/>
374         <xs:attribute name="Upper" type="xs:unsignedLong"
375 use="required"/>
376         <xs:attribute name="Lower" type="xs:unsignedLong"
377 use="required"/>
378         <xs:anyAttribute namespace="##other" processContents="lax"/>
379     </xs:complexType>
380 </xs:element>
381 <xs:element name="None" minOccurs="0">
382     <xs:complexType>
383         <xs:sequence/>
384     </xs:complexType>
385 </xs:element>
386 </xs:choice>
387 <xs:element name="Final" minOccurs="0">
388     <xs:complexType>
389         <xs:sequence/>
390     </xs:complexType>
391 </xs:element>
392 </xs:sequence>
393 <xs:element name="Nack" type="xs:unsignedLong"
394 maxOccurs="unbounded"/>
395 </xs:choice>
396 <xs:any namespace="##other" processContents="lax" minOccurs="0"
397 maxOccurs="unbounded"/>
398 </xs:sequence>
399 <xs:anyAttribute namespace="##other" processContents="lax"/>
400 </xs:complexType>
401 </xs:element>
402 <xs:complexType name="AckRequestedType">
403     <xs:sequence>
404         <xs:element ref="wsrm:Identifier"/>
405         <xs:any namespace="##other" processContents="lax" minOccurs="0"
406 maxOccurs="unbounded"/>
407     </xs:sequence>
408     <xs:anyAttribute namespace="##other" processContents="lax"/>
409 </xs:complexType>
410 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
411 <xs:element name="Identifier">
412     <xs:complexType>
413         <xs:annotation>
414             <xs:documentation>
415                 This type is for elements whose [children] is an anyURI and can have
416 arbitrary attributes.
417             </xs:documentation>
418         </xs:annotation>
419         <xs:simpleContent>
420             <xs:extension base="xs:anyURI">
421                 <xs:anyAttribute namespace="##other" processContents="lax"/>
422             </xs:extension>
423         </xs:simpleContent>
424     </xs:complexType>
425 </xs:element>
426 <xs:simpleType name="MessageNumberType">
427     <xs:restriction base="xs:unsignedLong">
428         <xs:minInclusive value="1"/>
429         <xs:maxInclusive value="9223372036854775807"/>
430     </xs:restriction>
431 </xs:simpleType>
432 <!-- Fault Container and Codes -->
433 <xs:simpleType name="FaultCodes">
434     <xs:restriction base="xs:QName">
435         <xs:enumeration value="wsrm:SequenceTerminated"/>

```

```

373     <xs:enumeration value="wsrm:UnknownSequence"/>
374     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
375     <xs:enumeration value="wsrm:MessageNumberRollover"/>
376     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
377     <xs:enumeration value="wsrm:SequenceClosed"/>
378     <xs:enumeration value="wsrm:WSRMRequired"/>
379   </xs:restriction>
380 </xs:simpleType>
381 <xs:complexType name="SequenceFaultType">
382   <xs:sequence>
383     <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
384     <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
385     <xs:any namespace="##other" processContents="lax" minOccurs="0"
386 maxOccurs="unbounded"/>
387   </xs:sequence>
388   <xs:anyAttribute namespace="##other" processContents="lax"/>
389 </xs:complexType>
390 <xs:complexType name="DetailType">
391   <xs:sequence>
392     <xs:any namespace="##other" processContents="lax" minOccurs="0"
393 maxOccurs="unbounded"/>
394   </xs:sequence>
395   <xs:anyAttribute namespace="##other" processContents="lax"/>
396 </xs:complexType>
397 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
398 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
399 <xs:element name="CreateSequenceResponse"
400 type="wsrm:CreateSequenceResponseType"/>
401 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
402 <xs:element name="CloseSequenceResponse"
403 type="wsrm:CloseSequenceResponseType"/>
404 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
405 <xs:element name="TerminateSequenceResponse"
406 type="wsrm:TerminateSequenceResponseType"/>
407 <xs:complexType name="CreateSequenceType">
408   <xs:sequence>
409     <xs:element ref="wsrm:AcksTo"/>
410     <xs:element ref="wsrm:Expires" minOccurs="0"/>
411     <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
412     <xs:any namespace="##other" processContents="lax" minOccurs="0"
413 maxOccurs="unbounded">
414       <xs:annotation>
415         <xs:documentation>
416           It is the authors intent that this extensibility be used to
417 transfer a Security Token Reference as defined in WS-Security.
418         </xs:documentation>
419       </xs:annotation>
420     </xs:any>
421   </xs:sequence>
422   <xs:anyAttribute namespace="##other" processContents="lax"/>
423 </xs:complexType>
424 <xs:complexType name="CreateSequenceResponseType">
425   <xs:sequence>
426     <xs:element ref="wsrm:Identifier"/>
427     <xs:element ref="wsrm:Expires" minOccurs="0"/>
428     <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>

```

```

373     <xs:element name="IncompleteSequenceBehaviour"
374 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
375     <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
376     <xs:any namespace="##other" processContents="lax" minOccurs="0"
377 maxOccurs="unbounded"/>
378   </xs:sequence>
379   <xs:anyAttribute namespace="##other" processContents="lax"/>
380 </xs:complexType>
381 <xs:complexType name="CloseSequenceType">
382   <xs:sequence>
383     <xs:element ref="wsrm:Identifier"/>
384     <xs:any namespace="##other" processContents="lax" minOccurs="0"
385 maxOccurs="unbounded"/>
386   </xs:sequence>
387   <xs:anyAttribute namespace="##other" processContents="lax"/>
388 </xs:complexType>
389 <xs:complexType name="CloseSequenceResponseType">
390   <xs:sequence>
391     <xs:element ref="wsrm:Identifier"/>
392     <xs:any namespace="##other" processContents="lax" minOccurs="0"
393 maxOccurs="unbounded"/>
394   </xs:sequence>
395   <xs:anyAttribute namespace="##other" processContents="lax"/>
396 </xs:complexType>
397 <xs:complexType name="TerminateSequenceType">
398   <xs:sequence>
399     <xs:element ref="wsrm:Identifier"/>
400     <xs:any namespace="##other" processContents="lax" minOccurs="0"
401 maxOccurs="unbounded"/>
402   </xs:sequence>
403   <xs:anyAttribute namespace="##other" processContents="lax"/>
404 </xs:complexType>
405 <xs:complexType name="TerminateSequenceResponseType">
406   <xs:sequence>
407     <xs:element ref="wsrm:Identifier"/>
408     <xs:any namespace="##other" processContents="lax" minOccurs="0"
409 maxOccurs="unbounded"/>
410   </xs:sequence>
411   <xs:anyAttribute namespace="##other" processContents="lax"/>
412 </xs:complexType>
413 <xs:element name="AcksTo"

```

```

373     type="wsa:EndpointReferenceType"/>
374     <xs:complexType name="OfferType">
375       <xs:sequence>
376         <xs:element ref="wsrm:Identifier"/>
377         <xs:element ref="wsrm:Expires" minOccurs="0"/>
378         <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
379         <xs:any namespace="##other" processContents="lax" minOccurs="0"
380 maxOccurs="unbounded"/>
381       </xs:sequence>
382       <xs:anyAttribute namespace="##other" processContents="lax"/>
383     </xs:complexType>
384     <xs:complexType name="AcceptType">
385       <xs:sequence>
386         <xs:element ref="wsrm:AcksTo"/>
387         <xs:any namespace="##other" processContents="lax" minOccurs="0"
388 maxOccurs="unbounded"/>
389       </xs:sequence>
390       <xs:anyAttribute namespace="##other" processContents="lax"/>
391     </xs:complexType>
392     <xs:element name="Expires">
393       <xs:complexType>
394         <xs:simpleContent>
395           <xs:extension base="xs:duration">
396             <xs:anyAttribute namespace="##other" processContents="lax"/>
397           </xs:extension>
398         </xs:simpleContent>
399       </xs:complexType>
400     </xs:element>
401     <xs:element name="AcknowledgementInterval">
402       <xs:complexType>
403         <xs:sequence/>
404         <xs:attribute name="Milliseconds" type="xs:unsignedLong"
405 use="required"/>
406         <xs:anyAttribute namespace="##other" processContents="lax"/>
407       </xs:complexType>
408     </xs:element>
373     <xs:simpleType name="IncompleteSequenceBehaviorType">
374       <xs:restriction base="xs:string">
375         <xs:enumeration value="DiscardEntireSequence"/>
376         <xs:enumeration value="DiscardFollowingFirstGap"/>
377         <xs:enumeration value="NoDiscard"/>
378       </xs:restriction>
379     </xs:simpleType>
380 </xs:schema>

```

373 B. Message Examples

373 B.1 Create Sequence

373 Create Sequence

```
373 <?xml version="1.0" encoding="UTF-8"?>
373 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
373 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
373 xmlns:wsa="http://www.w3.org/2005/08/addressing">
373   <S:Header>
373     <wsa:MessageID>
373       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546817
373     </wsa:MessageID>
373     <wsa:To>http://example.com/serviceB/123</wsa:To>
373     <wsa:Action>http://docs.oasis-open.org/ws-
374 rx/wsmr/200604/CreateSequence</wsa:Action>
373     <wsa:ReplyTo>
373       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
373     </wsa:ReplyTo>
373   </S:Header>
373   <S:Body>
373     <wsmr:CreateSequence>
373       <wsmr:AcksTo>
373         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
373       </wsmr:AcksTo>
373     </wsmr:CreateSequence>
373   </S:Body>
373 </S:Envelope>
```

373 Create Sequence Response

```
373 <?xml version="1.0" encoding="UTF-8"?>
373 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
374 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
375 xmlns:wsa="http://www.w3.org/2005/08/addressing">
373   <S:Header>
373     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
373     <wsa:RelatesTo>
373       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8a7c2eb546817
373     </wsa:RelatesTo>
373     <wsa:Action>
373       http://docs.oasis-open.org/ws-rx/wsmr/200604/CreateSequenceResponse
373     </wsa:Action>
373   </S:Header>
373   <S:Body>
373     <wsmr:CreateSequenceResponse>
373       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
373     </wsmr:CreateSequenceResponse>
373   </S:Body>
373 </S:Envelope>
```

373 B.2 Initial Transmission

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

373 **Message 1**

```
373 <?xml version="1.0" encoding="UTF-8"?>
373 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
373 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
373 xmlns:wsa="http://www.w3.org/2005/08/addressing">
373   <S:Header>
373     <wsa:MessageID>
373       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
373     </wsa:MessageID>
373     <wsa:To>http://example.com/serviceB/123</wsa:To>
373     <wsa:From>
373       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
373     </wsa:From>
373     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
373     <wsmr:Sequence>
373       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
373       <wsmr:MessageNumber>1</wsmr:MessageNumber>
373     </wsmr:Sequence>
373   </S:Header>
373   <S:Body>
373     <!-- Some Application Data -->
373   </S:Body>
373 </S:Envelope>
```

373 **Message 2**

```
373 <?xml version="1.0" encoding="UTF-8"?>
373 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
373 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
373 xmlns:wsa="http://www.w3.org/2005/08/addressing">
373   <S:Header>
373     <wsa:MessageID>
373       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
373     </wsa:MessageID>
373     <wsa:To>http://example.com/serviceB/123</wsa:To>
373     <wsa:From>
373       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
373     </wsa:From>
373     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
373     <wsmr:Sequence>
373       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
373       <wsmr:MessageNumber>2</wsmr:MessageNumber>
373     </wsmr:Sequence>
373   </S:Header>
373   <S:Body>
373     <!-- Some Application Data -->
373   </S:Body>
373 </S:Envelope>
```

373 **Message 3**

```
373 <?xml version="1.0" encoding="UTF-8"?>
373 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
373 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
373 xmlns:wsa="http://www.w3.org/2005/08/addressing">
373   <S:Header>
373     <wsa:MessageID>
373       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
373     </wsa:MessageID>
373     <wsa:To>http://example.com/serviceB/123</wsa:To>
373     <wsa:From>
373       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
```

```

373 </wsa:From>
373 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
373 <wsrm:Sequence>
373 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
373 <wsrm:MessageNumber>3</wsrm:MessageNumber>
373 </wsrm:Sequence>
373 <wsrm:AckRequested>
373 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
373 </wsrm:AckRequested>
373 </S:Header>
373 <S:Body>
373 <!-- Some Application Data -->
373 </S:Body>
373 </S:Envelope>

```

373 B.3 First Acknowledgement

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

```

373 <?xml version="1.0" encoding="UTF-8"?>
373 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
373 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
373 xmlns:wsa="http://www.w3.org/2005/08/addressing">
374 <S:Header>
375 <wsa:MessageID>
376 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
377 </wsa:MessageID>
378 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
379 <wsa:From>
380 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
381 </wsa:From>
382 <wsa:Action>
383 http://docs.oasis-open.org/ws-rx/wsrn/200604/SequenceAcknowledgement
384 </wsa:Action>
385 <wsrm:SequenceAcknowledgement>
386 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
387 <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
388 <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
389 </wsrm:SequenceAcknowledgement>
390 </S:Header>
391 <S:Body/>
392 </S:Envelope>

```

393 B.4 Retransmission

394 The RM Sourcediscovers that message number 2 was not received so it resends the message and
395 requests an acknowledgement:

```

396 <?xml version="1.0" encoding="UTF-8"?>
397 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
398 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
399 xmlns:wsa="http://www.w3.org/2005/08/addressing">
400 <S:Header>
401 <wsa:MessageID>
402 http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
403 </wsa:MessageID>
404 <wsa:To>http://example.com/serviceB/123</wsa:To>
405 <wsa:From>
406 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
407 </wsa:From>

```

```

408 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
409 <wsrm:Sequence>
410 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
411 <wsrm:MessageNumber>2</wsrm:MessageNumber>
412 </wsrm:Sequence>
413 <wsrm:AckRequested>
414 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
415 </wsrm:AckRequested>
416 </S:Header>
417 <S:Body>
418 <!-- Some Application Data -->
419 </S:Body>
420 </S:Envelope>

```

421 B.5 Termination

422 The RM Destination now responds with an acknowledgement for the complete Sequence which can then
423 be terminated:

```

424 <?xml version="1.0" encoding="UTF-8"?>
425 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
426 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
427 xmlns:wsa="http://www.w3.org/2005/08/addressing">
428 <S:Header>
429 <wsa:MessageID>
430 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
431 </wsa:MessageID>
432 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
433 <wsa:From>
434 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
435 </wsa:From>
436 <wsa:Action>
437 http://docs.oasis-open.org/ws-rx/wsmr/200604/SequenceAcknowledgement
438 </wsa:Action>
439 <wsmr:SequenceAcknowledgement>
440 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
441 <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
442 </wsmr:SequenceAcknowledgement>
443 </S:Header>
444 <S:Body/>
445 </S:Envelope>

```

446 Terminate Sequence

```

447 <?xml version="1.0" encoding="UTF-8"?>
448 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
449 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
450 xmlns:wsa="http://www.w3.org/2005/08/addressing">
451 <S:Header>
452 <wsa:MessageID>
453 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
454 </wsa:MessageID>
455 <wsa:To>http://example.com/serviceB/123</wsa:To>
456 <wsa:Action>
457 http://docs.oasis-open.org/ws-rx/wsmr/200604/TerminateSequence
458 </wsa:Action>
459 <wsa:From>
460 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
461 </wsa:From>
462 </S:Header>
463 <S:Body>
464 <wsmr:TerminateSequence>

```

```
465     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
466     </wsrm:TerminateSequence>
467     </S:Body>
468 </S:Envelope>
```

469 Terminate Sequence Response

```
470 <?xml version="1.0" encoding="UTF-8"?>
471 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
472 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
473 xmlns:wsa="http://www.w3.org/2005/08/addressing">
474   <S:Header>
475     <wsa:MessageID>
476       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
477     </wsa:MessageID>
478     <wsa:To>http://example.com/serviceA/789</wsa:To>
479     <wsa:Action>
480       http://docs.oasis-open.org/ws-rx/wsr/200604/TerminateSequenceResponse
481     </wsa:Action>
482     <wsa:RelatesTo>
483       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
484     </wsa:RelatesTo>
485     <wsa:From>
486       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
487     </wsa:From>
488   </S:Header>
489   <S:Body>
490     <wsrm:TerminateSequenceResponse>
491       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
492     </wsrm:TerminateSequenceResponse>
492   </S:Body>
492 </S:Envelope>
```

492 C. WSDL

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

493 <http://docs.oasis-open.org/ws-rx/wsrn/200604/wsd/wsrn-1.1-wsd-200604.wsd>

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

```

493 <?xml version="1.0" encoding="utf-8"?>
494 <!--
495 OASIS takes no position regarding the validity or scope of any intellectual
496 property or other rights that might be claimed to pertain to the
497 implementation or use of the technology described in this document or the
498 extent to which any license under such rights might or might not be available;
499 neither does it represent that it has made any effort to identify any such
500 rights. Information on OASIS's procedures with respect to rights in OASIS
501 specifications can be found at the OASIS website. Copies of claims of rights
502 made available for publication and any assurances of licenses to be made
503 available, or the result of an attempt made to obtain a general license or
504 permission for the use of such proprietary rights by implementors or users of
505 this specification, can be obtained from the OASIS Executive Director.
506 OASIS invites any interested party to bring to its attention any copyrights,
507 patents or patent applications, or other proprietary rights which may cover
508 technology that may be required to implement this specification. Please
509 address the information to the OASIS Executive Director.
510 Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
511 This document and translations of it may be copied and furnished to others,
512 and derivative works that comment on or otherwise explain it or assist in its
513 implementation may be prepared, copied, published and distributed, in whole or
514 in part, without restriction of any kind, provided that the above copyright
515 notice and this paragraph are included on all such copies and derivative
516 works. However, this document itself does not be modified in any way, such as
517 by removing the copyright notice or references to OASIS, except as needed for
518 the purpose of developing OASIS specifications, in which case the procedures
519 for copyrights defined in the OASIS Intellectual Property Rights document must
520 be followed, or as required to translate it into languages other than English.
521 The limited permissions granted above are perpetual and will not be revoked by
522 OASIS or its successors or assigns.
523 This document and the information contained herein is provided on an "AS
524 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
525 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
526 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
527 FITNESS FOR A PARTICULAR PURPOSE.
528 -->
529 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
530 xmlns:xs="http://www.w3.org/2001/XMLSchema"
531 xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
532 open.org/ws-rx/wsrn/200604" xmlns:tns="http://docs.oasis-open.org/ws-
533 rx/wsrn/200604/wsdl" targetNamespace="http://docs.oasis-open.org/ws-
534 rx/wsrn/200604/wsdl">
535
536   <wsdl:types>
537     <xs:schema
538       <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsrn/200604"
539       schemaLocation="http://docs.oasis-open.org/ws-rx/wsrn/200604/wsrn-1.1-schema-
540       200604.xsd"/>
541     </xs:schema>
542   </wsdl:types>
543
544   <wsdl:message name="CreateSequence">
545     <wsdl:part name="create" element="rm:CreateSequence"/>
546   </wsdl:message>
547   <wsdl:message name="CreateSequenceResponse">
548     <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
549   </wsdl:message>
550   <wsdl:message name="CloseSequence">
551     <wsdl:part name="close" element="rm:CloseSequence"/>
552   </wsdl:message>
553   <wsdl:message name="CloseSequenceResponse">
554     <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
555   </wsdl:message>

```

```

493 <wsdl:message name="TerminateSequence">
494 <wsdl:part name="terminate" element="rm:TerminateSequence"/>
495 </wsdl:message>
496 <wsdl:message name="TerminateSequenceResponse">
497 <wsdl:part name="terminateResponse"
498 element="rm:TerminateSequenceResponse"/>
499 </wsdl:message>

500 <wsdl:portType name="SequenceAbstractPortType">
501 <wsdl:operation name="CreateSequence">
502 <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
503 open.org/ws-rx/wsrn/200604/CreateSequence"/>
504 <wsdl:output message="tns:CreateSequenceResponse"
505 wsa:Action="http://docs.oasis-open.org/ws-
506 rx/wsrn/200604/CreateSequenceResponse"/>
507 </wsdl:operation>
508 <wsdl:operation name="CloseSequence">
509 <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
510 open.org/ws-rx/wsrn/200604/CloseSequence"/>
511 <wsdl:output message="tns:CloseSequenceResponse"
512 wsa:Action="http://docs.oasis-open.org/ws-
513 rx/wsrn/200604/CloseSequenceResponse"/>
514 </wsdl:operation>
515 <wsdl:operation name="TerminateSequence">
516 <wsdl:input message="tns:TerminateSequence"
517 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequence"/>
518 <wsdl:output message="tns:TerminateSequenceResponse"
519 wsa:Action="http://docs.oasis-open.org/ws-
520 rx/wsrn/200604/TerminateSequenceResponse"/>
521 </wsdl:operation>
522 </wsdl:portType>

523 </wsdl:definitions>

```

493 D. State Tables

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

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

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

493 Table 2 RM Source State Transition Table

Events	States							
	None	Connecting	Connected	Rollover	Closing	Closed	Terminating	Terminated
Create Sequence	<i>Transmit Create Sequence</i> Connecting	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Create Sequence Response	N/A	No action Connected	N/A	N/A	N/A	N/A	N/A	N/A
Create Sequence Refused Fault	N/A	No action Terminated	N/A	N/A	N/A	N/A	N/A	N/A
New Message	N/A	N/A	<i>Transmit message</i> Connected	<i>no action</i> Rollover	<i>No action</i> Closing	N/A	N/A	N/A
Retransmit of unack message	N/A	N/A	<i>Transmit message</i> Connected	<i>Transmit message</i> Rollover	<i>Trasmit message?</i> Closing	<i>No action</i> Closed	N/A	N/A
SeqAck (non-final)	N/A	N/A	<i>Process Ack ranges</i> Connected	<i>Process Ack ranges</i> Rollover	<i>Process Ack ranges</i> Closing	<i>Process Ack ranges</i> Closed	<i>Process Ack ranges</i> Terminating	<i>Transmit Unknown Sequence Fault</i> Terminated
Nack	N/A	N/A	<i>Transmit message(s)</i> Connected	<i>Transmit message(s)</i> Rollover	<i>Transmit message(s)</i> Closing	<i>No action</i> Closed	<i>No action</i> Terminating	<i>Transmit Unknown Sequence fault</i> Terminated
Reached max msg number	N/A	N/A	<i>No action</i> Rollover	<i>No action</i> Rollover	N/A	N/A	N/A	N/A

Events	States							
	None	Connecting	Connected	Rollover	Closing	Closed	Terminating	Terminated
Message Number Rollover Fault	N/A	N/A	No action Rollover	No action Rollover	No action Closing	No action Closed	No action Terminating	Transmit Unknown Sequence Fault Terminated
Close Sequence	N/A	N/A	Transmit Close Sequence Closing	Transmit Close Sequence Closing	Transmit Close Sequence Closing	No action Closed	No action Terminating	N/A
Close Sequence Response	N/A	N/A	N/A	N/A	No action Closed	No action Closed	No action Terminating	Transmit Unknown Sequence Fault Terminated
SeqAck (final)	N/A	N/A	Process Ack/Nack ranges Closed	Process Ack/Nack ranges Closed	Process Ack/Nack ranges Closed	Process Ack/Nack ranges Closed	Process Ack/Nack ranges Terminating	Transmit Unknown Sequence fault Terminated
Sequence Closed Fault	N/A	N/A	No action Closed	No action Closed	No action Closed	No action Closed	No action Terminating	Transmit Unknown Sequence Fault Terminated
Unknown Sequence Fault	N/A	N/A	No action Terminated					
Sequence Terminated Fault	N/A	N/A	No action Terminated					
Terminate Sequence	N/A	N/A	Transmit Terminate Sequence Terminating	N/A				
Terminate Sequence Response	N/A	N/A	N/A	N/A	N/A	N/A	No action Terminated	No action Terminated
Elapse Expires duration	N/A	N/A	Send Sequence Terminated Fault Terminated	N/A				

493 In Table 2 above, the rows consists of events that occur at the RM Source throughout the lifetime of an
494 RM Sequence and the columns consists of various RM Source states. Each cell in the table above lists

495 the action that the RM Source takes on occurrence of a particular event and the next state that it
 496 transitions.

497 Table 3 RM Destination State Transition Table

Events	States						
	None	Connecting	Connected			Closed	Terminated
Creation request not satisfied	N/A	<i>Send Create Sequence Refused Fault</i> Terminated	N/A			N/A	
Message (with message number within range)	N/A	N/A	<i>No action</i> Connected			<i>Send Sequence Closed Fault (with SeqAck+Final)</i> Closed	<i>Send Unknown Seq Fault</i> Terminated
Ack requested	N/A	N/A	<i>Send SequenceAck</i> Connected			<i>Send SeqAck+Final</i> Closed	<i>Send Unknown Seq Fault</i> Terminated
Message (with message number outside of range)	N/A	N/A	<i>Send Message Number Rollover Fault</i> Connected			N/A	N/A
Close Sequence	N/A	N/A	<i>Send CloseSequenceResponse with SequenceAck (Final)</i> Closed			<i>Send Close Sequence Response with SeqAck+Final</i> Closed	<i>Send Unknown Sequence Fault</i> Terminated
Close Sequence itself	N/A	N/A	Closed			<i>Send Sequence Closed Fault</i> Closed	N/A
Terminate Sequence	N/A	N/A	<i>Send Terminate Sequence Response</i> Terminated			<i>Send Terminate Sequence Response</i> Terminated	<i>Send Unknown Sequence Fault</i> Terminated

Events	States						
	None	Connecting	Connected			Closed	Terminated
Unknown Sequence Fault	N/A	N/A	<i>No action</i> Terminated			<i>No action</i> Terminated	<i>No action</i> Terminated
Sequence Terminated Fault	N/A	N/A	<i>No action</i> Terminated			<i>No action</i> Terminated	<i>No action</i> Terminated
EIapse Expires duration	N/A	N/A	<i>Send Sequence Terminated Fault</i> Terminated			<i>Send Sequence Terminated Fault</i> Terminated	N/A

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

501 E. Acknowledgments

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

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

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

501 Keith Ballinger, Microsoft, Stefan Batres, Microsoft, Allen Brown, Microsoft, Michael Conner, IBM,
502 George Copeland, Microsoft, Francisco Curbera, IBM, Paul Fremantle, IBM, Steve Graham, IBM,
503 Pat Helland, Microsoft, Rick Hill, Microsoft, Scott Hinkelman, IBM, Tim Holloway, IBM, Efim Hudis,
504 Microsoft, Gopal Kakivaya, Microsoft, Johannes Klein, Microsoft, Frank Leymann, IBM, Martin
505 Nally, IBM, Peter Niblett, IBM, Jeffrey Schlimmer, Microsoft, James Snell, IBM, Keith Stobie,
506 Microsoft, Satish Thatte, Microsoft, Stephen Todd, IBM, Sanjiva Weerawarana, IBM, Roger
507 Wolter, Microsoft.

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

501 *TBD*

F. Revision History

Rev	Date	By Whom	What
wd-01	2005-07-07	Christopher Ferris	Initial version created based on submission by the authors.
ws-02	2005-07-21	Doug Davis	I011 (PT0S) added
wd-02	2005-08-16	Anish Karmarkar	Trivial editorial changes
ws-03	2005-09-15	Doug Davis	I019 and i028 (CloseSeq) added
wd-05	2005-09-26	Gilbert Pilz	i005 (Source resend of nacks messages when ack already received) added.
wd-05	2005-09-27	Doug Davis	i027 (InOrder delivery assurance spanning multiple sequences) added
wd-05	2005-09-27	Doug Davis	i020 (Semantics of "At most once" Delivery Assurance) added
wd-05	2005-09-27	Doug Davis	i034 (Fault while processing a piggy-backed RM header) added
wd-05	2005-09-27	Doug Davis	i033 (Processing model of NACKs) added
wd-05	2005-09-27	Doug Davis	i031 (AckRequested schema inconsistency) added
wd-05	2005-09-27	Doug Davis	i025 (SeqAck/None) added
wd-05	2005-09-27	Doug Davis	i029 (Remove dependency on WS-Security) added
wd-05	2005-09-27	Doug Davis	i039 (What does 'have a mU attribute' mean) added
wd-05	2005-09-27	Doug Davis	i040 (Change 'optiona'/'required' to 'OPTIONAL'/'REQUIRED') added
wd-05	2005-09-30	Anish Karmarkar	i017 (Change NS to http://docs.oasis-open.org/wsrn/200510/)
wd-05	2005-09-30	Anish Karmarkar	i045 (Include SecureConversation as a reference and move it to non-normative citation)
wd-05	2005-09-30	Anish Karmarkar	i046 (change the type of wsrn:FaultCode element)
wd-06	2005-11-02	Gilbert Pilz	Start wd-06 by changing title page from cd-01.
wd-06	2005-11-03	Gilbert Pilz	i047 (Reorder spec sections)
wd-07	2005-11-17	Gilbert Pilz	Start wd-07
wd-07	2005-11-28	Doug Davis	i071 – except for period in Appendix headings
wd-07	2005-11-28	Doug Davis	i10
wd-07	2005-11-28	Doug Davis	i030
wd-07	2005-11-28	Doug Davis	i037
wd-07	2005-11-28	Doug Davis	i038
wd-07	2005-11-28	Doug Davis	i041
wd-07	2005-11-28	Doug Davis	i043
wd-07	2005-11-28	Doug Davis	i044

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

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

Rev	Date	By Whom	What
wd-12	2006-03-20	Doug Davis	Added space in "SOAP1.x" – PaulCotton
wd-12	2006-04-11	Doug Davis	Issue 007 applied
wd-12	2006-04-11	Doug Davis	Issue 090 applied
wd-12	2006-04-11	Doug Davis	Issue 098 applied
wd-12	2006-04-11	Doug Davis	Issue 099 applied
wd-12	2006-04-11	Doug Davis	Issue 101 applied
wd-12	2006-04-11	Doug Davis	Issue 103 applied
wd-12	2006-04-11	Doug Davis	Issue 104 applied
wd-12	2006-04-11	Doug Davis	Issue 105 applied
wd-12	2006-04-11	Doug Davis	Issue 107 applied
wd-12	2006-04-11	Doug Davis	Issue 109 applied
wd-12	2006-04-11	Doug Davis	Issue 110 applied
wd-12	2006-04-12	Doug Davis	Used "generated" instead of "issue" or "send" when talking about faults.
wd-12	2006-04-24	Gilbert Pilz	Update references to WS-Addressing to the Proposed Recommendations; update WS-RM namespace to "200604".
wd-13	2006-05-08	Gilbert Pilz	i093 part 1; more work needed
wd-13	2006-05-10	Doug Davis	Issue 096 applied
wd-13	2006-05-26	Gilbert Pilz	i093 part 2; reflects decisions from 2006-05-25 meeting
wd-13	2006-05-28	Gilbert Pilz	Issue 106 applied
wd-13	2006-05-29	Gilbert Pilz	Issue 118 applied
wd-13	2006-05-29	Gilbert Pilz	Issue 120 applied
wd-13	2006-05-30	Gilbert Pilz	Issue 114 applied
wd-13	2006-05-30	Gilbert Pilz	Issue 116 applied
wd-14	2006-06-05	Gilbert Pilz	Accept all changes; bump WD number
wd-14	2006-06-07	Doug Davis	Applied lots of minor edits from Marc Goodner
wd-14	2006-06-07	Doug Davis	Change a couple of period/sp/sp to period/sp
wd-14	2006-06-07	Doug Davis	Added a space in "URI]of" – per Marc Goodner
wd-14	2006-06-07	Doug Davis	Issue 131 applied
wd-14	2006-06-07	Doug Davis	Issue 132 applied
wd-14	2006-06-07	Doug Davis	Issue 119 applied
wd-14	2006-06-07	Doug Davis	Applied lots of minor edits from Doug Davis

501 G. Notices

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

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

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

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

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

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