



1 Web Services Reliable Messaging 2 (WS-ReliableMessaging)

3 Working Draft 1~~3~~⁴, June 12, 2006

4 Document identifier:

5 wsrn-1.1-spec-wd-1~~3~~⁴

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

13 Contributors:

14 TBD

15 Abstract:

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

Table of Contents

33		
34	1 Introduction.....	4
35	1.1 Notational Conventions.....	4
36	1.2 Namespace.....	4
37	1.3 Compliance.....	5
38	2 Reliable Messaging Model.....	6
39	2.1 Glossary.....	6
40	2.2 Protocol Preconditions.....	7
41	2.3 Protocol Invariants.....	7
42	2.4 Example Message Exchange.....	7
43	3 RM Protocol Elements.....	10
44	3.1 Sequence Creation.....	10
45	3.2 Closing A Sequence.....	14
46	3.3 Sequence Termination.....	16
47	3.4 Sequences.....	17
48	3.5 Request Acknowledgement.....	18
49	3.6 Sequence Acknowledgement.....	19
50	3.7 MakeConnection.....	21
51	4 Faults.....	24
52	4.1 SequenceFault Element.....	25
53	4.2 Sequence Terminated.....	26
54	4.3 Unknown Sequence.....	27
55	4.4 Invalid Acknowledgement.....	27
56	4.5 Message Number Rollover.....	27
57	4.6 Create Sequence Refused.....	27
58	4.7 Sequence Closed.....	28
59	4.8 WSRM Required.....	28
60	4.9 UnsupportedSelection	28
61	5 Security Considerations.....	29
62	6 References.....	31
63	6.1 Normative.....	31
64	6.2 Non-Normative.....	31
65	A. Schema.....	33
66	B. Message Examples.....	39
67	B.1 Create Sequence.....	39
68	B.2 Initial Transmission.....	39
69	B.3 First Acknowledgement.....	41
70	B.4 Retransmission.....	41

71	B.5 Termination.....	42
72	C. WSDL.....	44
73	D. State Tables.....	47
74	E. Acknowledgments.....	51
75	F. Revision History.....	52
76	G. Notices.....	56

1 Introduction

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

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

1.1 Goals and Requirements

1.1.1 Requirements

1.2 Notational Conventions

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

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

- The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- Characters are appended to elements and attributes to indicate cardinality:
 - "?" (0 or 1)
 - "*" (0 or more)
 - "+" (1 or more)
- The character "|" is used to indicate a choice between alternatives.
- The characters "[" and "]" are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content specified in this document. Additional children elements and/or attributes MAY be added at the indicated extension points but they MUST NOT contradict the semantics of the parent and/or owner, respectively. If an extension is not recognized it SHOULD be ignored.
- XML namespace prefixes (See Section [Namespace1.2](#)) are used to indicate the namespace of the element being defined.

1.3 Namespace

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

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

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

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

115 [The following namespaces are used in this document:](#)

116 *Table 1*

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

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

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

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

120 **1.4 Compliance**

121 An implementation is not compliant with this specification if it fails to satisfy one or more of the MUST or
122 REQUIRED level requirements defined herein. A SOAP Node MUST NOT use the XML namespace
123 identifier for this specification (listed in Section [Namespace1.2](#)) within SOAP Envelopes unless it is
124 compliant with this specification.

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

2 Reliable Messaging Model

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

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

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

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

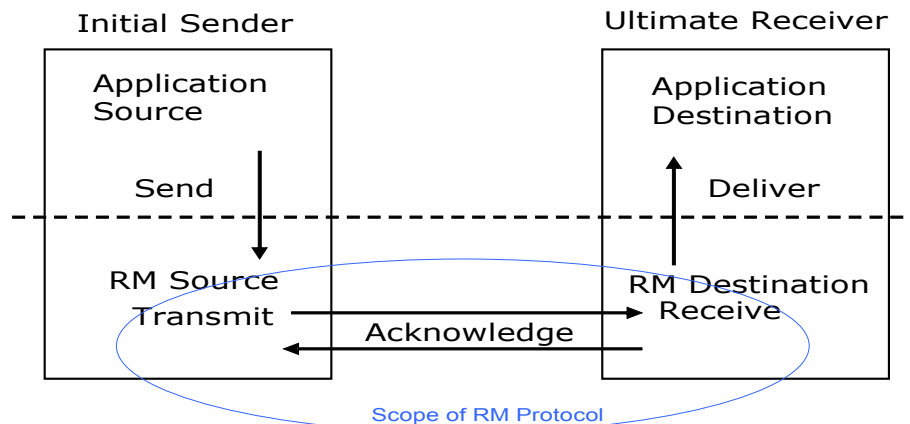


Figure 1: Reliable Messaging Model

2.1 Glossary

The following definitions are used throughout this specification:

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

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

159 **Application Source:** The endpoint that ~~s~~Sends a message.

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

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

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

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

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

169 **Send:** The act of submitting a message to the RM Source for reliable ~~transfer~~delivery. ~~The reliability~~
170 ~~guarantee begins at this point.~~

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

172 2.2 Protocol Preconditions

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

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

179 If a secure exchange of messages is ~~required~~REQUIRED, then the RM Source and RM Destination
180 MUST have a security context.

181 2.3 Protocol Invariants

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

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

190 2.4 Example Message Exchange

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



Figure 2: The WS-ReliableMessaging Protocol

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

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

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

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

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

204 Now that the basic model has been outlined, the details of the elements used in this protocol are now
205 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 [\[WS-Addressing\]](#), 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 [\[WS-Addressing\]](#)). It specifies the endpoint reference to which messages containing `<wsrm:SequenceAcknowledgement>` header blocks

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

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

229 /wsrm:CreateSequence/wsrm:Expires

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

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

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

232 /wsrm:CreateSequence/wsrm:Offer

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

233 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier

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

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

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

239 /wsrm:CreateSequence/wsrm:Offer/wsrm:Endpoint

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

243 Implementations MUST NOT use an endpoint reference in the Endpoint element that would prevent the
244 sending of WS-RM protocol messages. For example, using the WS-Addressing
245 "http://www.w3.org/2005/08/addressing/none" IRI would make it impossible for the RM Destination to ever
246 send WS-RM protocol messages (e.g. wsrm:TerminateSequence) to the RM Source for the Offered
247 Sequence. Implementations MAY use the WS-RM anonymous URI template and doing so implies that
248 messages will be retrieved using a mechanism such as the wsrm:MakeConnection message (see section
249 3.7).

250 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires

250 This element, if present, of type `xs:duration` specifies the duration for the [offered](#) Sequence. A value of
251 'PT0S' indicates that the [offered](#) Sequence will never expire. Absence of the element indicates an implied
252 value of 'PT0S'.

253 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires/@{any}

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

253 /wsrm:CreateSequence/wsrm:Offer/{any}

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

253 /wsrm:CreateSequence/wsrm:Offer/@{any}

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

253 /wsrm:CreateSequence/{any}

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

253 /wsrm:CreateSequence/@{any}

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

253 A <wsrm:CreateSequenceResponse> is sent in the body of a response message by an RM
254 Destination in response to receipt of a <wsrm:CreateSequence> request message. It carries the
255 <wsrm:Identifier> of the created Sequence and indicates that the RM Source [maycan](#) begin sending
256 messages in the context of the identified Sequence.

257 The following exemplar defines the <wsrm:CreateSequenceResponse> syntax:

```
257 <wsrm:CreateSequenceResponse ...>
257   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
257   <wsrm:Expires> xs:duration </wsrm:Expires> ?
257   <wsrm:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?
257   <wsrm:IncompleteSequenceBehavior> wsrm:IncompleteSequenceBehaviorType
258 </wsrm:IncompleteSequenceBehavior> ?
257   <wsrm:Accept ...>
257     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
257     ...
257   </wsrm:Accept> ?
257   ...
257 </wsrm:CreateSequenceResponse>
```

257 /wsrm:CreateSequenceResponse

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

257 /wsrm:CreateSequenceResponse/wsrm:Identifier

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

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

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

257 /wsrm:CreateSequenceResponse/wsrm:Expires

257 This element, if present, of type `xs:duration` accepts or refines the RM Source's requested duration for
258 the Sequence. A value of 'PT0S' indicates that the Sequence will never expire. Absence of the element
259 indicates an implied value of 'PT0S'. The RM Destination MUST set the value of this element to be equal
260 to or less than the value requested by the RM Source in the corresponding `<wsrm:CreateSequence>`
261 message.

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

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

257 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval`

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

257 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@Milliseconds`

257 The acknowledgement interval, specified in milliseconds.

257 `/wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval/@{any}`

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

257 `/wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior`

257 This ~~optional~~[OPTIONAL](#) element, if present, specifies the behavior that the RM Destination will exhibit
258 upon the closure of an incomplete sequence.

259 A value of "DiscardEntireSequence" indicates that the entire sequence will be discarded by the RM
260 Destination if the sequence is closed when there are one or more gaps in the [final](#)
261 `SequenceAcknowledgement/Final`.

262 A value of "DiscardFollowingFirstGap" indicates that messages in the sequence beyond the first gap will
263 be discarded by the RM Destination when there are one or more gaps in the [final](#)
264 `SequenceAcknowledgement/Final`.

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

265 `/wsrm:CreateSequenceResponse/wsrm:Accept`

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

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

265 `/wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo`

265 The RM Destination MUST include this element, of type `wsa:EndpointReferenceType` (as specified
266 by WS-Addressing [\[WS-Addressing\]](#)). The RM Source SHOULD send messages with
267 `<wsrm:SequenceAcknowledgement>` header blocks related to the accepted Sequence to the
268 referenced endpoint.

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

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

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

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

269 /wsrm:CreateSequenceResponse/{any}

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

269 /wsrm:CreateSequenceResponse/@{any}

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

269 3.2 Closing A Sequence

270 There ~~may~~are be times during the use of an RM Sequence that the RM Source or RM Destination will
271 wish to discontinue using a Sequence. Simply terminating the Sequence discards the state managed by
272 the RM Destination, leaving the RM Source unaware of the final ranges of messages that were
273 successfully ~~delivered~~transferred to the RM Destination. ~~To ensure that the Sequence ends with a known~~
274 final state ~~both~~either the RM Source ~~and~~or RM Destination ~~may~~ MAY choose to close the Sequence
275 before terminating it.

276 If the RM Source wishes to close the Sequence, then it sends a <wsrm:CloseSequence> element, in
277 the body of a message, to the RM Destination. ~~This message indicates that the RM Destination MUST~~
278 NOT receive any new messages for the specified Sequence, other than those already received at the time
279 the <wsrm:CloseSequence> element is interpreted by the RM Destination. Upon receipt of this
280 message, or subsequent to the RM Destination closing the Sequence of its own volition, the RM
281 Destination MUST include a final <wsrm:SequenceAcknowledgement> (within which the RM
282 Destination MUST include the <wsrm:Final> element) header block on any messages associated with
283 the Sequence destined to the RM Source, including the CloseSequenceResponse message or on any
284 Sequence Fault transmitted to the RM Source.

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

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

293 The following exemplar defines the CloseSequence syntax:

```
294 <wsrm:CloseSequence ...>  
295   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
296   ...  
297 </wsrm:CloseSequence>
```

298 /wsrm:CloseSequence

299 This element is sent by an RM Source to indicate that the RM Destination MUST NOT receive any new
300 messages for this Sequence. A SequenceClosed fault MUST be generated by the RM Destination when it
301 receives a message for a Sequence that is [already](#) closed.

302 /wsrm:CloseSequence/wsrm:Identifier

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

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

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

302 /wsrm:CloseSequence/{any}

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

302 /wsrm:CloseSequence@{any}

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

302 A <wsrm:CloseSequenceResponse> is sent in the body of a response message by an RM Destination
303 in response to receipt of a <wsrm:CloseSequence> request message. It indicates that the RM
304 Destination has closed the Sequence.

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

```
302 <wsrm:CloseSequenceResponse ...>  
302   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
302   ...  
302 </wsrm:CloseSequenceResponse>
```

302 /wsrm:CloseSequenceResponse

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

302 /wsrm:CloseSequenceResponse/wsrm:Identifier

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

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

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

302 /wsrm:CloseSequenceResponse/{any}

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

302 /wsrm:CloseSequenceResponse@{any}

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

3.3 Sequence Termination

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

The following exemplar defines the TerminateSequence syntax:

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

`/wsrm:TerminateSequence`

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

`/wsrm:TerminateSequence/wsrm:Identifier`

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

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

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

`/wsrm:TerminateSequence/{any}`

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

`/wsrm:TerminateSequence/@{any}`

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

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

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

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

`/wsrm:TerminateSequenceResponse`

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

309 `/wsrm:TerminateSequenceResponse/wsrm:Identifier`

309 The RM Destination MUST include this element in any `TerminateSequenceResponse` message it sends.
310 The RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986
311 [\[URI\]](#)) of the Sequence that is being terminated.

312 `/wsrm:TerminateSequenceResponse/wsrm:Identifier/@{any}`

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

312 `/wsrm:TerminateSequenceResponse/{any}`

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

312 `/wsrm:TerminateSequenceResponse/@{any}`

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

312 On receipt of a `<wsrm:TerminateSequence>` message an RM Destination MUST respond with a
313 corresponding `<wsrm:TerminateSequenceResponse>` message or generate a fault.

312 3.4 Sequences

313 The RM protocol uses a `<wsrm:Sequence>` header block to track and manage the reliable
314 [deliverytransfer](#) of messages. The RM Source MUST include a `<wsrm:Sequence>` header block in
315 all messages for which reliable [deliverytransfer](#) is [requiredREQUIRED](#). The RM Source MUST
316 identify Sequences with unique `<wsrm:Identifier>` elements and the RM Source MUST assign
317 each message within a Sequence a `<wsrm:MessageNumber>` element that increments by 1 from an
318 initial value of 1. These values are contained within a `<wsrm:Sequence>` header block accompanying
319 each message being [deliveredtransferred](#) in the context of a Sequence.

320 The RM Source MUST NOT include more than one `<wsrm:Sequence>` header block in any message.

320 A following exemplar defines its syntax:

```
320 <wsrm:Sequence ...>  
320   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
320   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>  
320   ...  
320 </wsrm:Sequence>
```

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

320 `/wsrm:Sequence`

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

320 /wsrm:Sequence/wsrm:Identifier

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

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

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

323 /wsrm:Sequence/wsrm:MessageNumber

323 The RM Source MUST include this element within any Sequence headers it creates. This element is of
324 type `wsrm:MessageNumberType`. It represents the ordinal position of the message within a Sequence.
325 Sequence message numbers start at 1 and monotonically increase [by 1](#) throughout the Sequence. If the
326 message number exceeds the internal limitations of an RM Source or RM Destination or reaches the
327 maximum value of 9,223,372,036,854,775,807 the RM Source or Destination MUST generate a
328 MessageNumberRollover fault.

329 /wsrm:Sequence/{any}

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

329 /wsrm:Sequence/@{any}

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

329 The following example illustrates a Sequence header block.

```
329 <wsrm:Sequence>  
329   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
329   <wsrm:MessageNumber>10</wsrm:MessageNumber>  
329 </wsrm:Sequence>
```

329 3.5 Request Acknowledgement

330 The purpose of the <wsrm:AckRequested> header block is to signal to the RM Destination that the RM
331 Source is requesting that a <wsrm:SequenceAcknowledgement> be sent.

330 The RM Source MAY request an acknowledgement message from the RM Destination at any time by
331 including an <wsrm:AckRequested> header block in any message targeted to the RM Destination. An
332 RM Destination that receives a message that contains an <wsrm:AckRequested> header block MUST
333 send a message containing a <wsrm:SequenceAcknowledgement> header block to the wsrm:AcksTo
334 endpoint reference (see Section 3.1). If a non-mustUnderstand fault occurs when processing an RM
335 [h](#)Header that was piggy-backed on another message, a fault MUST be generated, but the processing of
336 the original message MUST NOT be affected. It is RECOMMENDED that the RM [Destination](#) return a
337 <wsrm:AcknowledgementRange> or <wsrm:None> element instead of a <wsrm:Nack> element (see
338 [Section 3.6below](#)).

339 The following exemplar defines its syntax:

```
339 <wsrm:AckRequested ...>  
339   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
339   ...  
339 </wsrm:AckRequested>
```

339 /wsrm:AckRequested

339 This element requests an acknowledgement for the identified Sequence.

339 /wsrm:AckRequested/wsrm:Identifier

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

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

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

339 /wsrm:AckRequested/{any}

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

339 /wsrm:AckRequested/@{any}

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

339 3.6 Sequence Acknowledgement

340 The RM Destination informs the RM Source of successful message receipt using a
341 <wsrm:SequenceAcknowledgement> header block. The RM Destination MAY transmit the
342 <wsrm:SequenceAcknowledgement> header block independently or it MAY include the
343 <wsrm:SequenceAcknowledgement> header block on any message targeted to the AcksTo EPR. ~~The~~
344 ~~RM Destination MAY send a <wsrm:SequenceAcknowledgement> header block at any point~~
345 ~~during which the Sequence is valid.~~ Acknowledgements can be explicitly requested using the
346 <wsrm:AckRequested> directive (see Section [3.5 Request Acknowledgement](#)). If a non-
347 mustUnderstand fault occurs when processing an RM ~~h~~Header that was piggy-backed on another
348 message, a fault MUST be generated, but the processing of the original message MUST NOT be
349 affected.

350 A RM [Destination](#) MAY include a wsrm:SequenceAcknowledgement header block on any SOAP envelope
351 targetted to the endpoint referenced by the wsrm:AcksTo EPR. ~~This concept is often referred to as "piggy-~~
352 ~~backing"—Sequence acknowledgements.~~

353 During creation of a Sequence the RM Source MAY specify the WS-Addressing anonymous IRI as the
354 address of the <wsrm:AcksTo> EPR for that Sequence. When the RM Source specifies the WS-
355 Addressing anonymous IRI as the address of the <wsrm:AcksTo> EPR, the RM Destination MUST
356 transmit any <wsrm:SequenceAcknowledgement> headers for the created Sequence in a SOAP
357 envelope to be transmitted on the protocol binding-specific channel. Such a channel is provided by the
358 context of a received message containing a SOAP envelope that contains a <wsrm:Sequence> header
359 block and/or a <wsrm:AckRequested> header block for that same Sequence identifier.

360 The following exemplar defines its syntax:

```
361 <wsrm:SequenceAcknowledgement ...>
362   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
363   [ [ [ <wsrm:AcknowledgementRange ...
364         Upper="wsrm:MessageNumberType"
365         Lower="wsrm:MessageNumberType" /> +
```

```

366 | <wsrm:None/> ]
367 <wsrm:Final/> ? ]
368 | <wsrm:Nack> wsrm:MessageNumberType </wsrm:Nack> + ]
369
370 ...
371 </wsrm:SequenceAcknowledgement>

```

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

373 `/wsrm:SequenceAcknowledgement`

374 This element contains the Sequence acknowledgement information.

375 `/wsrm:SequenceAcknowledgement/wsrm:Identifier`

376 An RM Destination that includes a `<wsrm:SequenceAcknowledgement>` header block in a SOAP
 377 envelope MUST include this element in that header block. The RM Destination MUST set the value of this
 378 element to the absolute URI (conformant with RFC3986 [URI]) that uniquely identifies the Sequence. The
 379 RM Destination MUST NOT include multiple `<wsrm:SequenceAcknowledgement>` header blocks that
 380 share the same value for `<wsrm:Identifier>` within the same SOAP envelope.

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

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

384 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

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

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

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

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

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

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

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

399 `/wsrm:SequenceAcknowledgement/wsrm:Final`

400 The RM Destination MAY include this element within a `<wsrm:SequenceAcknowledgement>` header
 401 block. This element indicates that the RM Destination is not receiving new messages for the specified
 402 Sequence. The RM Source can be assured that the ranges of messages acknowledged by this
 403 SequenceAcknowledgement header block will not change in the future. The RM Destination MUST
 404 include this element when the Sequence is closed. ~~Note:~~ the RM Destination MUST NOT include this
 405 element when sending a Nack; it can only be used when sending `<wsrm:AcknowledgementRange>`s or
 406 `<wsrm:None>`.

407 `/wsrm:SequenceAcknowledgement/wsrm:Nack`

408 The RM Destination MAY include this element within a `<wsrm:SequenceAcknowledgement>` header
 409 block. If used, the RM Destination MUST set the value of this element to a `wsrm:MessageNumberType`
 410 representing the `<wsrm:MessageNumber>` of an unreceived message in a Sequence. The RM
 411 Destination MUST NOT include a `<wsrm:Nack>` element if a sibling
 412 `<wsrm:AcknowledgementRange>` or `<wsrm:None>` element is also present as a child of
 413 `<wsrm:SequenceAcknowledgement>`. Upon the receipt of a Nack, an RM Source SHOULD retransmit
 414 the message identified by the Nack. The RM Destination MUST NOT issue a
 415 `<wsrm:SequenceAcknowledgement>` containing a `<wsrm:Nack>` for a message that it has previously
 416 acknowledged within a `<wsrm:AcknowledgementRange>`. The RM Source SHOULD ignore a
 417 `<wsrm:SequenceAcknowledgement>` containing a `<wsrm:Nack>` for a message that has previously
 418 been acknowledged within a `<wsrm:AcknowledgementRange>`.

419 `/wsrm:SequenceAcknowledgement/wsrm:None`

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

419 `/wsrm:SequenceAcknowledgement/{any}`

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

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

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

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

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

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

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

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

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

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

419 3.7 MakeConnection

420 [When an endpoint is not directly addressable \(e.g. behind a firewall or not able to allow incoming](#)
 421 [connections\), an anonymous URI in the EPR address property can indicate such an endpoint. The WS-](#)

420 [Addressing anonymous URI](#) is one such anonymous URI. This specification defines a URI template (the
421 [WS-RM anonymous URI](#)) which may be used to uniquely identify anonymous endpoint.

420 <http://docs.oasis-open.org/ws-rx/wsrn/200604/anonymous?id={uuid}>

420 This URI template in an EPR indicates a protocol-specific back-channel will be established through a
421 mechanism such as [wsrm:MakeConnection](#), defined below. When using this URI template, "{uudi}"
422 MUST be replaced by a UUID value as defined by RFC4122[UUID]. This UUID value uniquely
423 distinguishes the endpoint. A sending endpoint SHOULD transmit messages at endpoints identified with
424 the URI template using a protocol-specific back-channel, including but not limited to those established with
425 a [wsrm:MakeConnection](#) message. Note, this URI is semantically similar to the [WS-Addressing](#)
426 [anonymous URI](#) if a protocol-specific back-channel is available.

427 The [wsrm:MakeConnection](#) is a one-way operation that establishes a contextualized back-channel for
428 the transmission of messages according to matching criteria (defined below). In the non-faulting case, if
429 no matching message is available then no SOAP envelopes will be returned on the back-channel. A
430 common usage will be a client RM Destination sending [wsrm:MakeConnection](#) to a server RM Source
431 for the purpose of receiving asynchronous response messages.

427 The following exemplar defines the [<wsrm:MakeConnection>](#) syntax:

```
427 <wsrm:MakeConnection ...>  
427   <wsrm:Identifier> xs:anyURI </wsrm:Identifier> ?  
427   <wsrm:Address> xs:anyURI </wsrm:Address> ?  
427   ...  
427 </wsrm:MakeConnection>
```

427 [/wsrm:MakeConnection](#)

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

427 [/wsrm:MakeConnection/wsrm:Identifier](#)

427 This element specifies the WS-RM Sequence Identifier that establishes the context for the transport-
428 specific back-channel. The Sequence Identifier should be compared with the Sequence Identifiers
429 associated with the messages held by the sending endpoint, and if there is a matching message it will be
430 returned. If this element is omitted from the message then the [wsrm:Address](#) MUST be included in the
431 message.

427 [/wsrm:MakeConnection/wsrm:Address](#)

427 This element specifies the URI ([wsa:Address](#)) of the initiating endpoint. Endpoints MUST NOT return
428 messages on the transport-specific back-channel unless they have been addressed to this URI. The
429 [/wsrm:MakeConnection/wsrm:Address](#) property and a message's WS-Addressing destination
430 property are considered identical when they are exactly the same character-for-character. Note that URIs
431 which are not identical in this sense may in fact be functionally equivalent. Examples include URI
432 references which differ only in case, or which are in external entities which have different effective base
433 URIs. If this element is omitted from the message then the [wsrm:Identifier](#) MUST be included in the
434 message.

427 [/wsrm:MakeConnection/{any}](#)

427 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,
428 to be passed. This allows fine-tuning of the messages to be returned, additional selection criteria included
429 here are logically ANDed with the [wsrm:Address](#) and/or [wsrm:Identifier](#). If an extension is not
430 supported by the endpoint then it should return a [wsrm:UnsupportedSelection](#) fault.

427 /wsrm:MakeConnection/@{any}

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

427 If both `wsrm:Identifier` and `wsrm:Address` are present, then the endpoint processing the
428 `wsrm:MakeConnection` message MUST insure that any SOAP Envelope flowing on the backchannel
429 MUST be associated with the given Sequence and MUST be addressed to the given URI.

427 The management of messages that are awaiting the establishment of a back-channel to their receiving
428 endpoint is an implementation detail that is outside the scope of this specification. Note, however, that
429 these messages form a class of asynchronous messages that is not dissimilar from “ordinary”
430 asynchronous messages that are waiting for the establishment of a connection to their destination
431 endpoints.

427 This specification places no constraint on the types of messages that can be returned on the transport-
428 specific back-channel. As in an asynchronous environment, it is up to the recipient of the
429 `wsrm:MakeConnection` message to decide which messages are appropriate for transmission to any
430 particular endpoint. However, the endpoint processing the `wsrm:MakeConnection` message MUST
431 insure that the messages match the selection criteria as specified by the child elements of the
432 `wsrm:MakeConnection` element.

|
|
|
|
|

4 Faults

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

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

Entities that generate WS-ReliableMessaging faults MUST include as the [action] property the default fault action IRI defined ~~in the version of WS-Addressing used in the message below~~. The value from the ~~current version~~ [W3C Recommendation](#) is below for informational purposes:

```
http://schemas.xmlsoap.org/wsdocs.oasis-open.org/ws-  
rm/2004/08/addressing/fault
```

The faults defined in this section are generated if the condition stated in the preamble is met. Fault handling rules are defined in section 64 of WS-Addressing [SOAP Binding](#).

The definitions of faults use the following properties:

[Code] The fault code.

[Subcode] The fault subcode.

[Reason] The English language reason element.

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

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

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

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

```
<S:Envelope>  
<S:Header>  
  <wsa:Action>  
    http://schemas.xmlsoap.org/ws/2004/08/addressing/faulthttp://docs.oasis-  
  </wsa:Action>  
  <!-- Headers elided for clarity. -->  
</S:Header>  
<S:Body>  
  <S:Fault>
```



```

428     <S:Code>
428         <S:Value> [Code] </S:Value>
428         <S:Subcode>
428             <S:Value> [Subcode] </S:Value>
428         </S:Subcode>
428     </S:Code>
428     <S:Reason>
428         <S:Text xml:lang="en"> [Reason] </S:Text>
428     </S:Reason>
428     <S:Detail>
428         [Detail]
428         ...
428     </S:Detail>
428 </S:Fault>
428 </S:Body>
428 </S:Envelope>

```

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

```

428 <S11:Envelope>
428   <S11:Header>
428     <wsrm:SequenceFault>
428       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
428       <wsrm:Detail> [Detail] </wsrm:Detail>
428       ...
428     </wsrm:SequenceFault>
428     <!-- Headers elided for clarity. -->
428   </S11:Header>
428   <S11:Body>
428     <S11:Fault>
428       <faultcode> [Code] </faultcode>
428       <faultstring> [Reason] </faultstring>
428     </S11:Fault>
428   </S11:Body>
428 </S11:Envelope>

```

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

```

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

```

428 4.1 SequenceFault Element

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

429 The following exemplar defines its syntax:

```

429 <wsrm:SequenceFault ...>

```

```

429 <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
429 <wsrm:Detail> ... </wsrm:Detail> ?
429 ...
429 </wsrm:SequenceFault>

```

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

429 `/wsrm:SequenceFault`

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

429 `/wsrm:SequenceFault/wsrm:FaultCode`

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

429 `/wsrm:SequenceFault/wsrm:Detail`

429 This ~~optional~~[OPTIONAL](#) element is intended for carrying application specific error information related to
430 the fault being described.

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

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

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

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

429 `/wsrm:SequenceFault/{any}`

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

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

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

429 4.2 Sequence Terminated

430 This fault is generated by either the RM Source or the RM Destination to indicate that it has either
431 encountered an unrecoverable condition, or has detected a violation of the protocol and as a
432 consequence, has chosen to terminate the Sequence. The endpoint that generates this fault
433 ~~should~~[SHOULD](#) make every reasonable effort to notify the corresponding endpoint of this decision.

430 Receipt of `SequenceTerminated` by either the `RM_Destination` or the `RM_Source` ~~shall~~[SHALL](#) terminate the
431 Sequence if it is not otherwise terminated.

430 Properties:

430 [Code] Sender or Receiver

430 [Subcode] `wsrm:SequenceTerminated`

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

430 [Detail]

```

430 <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>

```

4.3 Unknown Sequence

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

Properties:

[Code] Sender

[Subcode] wsrn:UnknownSequence

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

[Detail]

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

4.4 Invalid Acknowledgement

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

[Code] Sender

[Subcode] wsrn:InvalidAcknowledgement

[Reason] The SequenceAcknowledgement violates the cumulative acknowledgement invariant.

[Detail]

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

4.5 Message Number Rollover

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

Properties:

[Code] Sender

[Subcode] wsrn:MessageNumberRollover

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

[Detail]

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

4.6 Create Sequence Refused

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

Properties:

[Code] Sender

[Subcode] wsrn:CreateSequenceRefused

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

434 [Detail]

434 `xs:any`

434 4.7 Sequence Closed

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

436 This fault MUST be generated when an RM Destination is asked to receive a message for a Sequence
437 that is closed [or when an RM Destination is asked to close a Sequence that is already closed](#).

435 Properties:

435 [Code] Sender

435 [Subcode] wsrn:SequenceClosed

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

435 [Detail]

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

435 4.8 WSRM Required

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

436 Properties:

436 [Code] Sender

436 [Subcode] wsrn:WSRMRequired

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

436 [Detail]

436 `xs:any`

436 4.9 Unsupported Selection

437 [This fault is generated to indicate that endpoint processing the wsrn:MakeConnection message does not
438 support the selection criteria included in the extensibility section of the wsrn:MakeConnection message.](#)

437 [The QName of the unsupported element\(s\) are included in the detail.](#)

437 [Properties:](#)

437 [\[Code\] Receiver](#)

437 [\[Subcode\] wsrn:UnsupportedSelection](#)

437 [\[Reason\] The extension element used in the message selection is not supported by the RM Source](#)

437 [\[Detail\]](#)

437 `<wsrm:UnsupportedElement> xs:QName </wsrm:UnsupportedElement>+`

5 Security Considerations

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

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

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

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

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

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

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

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

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

6 References

6.1 Normative

[KEYWORDS]

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

[SOAP 1.1]

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

[SOAP 1.2]

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

[URI]

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

[UUID]

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

[XML]

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

[XML-ns]

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

[XML-Schema Part1]

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

[XML-Schema Part2]

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

[WSDL 1.1]

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

[WS-Addressing]

W3C ~~Proposed~~ Recommendation, "[Web Services Addressing 1.0 - Core](#)", ~~March~~ 2006.

W3C ~~Proposed~~ Recommendation, "[Web Services Addressing 1.0 – SOAP Binding](#)", ~~March~~ 2006.

6.2 Non-Normative

[RDDL 2.0]

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

[WS-Policy]

~~W3C Member Submission, "Web Services Policy Framework (WS-Policy)," April 2006D. Box, et al, "Web Services Policy Framework (WS-Policy)," September 2004.~~

481 **[WS-PolicyAttachment]**

481 [W3C Member Submission, "Web Services Policy Attachment \(WS-PolicyAttachment\)," April 2006](#)~~Box, et-~~
482 ~~al, "Web Services Policy Attachment (WS-PolicyAttachment)," September 2004.~~

483 **[WS-Security]**

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

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

488 **[RTTM]**

488 V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", RFC 1323, May
489 1992.

490 **[SecurityPolicy]**

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

491 **[SecureConversation]**

492 S. Anderson, et al, "Web Services Secure Conversation Language (WS-SecureConversation)," February
493 2005.

494 **[Trust]**

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

494 **A. Schema**

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

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

497 The following copy is provided for reference.

```

497 <?xml version="1.0" encoding="UTF-8"?>
498 <!--
499 OASIS takes no position regarding the validity or scope of any intellectual
500 property or other rights that might be claimed to pertain to the
501 implementation or use of the technology described in this document or the
502 extent to which any license under such rights might or might not be available;
503 neither does it represent that it has made any effort to identify any such
504 rights. Information on OASIS's procedures with respect to rights in OASIS
505 specifications can be found at the OASIS website. Copies of claims of rights
506 made available for publication and any assurances of licenses to be made
507 available, or the result of an attempt made to obtain a general license or
508 permission for the use of such proprietary rights by implementors or users of
509 this specification, can be obtained from the OASIS Executive Director.
510 OASIS invites any interested party to bring to its attention any copyrights,
511 patents or patent applications, or other proprietary rights which may cover
512 technology that may be required to implement this specification. Please
513 address the information to the OASIS Executive Director.
514 Copyright &copy; OASIS Open 2002-2006. All Rights Reserved.
515 This document and translations of it may be copied and furnished to others,
516 and derivative works that comment on or otherwise explain it or assist in its
517 implementation may be prepared, copied, published and distributed, in whole or
518 in part, without restriction of any kind, provided that the above copyright
519 notice and this paragraph are included on all such copies and derivative
520 works. However, this document itself does not be modified in any way, such as
521 by removing the copyright notice or references to OASIS, except as needed for
522 the purpose of developing OASIS specifications, in which case the procedures
523 for copyrights defined in the OASIS Intellectual Property Rights document must
524 be followed, or as required to translate it into languages other than English.
525 The limited permissions granted above are perpetual and will not be revoked by
526 OASIS or its successors or assigns.
527 This document and the information contained herein is provided on an "AS
528 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
529 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
530 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
531 FITNESS FOR A PARTICULAR PURPOSE.
532 -->
533 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
534 xmlns:wsa="http://www.w3.org/2005/08/addressing"
535 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200604"
536 targetNamespace="http://docs.oasis-open.org/ws-rx/wsm/200604"
537 elementFormDefault="qualified" attributeFormDefault="unqualified">
538 <xs:import namespace="http://www.w3.org/2005/08/addressing"
539 schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
540 <!-- Protocol Elements -->
541 <xs:complexType name="SequenceType">
542 <xs:sequence>
543 <xs:element ref="wsm:Identifier"/>
544 <xs:element name="MessageNumber" type="wsm:MessageNumberType"/>
545 <xs:any namespace="##other" processContents="lax" minOccurs="0"
546 maxOccurs="unbounded"/>
547 </xs:sequence>
548 <xs:anyAttribute namespace="##other" processContents="lax"/>
549 </xs:complexType>
550 <xs:element name="Sequence" type="wsm:SequenceType"/>
551 <xs:element name="SequenceAcknowledgement">
552 <xs:complexType>
553 <xs:sequence>
554 <xs:element ref="wsm:Identifier"/>
555 <xs:choice>
556 <xs:sequence>
557 <xs:choice>
558 <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
559 <xs:complexType>

```

```

497         <xs:sequence/>
498         <xs:attribute name="Upper" type="xs:unsignedLong"
499 use="required"/>
500         <xs:attribute name="Lower" type="xs:unsignedLong"
501 use="required"/>
502         <xs:anyAttribute namespace="##other" processContents="lax"/>
503     </xs:complexType>
504 </xs:element>
505     <xs:element name="None" minOccurs="0">
506         <xs:complexType>
507             <xs:sequence/>
508         </xs:complexType>
509     </xs:element>
510 </xs:choice>
511     <xs:element name="Final" minOccurs="0">
512         <xs:complexType>
513             <xs:sequence/>
514         </xs:complexType>
515     </xs:element>
516 </xs:sequence>
517     <xs:element name="Nack" type="xs:unsignedLong"
518 maxOccurs="unbounded"/>
519 </xs:choice>
520     <xs:any namespace="##other" processContents="lax" minOccurs="0"
521 maxOccurs="unbounded"/>
522 </xs:sequence>
523     <xs:anyAttribute namespace="##other" processContents="lax"/>
524 </xs:complexType>
525 </xs:element>
526 <xs:complexType name="AckRequestedType">
527     <xs:sequence>
528         <xs:element ref="wsrm:Identifier"/>
529         <xs:any namespace="##other" processContents="lax" minOccurs="0"
530 maxOccurs="unbounded"/>
531     </xs:sequence>
532     <xs:anyAttribute namespace="##other" processContents="lax"/>
533 </xs:complexType>
534 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
535 <xs:element name="Identifier">
536     <xs:complexType>
537         <xs:annotation>
538             <xs:documentation>
539                 This type is for elements whose [children] is an anyURI and can have
540 arbitrary attributes.
541             </xs:documentation>
542         </xs:annotation>
543         <xs:simpleContent>
544             <xs:extension base="xs:anyURI">
545                 <xs:anyAttribute namespace="##other" processContents="lax"/>
546             </xs:extension>
547         </xs:simpleContent>
548     </xs:complexType>
549 </xs:element>
550 <xs:simpleType name="MessageNumberType">
551     <xs:restriction base="xs:unsignedLong">
552         <xs:minInclusive value="1"/>
553         <xs:maxInclusive value="9223372036854775807"/>
554     </xs:restriction>
555 </xs:simpleType>
556 <!-- Fault Container and Codes -->
557 <xs:simpleType name="FaultCodes">
558     <xs:restriction base="xs:QName">
559         <xs:enumeration value="wsrm:SequenceTerminated"/>

```

```

497 <xs:enumeration value="wsrm:UnknownSequence"/>
498 <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
499 <xs:enumeration value="wsrm:MessageNumberRollover"/>
500 <xs:enumeration value="wsrm:CreateSequenceRefused"/>
501 <xs:enumeration value="wsrm:SequenceClosed"/>
502 <xs:enumeration value="wsrm:WSRMRequired"/>
503 </xs:restriction>
504 </xs:simpleType>
505 <xs:complexType name="SequenceFaultType">
506 <xs:sequence>
507 <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
508 <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
509 <xs:any namespace="##other" processContents="lax" minOccurs="0"
510 maxOccurs="unbounded"/>
511 </xs:sequence>
512 <xs:anyAttribute namespace="##other" processContents="lax"/>
513 </xs:complexType>
514 <xs:complexType name="DetailType">
515 <xs:sequence>
516 <xs:any namespace="##other" processContents="lax" minOccurs="0"
517 maxOccurs="unbounded"/>
518 </xs:sequence>
519 <xs:anyAttribute namespace="##other" processContents="lax"/>
520 </xs:complexType>
521 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
522 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
523 <xs:element name="CreateSequenceResponse"
524 type="wsrm:CreateSequenceResponseType"/>
525 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
526 <xs:element name="CloseSequenceResponse"
527 type="wsrm:CloseSequenceResponseType"/>
528 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
529 <xs:element name="TerminateSequenceResponse"
530 type="wsrm:TerminateSequenceResponseType"/>
531 <xs:complexType name="CreateSequenceType">
532 <xs:sequence>
533 <xs:element ref="wsrm:AcksTo"/>
534 <xs:element ref="wsrm:Expires" minOccurs="0"/>
535 <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
536 <xs:any namespace="##other" processContents="lax" minOccurs="0"
537 maxOccurs="unbounded"/>
538 <xs:annotation>
539 <xs:documentation>
540 It is the authors intent that this extensibility be used to
541 transfer a Security Token Reference as defined in WS-Security.
542 </xs:documentation>
543 </xs:annotation>
544 </xs:any>
545 </xs:sequence>
546 <xs:anyAttribute namespace="##other" processContents="lax"/>
547 </xs:complexType>
548 <xs:complexType name="CreateSequenceResponseType">
549 <xs:sequence>
550 <xs:element ref="wsrm:Identifier"/>
551 <xs:element ref="wsrm:Expires" minOccurs="0"/>
552 <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>

```

```

497 <xs:element name="IncompleteSequenceBehaviour"
498 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
499 <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
500 <xs:any namespace="##other" processContents="lax" minOccurs="0"
501 maxOccurs="unbounded"/>
502 </xs:sequence>
503 <xs:anyAttribute namespace="##other" processContents="lax"/>
504 </xs:complexType>
505 <xs:complexType name="CloseSequenceType">
506 <xs:sequence>
507 <xs:element ref="wsrm:Identifier"/>
508 <xs:any namespace="##other" processContents="lax" minOccurs="0"
509 maxOccurs="unbounded"/>
510 </xs:sequence>
511 <xs:anyAttribute namespace="##other" processContents="lax"/>
512 </xs:complexType>
513 <xs:complexType name="CloseSequenceResponseType">
514 <xs:sequence>
515 <xs:element ref="wsrm:Identifier"/>
516 <xs:any namespace="##other" processContents="lax" minOccurs="0"
517 maxOccurs="unbounded"/>
518 </xs:sequence>
519 <xs:anyAttribute namespace="##other" processContents="lax"/>
520 </xs:complexType>
521 <xs:complexType name="TerminateSequenceType">
522 <xs:sequence>
523 <xs:element ref="wsrm:Identifier"/>
524 <xs:any namespace="##other" processContents="lax" minOccurs="0"
525 maxOccurs="unbounded"/>
526 </xs:sequence>
527 <xs:anyAttribute namespace="##other" processContents="lax"/>
528 </xs:complexType>
529 <xs:complexType name="TerminateSequenceResponseType">
530 <xs:sequence>
531 <xs:element ref="wsrm:Identifier"/>
532 <xs:any namespace="##other" processContents="lax" minOccurs="0"
533 maxOccurs="unbounded"/>
534 </xs:sequence>
535 <xs:anyAttribute namespace="##other" processContents="lax"/>
536 </xs:complexType>
537 <xs:element name="AcksTo"

```

```

497 type="wsa:EndpointReferenceType"/>
498 <xs:complexType name="OfferType">
499   <xs:sequence>
500     <xs:element ref="wsrm:Identifier"/>
501     <xs:element ref="wsrm:Expires" minOccurs="0"/>
502     <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
503     <xs:any namespace="##other" processContents="lax" minOccurs="0"
504 maxOccurs="unbounded"/>
505   </xs:sequence>
506   <xs:anyAttribute namespace="##other" processContents="lax"/>
507 </xs:complexType>
508 <xs:complexType name="AcceptType">
509   <xs:sequence>
510     <xs:element ref="wsrm:AcksTo"/>
511     <xs:any namespace="##other" processContents="lax" minOccurs="0"
512 maxOccurs="unbounded"/>
513   </xs:sequence>
514   <xs:anyAttribute namespace="##other" processContents="lax"/>
515 </xs:complexType>
516 <xs:element name="Expires">
517   <xs:complexType>
518     <xs:simpleContent>
519       <xs:extension base="xs:duration">
520         <xs:anyAttribute namespace="##other" processContents="lax"/>
521       </xs:extension>
522     </xs:simpleContent>
523   </xs:complexType>
524 </xs:element>
525 <xs:element name="AcknowledgementInterval">
526   <xs:complexType>
527     <xs:sequence/>
528     <xs:attribute name="Milliseconds" type="xs:unsignedLong"
529 use="required"/>
530     <xs:anyAttribute namespace="##other" processContents="lax"/>
531   </xs:complexType>
532 </xs:element>

```

```

497 <xs:simpleType name="IncompleteSequenceBehaviorType">
498 <xs:restriction base="xs:string">
499 <xs:enumeration value="DiscardEntireSequence"/>
500 <xs:enumeration value="DiscardFollowingFirstGap"/>
501 <xs:enumeration value="NoDiscard"/>
502 </xs:restriction>
503 </xs:simpleType>
504 </xs:schema><?xml version="1.0" encoding="UTF-8"?>
505 <!--
506 OASIS takes no position regarding the validity or scope of any intellectual
507 property or other rights that might be claimed to pertain to the
508 implementation or use of the technology described in this document or the
509 extent to which any license under such rights might or might not be available;
510 neither does it represent that it has made any effort to identify any such
511 rights. Information on OASIS's procedures with respect to rights in OASIS
512 specifications can be found at the OASIS website. Copies of claims of rights
513 made available for publication and any assurances of licenses to be made
514 available, or the result of an attempt made to obtain a general license or
515 permission for the use of such proprietary rights by implementors or users of
516 this specification, can be obtained from the OASIS Executive Director.
517 OASIS invites any interested party to bring to its attention any copyrights,
518 patents or patent applications, or other proprietary rights which may cover
519 technology that may be required to implement this specification. Please
520 address the information to the OASIS Executive Director.
521 Copyright &copy; OASIS Open 2002-2006. All Rights Reserved.
522 This document and translations of it may be copied and furnished to others,
523 and derivative works that comment on or otherwise explain it or assist in its
524 implementation may be prepared, copied, published and distributed, in whole or
525 in part, without restriction of any kind, provided that the above copyright
526 notice and this paragraph are included on all such copies and derivative
527 works. However, this document itself does not be modified in any way, such as
528 by removing the copyright notice or references to OASIS, except as needed for
529 the purpose of developing OASIS specifications, in which case the procedures
530 for copyrights defined in the OASIS Intellectual Property Rights document must
531 be followed, or as required to translate it into languages other than English.
532 The limited permissions granted above are perpetual and will not be revoked by
533 OASIS or its successors or assigns.
534 This document and the information contained herein is provided on an "AS IS"
535 basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
536 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT
537 INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS
538 FOR A PARTICULAR PURPOSE.
539 -->
540 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
541 xmlns:wsa="http://www.w3.org/2005/08/addressing"
542 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200604"
543 targetNamespace="http://docs.oasis-open.org/ws-rx/wsm/200604"
544 elementFormDefault="qualified" attributeFormDefault="unqualified">
545 <xs:import namespace="http://www.w3.org/2005/08/addressing"
546 schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
547 <!-- Protocol Elements -->
548 <xs:complexType name="SequenceType">
549 <xs:sequence>
550 <xs:element ref="wsm:Identifier"/>
551 <xs:element name="MessageNumber" type="wsm:MessageNumberType"/>
552 <xs:any namespace="##other" processContents="lax" minOccurs="0"
553 maxOccurs="unbounded"/>
554 </xs:sequence>
555 <xs:anyAttribute namespace="##other" processContents="lax"/>
556 </xs:complexType>
557 <xs:element name="Sequence" type="wsm:SequenceType"/>
558 <xs:element name="SequenceAcknowledgement">
559 <xs:complexType>

```



```

497     <xs:sequence>
498         <xs:element ref="wsrm:Identifier"/>
499         <xs:choice>
500             <xs:sequence>
501                 <xs:choice>
502                     <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
503                         <xs:complexType>
504                             <xs:sequence/>
505                             <xs:attribute name="Upper" type="xs:unsignedLong"
506 use="required"/>
507                             <xs:attribute name="Lower" type="xs:unsignedLong"
508 use="required"/>
509                             <xs:anyAttribute namespace="##other" processContents="lax"/>
510                         </xs:complexType>
511                     </xs:element>
512                     <xs:element name="None" minOccurs="0">
513                         <xs:complexType>
514                             <xs:sequence/>
515                         </xs:complexType>
516                     </xs:element>
517                 </xs:choice>
518                 <xs:element name="Final" minOccurs="0">
519                     <xs:complexType>
520                         <xs:sequence/>
521                     </xs:complexType>
522                 </xs:element>
523             </xs:sequence>
524             <xs:element name="Nack" type="xs:unsignedLong"
525 maxOccurs="unbounded"/>
526         </xs:choice>
527         <xs:any namespace="##other" processContents="lax" minOccurs="0"
528 maxOccurs="unbounded"/>
529     </xs:sequence>
530     <xs:anyAttribute namespace="##other" processContents="lax"/>
531 </xs:complexType>
532 </xs:element>
533 <xs:complexType name="AckRequestedType">
534     <xs:sequence>
535         <xs:element ref="wsrm:Identifier"/>
536         <xs:any namespace="##other" processContents="lax" minOccurs="0"
537 maxOccurs="unbounded"/>
538     </xs:sequence>
539     <xs:anyAttribute namespace="##other" processContents="lax"/>
540 </xs:complexType>
541 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
542 <xs:element name="Identifier">
543     <xs:complexType>
544         <xs:annotation>
545             <xs:documentation>
546                 This type is for elements whose [children] is an anyURI and can have
547 arbitrary attributes.
548             </xs:documentation>
549         </xs:annotation>
550         <xs:simpleContent>
551             <xs:extension base="xs:anyURI">
552                 <xs:anyAttribute namespace="##other" processContents="lax"/>
553             </xs:extension>
554         </xs:simpleContent>
555     </xs:complexType>
556 </xs:element>
557 <xs:element name="Address">
558     <xs:complexType>
559         <xs:simpleContent>

```



```

497     <xs:extension base="xs:anyURI">
498         <xs:anyAttribute namespace="##other" processContents="lax"/>
499     </xs:extension>
500 </xs:simpleContent>
501 </xs:complexType>
502 </xs:element>
503 <xs:complexType name="MakeConnectionType">
504     <xs:sequence>
505         <xs:element ref="wsrm:Identifier" minOccurs="0" maxOccurs="1"/>
506         <xs:element ref="wsrm:Address" minOccurs="0" maxOccurs="1"/>
507         <xs:any namespace="##other" processContents="lax" minOccurs="0"
508 maxOccurs="unbounded"/>
509     </xs:sequence>
510     <xs:anyAttribute namespace="##other" processContents="lax"/>
511 </xs:complexType>
512 <xs:element name="MakeConnection" type="wsrm:MakeConnectionType"/>
513 <xs:simpleType name="MessageNumberType">
514     <xs:restriction base="xs:unsignedLong">
515         <xs:minInclusive value="1"/>
516         <xs:maxInclusive value="9223372036854775807"/>
517     </xs:restriction>
518 </xs:simpleType>
519 <!-- Fault Container and Codes -->
520 <xs:simpleType name="FaultCodes">
521     <xs:restriction base="xs:QName">
522         <xs:enumeration value="wsrm:SequenceTerminated"/>
523         <xs:enumeration value="wsrm:UnknownSequence"/>
524         <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
525         <xs:enumeration value="wsrm:MessageNumberRollover"/>
526         <xs:enumeration value="wsrm:CreateSequenceRefused"/>
527         <xs:enumeration value="wsrm:SequenceClosed"/>
528         <xs:enumeration value="wsrm:WSRMRequired"/>
529         <xs:enumeration value="wsrm:UnsupportedSelection"/>
530     </xs:restriction>
531 </xs:simpleType>
532 <xs:complexType name="SequenceFaultType">
533     <xs:sequence>
534         <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
535         <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
536         <xs:any namespace="##other" processContents="lax" minOccurs="0"
537 maxOccurs="unbounded"/>
538     </xs:sequence>
539     <xs:anyAttribute namespace="##other" processContents="lax"/>
540 </xs:complexType>
541 <xs:complexType name="DetailType">
542     <xs:sequence>
543         <xs:any namespace="##other" processContents="lax" minOccurs="0"
544 maxOccurs="unbounded"/>
545     </xs:sequence>
546     <xs:anyAttribute namespace="##other" processContents="lax"/>
547 </xs:complexType>
548 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
549 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
550 <xs:element name="CreateSequenceResponse"
551 type="wsrm:CreateSequenceResponseType"/>
552 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
553 <xs:element name="CloseSequenceResponse"
554 type="wsrm:CloseSequenceResponseType"/>
555 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
556 <xs:element name="TerminateSequenceResponse"
557 type="wsrm:TerminateSequenceResponseType"/>
558 <xs:complexType name="CreateSequenceType">
559     <xs:sequence>

```

```

497     <xs:element ref="wsrm:AcksTo"/>
498     <xs:element ref="wsrm:Expires" minOccurs="0"/>
499     <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
500     <xs:any namespace="##other" processContents="lax" minOccurs="0"
501     maxOccurs="unbounded">
502         <xs:annotation>
503             <xs:documentation>
504                 It is the authors intent that this extensibility be used to
505                 transfer a Security Token Reference as defined in WS-Security.
506             </xs:documentation>
507         </xs:annotation>
508     </xs:any>
509 </xs:sequence>
510 <xs:anyAttribute namespace="##other" processContents="lax"/>
511 </xs:complexType>
512 <xs:complexType name="CreateSequenceResponseType">
513     <xs:sequence>
514         <xs:element ref="wsrm:Identifier"/>
515         <xs:element ref="wsrm:Expires" minOccurs="0"/>
516         <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>
517         <xs:element name="IncompleteSequenceBehavior"
518         type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
519         <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
520         <xs:any namespace="##other" processContents="lax" minOccurs="0"
521         maxOccurs="unbounded"/>
522     </xs:sequence>
523     <xs:anyAttribute namespace="##other" processContents="lax"/>
524 </xs:complexType>
525 <xs:complexType name="CloseSequenceType">
526     <xs:sequence>
527         <xs:element ref="wsrm:Identifier"/>
528         <xs:any namespace="##other" processContents="lax" minOccurs="0"
529         maxOccurs="unbounded"/>
530     </xs:sequence>
531     <xs:anyAttribute namespace="##other" processContents="lax"/>
532 </xs:complexType>
533 <xs:complexType name="CloseSequenceResponseType">

```

```

497     <xs:sequence>
498         <xs:element ref="wsrm:Identifier"/>
499         <xs:any namespace="##other" processContents="lax" minOccurs="0"
500 maxOccurs="unbounded"/>
501     </xs:sequence>
502     <xs:anyAttribute namespace="##other" processContents="lax"/>
503 </xs:complexType>
504 <xs:complexType name="TerminateSequenceType">
505     <xs:sequence>
506         <xs:element ref="wsrm:Identifier"/>
507         <xs:any namespace="##other" processContents="lax" minOccurs="0"
508 maxOccurs="unbounded"/>
509     </xs:sequence>
510     <xs:anyAttribute namespace="##other" processContents="lax"/>
511 </xs:complexType>
512 <xs:complexType name="TerminateSequenceResponseType">
513     <xs:sequence>
514         <xs:element ref="wsrm:Identifier"/>
515         <xs:any namespace="##other" processContents="lax" minOccurs="0"
516 maxOccurs="unbounded"/>
517     </xs:sequence>
518     <xs:anyAttribute namespace="##other" processContents="lax"/>
519 </xs:complexType>
520 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
521 <xs:complexType name="OfferType">
522     <xs:sequence>
523         <xs:element ref="wsrm:Identifier"/>
524         <xs:element ref="wsrm:Expires" minOccurs="0"/>
525         <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
526         <xs:any namespace="##other" processContents="lax" minOccurs="0"
527 maxOccurs="unbounded"/>
528     </xs:sequence>
529     <xs:anyAttribute namespace="##other" processContents="lax"/>
530 </xs:complexType>
531 <xs:complexType name="AcceptType">
532     <xs:sequence>
533         <xs:element ref="wsrm:AcksTo"/>
534         <xs:any namespace="##other" processContents="lax" minOccurs="0"
535 maxOccurs="unbounded"/>
536     </xs:sequence>
537     <xs:anyAttribute namespace="##other" processContents="lax"/>
538 </xs:complexType>
539 <xs:element name="Expires">
540     <xs:complexType>
541         <xs:simpleContent>
542             <xs:extension base="xs:duration">
543                 <xs:anyAttribute namespace="##other" processContents="lax"/>
544             </xs:extension>
545         </xs:simpleContent>
546     </xs:complexType>
547 </xs:element>
548 <xs:element name="AcknowledgementInterval">
549     <xs:complexType>
550         <xs:sequence/>
551         <xs:attribute name="Milliseconds" type="xs:unsignedLong"
552 use="required"/>
553         <xs:anyAttribute namespace="##other" processContents="lax"/>
554     </xs:complexType>
555 </xs:element>
556 <xs:simpleType name="IncompleteSequenceBehaviorType">
557     <xs:restriction base="xs:string">
558         <xs:enumeration value="DiscardEntireSequence"/>
559         <xs:enumeration value="DiscardFollowingFirstGap"/>

```

```
497     <xs:enumeration value="NoDiscard"/>
498   </xs:restriction>
499 </xs:simpleType>
500 <xs:simpleType name="UnsupportedElement">
501   <xs:restriction base="xs:QName"/>
502 </xs:element>
503 </xs:schema>
```

B. Message Examples

B.1 Create Sequence

Create Sequence

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

Create Sequence Response

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

B.2 Initial Transmission

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

497 Message 1

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

497 Message 2

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

497 Message 3

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

```

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

```

497 B.3 First Acknowledgement

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

```

497 <?xml version="1.0" encoding="UTF-8"?>
497 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
497 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
497 xmlns:wsa="http://www.w3.org/2005/08/addressing">
497 <S:Header>
497 <wsa:MessageID>
497 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
497 </wsa:MessageID>
497 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
497 <wsa:From>
497 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
497 </wsa:From>
497 <wsa:Action>
497 http://docs.oasis-open.org/ws-rx/wsr/200604/SequenceAcknowledgement
497 </wsa:Action>
497 <wsrm:SequenceAcknowledgement>
497 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
497 <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
497 <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
497 </wsrm:SequenceAcknowledgement>
497 </S:Header>
497 <S:Body/>
497 </S:Envelope>

```

497 B.4 Retransmission

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

```

497 <?xml version="1.0" encoding="UTF-8"?>
497 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
497 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
497 xmlns:wsa="http://www.w3.org/2005/08/addressing">
497 <S:Header>
497 <wsa:MessageID>
497 http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
497 </wsa:MessageID>
497 <wsa:To>http://example.com/serviceB/123</wsa:To>
497 <wsa:From>
497 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
497 </wsa:From>

```

```

497 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
497 <wsrm:Sequence>
497   <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
497   <wsrm:MessageNumber>2</wsrm:MessageNumber>
497 </wsrm:Sequence>
497 <wsrm:AckRequested>
497   <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
497 </wsrm:AckRequested>
497 </S:Header>
497 <S:Body>
497   <!-- Some Application Data -->
497 </S:Body>
497 </S:Envelope>

```

497 B.5 Termination

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

```

497 <?xml version="1.0" encoding="UTF-8"?>
497 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
497 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
497 xmlns:wsa="http://www.w3.org/2005/08/addressing">
497   <S:Header>
497     <wsa:MessageID>
497       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
497     </wsa:MessageID>
497     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
497     <wsa:From>
497       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
497     </wsa:From>
497     <wsa:Action>
497       http://docs.oasis-open.org/ws-rx/wsrn/200604/SequenceAcknowledgement
497     </wsa:Action>
497     <wsrm:SequenceAcknowledgement>
497       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
497       <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
497     </wsrm:SequenceAcknowledgement>
497   </S:Header>
497   <S:Body/>
497 </S:Envelope>

```

497 Terminate Sequence

```

497 <?xml version="1.0" encoding="UTF-8"?>
497 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
497 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
497 xmlns:wsa="http://www.w3.org/2005/08/addressing">
497   <S:Header>
497     <wsa:MessageID>
497       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
497     </wsa:MessageID>
497     <wsa:To>http://example.com/serviceB/123</wsa:To>
497     <wsa:Action>
497       http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequence
497     </wsa:Action>
497     <wsa:From>
497       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
497     </wsa:From>
497   </S:Header>
497   <S:Body>
497     <wsrm:TerminateSequence>

```



```
497     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
497     </wsrm:TerminateSequence>
497 </S:Body>
497 </S:Envelope>
```

497 Terminate Sequence Response

```
497 <?xml version="1.0" encoding="UTF-8"?>
497 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
497   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
497   xmlns:wsa="http://www.w3.org/2005/08/addressing">
497   <S:Header>
497     <wsa:MessageID>
497       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
497     </wsa:MessageID>
497     <wsa:To>http://example.com/serviceA/789</wsa:To>
497     <wsa:Action>
497       http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequenceResponse
497     </wsa:Action>
498     <wsa:RelatesTo>
499       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
500     </wsa:RelatesTo>
501     <wsa:From>
502       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
503     </wsa:From>
504   </S:Header>
505   <S:Body>
506     <wsrm:TerminateSequenceResponse>
507       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
508     </wsrm:TerminateSequenceResponse>
509   </S:Body>
510 </S:Envelope>
```

511 C. WSDL

- 512 The ~~non~~-normative WSDL 1.1 definition for WS-ReliableMessaging is located at:
513 <http://docs.oasis-open.org/ws-rx/wsrn/200604/wsd/wsrn-1.1-wsd-200604.wsd>
514 The following non-normative copy is provided for reference.

```

515 <?xml version="1.0" encoding="utf-8"?>
516 <!--
517 OASIS takes no position regarding the validity or scope of any intellectual
518 property or other rights that might be claimed to pertain to the
519 implementation or use of the technology described in this document or the
520 extent to which any license under such rights might or might not be available;
521 neither does it represent that it has made any effort to identify any such
522 rights. Information on OASIS's procedures with respect to rights in OASIS
523 specifications can be found at the OASIS website. Copies of claims of rights
524 made available for publication and any assurances of licenses to be made
525 available, or the result of an attempt made to obtain a general license or
526 permission for the use of such proprietary rights by implementors or users of
527 this specification, can be obtained from the OASIS Executive Director.
528 OASIS invites any interested party to bring to its attention any copyrights,
529 patents or patent applications, or other proprietary rights which may cover
530 technology that may be required to implement this specification. Please
531 address the information to the OASIS Executive Director.
532 Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
533 This document and translations of it may be copied and furnished to others,
534 and derivative works that comment on or otherwise explain it or assist in its
535 implementation may be prepared, copied, published and distributed, in whole or
536 in part, without restriction of any kind, provided that the above copyright
537 notice and this paragraph are included on all such copies and derivative
538 works. However, this document itself does not be modified in any way, such as
539 by removing the copyright notice or references to OASIS, except as needed for
540 the purpose of developing OASIS specifications, in which case the procedures
541 for copyrights defined in the OASIS Intellectual Property Rights document must
542 be followed, or as required to translate it into languages other than English.
543 The limited permissions granted above are perpetual and will not be revoked by
544 OASIS or its successors or assigns.
545 This document and the information contained herein is provided on an "AS
546 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
547 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
548 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
549 FITNESS FOR A PARTICULAR PURPOSE.
550 -->
551 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
552 xmlns:xs="http://www.w3.org/2001/XMLSchema"
553 xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
554 open.org/ws-rx/wsr/200604" xmlns:tns="http://docs.oasis-open.org/ws-
555 rx/wsr/200604/wsdl" targetNamespace="http://docs.oasis-open.org/ws-
556 rx/wsr/200604/wsdl">
557   <wsdl:types>
558     <xs:schema
559       <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsr/200604"
560       schemaLocation="http://docs.oasis-open.org/ws-rx/wsr/200604/wsr-1.1-schema-
561       200604.xsd"/>
562     </xs:schema>
563   </wsdl:types>
564
565   <wsdl:message name="CreateSequence">
566     <wsdl:part name="create" element="rm:CreateSequence"/>
567   </wsdl:message>
568   <wsdl:message name="CreateSequenceResponse">
569     <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
570   </wsdl:message>
571   <wsdl:message name="CloseSequence">
572     <wsdl:part name="close" element="rm:CloseSequence"/>
573   </wsdl:message>
574   <wsdl:message name="CloseSequenceResponse">
575     <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
576   </wsdl:message>

```

```

576 <wsdl:message name="TerminateSequence">
577 <wsdl:part name="terminate" element="rm:TerminateSequence"/>
578 </wsdl:message>
579 <wsdl:message name="TerminateSequenceResponse">
580 <wsdl:part name="terminateResponse"
581 element="rm:TerminateSequenceResponse"/>
582 </wsdl:message>

583 <wsdl:portType name="SequenceAbstractPortType">
584 <wsdl:operation name="CreateSequence">
585 <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
586 open.org/ws-rx/wsrn/200604/CreateSequence"/>
587 <wsdl:output message="tns:CreateSequenceResponse"
588 wsa:Action="http://docs.oasis-open.org/ws-
589 rx/wsrn/200604/CreateSequenceResponse"/>
590 </wsdl:operation>
591 <wsdl:operation name="CloseSequence">
592 <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
593 open.org/ws-rx/wsrn/200604/CloseSequence"/>
594 <wsdl:output message="tns:CloseSequenceResponse"
595 wsa:Action="http://docs.oasis-open.org/ws-
596 rx/wsrn/200604/CloseSequenceResponse"/>
597 </wsdl:operation>
598 <wsdl:operation name="TerminateSequence">
599 <wsdl:input message="tns:TerminateSequence"
600 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequence"/>
601 <wsdl:output message="tns:TerminateSequenceResponse"
602 wsa:Action="http://docs.oasis-open.org/ws-
603 rx/wsrn/200604/TerminateSequenceResponse"/>
604 </wsdl:operation>
605 </wsdl:portType>

606 </wsdl:definitions><?xml version="1.0" encoding="utf-8"?>
607 <!--
608 OASIS takes no position regarding the validity or scope of any intellectual
609 property or other rights that might be claimed to pertain to the
610 implementation or use of the technology described in this document or the
611 extent to which any license under such rights might or might not be available;
612 neither does it represent that it has made any effort to identify any such
613 rights. Information on OASIS's procedures with respect to rights in OASIS
614 specifications can be found at the OASIS website. Copies of claims of rights
615 made available for publication and any assurances of licenses to be made
616 available, or the result of an attempt made to obtain a general license or
617 permission for the use of such proprietary rights by implementors or users of
618 this specification, can be obtained from the OASIS Executive Director.
619 OASIS invites any interested party to bring to its attention any copyrights,
620 patents or patent applications, or other proprietary rights which may cover
621 technology that may be required to implement this specification. Please
622 address the information to the OASIS Executive Director.
623 Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
624 This document and translations of it may be copied and furnished to others,
625 and derivative works that comment on or otherwise explain it or assist in its
626 implementation may be prepared, copied, published and distributed, in whole or
627 in part, without restriction of any kind, provided that the above copyright
628 notice and this paragraph are included on all such copies and derivative
629 works. However, this document itself does not be modified in any way, such as
630 by removing the copyright notice or references to OASIS, except as needed for
631 the purpose of developing OASIS specifications, in which case the procedures
632 for copyrights defined in the OASIS Intellectual Property Rights document must
633 be followed, or as required to translate it into languages other than English.
634 The limited permissions granted above are perpetual and will not be revoked by
635 OASIS or its successors or assigns.
636 This document and the information contained herein is provided on an "AS IS"

```

basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

```
-->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-open.org/ws-rx/wsr/200604" xmlns:tns="http://docs.oasis-open.org/ws-rx/wsr/200604/wsdl" targetNamespace="http://docs.oasis-open.org/ws-rx/wsr/200604/wsdl">

  <wsdl:types>
    <xs:schema>
      <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsr/200604"
schemaLocation="http://docs.oasis-open.org/ws-rx/wsr/200604/wsr-1.1-schema-200604.xsd"/>
    </xs:schema>
  </wsdl:types>

  <wsdl:message name="CreateSequence">
    <wsdl:part name="create" element="rm:CreateSequence"/>
  </wsdl:message>
  <wsdl:message name="CreateSequenceResponse">
    <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
  </wsdl:message>
  <wsdl:message name="CloseSequence">
    <wsdl:part name="close" element="rm:CloseSequence"/>
  </wsdl:message>
  <wsdl:message name="CloseSequenceResponse">
    <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
  </wsdl:message>
  <wsdl:message name="TerminateSequence">
    <wsdl:part name="terminate" element="rm:TerminateSequence"/>
  </wsdl:message>
  <wsdl:message name="TerminateSequenceResponse">
    <wsdl:part name="terminateResponse"
element="rm:TerminateSequenceResponse"/>
  </wsdl:message>
  <wsdl:message name="MakeConnection">
    <wsdl:part name="makConnection" element="rm:MakeConnection"/>
  </wsdl:message>

  <wsdl:portType name="SequenceAbstractPortType">
    <wsdl:operation name="CreateSequence">
      <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-open.org/ws-rx/wsr/200604/CreateSequence"/>
      <wsdl:output message="tns:CreateSequenceResponse"
wsa:Action="http://docs.oasis-open.org/ws-rx/wsr/200604/CreateSequenceResponse"/>
    </wsdl:operation>
    <wsdl:operation name="CloseSequence">
      <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-open.org/ws-rx/wsr/200604/CloseSequence"/>
      <wsdl:output message="tns:CloseSequenceResponse"
wsa:Action="http://docs.oasis-open.org/ws-rx/wsr/200604/CloseSequenceResponse"/>
    </wsdl:operation>
    <wsdl:operation name="TerminateSequence">
      <wsdl:input message="tns:TerminateSequence"
wsa:Action="http://docs.oasis-open.org/ws-rx/wsr/200604/TerminateSequence"/>
      <wsdl:output message="tns:TerminateSequenceResponse"
wsa:Action="http://docs.oasis-open.org/ws-
```

```
697 rx/wsrn/200604/TerminateSequenceResponse"/>
698 </wsdl:operation>
699 <wsdl:operation name="MakeConnection">
700 <wsdl:input message="tns:MakeConnection" wsa:Action="http://docs.oasis-
701 open.org/ws-rx/wsrn/200604/MakeConnection"/>
702 </wsdl:operation>
703 </wsdl:portType>
704 </wsdl:definitions>
```

D. State Tables

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

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

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

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	No action Terminated	No action Terminated	No action Terminated	No action Terminated	No action Terminated
Sequence Terminated Fault	N/A	N/A	No action Terminated	No action Terminated	No action Terminated	No action Terminated	No action Terminated	No Action Terminated
Terminate Sequence	N/A	N/A	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	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 SequenceTerminated Fault Terminated	Send SequenceTerminated Fault Terminated	Send SequenceTerminated Fault Terminated	Send SequenceTerminated Fault Terminated	Send SequenceTerminated Fault Terminated	N/A

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

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

711 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

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

E. Acknowledgments

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

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

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

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

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

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
wd-14	2006-06-07	Doug Davis	s/"none"/"full-uri"/ - per Marc Goodner
wd-14	2006-06-12	Doug Davis	Complete i106
wd-14	2006-06-12	Doug Davis	Issues 089 applied
wd-14	2006-06-12	Doug Davis	Fix for several RFC2119 keywords – per Anish

716 G. Notices

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

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

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

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

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

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