



1 Web Services Reliable Messaging 2 (WS-ReliableMessaging)

3 Working Draft 1~~3~~4, June 9, 2006

4 Document identifier:

5 wsrn-1.1-spec-wd-1~~3~~

6 Location:

7 Editors:

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

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

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

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

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

13 Contributors:

14 TBD

15 Abstract:

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

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

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

30 Status:

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

Table of Contents

1	Introduction.....	4
1.1	Goals and Requirements.....	4
1.1.1	Requirements.....	4
1.2	Notational Conventions.....	4
1.3	Namespace.....	4
1.4	Compliance.....	5
2	Reliable Messaging Model.....	6
2.1	Glossary.....	6
2.2	Protocol Preconditions.....	7
2.3	Protocol Invariants.....	7
2.4	Example Message Exchange.....	7
3	RM Protocol Elements.....	10
3.1	Sequence Creation.....	10
3.2	Closing A Sequence.....	13
3.3	Sequence Termination.....	15
3.4	Sequences.....	16
3.5	Request Acknowledgement.....	17
3.6	Sequence Acknowledgement.....	18
4	Faults.....	21
4.1	SequenceFault Element.....	22
4.2	Sequence Terminated.....	23
4.3	Unknown Sequence.....	23
4.4	Invalid Acknowledgement.....	24
4.5	Message Number Rollover.....	24
4.6	Create Sequence Refused.....	24
4.7	Sequence Closed.....	25
4.8	WSRM Required.....	25
5	Security Considerations.....	26
6	References.....	28
6.1	Normative.....	28
6.2	Non-Normative.....	28
A.	Schema.....	30
B.	Message Examples.....	35
B.1	Create Sequence.....	35
B.2	Initial Transmission.....	35
B.3	First Acknowledgement.....	37
B.4	Retransmission.....	37

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

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 delivery of messages. It defines a messaging protocol to identify, track, and manage the reliable delivery of messages between a source and a destination. It also defines a SOAP binding that is required for interoperability. Additional bindings may 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 may 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. 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 reliability features that enable ordered delivery, duplicate elimination, and guaranteed receipt for the RM Destination. It is expected that the Application Destination and RM Destination will implement as many of these or as few of these characteristics as necessary to implement the AD. Regardless of which of the reliability features 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 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.

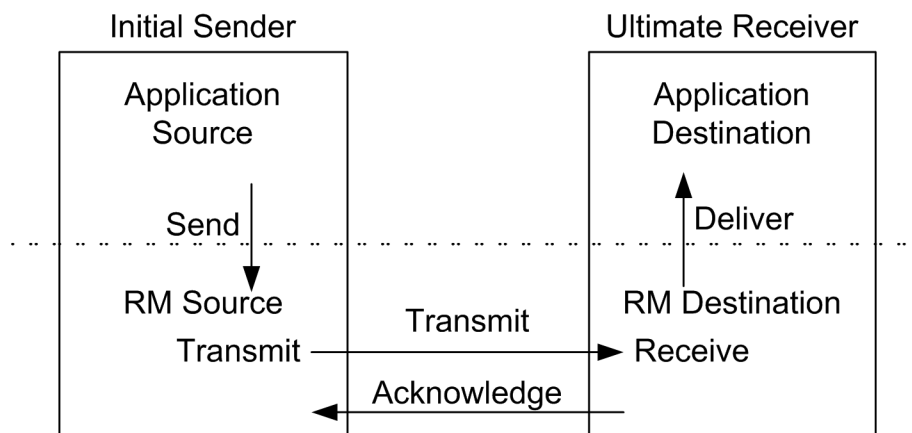


Figure 1: Reliable Messaging Model

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.

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

Application Source: The endpoint that sends a message.

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

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

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

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

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

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

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

167 2.2 Protocol Preconditions

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

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

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

172 2.3 Protocol Invariants

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

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

173 2.4 Example Message Exchange

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



Figure 2: The WS-ReliableMessaging Protocol

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

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

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

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

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

174 Now that the basic model has been outlined, the details of the elements used in this protocol are now
175 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

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

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

195 /wsrm:CreateSequence/wsrm:Expires

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

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

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

195 /wsrm:CreateSequence/wsrm:Offer

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

195 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier

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

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

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

195 /wsrm:CreateSequence/wsrm:Offer/wsrm:Endpoint

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

195 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires

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

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

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

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

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

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

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

198 /wsrm:CreateSequence/{any}

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

198 /wsrm:CreateSequence/@{any}

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

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

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

```
198 <wsrm:CreateSequenceResponse ...>
198   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
198   <wsrm:Expires> xs:duration </wsrm:Expires> ?
198   <wsrm:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?
198   <wsrm:IncompleteSequenceBehavior> wsrm:IncompleteSequenceBehaviorType
199 </wsrm:IncompleteSequenceBehavior> ?
198   <wsrm:Accept ...>
198     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
198     ...
198   </wsrm:Accept> ?
198   ...
198 </wsrm:CreateSequenceResponse>
```

198 /wsrm:CreateSequenceResponse

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

198 /wsrm:CreateSequenceResponse/wsrm:Identifier

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

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

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

198 /wsrm:CreateSequenceResponse/wsrm:Expires

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

198 /wsrm:CreateSequenceResponse/wsrm:Expires/@{any}

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

198 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval

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

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

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

198 /wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior
 198 This ~~optional~~[OPTIONAL](#) element, if present, specifies the behavior that the RM Destination will exhibit
 199 upon the closure of an incomplete sequence.

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

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

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

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

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

206 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo
 206 The RM Destination MUST include this element, of type `wsa:EndpointReferenceType` (as specified
 207 by WS-Addressing-[\[WS-Addressing\]](#)). The RM Source SHOULD send messages with
 208 <wsrm:SequenceAcknowledgement> header blocks related to the accepted Sequence to the
 209 referenced endpoint.

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

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

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

210 /wsrm:CreateSequenceResponse/@{any}

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

212 3.2 Closing A Sequence

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

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

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

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

227 The following exemplar defines the `CloseSequence` syntax:

```
227 <wsrm:CloseSequence ...>  
227   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
227   ...  
227 </wsrm:CloseSequence>
```

227 `/wsrm:CloseSequence`

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

230 `/wsrm:CloseSequence/wsrm:Identifier`

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

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

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

230 `/wsrm:CloseSequence/{any}`

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

230 /wsrm:CloseSequence@{any}

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

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

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

```
230 <wsrm:CloseSequenceResponse ...>  
230   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
230   ...  
230 </wsrm:CloseSequenceResponse>
```

230 /wsrm:CloseSequenceResponse

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

230 /wsrm:CloseSequenceResponse/wsrm:Identifier

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

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

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

230 /wsrm:CloseSequenceResponse/{any}

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

230 /wsrm:CloseSequenceResponse@{any}

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

230 3.3 Sequence Termination

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

231 The following exemplar defines the TerminateSequence syntax:

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


231
231

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

231 /wsrm:TerminateSequence

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

231 /wsrm:TerminateSequence/wsrm:Identifier

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

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

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

234 /wsrm:TerminateSequence/{any}

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

234 /wsrm:TerminateSequence/@{any}

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

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

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

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

234 /wsrm:TerminateSequenceResponse

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

234 /wsrm:TerminateSequenceResponse/wsrm:Identifier

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

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

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

237 /wsrm:TerminateSequenceResponse/{any}

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

237 /wsrm:TerminateSequenceResponse/@{any}

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

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

237 3.4 Sequences

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

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

238 A following exemplar defines its syntax:

```
238 <wsrm:Sequence ...>  
238   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
238   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>  
238   ...  
238 </wsrm:Sequence>
```

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

238 /wsrm:Sequence

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

238 /wsrm:Sequence/wsrm:Identifier

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

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

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

241 /wsrm:Sequence/wsrm:MessageNumber

241 The RM Source MUST include this element within any Sequence headers it creates. This element is of
242 type `wsrm:MessageNumberType`. It represents the ordinal position of the message within a Sequence.
243 Sequence message numbers start at 1 and monotonically increase [by 1](#) throughout the Sequence. If the

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

244 /wsrm:Sequence/{any}

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

244 /wsrm:Sequence/@{any}

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

244 The following example illustrates a Sequence header block.

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

244 3.5 Request Acknowledgement

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

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

254 The following exemplar defines its syntax:

```
254 <wsrm:AckRequested ...>  
254   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
  
254   ...  
254 </wsrm:AckRequested>
```

254 /wsrm:AckRequested

254 This element requests an acknowledgement for the identified Sequence.

254 /wsrm:AckRequested/wsrm:Identifier

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

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

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

254 /wsrm:AckRequested/{any}

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

254 /wsrm:AckRequested/@{any}

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

254 3.6 Sequence Acknowledgement

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

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

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

275 The following exemplar defines its syntax:

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

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

288 /wsrm:SequenceAcknowledgement

289 This element contains the Sequence acknowledgement information.

290 /wsrm:SequenceAcknowledgement/wsrm:Identifier

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

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

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

299 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

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

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

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

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

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

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

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

314 `/wsrm:SequenceAcknowledgement/wsrm:Final`

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

322 `/wsrm:SequenceAcknowledgement/wsrm:Nack`

323 The RM Destination MAY include this element within a `<wsrm:SequenceAcknowledgement>` header
324 block. If used, the RM Destination MUST set the value of this element to a `wsrm:MessageNumberType`
325 representing the `<wsrm:MessageNumber>` of an unreceived message in a Sequence. The RM
326 Destination MUST NOT include a `<wsrm:Nack>` element if a sibling
327 `<wsrm:AcknowledgementRange>` or `<wsrm:None>` element is also present as a child of
328 `<wsrm:SequenceAcknowledgement>`. Upon the receipt of a Nack, an RM Source SHOULD retransmit
329 the message identified by the Nack. The RM Destination MUST NOT issue a
330 `<wsrm:SequenceAcknowledgement>` containing a `<wsrm:Nack>` for a message that it has previously
331 acknowledged within a `<wsrm:AcknowledgementRange>`. The RM Source SHOULD ignore a
332 `<wsrm:SequenceAcknowledgement>` containing a `<wsrm:Nack>` for a message that has previously
333 been acknowledged within a `<wsrm:AcknowledgementRange>`.

334 /wsrm:SequenceAcknowledgement/wsrm:None

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

339 /wsrm:SequenceAcknowledgement/{any}

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

342 /wsrm:SequenceAcknowledgement/@{any}

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

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

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

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

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

```
353 <wsrm:SequenceAcknowledgement>  
354     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
355     <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
356     <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
357     <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
358 </wsrm:SequenceAcknowledgement>
```

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

```
360 <wsrm:SequenceAcknowledgement>  
361     <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
362     <wsrm:Nack>3</wsrm:Nack>  
363 </wsrm:SequenceAcknowledgement>
```

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 ~~used in the message WS-Addressing the version of in below.~~ The value from the ~~current~~

```
http://schemas.xmlsoap.org/wsdocs.oasis-open.org/ws-  
rm/2004/08/08/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>  
    /08/addressing/fault4/200schemas.xmlsoap.org/wshttp://http://docs.oasis-  
  </wsa:Action>  
  <!-- Headers elided for clarity. -->  
</S:Header>  
<S:Body>  
  <S:Fault>  
    <S:Code>  
      <S:Value> [Code] </S:Value>
```

```

365     <S:Subcode>
365         <S:Value> [Subcode] </S:Value>
365     </S:Subcode>
365 </S:Code>
365 <S:Reason>
365     <S:Text xml:lang="en"> [Reason] </S:Text>
365 </S:Reason>
365 <S:Detail>
365     [Detail]
365     ...
365 </S:Detail>
365 </S:Fault>
365 </S:Body>
365 </S:Envelope>

```

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

```

365 <S11:Envelope>
365   <S11:Header>
365     <wsrm:SequenceFault>
365       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
365       <wsrm:Detail> [Detail] </wsrm:Detail>
365       ...
365     </wsrm:SequenceFault>
365   <!-- Headers elided for clarity. -->
365 </S11:Header>
365 <S11:Body>
365   <S11:Fault>
365     <faultcode> [Code] </faultcode>
365     <faultstring> [Reason] </faultstring>
365   </S11:Fault>
365 </S11:Body>
365 </S11:Envelope>

```

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

```

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

```

365 4.1 SequenceFault Element

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

366 The following exemplar defines its syntax:

```

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

```



```
366     ...
366 </wsrm:SequenceFault>
```

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

366 `/wsrm:SequenceFault`

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

366 `/wsrm:SequenceFault/wsrm:FaultCode`

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

366 `/wsrm:SequenceFault/wsrm:Detail`

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

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

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

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

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

366 `/wsrm:SequenceFault/{any}`

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

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

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

366 4.2 Sequence Terminated

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

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

367 Properties:

367 [Code] Sender or Receiver

367 [Subcode] `wsrm:SequenceTerminated`

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

367 [Detail]

```
367 <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 shall 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.

371 [Detail]

371 `xs:any`

371 4.7 Sequence Closed

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

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

372 Properties:

372 [Code] Sender

372 [Subcode] wsrn:SequenceClosed

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

372 [Detail]

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

372 4.8 WSRM Required

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

373 Properties:

373 [Code] Sender

373 [Subcode] wsrn:WSRMRequired

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

373 [Detail]

373 `xs:any`

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.

- 384 • **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by comparing
385 secured policies – see WS-Policy and WS-SecurityPolicy).
- 384 • **Authentication** – Authentication is established using the mechanisms described in WS-Security
385 and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- 384 • **Accountability** – Accountability is a function of the type of and string of the key and algorithms
385 being used. In many cases, a strong symmetric key provides sufficient accountability. However, in
386 some environments, strong PKI signatures are required.
- 384 • **Availability** – All reliable messaging services are subject to a variety of availability attacks. Replay
385 detection is a common attack and it is recommended that this be addressed by the mechanisms
386 described in WS-Security. (Note that because of legitimate message replays, detection should
387 include a differentiator besides message id such as a timestamp). Other attacks, such as network-
388 level denial of service attacks are harder to avoid and are outside the scope of this specification.
389 That said, care should be taken to ensure that minimal state is saved prior to any authenticating
390 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.

[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 2006~~~~D. Box, et al, "[Web Services Policy Framework \(WS-Policy\)](#)," September 2004.~~

[WS-PolicyAttachment]

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

416 **[WS-Security]**

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

419 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "[OASIS Web Services Security:](#)
420 [SOAP Message Security 1.1 \(WS-Security 2004\)](#)", OASIS Standard 200602, February 2006.

421 **[RTTM]**

421 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May
422 1992.

423 **[SecurityPolicy]**

423 G. Della-Libra, et. al. "[Web Services Security Policy Language \(WS-SecurityPolicy\)](#)", July 2005

424 **[SecureConversation]**

425 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," February
426 2005.

427 **[Trust]**

427 S. Anderson, et al, "[Web Services Trust Language \(WS-Trust\)](#)," February 2005.

427 **A. Schema**

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

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

430 The following copy is provided for reference.

```

430 <?xml version="1.0" encoding="UTF-8"?>
431 <!--
432 OASIS takes no position regarding the validity or scope of any intellectual
433 property or other rights that might be claimed to pertain to the
434 implementation or use of the technology described in this document or the
435 extent to which any license under such rights might or might not be available;
436 neither does it represent that it has made any effort to identify any such
437 rights. Information on OASIS's procedures with respect to rights in OASIS
438 specifications can be found at the OASIS website. Copies of claims of rights
439 made available for publication and any assurances of licenses to be made
440 available, or the result of an attempt made to obtain a general license or
441 permission for the use of such proprietary rights by implementors or users of
442 this specification, can be obtained from the OASIS Executive Director.
443 OASIS invites any interested party to bring to its attention any copyrights,
444 patents or patent applications, or other proprietary rights which may cover
445 technology that may be required to implement this specification. Please
446 address the information to the OASIS Executive Director.
447 Copyright © OASIS Open 2002-2006. All Rights Reserved.
448 This document and translations of it may be copied and furnished to others,
449 and derivative works that comment on or otherwise explain it or assist in its
450 implementation may be prepared, copied, published and distributed, in whole or
451 in part, without restriction of any kind, provided that the above copyright
452 notice and this paragraph are included on all such copies and derivative
453 works. However, this document itself does not be modified in any way, such as
454 by removing the copyright notice or references to OASIS, except as needed for
455 the purpose of developing OASIS specifications, in which case the procedures
456 for copyrights defined in the OASIS Intellectual Property Rights document must
457 be followed, or as required to translate it into languages other than English.
458 The limited permissions granted above are perpetual and will not be revoked by
459 OASIS or its successors or assigns.
460 This document and the information contained herein is provided on an "AS
461 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
462 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
463 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
464 FITNESS FOR A PARTICULAR PURPOSE.
465 -->
466 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
467 xmlns:wsa="http://www.w3.org/2005/08/addressing"
468 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200604"
469 targetNamespace="http://docs.oasis-open.org/ws-rx/wsm/200604"
470 elementFormDefault="qualified" attributeFormDefault="unqualified">
471   <xs:import namespace="http://www.w3.org/2005/08/addressing"
472   schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
473   <!-- Protocol Elements -->
474   <xs:complexType name="SequenceType">
475     <xs:sequence>
476       <xs:element ref="wsm:Identifier"/>
477       <xs:element name="MessageNumber" type="wsm:MessageNumberType"/>
478       <xs:any namespace="##other" processContents="lax" minOccurs="0"
479 maxOccurs="unbounded"/>
480     </xs:sequence>
481     <xs:anyAttribute namespace="##other" processContents="lax"/>
482   </xs:complexType>
483   <xs:element name="Sequence" type="wsm:SequenceType"/>
484   <xs:element name="SequenceAcknowledgement">
485     <xs:complexType>
486       <xs:sequence>
487         <xs:element ref="wsm:Identifier"/>
488         <xs:choice>
489           <xs:sequence>
490             <xs:choice>
491               <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
492                 <xs:complexType>

```



```

430         <xs:sequence/>
431         <xs:attribute name="Upper" type="xs:unsignedLong"
432 use="required"/>
433         <xs:attribute name="Lower" type="xs:unsignedLong"
434 use="required"/>
435         <xs:anyAttribute namespace="##other" processContents="lax"/>
436     </xs:complexType>
437 </xs:element>
438     <xs:element name="None" minOccurs="0">
439         <xs:complexType>
440             <xs:sequence/>
441         </xs:complexType>
442     </xs:element>
443 </xs:choice>
444     <xs:element name="Final" minOccurs="0">
445         <xs:complexType>
446             <xs:sequence/>
447         </xs:complexType>
448     </xs:element>
449 </xs:sequence>
450     <xs:element name="Nack" type="xs:unsignedLong"
451 minOccurs="unbounded"/>
452 </xs:choice>
453     <xs:any namespace="##other" processContents="lax" minOccurs="0"
454 minOccurs="unbounded"/>
455 </xs:sequence>
456     <xs:anyAttribute namespace="##other" processContents="lax"/>
457 </xs:complexType>
458 </xs:element>
459 <xs:complexType name="AckRequestedType">
460     <xs:sequence>
461         <xs:element ref="wsrm:Identifier"/>
462         <xs:any namespace="##other" processContents="lax" minOccurs="0"
463 minOccurs="unbounded"/>
464     </xs:sequence>
465     <xs:anyAttribute namespace="##other" processContents="lax"/>
466 </xs:complexType>
467 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
468 <xs:element name="Identifier">
469     <xs:complexType>
470         <xs:annotation>
471             <xs:documentation>
472                 This type is for elements whose [children] is an anyURI and can have
473 arbitrary attributes.
474             </xs:documentation>
475         </xs:annotation>
476         <xs:simpleContent>
477             <xs:extension base="xs:anyURI">
478                 <xs:anyAttribute namespace="##other" processContents="lax"/>
479             </xs:extension>
480         </xs:simpleContent>
481     </xs:complexType>
482 </xs:element>
483 <xs:simpleType name="MessageNumberType">
484     <xs:restriction base="xs:unsignedLong">
485         <xs:minInclusive value="1"/>
486         <xs:maxInclusive value="9223372036854775807"/>
487     </xs:restriction>
488 </xs:simpleType>
489 <!-- Fault Container and Codes -->
490 <xs:simpleType name="FaultCodes">
491     <xs:restriction base="xs:QName">
492         <xs:enumeration value="wsrm:SequenceTerminated"/>

```

```

430     <xs:enumeration value="wsrm:UnknownSequence"/>
431     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
432     <xs:enumeration value="wsrm:MessageNumberRollover"/>
433     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
434     <xs:enumeration value="wsrm:SequenceClosed"/>
435     <xs:enumeration value="wsrm:WSRMRequired"/>
436   </xs:restriction>
437 </xs:simpleType>
438 <xs:complexType name="SequenceFaultType">
439   <xs:sequence>
440     <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
441     <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
442     <xs:any namespace="##other" processContents="lax" minOccurs="0"
443 maxOccurs="unbounded"/>
444   </xs:sequence>
445   <xs:anyAttribute namespace="##other" processContents="lax"/>
446 </xs:complexType>
447 <xs:complexType name="DetailType">
448   <xs:sequence>
449     <xs:any namespace="##other" processContents="lax" minOccurs="0"
450 maxOccurs="unbounded"/>
451   </xs:sequence>
452   <xs:anyAttribute namespace="##other" processContents="lax"/>
453 </xs:complexType>
454 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
455 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
456 <xs:element name="CreateSequenceResponse"
457 type="wsrm:CreateSequenceResponseType"/>
458 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
459 <xs:element name="CloseSequenceResponse"
460 type="wsrm:CloseSequenceResponseType"/>
461 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
462 <xs:element name="TerminateSequenceResponse"
463 type="wsrm:TerminateSequenceResponseType"/>
464 <xs:complexType name="CreateSequenceType">
465   <xs:sequence>
466     <xs:element ref="wsrm:AcksTo"/>
467     <xs:element ref="wsrm:Expires" minOccurs="0"/>
468     <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
469     <xs:any namespace="##other" processContents="lax" minOccurs="0"
470 maxOccurs="unbounded">
471       <xs:annotation>
472         <xs:documentation>
473           It is the authors intent that this extensibility be used to
474 transfer a Security Token Reference as defined in WS-Security.
475         </xs:documentation>
476       </xs:annotation>
477     </xs:any>
478   </xs:sequence>
479   <xs:anyAttribute namespace="##other" processContents="lax"/>
480 </xs:complexType>
481 <xs:complexType name="CreateSequenceResponseType">
482   <xs:sequence>
483     <xs:element ref="wsrm:Identifier"/>
484     <xs:element ref="wsrm:Expires" minOccurs="0"/>
485     <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>

```

```

430     <xs:element name="IncompleteSequenceBehaviour"
431 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
432     <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
433     <xs:any namespace="##other" processContents="lax" minOccurs="0"
434 maxOccurs="unbounded"/>
435   </xs:sequence>
436   <xs:anyAttribute namespace="##other" processContents="lax"/>
437 </xs:complexType>
438 <xs:complexType name="CloseSequenceType">
439   <xs:sequence>
440     <xs:element ref="wsrm:Identifier"/>
441     <xs:any namespace="##other" processContents="lax" minOccurs="0"
442 maxOccurs="unbounded"/>
443   </xs:sequence>
444   <xs:anyAttribute namespace="##other" processContents="lax"/>
445 </xs:complexType>
446 <xs:complexType name="CloseSequenceResponseType">
447   <xs:sequence>
448     <xs:element ref="wsrm:Identifier"/>
449     <xs:any namespace="##other" processContents="lax" minOccurs="0"
450 maxOccurs="unbounded"/>
451   </xs:sequence>
452   <xs:anyAttribute namespace="##other" processContents="lax"/>
453 </xs:complexType>
454 <xs:complexType name="TerminateSequenceType">
455   <xs:sequence>
456     <xs:element ref="wsrm:Identifier"/>
457     <xs:any namespace="##other" processContents="lax" minOccurs="0"
458 maxOccurs="unbounded"/>
459   </xs:sequence>
460   <xs:anyAttribute namespace="##other" processContents="lax"/>
461 </xs:complexType>
462 <xs:complexType name="TerminateSequenceResponseType">
463   <xs:sequence>
464     <xs:element ref="wsrm:Identifier"/>
465     <xs:any namespace="##other" processContents="lax" minOccurs="0"
466 maxOccurs="unbounded"/>
467   </xs:sequence>
468   <xs:anyAttribute namespace="##other" processContents="lax"/>
469 </xs:complexType>
470 <xs:element name="AcksTo"

```

```

430     type="wsa:EndpointReferenceType"/>
431     <xs:complexType name="OfferType">
432         <xs:sequence>
433             <xs:element ref="wsrm:Identifier"/>
434             <xs:element ref="wsrm:Expires" minOccurs="0"/>
435             <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
436             <xs:any namespace="##other" processContents="lax" minOccurs="0"
437 maxOccurs="unbounded"/>
438         </xs:sequence>
439         <xs:anyAttribute namespace="##other" processContents="lax"/>
440     </xs:complexType>
441     <xs:complexType name="AcceptType">
442         <xs:sequence>
443             <xs:element ref="wsrm:AcksTo"/>
444             <xs:any namespace="##other" processContents="lax" minOccurs="0"
445 maxOccurs="unbounded"/>
446         </xs:sequence>
447         <xs:anyAttribute namespace="##other" processContents="lax"/>
448     </xs:complexType>
449     <xs:element name="Expires">
450         <xs:complexType>
451             <xs:simpleContent>
452                 <xs:extension base="xs:duration">
453                     <xs:anyAttribute namespace="##other" processContents="lax"/>
454                 </xs:extension>
455             </xs:simpleContent>
456         </xs:complexType>
457     </xs:element>
458     <xs:element name="AcknowledgementInterval">
459         <xs:complexType>
460             <xs:sequence/>
461             <xs:attribute name="Milliseconds" type="xs:unsignedLong"
462 use="required"/>
463             <xs:anyAttribute namespace="##other" processContents="lax"/>
464         </xs:complexType>
465     </xs:element>
430     <xs:simpleType name="IncompleteSequenceBehaviorType">
431         <xs:restriction base="xs:string">
432             <xs:enumeration value="DiscardEntireSequence"/>
433             <xs:enumeration value="DiscardFollowingFirstGap"/>
434             <xs:enumeration value="NoDiscard"/>
435         </xs:restriction>
436     </xs:simpleType>
437 </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:

430 Message 1

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

430 Message 2

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

430 Message 3

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

```

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

```

430 B.3 First Acknowledgement

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

```

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

```

430 B.4 Retransmission

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

```

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

```

```

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

```

B.5 Termination

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

```

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

```

Terminate Sequence

```

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

```



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

430 Terminate Sequence Response

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

430 C. WSDL

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

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

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

```

431 <?xml version="1.0" encoding="utf-8"?>
432 <!--
433 OASIS takes no position regarding the validity or scope of any intellectual
434 property or other rights that might be claimed to pertain to the
435 implementation or use of the technology described in this document or the
436 extent to which any license under such rights might or might not be available;
437 neither does it represent that it has made any effort to identify any such
438 rights. Information on OASIS's procedures with respect to rights in OASIS
439 specifications can be found at the OASIS website. Copies of claims of rights
440 made available for publication and any assurances of licenses to be made
441 available, or the result of an attempt made to obtain a general license or
442 permission for the use of such proprietary rights by implementors or users of
443 this specification, can be obtained from the OASIS Executive Director.
444 OASIS invites any interested party to bring to its attention any copyrights,
445 patents or patent applications, or other proprietary rights which may cover
446 technology that may be required to implement this specification. Please
447 address the information to the OASIS Executive Director.
448 Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
449 This document and translations of it may be copied and furnished to others,
450 and derivative works that comment on or otherwise explain it or assist in its
451 implementation may be prepared, copied, published and distributed, in whole or
452 in part, without restriction of any kind, provided that the above copyright
453 notice and this paragraph are included on all such copies and derivative
454 works. However, this document itself does not be modified in any way, such as
455 by removing the copyright notice or references to OASIS, except as needed for
456 the purpose of developing OASIS specifications, in which case the procedures
457 for copyrights defined in the OASIS Intellectual Property Rights document must
458 be followed, or as required to translate it into languages other than English.
459 The limited permissions granted above are perpetual and will not be revoked by
460 OASIS or its successors or assigns.
461 This document and the information contained herein is provided on an "AS
462 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
463 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
464 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
465 FITNESS FOR A PARTICULAR PURPOSE.
466 -->
467 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
468 xmlns:xs="http://www.w3.org/2001/XMLSchema"
469 xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
470 open.org/ws-rx/wsr/200604" xmlns:tns="http://docs.oasis-open.org/ws-
471 rx/wsr/200604/wsdl" targetNamespace="http://docs.oasis-open.org/ws-
472 rx/wsr/200604/wsdl">
473
474   <wsdl:types>
475     <xs:schema
476       <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsr/200604"
477       schemaLocation="http://docs.oasis-open.org/ws-rx/wsr/200604/wsr-1.1-schema-
478       200604.xsd"/>
479     </xs:schema>
480   </wsdl:types>
481
482   <wsdl:message name="CreateSequence">
483     <wsdl:part name="create" element="rm:CreateSequence"/>
484   </wsdl:message>
485   <wsdl:message name="CreateSequenceResponse">
486     <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
487   </wsdl:message>
488   <wsdl:message name="CloseSequence">
489     <wsdl:part name="close" element="rm:CloseSequence"/>
490   </wsdl:message>
491   <wsdl:message name="CloseSequenceResponse">
492     <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
493   </wsdl:message>

```

```

431     <wsdl:message name="TerminateSequence">
432         <wsdl:part name="terminate" element="rm:TerminateSequence"/>
433     </wsdl:message>
434     <wsdl:message name="TerminateSequenceResponse">
435         <wsdl:part name="terminateResponse"
436 element="rm:TerminateSequenceResponse"/>
437     </wsdl:message>

438     <wsdl:portType name="SequenceAbstractPortType">
439         <wsdl:operation name="CreateSequence">
440             <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
441 open.org/ws-rx/wsrn/200604/CreateSequence"/>
442             <wsdl:output message="tns:CreateSequenceResponse"
443 wsa:Action="http://docs.oasis-open.org/ws-
444 rx/wsrn/200604/CreateSequenceResponse"/>
445         </wsdl:operation>
446         <wsdl:operation name="CloseSequence">
447             <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
448 open.org/ws-rx/wsrn/200604/CloseSequence"/>
449             <wsdl:output message="tns:CloseSequenceResponse"
450 wsa:Action="http://docs.oasis-open.org/ws-
451 rx/wsrn/200604/CloseSequenceResponse"/>
452         </wsdl:operation>
453         <wsdl:operation name="TerminateSequence">
454             <wsdl:input message="tns:TerminateSequence"
455 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequence"/>
456             <wsdl:output message="tns:TerminateSequenceResponse"
457 wsa:Action="http://docs.oasis-open.org/ws-
458 rx/wsrn/200604/TerminateSequenceResponse"/>
459         </wsdl:operation>
460     </wsdl:portType>

461 </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

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

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

434 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

435 In Table 3 above, the rows consists of events that occur at the RM Destination throughout the lifetime of
436 an RM Sequence and the columns consists of various RM Destination states. -Each cell in the table above
437 lists the action that the RM Destination takes on occurrence of a particular event and the next state that it
438 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

G. Notices

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

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

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

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

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

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