



# Web Services Reliable Messaging (WS-ReliableMessaging)

Committee Draft ~~05, February 1, 2007~~ ~~4, August 11, 2006~~

## Document identifier:

wsrn-1.1-spec-cd-054

## Location:

<http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-spec-cd-05608/wsrn-1.1-spec-cd-04.pdf>

## Editors:

Doug Davis, IBM <[dug@us.ibm.com](mailto:dug@us.ibm.com)>  
Anish Karmarkar, Oracle <[Anish.Karmarkar@oracle.com](mailto:Anish.Karmarkar@oracle.com)>  
Gilbert Pilz, BEA <[gpilz@bea.com](mailto:gpilz@bea.com)>  
Steve Winkler, SAP <[steve.winkler@sap.com](mailto:steve.winkler@sap.com)>  
Ümit Yalçınalp, SAP <[umit.yalcinalp@sap.com](mailto:umit.yalcinalp@sap.com)>

## Contributors:

See the Acknowledgments (Appendix E).

## Abstract:

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

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

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

## Status:

This document was last revised or approved by the WS-RX on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule. Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/ws-rx>. For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/ws-rx/ipr.php>). The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/ws-rx>.

# Table of Contents

42		
43	1	Introduction.....4
44	1.1	Notational Conventions.....4
45	1.2	Namespace.....5
46	1.3	Conformance.....5
47	2	Reliable Messaging Model.....6
48	2.1	Glossary.....6
49	2.2	Protocol Preconditions.....7
50	2.3	Protocol Invariants.....8
51	2.4	Delivery Assurances.....8
52	2.5	Example Message Exchange.....9
53	3	RM Protocol Elements.....11
54	3.1	Considerations on the Use of Extensibility Points.....11
55	3.2	Considerations on the Use of "Piggy-Backing".....11
56	3.3	Composition with WS-Addressing.....11
57	3.4	Sequence Creation.....12
58	3.5	Closing A Sequence.....16
59	3.6	Sequence Termination.....18
60	3.7	Sequences.....20
61	3.8	Request Acknowledgement.....21
62	3.9	Sequence Acknowledgement.....22
63	4	Faults.....25
64	4.1	SequenceFault Element.....26
65	4.2	Sequence Terminated.....27
66	4.3	Unknown Sequence.....27
67	4.4	Invalid Acknowledgement.....28
68	4.5	Message Number Rollover.....28
69	4.6	Create Sequence Refused.....29
70	4.7	Sequence Closed.....29
71	4.8	WSRM Required.....30
72	5	Security Threats and Countermeasures.....31
73	5.1	Threats and Countermeasures.....31
74	5.2	Security Solutions and Technologies.....33
75	6	Securing Sequences.....37
76	6.1	Securing Sequences Using WS-Security.....37
77	6.2	Securing Sequences Using SSL/TLS.....38
78	7	References.....40
79	7.1	Normative.....40

80	7.2 <a href="#">Non-Normative</a> .....	41
81	Appendix A. Schema.....	43
82	Appendix B. WSDL.....	48
83	Appendix C. Message Examples.....	50
84	Appendix C.1 Create Sequence.....	50
85	Appendix C.2 Initial Transmission.....	50
86	Appendix C.3 First Acknowledgement.....	52
87	Appendix C.4 Retransmission.....	52
88	Appendix C.5 Termination.....	53
89	Appendix D. State Tables.....	55
90	Appendix E. Acknowledgments.....	60
91	Appendix F. Revision History.....	61
92	Appendix G. Notices.....	67

# 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 transfer of messages. It defines a messaging protocol to identify, track, and manage the reliable transfer of messages between a source and a destination. It also defines a SOAP binding that is required for interoperability. Additional bindings can 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 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 1.2) are used to indicate the namespace of the element being defined.

Elements and Attributes defined by this specification are referred to in the text of this document using XPath 1.0 [XPath 1.0] expressions. Extensibility points are referred to using an extended version of this syntax:

- An element extensibility point is referred to using {any} in place of the element name. This indicates that any element name can be used, from any namespace other than the wsrn: namespace.
- An attribute extensibility point is referred to using @{any} in place of the attribute name. This indicates that any attribute name can be used, from any namespace other than the wsrn: namespace.

## 1.2 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/200702608>

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

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

Table 1

Prefix	Namespace
S	(Either SOAP 1.1 or 1.2)
S11	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>
S12	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>
wsrn	<a href="http://docs.oasis-open.org/ws-rx/wsrn/200702">http://docs.oasis-open.org/ws-rx/wsrn/200702</a>
wsa	<a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a>
wsaw	<a href="http://www.w3.org/2006/05/addressing/wsdl">http://www.w3.org/2006/05/addressing/wsdl</a>
wsse	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

The normative schema for WS-ReliableMessaging can be found linked from the namespace document that is located at the namespace URI specified above.

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

## 1.3 Conformance

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

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

## 2 Reliable Messaging Model

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

The WS-ReliableMessaging specification defines an interoperable protocol that enables a Reliable Messaging (RM) Source to accurately determine the disposition of each message it Transmits as perceived by the RM Destination, so as to allow it to resolve any in-doubt status regarding receipt of the message Transmitted. The protocol also enables an RM Destination to efficiently determine which of those messages it Receives have been previously Received, enabling it to filter out duplicate message transmissions caused by the retransmission, by the RM Source, of [an](#) unacknowledged message. It also enables an RM Destination to Deliver the messages it Receives to the Application Destination in the order in which they were sent by an Application Source, in the event that they are Received out of order. Note that this specification places no restriction on the scope of the RM Source or RM Destination entities. For example, either can span multiple WSDL Ports or Endpoints.

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

Figure 1 below illustrates the entities and events in a simple reliable exchange of messages. First, the Application Source Sends a message for reliable transfer. The Reliable Messaging Source accepts the message and Transmits it one or more times. After accepting 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.

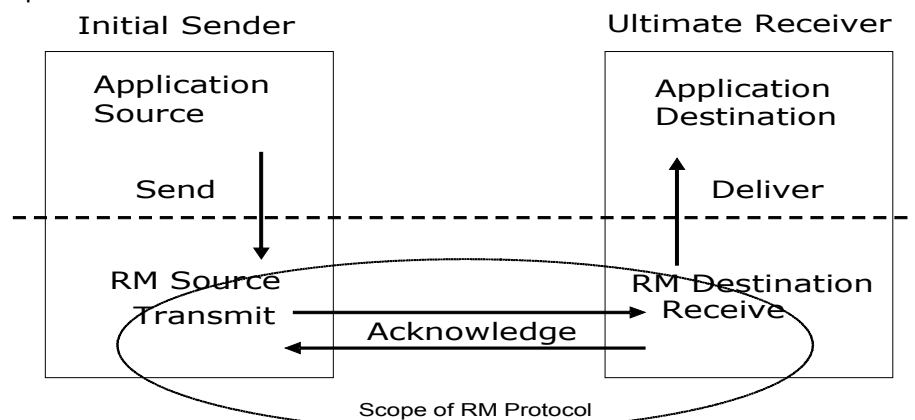


Figure 1: Reliable Messaging Model

### 2.1 Glossary

The following definitions are used throughout this specification:

**Accept:** The act of qualifying a message by the RM Destination such that it becomes eligible for Delivery and acknowledgement.

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

184 **Acknowledgement Message:** A message containing a `SequenceAcknowledgement` header block.  
185 Acknowledgement Messages may or may not contain a SOAP body.

186 **Acknowledgement Request:** A message containing an `AckRequested` header. Acknowledgement  
187 Requests may or may not contain a SOAP body.

188 **Application Destination:** The Endpoint to which a message is Delivered.

189 **Application Source:** The Endpoint that Sends a message.

190 **Back-channel:** When the underlying transport provides a mechanism to return a transport-protocol  
191 specific response, capable of carrying a SOAP message, without initiating a new connection, this  
192 specification refers to this mechanism as a back-channel.  
193 **Deliver:** The act of transferring a message from the RM Destination to the Application Destination.

194 **Deliver:** The act of transferring responsibility for a message from the RM Destination to the Application  
195 Destination.

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

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

202 **Receive:** The act of reading a message from a network connection and accepting it.

203 **RM Destination:** The Endpoint that Receives messages Transmitted reliably from an RM Source.

204 **RM Protocol Header Block:** One of `Sequence`, `SequenceAcknowledgement`, or `AckRequested`.

205 **RM Source:** The Endpoint that Transmits messages reliably to an RM Destination.

206 **Send:** The act of transferring a message from the Application Source to the RM Source for reliable  
207 transfer.

208 **Sequence Lifecycle Message:** A message that contains one of: `CreateSequence`,  
209 `CreateSequenceResponse`, `CloseSequence`, `CloseSequenceResponse`, `TerminateSequence`,  
210 `TerminateSequenceResponse` as the child element of the SOAP body element.

211 **Sequence Traffic Message:** A message containing a `Sequence` header block.

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

## 213 2.2 Protocol Preconditions

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

- 216 • For any single message exchange the RM Source MUST have an endpoint reference that uniquely  
217 identifies the RM Destination Endpoint.
- 218 • The RM Source MUST have successfully created a Sequence with the RM Destination.
- 219 • The RM Source MUST be capable of formulating messages that adhere to the RM Destination's  
220 policies.

- 221 • If a secure exchange of messages is REQUIRED, then the RM Source and RM Destination MUST  
222 have a security context.

## 223 2.3 Protocol Invariants

224 During the lifetime of a Sequence, the following invariants are REQUIRED for correctness:

- 225 • The RM Source MUST assign each message within a Sequence a message number (defined  
226 below) beginning at 1 and increasing by exactly 1 for each subsequent message. These numbers  
227 MUST be assigned in the same order in which messages are sent by the Application Source.
- 228 • Within every Acknowledgement Message it issues, the RM Destination MUST include one or more  
229 AcknowledgementRange child elements that contain, in their collective ranges, the message  
230 number of every message accepted by the RM Destination. The RM Destination MUST exclude, in  
231 the AcknowledgementRange elements, the message numbers of any messages it has not  
232 accepted. If no messages have been received the RM Destination MUST return None instead of an  
233 AcknowledgementRange(s). The RM Destination MAY transmit a Nack for a specific message  
234 or messages instead of an AcknowledgementRange(s).
- 235 • While the Sequence is not closed or terminated, the RM Source SHOULD retransmit  
236 unacknowledged messages.

## 237 2.4 Delivery Assurances

238 This section defines a number of Delivery Assurance assertions, which can be supported by RM Sources  
239 and RM Destinations. These assertions can be specified as policy assertions using the WS-Policy  
240 framework [[WS-Policy]]. For details on this see the WSRM Policy specification [WS-RM Policy].

### 241 AtLeastOnce

242 Each message is to be delivered at least once, or else an error MUST be raised by the RM Source and/or  
243 RM Destination. The requirement on an RM Source is that it SHOULD retry transmission of every  
244 message sent by the Application Source until it receives an acknowledgement from the RM Destination.  
245 The requirement on the RM Destination is that it SHOULD retry the transfer to the Application Destination  
246 of any message that it accepts from the RM Source, until that message has been successfully delivered.  
247 There is no requirement for the RM Destination to apply duplicate message filtering.

### 248 AtMostOnce

249 Each message is to be delivered at most once. The RM Source MAY retry transmission of  
250 unacknowledged messages, but is NOT REQUIRED to do so. The requirement on the RM Destination is  
251 that it MUST filter out duplicate messages, i.e. that it MUST NOT deliver a duplicate of a message that  
252 has already been delivered.

### 253 ExactlyOnce

254 Each message is to be delivered exactly once: if a message cannot be delivered then an error MUST be  
255 raised by the RM Source and/or RM Destination. The requirement on an RM Source is that it SHOULD  
256 retry transmission of every message sent by the Application Source until it receives an acknowledgement  
257 from the RM Destination. The requirement on the RM Destination is that it SHOULD retry the transfer to  
258 the Application Destination of any message that it accepts from the RM Source until that message has  
259 been successfully delivered, and that it MUST NOT deliver a duplicate of a message that has already  
260 been delivered.

### 261 InOrder



262 Messages from each individual sequence are to be delivered in the same order they have been sent by  
 263 the Application Source. The requirement on an RM Source is that it MUST ensure that the ordinal position  
 264 of each message in the sequence (as indicated by a message sequence number) is consistent with the  
 265 order in which the messages have been sent from the Application Source. The requirement on the RM  
 266 Destination is that it MUST deliver received messages for each sequence in the order indicated by the  
 267 message numbering. This DeliveryAssurance can be used in combination with any of the AtLeastOnce,  
 268 AtMostOnce or ExactlyOnce assertions, and the requirements of those assertions MUST also be met. In  
 269 particular if the AtLeastOnce or ExactlyOnce assertion applies and the RM Destination detects a gap in  
 270 the sequence then the RM Destination MUST NOT deliver any subsequent messages from that sequence  
 271 until the missing messages are received or until the sequence is closed.

## 272 2.5 Example Message Exchange

273 Figure 2 illustrates a possible message exchange between two reliable messaging Endpoints A and B.



Figure 2: The WS-ReliableMessaging Protocol

- 274 1. The protocol preconditions are established. These include policy exchange, endpoint resolution,  
 275 and establishing trust.
- 276 2. The RM Source requests creation of a new Sequence.
- 277 3. The RM Destination creates a new Sequence and returns its unique identifier.

- 278 4. The RM Source begins Transmitting messages in the Sequence beginning with MessageNumber 1.  
279 In the figure above, the RM Source sends 3 messages in the Sequence.
- 280 5. The 2<sup>nd</sup> message in the Sequence is lost in transit.
- 281 6. The 3<sup>rd</sup> message is the last in this Sequence and the RM Source includes an AckRequested  
282 header to ensure that it gets a timely SequenceAcknowledgement for the Sequence.
- 283 7. The RM Destination acknowledges receipt of message numbers 1 and 3 as a result of receiving the  
284 RM Source's AckRequested header.
- 285 8. The RM Source retransmits the unacknowledged message with MessageNumber 2. This is a new  
286 message from the perspective of the underlying transport, but it has the same Sequence Identifier  
287 and MessageNumber so the RM Destination can recognize it as a duplicate of the earlier message,  
288 in case the original and retransmitted messages are both Received. The RM Source includes an  
289 AckRequested header in the retransmitted message so the RM Destination will expedite an  
290 acknowledgement.
- 291 9. The RM Destination Receives the second transmission of the message with MessageNumber 2  
292 and acknowledges receipt of message numbers 1, 2, and 3.
- 293 10. The RM Source Receives this Acknowledgement and sends a TerminateSequence message to the  
294 RM Destination indicating that the Sequence is completed. The TerminateSequence message  
295 indicates that message number 3 was the last message in the Sequence. The RM Destination then  
296 and reclaims any resources associated with the Sequence.
- 297 11. The RM Destination Receives the TerminateSequence message indicating that the RM Source will  
298 not be sending any more messages. The RM Destination sends a TerminateSequenceResponse  
299 message to the RM Source and ~~and~~ reclaims any resources associated with the Sequence.
- 300 The RM Source will expect to Receive Acknowledgements from the RM Destination during the course of a  
301 message exchange at occasions described in Section 3 below. Should an Acknowledgement not be  
302 Received in a timely fashion, the RM Source MUST re-transmit the message since either the message or  
303 the associated Acknowledgement might have been lost. Since the nature and dynamic characteristics of  
304 the underlying transport and potential intermediaries are unknown in the general case, the timing of re-  
305 transmissions cannot be specified. Additionally, over-aggressive re-transmissions have been  
306 demonstrated to cause transport or intermediary flooding which are counterproductive to the intention of  
307 providing a reliable exchange of messages. Consequently, implementers are encouraged to utilize  
308 adaptive mechanisms that dynamically adjust re-transmission time and the back-off intervals that are  
309 appropriate to the nature of the transports and intermediaries envisioned. For the case of TCP/IP  
310 transports, a mechanism similar to that described as RTTM in RFC 1323 [RTTM] SHOULD be  
311 considered.
- 312 Now that the basic model has been outlined, the details of the elements used in this protocol are now  
313 provided in Section 3.

## 3 RM Protocol Elements

The following sub-sections define the various RM protocol elements, and prescribe their usage by a conformant implementations.

### 3.1 Considerations on the Use of Extensibility Points

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.

### 3.2 Considerations on the Use of "Piggy-Backing"

Some RM Protocol Header Blocks may be added to messages that are 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. The determination of if and when a Header Block will be piggy-backed onto another message is made by the entity (RM Source or RM Destination) that is sending the header. In order to ensure optimal and successful processing of RM Sequences, endpoints that receive RM-related messages SHOULD be prepared to process RM Protocol Header Blocks that are included in any message it receives. See the sections that define each RM Protocol Header Block to know which ones may be considered for piggy-backing. ~~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.~~

### 3.3 Composition with WS-Addressing

When the RM protocol, defined in this specification, is composed with the WS-Addressing specification, the following rules prescribe the constraints on the value of the `wsa:Action` header:

1. When an Endpoint generates a message that carries an RM protocol element, that is defined in the following sections, in the body of a SOAP envelope that Endpoint MUST include in that envelope a `wsa:Action` SOAP header block whose value is an IRI that is a concatenation of the WS-RM namespace URI, followed by a "/", followed by the value of the local name of the child element of the SOAP body. For example, for a Sequence creation request message as described in section 3.4~~section 3 below, in the body of a SOAP envelope that Endpoint MUST include in that envelope a `wsa:Action` SOAP header block whose value is an IRI that is a concatenation of the WS-RM namespace URI, followed by a "/", followed by the value of the local name of the child element of the SOAP body. For example, for a Sequence creation request message as described in section 3.4~~ below, the value of the `wsa:Action` IRI would be:

`http://docs.oasis-open.org/ws-rx/wsrn/200702/08/CreateSequence`

2. When an Endpoint generates an Acknowledgement Message that has no element content in the SOAP body, then the value of the `wsa:Action` IRI MUST be:

`http://docs.oasis-open.org/ws-rx/wsrn/200702/08/SequenceAcknowledgement`

3. When an Endpoint generates an Acknowledgement Request that has no element content in the SOAP body, then the value of the `wsa:Action` IRI MUST be:

<http://docs.oasis-open.org/ws-rx/wsrn/200702698/AckRequested>

4. When an Endpoint generates an RM fault as defined in section 4 below, the value of the `wsa:Action` IRI MUST be as defined in section 4 below.

### 3.4 Sequence Creation

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

The following describes the content model of the `CreateSequence` element.

`/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 `CreateSequenceResponse` response message or a `CreateSequenceRefused` fault.

`/wsrm:CreateSequence/wsrm:AcksTo`

The RM Source MUST include this element in any `CreateSequence` message it sends. This element is of type `wsa:EndpointReferenceType` (as specified by WS-Addressing). It specifies the endpoint reference to which messages containing `SequenceAcknowledgement` header blocks and faults related to the created Sequence are to be sent, unless otherwise noted in this specification (for example, see Section 3.52).

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

`/wsrm:CreateSequence/wsrm:Expires`

This element, if present, of type `xs:duration` specifies the RM Source's requested duration for the Sequence. The RM Destination MAY either accept the requested duration or assign a lesser value of its

400 choosing. A value of "PT0S" indicates that the Sequence will never expire. Absence of the element  
401 indicates an implied value of "PT0S".

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

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

405 /wsrm:CreateSequence/wsrm:Offer

406 This element, if present, enables an RM Source to offer a corresponding Sequence for the reliable  
407 exchange of messages Transmitted from RM Destination to RM Source.

408 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier

409 The RM Source MUST set the value of this element to an absolute URI (conformant with RFC3986 [URI])  
410 that uniquely identifies the offered Sequence.

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

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

414 /wsrm:CreateSequence/wsrm:Offer/wsrm:Endpoint

415 An RM Source MUST include this element, of type `wsa:EndpointReferenceType` (as specified by  
416 WS-Addressing). This element specifies the endpoint reference to which Sequence Lifecycle Messages,  
417 ~~Sequence Traffic Messages~~, Acknowledgement Requests, and fault messages related to the offered  
418 Sequence are to be sent.

419 Implementations MUST NOT use an endpoint reference in the Endpoint element that would prevent the  
420 sending of Sequence Lifecycle Message, etc. For example, using the WS-Addressing  
421 "http://www.w3.org/2005/08/addressing/none" IRI would make it impossible for the RM Destination to ever  
422 send Sequence Lifecycle Messages (e.g. TerminateSequence) to the RM Source for the Offered  
423 Sequence. Sequence Traffic Message, etc. For example, using the WS-Addressing  
424 "http://www.w3.org/2005/08/addressing/none" IRI would make it impossible for the RM Destination to ever  
425 send Sequence Lifecycle Messages (e.g. TerminateSequence) to the RM Source for the Offered  
426 Sequence. Implementations MAY use the WS-RM anonymous URI template and doing so implies that  
427 messages will be retrieved using a mechanism such as the MakeConnection message (see section  
428 3.7).-

429 The Offer of an Endpoint containing the "http://www.w3.org/2005/08/addressing/anonymous" IRI as its  
430 address is problematic due to the inability of a source to connect to this address and retry  
431 unacknowledged messages (as described in Section 2.3). Note that this specification does not define any  
432 mechanisms for providing this assurance. In the absence of an extension that addresses this issue, an  
433 RM Destination MUST NOT accept (via the /wsrm:CreateSequenceResponse/wsrm:Accept  
434 element described below) an Offer that contains the "http://www.w3.org/2005/08/addressing/anonymous"  
435 IRI as its address.

436 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires

437 This element, if present, of type `xs:duration` specifies the duration for the offered Sequence. A value of  
438 "PT0S" indicates that the offered Sequence will never expire. Absence of the element indicates an implied  
439 value of "PT0S".

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

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

443 `/wsrm:CreateSequence/wsrm:Offer/wsrm:IncompleteSequenceBehavior`

444 This element, if present, specifies the behavior that the destination will exhibit upon the closure or  
445 termination of an incomplete Sequence. For the purposes of defining the values used, the term "discard"  
446 refers to behavior equivalent to the Application Destination never processing a particular message.

447 A value of "DiscardEntireSequence" indicates that the entire Sequence MUST be discarded if the  
448 Sequence is closed, or terminated, when there are one or more gaps in the final  
449 `SequenceAcknowledgement`.

450 A value of "DiscardFollowingFirstGap" indicates that messages in the Sequence beyond the first gap  
451 MUST be discarded when there are one or more gaps in the final `SequenceAcknowledgement`.

452 The default value of "NoDiscard" indicates that no acknowledged messages in the Sequence will be  
453 discarded.

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

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

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

458 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
459 element~~different (extensible) types of information, based on a schema, to be passed.~~

460 `/wsrm:CreateSequence/{any}`

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

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

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

466 A `CreateSequenceResponse` is sent in the body of a response message by an RM Destination in  
467 response to receipt of a `CreateSequence` request message. It carries the `Identifier` of the created  
468 Sequence and indicates that the RM Source can begin sending messages in the context of the identified  
469 Sequence.

470 The following exemplar defines the `CreateSequenceResponse` syntax:

```
471 <wsrm:CreateSequenceResponse ...>
472   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
473   <wsrm:Expires ...> xs:duration </wsrm:Expires> ?
474   <wsrm:IncompleteSequenceBehavior>
475     wsrm:IncompleteSequenceBehaviorType
476   </wsrm:IncompleteSequenceBehavior> ?
477   <wsrm:Accept ...>
478     <wsrm:AcksTo wsa:EndpointReferenceType </wsrm:AcksTo>
479     ...
480   </wsrm:Accept> ?
481   ...
482 </wsrm:CreateSequenceResponse>
```

483 The following describes the content model of the `CreateSequenceResponse` element.

484 `/wsrm:CreateSequenceResponse`



485 This element is sent in the body of the response message in response to a `CreateSequence` request  
486 message. It indicates that the RM Destination has created a new Sequence at the request of the RM  
487 Source. The RM Destination MUST NOT send this element as a header block.

488 `/wsrm:CreateSequenceResponse/wsrm:Identifier`

489 The RM Destination MUST include this element within any `CreateSequenceResponse` message it sends.  
490 The RM Destination MUST set the value of this element to the absolute URI (conformant with RFC3986)  
491 that uniquely identifies the Sequence that has been created by the RM Destination.

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

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

495 `/wsrm:CreateSequenceResponse/wsrm:Expires`

496 This element, if present, of type `xs:duration` accepts or refines the RM Source's requested duration for  
497 the Sequence. It specifies the amount of time after which any resources associated with the Sequence  
498 SHOULD be reclaimed thus causing the Sequence to be silently terminated. At the RM Destination this  
499 duration is measured from a point proximate to Sequence creation and at the RM Source this duration is  
500 measured from a point approximate to the successful processing of the `CreateSequenceResponse`. A  
501 value of "PT0S" indicates that the Sequence will never expire. Absence of the element indicates an  
502 implied value of "PT0S". The RM Destination MUST set the value of this element to be equal to or less  
503 than the value requested by the RM Source in the corresponding `CreateSequence` message.

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

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

507 `/wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior`

508 This element, if present, specifies the behavior that the destination will exhibit upon the closure or  
509 termination of an incomplete Sequence. For the purposes of defining the values used, the term "discard"  
510 refers to behavior equivalent to the Application Destination never processing a particular message.

511 A value of "DiscardEntireSequence" indicates that the entire Sequence MUST be discarded if the  
512 Sequence is closed, or terminated, when there are one or more gaps in the final  
513 `SequenceAcknowledgement`.

514 A value of "DiscardFollowingFirstGap" indicates that messages in the Sequence beyond the first gap  
515 MUST be discarded when there are one or more gaps in the final `SequenceAcknowledgement`.

516 The default value of "NoDiscard" indicates that no acknowledged messages in the Sequence will be  
517 discarded.

518 `/wsrm:CreateSequenceResponse/wsrm:Accept`

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

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

524 `/wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo`

525 The RM Destination MUST include this element, of type `wsa:EndpointReferenceType` (as specified  
526 by WS-Addressing). It specifies the endpoint reference to which messages containing  
527 `SequenceAcknowledgement` header blocks and faults related to the created Sequence are to be sent,  
528 unless otherwise noted in this specification (for example, see Section 3.52).

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

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

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

536 `/wsrm:CreateSequenceResponse/wsrm:Accept/@{any}`

537 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
538 element, different (extensible) types of information, based on a schema, to be passed.

539 `/wsrm:CreateSequenceResponse/{any}`

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

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

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

## 545 3.5 Closing A Sequence

546 There are times during the use of an RM Sequence that the RM Source or RM Destination will wish to  
547 discontinue using a Sequence. Simply terminating the Sequence discards the state managed by the RM  
548 Destination, leaving the RM Source unaware of the final ranges of messages that were successfully  
549 transferred to the RM Destination. To ensure that the Sequence ends with a known final state either the  
550 RM Source or RM Destination MAY choose to close the Sequence before terminating it.

551 If the RM Source wishes to close the Sequence, then it sends a `CloseSequence` element, in the body of  
552 a message, to the RM Destination. This message indicates that the RM Destination MUST NOT accept  
553 any new messages for the specified Sequence, other than those already accepted at the time the  
554 `CloseSequence` element is interpreted by the RM Destination. Upon receipt of this message, or  
555 subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST  
556 include a final `SequenceAcknowledgement` (within which the RM Destination MUST include the `Final`  
557 element) header block on any messages associated with the Sequence destined to the RM Source,  
558 including the `CloseSequenceResponse` message or on any Sequence fault Transmitted to the RM  
559 Source.

560 To allow the RM Destination to determine if it has received all of the messages in a Sequence, the RM  
561 Source SHOULD include the `LastMsgNumber` element in any `CloseSequence` messages it sends. The  
562 RM Destination can use this information, for example, to implement the behavior indicated by /  
563 `wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior`. The value of the  
564 `LastMsgNumber` element MUST be the same in all the `CloseSequence` messages for the closing  
565 Sequence.



566 If the RM Destination decides to close a Sequence of its own volition, it MAY inform the RM Source of this  
567 event by sending a `CloseSequence` element, in the body of a message, to the AcksTo EPR of that  
568 Sequence. The RM Destination MUST include a final `SequenceAcknowledgement` (within which the RM  
569 Destination MUST include the `Final` element) header block in this message and any subsequent  
570 messages associated with the Sequence destined to the RM Source.

571 While the RM Destination MUST NOT accept any new messages for the specified Sequence it MUST still  
572 process Sequence Lifecycle Messages and Acknowledgement Requests. For example, it MUST respond to  
573 AckRequested, TerminateSequence as well as CloseSequence messages. Note, subsequent  
574 CloseSequence messages have no effect on the state of the Sequence.

575 In the case where the RM Destination wishes to discontinue use of a Sequence it is RECOMMENDED  
576 that it close the Sequence. Please see `Final` and the `SequenceClosed` fault. Whenever possible the  
577 `SequenceClosed` fault SHOULD be used in place of the `SequenceTerminated` fault to allow the RM  
578 Source to still Receive Acknowledgements.

579 The following exemplar defines the `CloseSequence` syntax:

```
580 <wsrm:CloseSequence ...>  
581   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
582   <wsrm:LastMsgNumber> wsrm:MessageNumberType </wsrm:LastMsgNumber> ?  
583   ...  
584 </wsrm:CloseSequence>
```

585 The following describes the content model of the `CloseSequence` element.

586 `/wsrm:CloseSequence`

587 This element MAY be sent by an RM Source to indicate that the RM Destination MUST NOT accept any  
588 new messages for this Sequence This element MAY also be sent by an RM Destination to indicate that it  
589 will not accept any new messages for this Sequenceis sent by an RM Source to indicate that the RM  
590 Destination MUST NOT accept any new messages for this Sequence. A `SequenceClosed` fault MUST be  
591 generated by the RM Destination when it Receives a message for a Sequence that is already closed.

592 `/wsrm:CloseSequence/wsrm:Identifier`

593 The RM Source or RM Destination MUST include this element in any `CloseSequence` messages it sends.  
594 The RM Source or RM Destination MUST set the value of this element to the absolute URI (conformant  
595 with RFC3986) of the closing SequenceMUST include this element in any `CloseSequence` messages it  
596 sends. The RM Source MUST set the value of this element to the absolute URI (conformant with  
597 RFC3986) of the Sequence that is being closed.

598 `/wsrm:CloseSequence/wsrm:LastMessageNumber`

599 The RM Source SHOULD include this element in any `CloseSequence` message it sends. The  
600 `LastMsgNumber` element specifies the highest assigned message number of all the Sequence Traffic  
601 Messages for the closing Sequence.

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

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

605 `/wsrm:CloseSequence/{any}`

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

608 `/wsrm:CloseSequence@{any}`

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

611 A `CloseSequenceResponse` is sent in the body of a message in response to receipt of a  
612 `CloseSequence` request message. It indicates that the responderresponse message by an RM  
613 Destination in response to receipt of a `CloseSequence` request message. It indicates that the RM  
614 Destination has closed the Sequence.

615 The following exemplar defines the `CloseSequenceResponse` syntax:

```
616 <wsrm:CloseSequenceResponse ...>  
617   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
618   ...  
619 </wsrm:CloseSequenceResponse>
```

620 The following describes the content model of the `CloseSequenceResponse` element.

621 `/wsrm:CloseSequenceResponse`

622 This element is sent in the body of a message in response to receipt of a `CloseSequence` request  
623 message. It indicates that the responderresponse message by an RM Destination in response to receipt  
624 of a `CloseSequence` request message. It indicates that the RM Destination has closed the Sequence.

625 `/wsrm:CloseSequenceResponse/wsrm:Identifier`

626 The responder (RM Source or RM Destination) MUST include this element in any  
627 `CloseSequenceResponse` message it sends. The responder MUST set the value of this element to the  
628 absolute URI (conformant with RFC3986) of the closing SequenceRM Destination MUST include this  
629 element in any `CloseSequenceResponse` message it sends. The RM Destination MUST set the value of  
630 this element to the absolute URI (conformant with RFC3986) of the Sequence that is being closed.

631 `/wsrm:CloseSequenceResponse/wsrm:Identifier/@{any}`

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

634 `/wsrm:CloseSequenceResponse/{any}`

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

637 `/wsrm:CloseSequenceResponse@{any}`

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

## 640 3.6 Sequence Termination

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

648 To allow the RM Destination to determine if it has received all of the messages in a Sequence, the RM  
649 Source SHOULD include the `LastMsgNumber` element in any `TerminateSequence` messages it sends.  
650 The RM Destination can use this information, for example, to implement the behavior indicated by /

651 wsrn:CreateSequenceResponse/wsrn:IncompleteSequenceBehavior. The value of the  
652 LastMsgNumber element in the TerminateSequence message MUST be equal to the value of the  
653 LastMsgNumber element in any CloseSequence message(s) sent by the RM Source for the same  
654 Sequence.

655 If the RM Destination decides to terminate a Sequence of its own volition, it MAY inform the RM Source of  
656 this event by sending a TerminateSequence element, in the body of a message, to the AcksTo EPR for  
657 that Sequence. The RM Destination MUST include a final SequenceAcknowledgement (within which  
658 the RM Destination MUST include the Final element) header block in this message.

659 The following exemplar defines the TerminateSequence syntax:

```
660 <wsrm:TerminateSequence ...>  
661   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
662   <wsrm>LastMsgNumber> wsrm:MessageNumberType </wsrm>LastMsgNumber> ?  
663   ...  
664 </wsrm:TerminateSequence>
```

665 The following describes the content model of the TerminateSequence element.

666 /wsrm:TerminateSequence

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

672 This element MAY also be sent by the RM Destination to indicate that it has unilaterally terminated the  
673 Sequence. Upon sending this message the RM Destination MUST NOT accept any additional messages  
674 (with the exception of the corresponding TerminateSequenceResponse) for this Sequence. Upon  
675 receipt of a TerminateSequence the RM Source MUST NOT send any additional messages (with the  
676 exception of the corresponding TerminateSequenceResponse) for this Sequence.

677 /wsrm:TerminateSequence/wsrn:Identifier

678 The RM Source or RM Destination MUST include this element in any TerminateSequence message it  
679 sends. The RM Source or RM Destination MUST set the value of this element to the absolute URI  
680 (conformant with RFC3986) of the terminating SequenceMUST include this element in any  
681 TerminateSequence message it sends. The RM Source MUST set the value of this element to the  
682 absolute URI (conformant with RFC3986) of the Sequence that is being terminated.

683 /wsrm:TerminateSequence/wsrn>LastMsgNumber

684 The RM Source SHOULD include this element in any TerminateSequence message it sends. The  
685 LastMsgNumber element specifies the highest assigned message number of all the Sequence Traffic  
686 Messages for the closing Sequence.

687 /wsrm:TerminateSequence/wsrn:Identifier/{any}

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

690 /wsrm:TerminateSequence/{any}

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

693 /wsrm:TerminateSequence/{any}

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

696 A `TerminateSequenceResponse` is sent in the body of a message in response to receipt of a  
697 `TerminateSequence` request message. It indicates that responderresponse message by an RM  
698 Destination in response to receipt of a `TerminateSequence` request message. It indicates that the RM  
699 Destination has terminated the Sequence.

700 The following exemplar defines the `TerminateSequenceResponse` syntax:

```
701 <wsrm:TerminateSequenceResponse ...>  
702   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
703   ...  
704 </wsrm:TerminateSequenceResponse>
```

705 The following describes the content model of the `TerminateSequence` element.

706 `/wsrm:TerminateSequenceResponse`

707 This element is sent in the body of a message in response to receipt of a `TerminateSequence` request  
708 message. It indicates that the responder has terminated the Sequence. The responderresponse message  
709 by an RM Destination in response to receipt of a `TerminateSequence` request message. It indicates  
710 that the RM Destination has terminated the Sequence. The RM Destination MUST NOT send this element  
711 as a header block.

712 `/wsrm:TerminateSequenceResponse/wsrm:Identifier`

713 The responder (RM Source or RM Destination) MUST include this element in any  
714 `TerminateSequenceResponse` message it sends. The responder MUST set the value of this element  
715 to the absolute URI (conformant with RFC3986) of the terminating SequenceRM Destination MUST  
716 include this element in any `TerminateSequenceResponse` message it sends. The RM Destination  
717 MUST set the value of this element to the absolute URI (conformant with RFC3986) of the Sequence that  
718 is being terminated.

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

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

722 `/wsrm:TerminateSequenceResponse/{any}`

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

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

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

728 On receipt of a `TerminateSequence` message the receiver (RM Source or RM Destination)an RM  
729 Destination MUST respond with a corresponding `TerminateSequenceResponse` message or generate  
730 a fault `UnknownSequenceFault` if the Sequence is not known.

## 731 3.7 Sequences

732 The RM protocol uses a Sequence header block to track and manage the reliable transfer of messages.  
733 The RM Source MUST include a `Sequence` header block in all messages for which reliable transfer is  
734 REQUIRED. The RM Source MUST identify Sequences with unique Identifier elements and the RM  
735 Source MUST assign each message within a Sequence a `MessageNumber` element that increments by 1

736 from an initial value of 1. These values are contained within a *Sequence* header block accompanying  
737 each message being transferred in the context of a *Sequence*.

738 The RM Source MUST NOT include more than one *Sequence* header block in any message.

739 A following exemplar defines its syntax:

```
740 <wsrm:Sequence ...>  
741   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
742   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>  
743   ...  
744 </wsrm:Sequence>
```

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

746 /wsrm:Sequence

747 This protocol element associates the message in which it is contained with a previously established RM  
748 *Sequence*. It contains the *Sequence*'s unique identifier and the containing message's ordinal position  
749 within that *Sequence*. The RM Destination MUST understand the *Sequence* header block. The RM  
750 Source MUST assign a *mustUnderstand* attribute with a value 1/true (from the namespace  
751 corresponding to the version of SOAP to which the *Sequence* SOAP header block is bound) to the  
752 *Sequence* header block element.

753 /wsrm:Sequence/wsrm:Identifier

754 An RM Source that includes a *Sequence* header block in a SOAP envelope MUST include this element in  
755 that header block. The RM Source MUST set the value of this element to the absolute URI (conformant  
756 with RFC3986) that uniquely identifies the *Sequence*.

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

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

760 /wsrm:Sequence/wsrm:MessageNumber

761 The RM Source MUST include this element within any *Sequence* headers it creates. This element is of  
762 type *MessageNumberType*. It represents the ordinal position of the message within a *Sequence*.  
763 *Sequence* message numbers start at 1 and monotonically increase by 1 throughout the *Sequence*. See  
764 Section 4.5 for Message Number Rollover fault.

765 /wsrm:Sequence/{any}

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

768 /wsrm:Sequence/@{any}

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

771 The following example illustrates a *Sequence* header block.

```
772 <wsrm:Sequence>  
773   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
774   <wsrm:MessageNumber>10</wsrm:MessageNumber>  
775 </wsrm:Sequence>
```

## 3.8 Request Acknowledgement

The purpose of the `AckRequested` header block is to signal to the RM Destination that the RM Source is requesting that a `SequenceAcknowledgement` be sent.

The RM Source MAY request an Acknowledgement Message from the RM Destination at any time by independently transmitting an `AckRequested` header block (i.e. as a header of a SOAP envelope with an empty body). Alternatively the RM Source MAY include an `AckRequested` header block in any message targeted to the RM Destination. The RM Destination SHOULD process `AckRequested` header blocks that are included in any message it receives. If a non-mustUnderstand fault occurs when processing an `AckRequested` header block that was piggy-backed, a fault MUST be generated, but the processing of the original message MUST NOT be affected. ~~including an `AckRequested` header block in any message targeted to the RM Destination. An RM Destination that Receives a message that contains an `AckRequested` header block MUST send a message containing a `SequenceAcknowledgement` header block to the `AcksTo` endpoint reference (see Section 3.1) for a known Sequence or else generate an `UnknownSequence` fault. If a non-mustUnderstand fault occurs when processing an RM header that was piggy-backed on another message, a fault MUST be generated, but the processing of the original message MUST NOT be affected. It is RECOMMENDED that the RM Destination return a `AcknowledgementRange` or `None` element instead of a `Nack` element (see Section 3.6).~~

An RM Destination that Receives a message that contains an `AckRequested` header block MUST send a message containing a `SequenceAcknowledgement` header block to the `AcksTo` endpoint reference (see Section 3.4) for a known Sequence or else generate an `UnknownSequence` fault. It is RECOMMENDED that the RM Destination return a `AcknowledgementRange` or `None` element instead of a `Nack` element (see Section 3.9).

The following exemplar defines its syntax:

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

The following describes the content model of the `AckRequested` header block.

`/wsrm:AckRequested`

This element requests an Acknowledgement for the identified Sequence.

`/wsrm:AckRequested/wsrm:Identifier`

An RM Source that includes an `AckRequested` header block in a SOAP envelope MUST include this element in that header block. The RM Source MUST set the value of this element to the absolute URI, (conformant with RFC3986), that uniquely identifies the Sequence to which the request applies.

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

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

`/wsrm:AckRequested/{any}`

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

`/wsrm:AckRequested/@{any}`

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

## 3.9 Sequence Acknowledgement

The RM Destination informs the RM Source of successful message receipt using a SequenceAcknowledgement header block. ~~Acknowledgements can be explicitly requested using the AckRequested directive (see Section 3.8). The RM Destination MAY Transmit the SequenceAcknowledgement header block independently or it MAY include the SequenceAcknowledgement header block on any message targeted to the AcksTo EPR. Acknowledgements can be explicitly requested using the AckRequested directive (see Section 3.5). If a non-mustUnderstand fault occurs when processing an RM header that was piggy-backed on another message, a fault MUST be generated, but the processing of the original message MUST NOT be affected.~~

~~The RM Destination MAY Transmit the SequenceAcknowledgement header block independently (i.e. As a header of a SOAP envelope with an empty body). Alternatively, an RM Destination MAY include a SequenceAcknowledgement header block on any SOAP envelope targeted to the endpoint referenced by the AcksTo EPR. The RM Source SHOULD process SequenceAcknowledgement header blocks that are included in any message it receives. If a non-mustUnderstand fault occurs when processing a SequenceAcknowledgement header that was piggy-backed, a fault MUST be generated, but the processing of the original message MUST NOT be affected. A RM Destination MAY include a SequenceAcknowledgement header block on any SOAP envelope targetted to the endpoint referenced by the AcksTo EPR.~~

During creation of a Sequence the RM Source MAY specify the WS-Addressing anonymous IRI as the address of the AcksTo EPR for that Sequence. When the RM Source specifies the WS-Addressing anonymous IRI as the address of the AcksTo EPR, the RM Destination MUST Transmit any SequenceAcknowledgement headers for the created Sequence in a SOAP envelope to be Transmitted on the protocol binding-specific ~~back-channel. Such a channel is provided by the context of a Received message containing a SOAP envelope that contains a Sequence header block and/or an AckRequested header block for that same Sequence identifier. When the RM Destination receives an AckRequested header, and the AckTo EPR for that sequence is the WS-Addressing anonymous IRI, the RM Destination SHOULD respond on the protocol binding-specific back-channel provided by the Received message containing the AckRequested header block.~~ channel. ~~Such a channel is provided by the context of a Received message containing a SOAP envelope that contains a Sequence header block and/or a AckRequested header block for that same Sequence identifier.~~

The following exemplar defines its syntax:

```
<wsrm:SequenceAcknowledgement ...>
  <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
  [ [ [ <wsrm:AcknowledgementRange ...
    Upper="wsrm:MessageNumberType"
    Lower="wsrm:MessageNumberType" /> +
    | <wsrm:None/> ]
    <wsrm:Final/> ? ]
    | <wsrm:Nack> wsrm:MessageNumberType </wsrm:Nack> + ]
  ...
</wsrm:SequenceAcknowledgement>
```

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

/wsrm:SequenceAcknowledgement

This element contains the Sequence Acknowledgement information.

/wsrm:SequenceAcknowledgement/wsrm:Identifier



866 An RM Destination that includes a `SequenceAcknowledgement` header block in a SOAP envelope  
867 MUST include this element in that header block. The RM Destination MUST set the value of this element  
868 to the absolute URI (conformant with RFC3986) that uniquely identifies the Sequence. The RM  
869 Destination MUST NOT include multiple `SequenceAcknowledgement` header blocks that share the  
870 same value for `Identifier` within the same SOAP envelope.

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

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

874 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

875 The RM Destination MAY include one or more instances of this element within a  
876 `SequenceAcknowledgement` header block. It contains a range of Sequence message numbers  
877 successfully accepted by the RM Destination. The ranges MUST~~MessageNumbers successfully accepted~~  
878 ~~by the RM Destination. The ranges SHOULD~~ NOT overlap. The RM Destination MUST NOT include this  
879 element if a sibling `Nack` or `None` element is also present as a child of `SequenceAcknowledgement`.

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

881 The RM Destination MUST set the value of this attribute equal to the message number of the highest  
882 contiguous message in a Sequence range accepted by the RM Destination.

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

884 The RM Destination MUST set the value of this attribute equal to the message number of the lowest  
885 contiguous message in a Sequence range accepted by the RM Destination.

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

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

889 `/wsrm:SequenceAcknowledgement/wsrm:None`

890 The RM Destination MUST include this element within a `SequenceAcknowledgement` header block if  
891 the RM Destination has not accepted any messages for the specified Sequence. The RM Destination  
892 MUST NOT include this element if a sibling `AcknowledgementRange` or `Nack` element is also present  
893 as a child of the `SequenceAcknowledgement`.

894 `/wsrm:SequenceAcknowledgement/wsrm:Final`

895 The RM Destination MAY include this element within a `SequenceAcknowledgement` header block. This  
896 element indicates that the RM Destination is not receiving new messages for the specified Sequence. The  
897 RM Source can be assured that the ranges of messages acknowledged by this  
898 `SequenceAcknowledgement` header block will not change in the future. The RM Destination MUST  
899 include this element when the Sequence is closed. The RM Destination MUST NOT include this element  
900 when sending a `Nack`; it can only be used when sending `AcknowledgementRange` elements or a `None`.

901 `/wsrm:SequenceAcknowledgement/wsrm:Nack`

902 The RM Destination MAY include this element within a `SequenceAcknowledgement` header block. If  
903 used, the RM Destination MUST set the value of this element to a `MessageNumberType` representing  
904 the `MessageNumber` of an unreceived message in a Sequence. The RM Destination MUST NOT include  
905 a `Nack` element if a sibling `AcknowledgementRange` or `None` element is also present as a child of  
906 `SequenceAcknowledgement`. Upon the receipt of a `Nack`, an RM Source SHOULD retransmit the  
907 message identified by the `Nack`. The RM Destination MUST NOT issue a `SequenceAcknowledgement`



908 containing a `Nack` for a message that it has previously acknowledged within an  
909 `AcknowledgementRange`. The RM Source SHOULD ignore a `SequenceAcknowledgement` containing  
910 a `Nack` for a message that has previously been acknowledged within an `AcknowledgementRange`. The  
911 RM Source SHOULD ignore a `SequenceAcknowledgement` containing a `Nack` for a message that has  
912 previously been acknowledged within a `AcknowledgementRange`.

913 `/wsrm:SequenceAcknowledgement/{any}`

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

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

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

919 The following examples illustrate `SequenceAcknowledgement` elements:

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

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

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

```
927 <wsrm:SequenceAcknowledgement>  
928   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
929   <wsrm:AcknowledgementRange Upper="2" Lower="1"/>  
930   <wsrm:AcknowledgementRange Upper="6" Lower="4"/>  
931   <wsrm:AcknowledgementRange Upper="10" Lower="8"/>  
932 </wsrm:SequenceAcknowledgement>
```

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

```
934 <wsrm:SequenceAcknowledgement>  
935   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
936   <wsrm:Nack>3</wsrm:Nack>  
937 </wsrm:SequenceAcknowledgement>
```

## 938 1.1 MakeConnection

939 When an Endpoint is not directly addressable (e.g. behind a firewall or not able to allow incoming  
940 connections), an anonymous URI in the EPR address property can indicate such an Endpoint. The WS-  
941 Addressing anonymous URI is one such anonymous URI. This specification defines a URI template (the  
942 WS-RM anonymous URI) which may be used to uniquely identify anonymous Endpoints.

943 `http://docs.oasis-open.org/ws-rx/wsrn/200608/anonymous?id={uuid}`

944 This URI template in an EPR indicates a protocol-specific back-channel will be established through a  
945 mechanism such as `MakeConnection`, defined below. When using this URI template, "{uuid}" MUST be  
946 replaced by a UUID value as defined by RFC4122[UUID]. This UUID value uniquely distinguishes the  
947 Endpoint. A sending Endpoint SHOULD Transmit messages at Endpoints identified with the URI template  
948 using a protocol-specific back-channel, including but not limited to those established with a  
949 `MakeConnection` message. Note, this URI is semantically similar to the WS-Addressing anonymous  
950 URI if a protocol-specific back-channel is available.

951 The `MakeConnection` is a one-way operation that establishes a contextualized back-channel for the  
952 transmission of messages according to matching criteria (defined below). In the non-faulting case, if no

953 matching message is available then no SOAP envelopes will be returned on the back channel. A common  
954 usage will be a client RM Destination sending `MakeConnection` to a server RM Source for the purpose  
955 of receiving asynchronous response messages.

956 The following exemplar defines the `MakeConnection` syntax:

```
957 <wsrm:MakeConnection ...>  
958   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier> ?  
959   <wsrm:Address ...> xs:anyURI </wsrm:Address> ?  
960   ...  
961 </wsrm:MakeConnection>
```

962 `/wsrm:MakeConnection`

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

965 `/wsrm:MakeConnection/wsrm:Identifier`

966 This element specifies the WS-RM Sequence Identifier that establishes the context for the transport-  
967 specific back channel. The Sequence Identifier should be compared with the Sequence Identifiers  
968 associated with the messages held by the sending Endpoint, and if there is a matching message it will be  
969 returned. If this element is omitted from the message then the `Address` MUST be included in the  
970 message.

971 `/wsrm:MakeConnection/wsrm:Identifier/@{any}`

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

974 `/wsrm:MakeConnection/wsrm:Address`

975 This element specifies the URI (`wsa:Address`) of the initiating Endpoint. Endpoints MUST NOT return  
976 messages on the transport-specific back channel unless they have been addressed to this URI. This  
977 `Address` property and a message's WS-Addressing destination property are considered identical when  
978 they are exactly the same character-for-character. Note that URIs which are not identical in this sense  
979 may in fact be functionally equivalent. Examples include URI references which differ only in case, or  
980 which are in external entities which have different effective base URIs. If this element is omitted from the  
981 message then the `Identifier` MUST be included in the message.

982 `/wsrm:MakeConnection/wsrm:Address/@{any}`

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

985 `/wsrm:MakeConnection/{any}`

986 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,  
987 to be passed. This allows fine-tuning of the messages to be returned, additional selection criteria included  
988 here are logically ANDed with the `Address` and/or `Identifier`. If an extension is not supported by the  
989 Endpoint then it should return a `UnsupportedSelection` fault.

990 `/wsrm:MakeConnection/@{any}`

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

993 If both `Identifier` and `Address` are present, then the Endpoint processing the `MakeConnection`  
994 message MUST insure that any SOAP Envelope flowing on the backchannel MUST be associated with  
995 the given Sequence and MUST be addressed to the given URI.

996 The management of messages that are awaiting the establishment of a back-channel to their receiving  
997 Endpoint is an implementation detail that is outside the scope of this specification. Note, however, that  
998 these messages form a class of asynchronous messages that is not dissimilar from "ordinary"  
999 asynchronous messages that are waiting for the establishment of a connection to their destination  
1000 Endpoints.

1001 This specification places no constraint on the types of messages that can be returned on the transport-  
1002 specific back-channel. As in an asynchronous environment, it is up to the recipient of the  
1003 `MakeConnection` message to decide which messages are appropriate for transmission to any particular  
1004 Endpoint. However, the Endpoint processing the `MakeConnection` message MUST insure that the  
1005 messages match the selection criteria as specified by the child elements of the `MakeConnection`  
1006 element.

## 1007 1.2 MessagePending

1008 When `MakeConnection` is used, and a message is returned on the transport-specific back-channel, the  
1009 `MessagePending` header SHOULD be included on the returned message as an indicator whether there  
1010 are additional messages waiting to be retrieved using the same selection criteria that was specified in the  
1011 `MakeConnection` element.

1012 The following exemplar defines the `MessagePending` syntax:

```
1013 <wsrm:MessagePending pending="xs:boolean" ...>  
1014   ...  
1015 </wsrm:MessagePending>
```

1016 `/wsrm:MessagePending`

1017 This element indicates whether additional messages are waiting to be retrieved.

1018 `/wsrm:MessagePending@pending`

1019 This attribute, when set to "true", indicates that there is at least one message waiting to be retrieved.  
1020 When this attribute is set to "false" it indicates there are currently no messages waiting to be retrieved.

1021 `/wsrm:MessagePending/{any}`

1022 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,  
1023 to be passed.

1024 `/wsrm:MessagePending/@{any}`

1025 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
1026 element.

1027 The absence of the `MessagePending` header has no implication as to whether there are additional  
1028 messages waiting to be retrieved.

## 4 Faults

Faults for the `CreateSequence` message exchange are treated as defined in WS-Addressing. Create Sequence Refused is a possible fault reply for this operation. Unknown Sequence is a fault generated by Endpoints when messages carrying RM header blocks targeted at unrecognized or terminated Sequences are detected. ~~WSRMRequired is a fault generated an RM Destination that requires the use of WS-RM on a Received message that did not use the protocol. All other faults in this section relate to known Sequences. Destinations that generate faults related to known sequences SHOULD transmit those faults. If transmitted, such faults MUST be transmitted to the same [destination] as Acknowledgement m~~ Required is a fault generated an RM Destination that requires the use of WS-RM on a Received message that did not use the protocol. All other faults in this section relate to known Sequences. RM Destinations that generate Sequence faults SHOULD send those faults to the same [destination] as Acknowledgement Messages.

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

```
http://docs.oasis-open.org/ws-rx/wsrn/200702608/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 6 of WS-Addressing SOAP Binding.

The definitions of faults use the following properties:

[Code] The fault code.

[Subcode] The fault subcode.

[Reason] The English language reason element.

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

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

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

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

```
<S:Envelope>
  <S:Header>
    <wsa:Action>
      http://docs.oasis-open.org/ws-rx/wsrn/200702608/fault
    </wsa:Action>
    <!-- Headers elided for brevity. -->
  </S:Header>
  <S:Body>
    <S:Fault>
      <S:Code>
        <S:Value> [Code] </S:Value>
        <S:Subcode>
          <S:Value> [Subcode] </S:Value>
        </S:Subcode>
      </S:Code>
    <S:Reason>
```

```

1072     <S:Text xml:lang="en"> [Reason] </S:Text>
1073   </S:Reason>
1074   <S:Detail>
1075     [Detail]
1076     ...
1077   </S:Detail>
1078 </S:Fault>
1079 </S:Body>
1080 </S:Envelope>

```

1081 The properties above bind to a SOAP 1.1 fault as follows when the fault is triggered by processing an RM  
 1082 header block:

```

1083 <S11:Envelope>
1084   <S11:Header>
1085     <wsrm:SequenceFault>
1086       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
1087       <wsrm:Detail> [Detail] </wsrm:Detail>
1088       ...
1089     </wsrm:SequenceFault>
1090   <!-- Headers elided for brevity. -->
1091 </S11:Header>
1092 <S11:Body>
1093   <S11:Fault>
1094     <faultcode> [Code] </faultcode>
1095     <faultstring> [Reason] </faultstring>
1096   </S11:Fault>
1097 </S11:Body>
1098 </S11:Envelope>

```

1099 The properties bind to a SOAP 1.1 fault as follows when the fault is generated as a result of processing a  
 1100 CreateSequence request message:

```

1101 <S11:Envelope>
1102   <S11:Body>
1103     <S11:Fault>
1104       <faultcode> [Subcode] </faultcode>
1105       <faultstring> [Reason] </faultstring>
1106     </S11:Fault>
1107   </S11:Body>
1108 </S11:Envelope>

```

## 1109 4.1 SequenceFault Element

1110 The purpose of the `SequenceFault` element is to carry the specific details of a fault generated during  
 1111 the reliable messaging specific processing of a message belonging to a Sequence. WS-  
 1112 ReliableMessaging nodes MUST use the `SequenceFault` container only in conjunction with the SOAP  
 1113 1.1 fault mechanism. WS-ReliableMessaging nodes MUST NOT use the `SequenceFault` container in  
 1114 conjunction with the SOAP 1.2 binding.

1115 The following exemplar defines its syntax:

```

1116 <wsrm:SequenceFault ...>
1117   <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
1118   <wsrm:Detail> ... </wsrm:Detail> ?
1119   ...
1120 </wsrm:SequenceFault>

```

1121 The following describes the content model of the `SequenceFault` element.

1122 /wsrm:SequenceFault

1123 This is the element containing Sequence information for WS-ReliableMessaging

1124 /wsrm:SequenceFault/wsrm:FaultCode  
 1125 WS-ReliableMessaging nodes that generate a `SequenceFault` MUST set the value of this element to a  
 1126 qualified name from the set of fault [Subcodes] defined below.

1127 /wsrm:SequenceFault/wsrm:Detail  
 1128 This element, if present, carries application specific error information related to the fault being described.

1129 /wsrm:SequenceFault/wsrm:Detail/{any}  
 1130 The application specific error information related to the fault being described.

1131 /wsrm:SequenceFault/wsrm:Detail/@{any}  
 1132 The application specific error information related to the fault being described.

1133 /wsrm:SequenceFault/{any}  
 1134 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,  
 1135 to be passed.

1136 /wsrm:SequenceFault/@{any}  
 1137 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
 1138 element.

## 1139 4.2 Sequence Terminated

1140 The Endpoint that generates this fault SHOULD make every reasonable effort to notify the corresponding  
 1141 Endpoint of this decision.

1142 Properties:

1143 [Code] Sender or Receiver

1144 [Subcode] wsrm:SequenceTerminated

1145 [Reason] The Sequence has been terminated due to an unrecoverable error.

1146 [Detail]

1147 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`

Generated by	Condition	Action Upon Generation	Action Upon Receipt
RM Source or RM Destination.	Encountering an unrecoverable condition or detection of violation of the protocol.	Sequence termination.	MUST terminate the Sequence if not otherwise terminated.

## 1148 4.3 Unknown Sequence

1149 Properties:

1150 [Code] Sender

1151 [Subcode] wsrm:UnknownSequence

1152 [Reason] The value of wsrn:Identifier is not a known Sequence identifier.

1153 [Detail]

1154 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`

Generated by	Condition	Action Upon Generation	Action Upon Receipt
RM Source or RM Destination.	In response to a message containing an unknown or terminated Sequence identifier.	None.	MUST terminate the Sequence if not otherwise terminated.

#### 1155 4.4 Invalid Acknowledgement

1156 An example of when this fault is generated is when a message is Received by the RM Source containing  
1157 a SequenceAcknowledgement covering messages that have not been sent.

1158 [Code] Sender

1159 [Subcode] wsrn:InvalidAcknowledgement

1160 [Reason] The SequenceAcknowledgement violates the cumulative Acknowledgement invariant.

1161 [Detail]

1162 `<wsrm:SequenceAcknowledgement ...> ... </wsrm:SequenceAcknowledgement>`

Generated by	Condition	Action Upon Generation	Action Upon Receipt
RM Source.	In response to a SequenceAcknowledgement that violate the invariants stated in 2.3 or any of the requirements in 3.9 about valid combinations of AckRange, Nack and None in a single SequenceAcknowledgement element or with respect to already Received such elements.	Unspecified.	Unspecified.

#### 1163 4.5 Message Number Rollover

1164 If the condition listed below is reached, the RM Destination MUST generate this fault.

1165 Properties:

1166 [Code] Sender

1167 [Subcode] wsrn:MessageNumberRollover

1168 [Reason] The maximum value for wsrn:MessageNumber has been exceeded.

1169 [Detail]

1170 `<wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>`

1171 `<wsrm:MaxMessageNumber> wsrm:MessageNumberType </wsrm:MaxMessageNumber>`

Generated by	Condition	Action Upon Generation	Action Upon Receipt
RM Destination.	Message number in / wsrm:Sequence/wsrm: :MessageNumber of a Received message exceeds the internal limitations of an RM Destination or reaches the maximum value of 9,223,372,036,854,775,8 07.	RM Destination SHOULD continue to accept undelivered messages until the Sequence is closed or terminated.	RM Source SHOULD continue to retransmit undelivered messages until the Sequence is closed or terminated.

## 1172 4.6 Create Sequence Refused

1173 Properties:

1174 [Code] Sender or Receiver

1175 [Subcode] wsrm:CreateSequenceRefused

1176 [Reason] The Create Sequence request has been refused by the RM Destination.

1177 [Detail]

1178 `xs:any`

Generated by	Condition	Action Upon Generation	Action Upon Receipt
RM Destination.	In response to a CreateSequence message when the RM Destination does not wish to create a new Sequence.	Unspecified.	Sequence terminated.

## 1179 4.7 Sequence Closed

1180 This fault is generated by an RM Destination to indicate that the specified Sequence has been closed.

1181 This fault MUST be generated when an RM Destination is asked to accept a message for a Sequence that  
1182 is closed ~~or when an RM Destination is asked to close a Sequence that is already closed.~~

1183 Properties:

1184 [Code] Sender

1185 [Subcode] wsrm:SequenceClosed

1186 [Reason] The Sequence is closed and can-not accept new messages.

1187 [Detail]



1188 `<wsrm:Identifier...> xs:anyURI </wsrm:Identifier>`

Generated by	Condition	Action Upon Generation	Action Upon Receipt
RM Destination.	In response to a message that belongs to a Sequence that is already closed.	Unspecified.	Sequence closed.

## 1189 4.8 WSRM Required

1190 If an RM Destination requires the use of WS-RM, this fault is generated when it Receives an incoming  
1191 message that did not use this protocol.

1192 Properties:

1193 [Code] Sender

1194 [Subcode] wsrm:WSRMRequired

1195 [Reason] The RM Destination requires the use of WSRM.

1196 [Detail]

1197 `xs:any`

Generated-by	Condition	Action-Upon Generation	Action-Upon-Receipt
RM-Destination:	On receipt of a message that does not use this protocol and for which this protocol is required.	Unspecified:	Unspecified:

## 1198 1.3 ~~Unsupported Selection~~

1199 ~~The QName of the unsupported element(s) are included in the detail.~~

1200 ~~Properties:-~~

1201 ~~[Code] Receiver~~

1202 ~~[Subcode] wsrm:UnsupportedSelection~~

1203 ~~[Reason] The extension element used in the message selection is not supported by the RM Source~~

1204 ~~[Detail]~~

1205 ~~`<wsrm:UnsupportedElement> xs:QName </wsrm:UnsupportedElement>`~~

Generated by	Condition	Action Upon Generation	Action Upon Receipt
RM Source or RM Destination:	In response to a MakeConnection message containing a selection criteria in the extensibility section of the message that is not supported	Unspecified.	Unspecified.

## 5 Security Threats and Countermeasures

This specification considers two sets of security requirements, those of the applications that use the WS-RM protocol and those of the protocol itself.

This specification makes no assumptions about the security requirements of the applications that use WS-RM. However, once those requirements have been satisfied within a given operational context, the addition of WS-RM to this operational context should not undermine the fulfillment of those requirements; the use of WS-RM should not create additional attack vectors within an otherwise secure system.

There are many other security concerns that one may need to consider when implementing or using this protocol. The material below should not be considered as a "check list". Implementers and users of this protocol are urged to perform a security analysis to determine their particular threat profile and the appropriate responses to those threats.

Implementers are also advised that there is a core tension between security and reliable messaging that can be problematic if not addressed by implementations; one aspect of security is to prevent message replay but one of the invariants of this protocol is to resend messages until they are acknowledged. Consequently, if the security sub-system processes a message but a failure occurs before the reliable messaging sub-system Receives that message, 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 condition.

### 5.1 Threats and Countermeasures

The primary security requirement of this protocol is to protect the specified semantics and protocol invariants against various threats. The following sections describe several threats to the integrity and operation of this protocol and provide some general outlines of countermeasures to those threats. Implementers and users of this protocol should keep in mind that all threats are not necessarily applicable to all operational contexts.

#### 5.1.1 Integrity Threats

In general, any mechanism which allows an attacker to alter the information in a Sequence Traffic Message, Sequence Lifecycle Message, Acknowledgement Messages, Acknowledgement Request, or Sequence-related fault, or which allows an attacker to alter the correlation of a RM Protocol Header Block to its intended message represents a threat to the WS-RM protocol.

For example, if an attacker is able to swap `Sequence` headers on messages in transit between the RM Source and RM Destination then they have undermined the implementation's ability to guarantee the first invariant described in Section 2.3. The result is that there is no way of guaranteeing that messages will be Delivered to the Application Destination in the same order that they were sent by the Application Source.

##### 5.1.1.1 Countermeasures

Integrity threats are generally countered via the use of digital signatures some level of the communication protocol stack. Note that, in order to counter header swapping attacks, the signature **SHOULD** include both the SOAP body and any relevant SOAP headers (e.g. `Sequence` header). Because some headers (`AckRequested`, `SequenceAcknowledgement`) are independent of the body of the SOAP message in which they occur, implementations **MUST** allow for signatures that cover only these headers.

### 5.1.2 Resource Consumption Threats

The creation of a Sequence with an RM Destination consumes various resources on the systems used to implement that RM Destination. These resources can include network connections, database tables, message queues, etc. This behavior can be exploited to conduct denial of service attacks against an RM Destination. For example, a simple attack is to repeatedly send `CreateSequence` messages to an RM Destination. Another attack is to create a Sequence for a service that is known to require in-order message Delivery and use this Sequence to send a stream of very large messages to that service, making sure to omit message number “1” from that stream.

#### 5.1.2.1 Countermeasures

There are a number of countermeasures against the described resource consumption threats. The technique advocated by this specification is for the RM Destination to restrict the ability to create a Sequence to a specific set of entities/principals. This reduces the number of potential attackers and, in some cases, allows the identity of any attackers to be determined.

The ability to restrict Sequence creation depends, in turn, upon the RM Destination's ability identify and authenticate the RM Source that issued the `CreateSequence` message.

### 5.1.3 Sequence Spoofing Threats

Sequence spoofing is a class of threats in which the attacker uses knowledge of the `Identifier` for a particular Sequence to forge Sequence Lifecycle or Traffic Messages. For example the attacker creates a fake `TerminateSequence` message that references the target Sequence and sends this message to the appropriate RM Destination. Some sequence spoofing attacks also require up-to-date knowledge of the current `MessageNumber` for their target Sequence.

In general any Sequence Lifecycle Message, RM Protocol Header Block, or sequence-correlated SOAP fault (e.g. `InvalidAcknowledgement`) can be used by someone with knowledge of the Sequence identifier to attack the Sequence. These attacks are “two-way” in that an attacker may choose to target the RM Source by, for example, inserting a fake `SequenceAcknowledgement` header into a message that it sends to the `AcksTo` EPR of an RM Source.

#### 5.1.3.1 Sequence Hijacking

Sequence hijacking is a specific case of a sequence spoofing attack. The attacker attempts to inject Sequence Traffic Messages into an existing Sequence by inserting fake `Sequence` headers into those messages.

Note that “sequence hijacking” should not be equated with “security session hijacking”. Although a Sequence may be bound to some form of a security session in order to counter the threats described in this section, applications MUST NOT rely on WS-RM-related information to make determinations about the identity of the entity that created a message; applications SHOULD rely only upon information that is established by the security infrastructure to make such determinations. Failure to observe this rule creates, among other problems, a situation in which the absence of WS-RM may deprive an application of the ability to authenticate its peers even though the necessary security processing has taken place.

#### 5.1.3.2 Countermeasures

There are a number of countermeasures against sequence spoofing threats. The technique advocated by this specification is to consider the Sequence to be a shared resource that is jointly owned by the RM

1286 Source that initiated its creation (i.e. that sent the `CreateSequence` message) and the RM Destination that  
1287 serves as its terminus (i.e. that sent the `CreateSequenceResponse` message). To counter sequence  
1288 spoofing attempts the RM Destination SHOULD ensure that every message or fault that it Receives that  
1289 refers to a particular Sequence originated from the RM Source that jointly owns the referenced Sequence.  
1290 For its part the RM Source SHOULD ensure that every message or fault that it Receives that refers to a  
1291 particular Sequence originated from the RM Destination that jointly owns the referenced Sequence.

1292 For the RM Destination to be able to identify its sequence peer it MUST be able to identify and  
1293 authenticate the entity that sent the `CreateSequence` message. Similarly for the RM Source to identify its  
1294 sequence peer it MUST be able to identify and authenticate the entity that sent the  
1295 `CreateSequenceResponse` message. For either the RM Destination or the RM Source to determine if a  
1296 message was sent by its sequence peer it MUST be able to identify and authenticate the initiator of that  
1297 message and, if necessary, correlate this identity with the sequence peer identity established at sequence  
1298 creation time.

## 1299 **5.2 Security Solutions and Technologies**

1300 The security threats described in the previous sections are neither new nor unique. The solutions that  
1301 have been developed to secure other SOAP-based protocols can be used to secure WS-RM as well. This  
1302 section maps the facilities provided by common web services security solutions against countermeasures  
1303 described in the previous sections.

1304 Before continuing this discussion, however, some examination of the underlying requirements of the  
1305 previously described countermeasures is necessary. Specifically it should be noted that the technique  
1306 described in Section 5.1.2.1 has two components. Firstly, the RM Destination identifies and authenticates  
1307 the issuer of a `CreateSequence` message. Secondly, the RM Destination ~~to~~ performs an authorization  
1308 check against this authenticated identity and determines if the RM Source is permitted to create  
1309 Sequences with the RM Destination. Since the facilities for performing this authorization check (runtime  
1310 infrastructure, policy frameworks, etc.) lie completely within the domain of individual implementations, any  
1311 discussion of such facilities is considered to be beyond the scope of this specification.

### 1312 **5.2.1 Transport Layer Security**

1313 This section describes how the ~~the~~ facilities provided by SSL/TLS [RFC 4346] can be used to implement  
1314 the countermeasures described in the previous sections. The use of SSL/TLS is subject to the constraints  
1315 defined in Section 4 of the Basic Security Profile 1.0 [BSP 1.0].

1316 The description provided here is general in nature and is not intended to serve as a complete definition on  
1317 the use of SSL/TLS to protect WS-RM. In order to interoperate implementations need to agree on the  
1318 choice of features as well as the manner in which they will be used. The mechanisms described in the  
1319 Web Services Security Policy Language [SecurityPolicy] MAY be used by services to describe the  
1320 requirements and constraints of the use of SSL/TLS.

#### 1321 **5.2.1.1 Model**

1322 The basic model for using SSL/TLS is as follows:

- 1323 1. The RM Source establishes an SSL/TLS session with the RM Destination.
- 1324 2. The RM Source uses this SSL/TLS session to send a `CreateSequence` message to the RM  
1325 Destination.

3. The RM Destination establishes an SSL/TLS session with the RM Source and sends an asynchronous `CreateSequenceResponse` using this session. Alternately it may respond with a synchronous `CreateSequenceResponse` using the session established in (1).
4. For the lifetime of the Sequence the RM Source uses the SSL/TLS session from (1) to Transmit any and all messages or faults that refer to that Sequence.
5. For the lifetime of the Sequence the RM Destination either uses the SSL/TLS session established in (3) to Transmit any and all messages or faults that refer to that Sequence or, for synchronous exchanges, the RM Destination uses the SSL/TLS session established in (1).

#### 5.2.1.2 Countermeasure Implementation

Used in its simplest fashion (without relying upon any authentication mechanisms), SSL/TLS provides the necessary integrity qualities to counter the threats described in Section 5.1.1. Note, however, that the nature of SSL/TLS limits the scope of this integrity protection to a single transport level session. If SSL/TLS is the only mechanism used to provide integrity, any intermediaries between the RM Source and the RM Destination MUST be trusted to preserve the integrity of the messages that flow through them.

As noted, the technique described in Sections 5.1.2.1 involves the use of authentication. This specification advocates either of two mechanisms for authenticating entities using SSL/TLS. In both of these methods the SSL/TLS server (the party accepting the SSL/TLS connection) authenticates itself to the SSL/TLS client using an X.509 certificate that is exchanged during the SSL/TLS handshake.

- **HTTP Basic Authentication:** This method of authentication presupposes that a SOAP/HTTP binding is being used as part of the protocol stack beneath WS-RM. Subsequent to the establishment of the SSL/TLS session, the sending party authenticates itself to the receiving party using HTTP Basic Authentication [RFC 2617]. For example, a RM Source might authenticate itself to a RM Destination (e.g. when transmitting a Sequence Traffic Message) using BasicAuth. Similarly the RM Destination might authenticate itself to the RM Source (e.g. when sending an Acknowledgement) using BasicAuth.
- **SSL/TLS Client Authentication:** In this method of authentication, the party initiating the connection authenticates itself to the party accepting the connection using an X.509 certificate that is exchanged during the SSL/TLS handshake.

To implement the countermeasures described in section 5.1.2.1 the RM Source must authenticate itself using one the above mechanisms. The authenticated identity can then be used to determine if the RM Source is authorized to create a Sequence with the RM Destination.

This specification advocates implementing the countermeasures described in section 5.1.3.2 by requiring an RM node's Sequence peer to be equivalent to their SSL/TLS session peer. This allows the authorization decisions described in section 5.1.3.2 to be based on SSL/TLS session identity rather than on authentication information. For example, an RM Destination can determine that a Sequence Traffic Message rightfully belongs to its referenced Sequence if that message arrived over the same SSL/TLS session that was used to carry the `CreateSequence` message for that Sequence. Note that requiring a one-to-one relationship between SSL/TLS session peer and Sequence peer constrains the lifetime of a SSL/TLS-protected Sequence to be less than or equal to the lifetime of the SSL/TLS session that is used to protect that Sequence.

This specification does not preclude the use of other methods of using SSL/TLS to implement the countermeasures (such as associating specific authentication information with a Sequence) although such methods are not covered by this document.

1369 Issues specific to the life-cycle management of SSL/TLS sessions (such as the resumption of a SSL/TLS  
1370 session) are outside the scope of this specification.

## 1371 **5.2.2 SOAP Message Security**

1372 The mechanisms described in WS-Security may be used in various ways to implement the  
1373 countermeasures described in the previous sections. This specification advocates using the protocol  
1374 described by WS-SecureConversation [SecureConversation] (optionally in conjunction with WS-Trust  
1375 [Trust]) as a mechanism for protecting Sequences. The use of WS-Security (as an underlying component  
1376 of WS-SecureConversation) is subject to the constraints defined in the Basic Security Profile 1.0.

1377 The description provided here is general in nature and is not intended to serve as a complete definition on  
1378 the use of WS-SecureConversation/WS-Trust to protect WS-RM. In order to interoperate implementations  
1379 need to agree on the choice of features as well as the manner in which they will be used. The  
1380 mechanisms described in the Web Services Security Policy Language MAY be used by services to  
1381 describe the requirements and constraints of the use of WS-SecureConversation.

### 1382 **5.2.2.1 Model**

1383 The basic model for using WS-SecureConversation is as follows:

- 1384 1. The RM Source and the RM Destination create a WS-SecureConversation security context. This  
1385 may involve the participation of third parties such as a security token service. The tokens  
1386 exchanged may contain authentication claims (e.g. X.509 certificates or Kerberos service tickets).
- 1387 2. During the `CreateSequence` exchange, the RM Source SHOULD explicitly identify the security  
1388 context that will be used to protect the Sequence. This is done so that, in cases where the  
1389 `CreateSequence` message is signed by more than one security context, the RM Source can  
1390 indicate which security context should be used to protect the newly created Sequence.
- 1391 3. For the lifetime of the Sequence the RM Source and the RM Destination use the session key(s)  
1392 associated with the security context to sign (as defined by WS-Security) at least the body and any  
1393 relevant WS-RM-defined headers of any and all messages or faults that refer to that Sequence.

### 1394 **5.2.2.2 Countermeasure Implementation**

1395 Without relying upon any authentication information, the per-message signatures provide the necessary  
1396 integrity qualities to counter the threats described in Section 5.1.1.

1397 To implement the countermeasures described in section 5.1.2.1 some mutually agreed upon form of  
1398 authentication claims must be provided by the RM Source to the RM Destination during the establishment  
1399 of the Security Context. These claims can then be used to determine if the RM Source is authorized to  
1400 create a Sequence with the RM Destination.

1401 This specification advocates implementing the countermeasures described in section 5.1.3.2 by requiring  
1402 an RM node's Sequence peer to be equivalent to their security context session peer. This allows the  
1403 authorization decisions described in section 5.1.3.2 to be based on the identity of the message's security  
1404 context rather than on any authentication claims that may have been established during security context  
1405 initiation. Note that other methods of using WS-SecurityConversation to implement the countermeasures  
1406 (such as associating specific authentication claims to a Sequence) are possible but not covered by this  
1407 document.

1408 As with transport security, the requisite equivalence of a security context peer and with a Sequence peer  
1409 limits the lifetime of a Sequence to the lifetime of the protecting security context. Unlike transport security,

1410 the association between a Sequence and its protecting security context cannot always be established  
1411 implicitly at Sequence creation time. This is due to the fact that the `CreateSequence` and  
1412 `CreateSequenceResponse` messages may be signed by more than one security context.  
1413 Issues specific to the life-cycle management of WS-SecurityConversation security contexts (such as  
1414 amending or renewing contexts) are outside the scope of this specification.



## 6 Securing Sequences

As noted in Section 5, the RM Source and RM Destination should be able to protect their shared Sequences against the threat of Sequence Spoofing attacks. There are a number of OPTIONAL means of achieving this objective depending upon the underlying security infrastructure.

### 6.1 Securing Sequences Using WS-Security

One mechanism for protecting a Sequence is to include a security token using a `wsse:SecurityTokenReference` element from WS-Security (see section 9 in WS-SecureConversation) in the `CreateSequence` element. This establishes an association between the created (and, if present, offered) Sequence(s) and the referenced security token, such that the RM Source and Destination MUST use the security token as the basis for authorization of all subsequent interactions related to the Sequence(s). The `wsse:SecurityTokenReference` explicitly identifies the token as there may be more than one token on a `CreateSequence` message or inferred from the communication context (e.g. transport protection).

It is RECOMMENDED that a message independent referencing mechanism be used to identify the token, if the token being referenced supports such mechanism.

The following exemplar defines the `CreateSequence` syntax when extended to include a `wsse:SecurityTokenReference`:

```
<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:IncompleteSequenceBehavior>
      wsrml:IncompleteSequenceBehaviorType
    </wsrm:IncompleteSequenceBehavior> ?
    ...
  </wsrm:Offer> ?
  ...
  <wsse:SecurityTokenReference>
    ...
  </wsse:SecurityTokenReference> ?
  ...
</wsrm:CreateSequence>
```

The following describes the content model of the additional `CreateSequence` elements.

`/wsrm:CreateSequence/wsse:SecurityTokenReference`

This element uses the extensibility mechanism defined for the `CreateSequence` element (defined in section 3.41) to communicate an explicit reference to the security token, using a `wsse:SecurityTokenReference` as documented in WS-Security, that the RM Source and Destination MUST use to authorize messages for the created (and, if present, the offered) Sequence(s). All subsequent messages related to the created (and, if present, the offered) Sequence(s) MUST demonstrate proof-of-possession of the secret associated with the token (e.g., by using or deriving from a private or secret key).

When a RM Source transmits a `CreateSequence` that has been extended to include a `wsse:SecurityTokenReference` it SHOULD ensure that the RM Destination both understands and will conform to  
~~transmits a `CreateSequence` that has been extended to include a `wsse:SecurityTokenReference` it SHOULD ensure that the RM Destination both understands and~~

1463 ~~will conform with~~ the requirements listed above. In order to achieve this, the RM Source SHOULD include  
1464 the `UsesSequenceSTR` element as a SOAP header block within the `CreateSequence` message. This  
1465 element MUST include a `soap:mustUnderstand` attribute with a value of 'true'. Thus the RM Source  
1466 can be assured that a RM Destination that responds with a `CreateSequenceResponse` understands  
1467 and conforms with the requirements listed above. Note that an RM Destination understanding this header  
1468 does not mean that it has processed and understood any WS-Security headers, the fault behavior defined  
1469 in WS-Security still applies.

1470 The following exemplar defines the `UsesSequenceSTR` syntax:

```
1471 <wsrm:UsesSequenceSTR ... />
```

1472 The following describes the content model of the `UsesSequenceSTR` header block.

1473 `/wsrm:UsesSequenceSTR`

1474 This element SHOULD be included as a SOAP header block in `CreateSequence` messages that use the  
1475 extensibility mechanism described above in this section. The `soap:mustUnderstand` attribute value  
1476 MUST be 'true'. The receiving RM Destination MUST understand and correctly implement the extension  
1477 described above or else generate a `soap:MustUnderstand` fault, thus aborting the requested  
1478 Sequence creation.

1479 The following is an example of a `CreateSequence` message using the

1480 `wsse:SecurityTokenReference` extension and the `UsesSequenceSTR` header block:

```
1481 <soap:Envelope ...>  
1482   <soap:Header>  
1483     ...  
1484     <wsrm:UsesSequenceSTR soap:mustUnderstand='true' />  
1485     ...  
1486   </soap:Header>  
1487   <soap:Body>  
1488     <wsrm:CreateSequence>  
1489       <wsrm:AcksTo>  
1490         <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>  
1491       </wsrm:AcksTo>  
1492       <wsse:SecurityTokenReference>  
1493         ...  
1494       </wsse:SecurityTokenReference>  
1495     </wsrm:CreateSequence>  
1496   </soap:Body>  
1497 </soap:Envelope>
```

## 1498 6.2 Securing Sequences Using SSL/TLS

1499 One mechanism for protecting a Sequence is to bind the Sequence to the underlying SSL/TLS session(s).  
1500 The RM Source indicates to the RM Destination that a Sequence is to be bound to the underlying  
1501 SSL/TLS session(s) via the `UsesSequenceSSL` header block. If the RM Source wishes to bind a  
1502 Sequence to the underlying SSL/TLS sessions(s) it MUST include the `UsesSequenceSSL` element as a  
1503 SOAP header block within the `CreateSequence` message.

1504 The following exemplar defines the `UsesSequenceSSL` syntax:

```
1505 <wsrm:UsesSequenceSSL soap:mustUnderstand="true" ... />
```

1506 The following describes the content model of the `UsesSequenceSSL` header block.

1507 `/wsrm:UsesSequenceSSL`

1508 The RM Source MAY include this element as a SOAP header block of a `CreateSequence` message to  
1509 indicate to the RM Destination that the resulting Sequence is to be bound to the SSL/TLS session that was

1510 used to carry the `CreateSequence` message. If included, the RM Source MUST mark this header with a  
1511 `soap:mustUnderstand` attribute with a value of 'true'. The receiving RM Destination MUST understand  
1512 and correctly implement the functionality described in Section 5.2.1 or else generate a  
1513 `soap:MustUnderstand` fault, thus aborting the requested Sequence creation.

1514 Note that the use inclusion of the above header by the RM Source implies that all Sequence-related  
1515 information (Sequence Lifecycle or Acknowledgment messages or Sequence-related faults) flowing from  
1516 the RM Destination to the RM Source will be bound to the SSL/TLS session that is used to carry the  
1517 `CreateSequenceResponse` message.

## 7 References

### 7.1 Normative

#### [KEYWORDS]

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

<http://www.ietf.org/rfc/rfc2119.txt>

#### [WS-RM Policy]

OASIS WS-RX Technical Committee Draft, "Web Services ReliableMessaging Policy Assertion( WS-RM Policy)" February 2007

<http://docs.oasis-open.org/ws-rx/wsrmp/200702/wsrmp-1.1-spec-cd-05.pdf>

#### [SOAP 1.1]

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

<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

#### [SOAP 1.2]

W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging Framework" June 2003.-

<http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>

#### [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.-

<http://ietf.org/rfc/rfc3986>

#### [UUID]

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

<http://www.ietf.org/rfc/rfc4122.txt>

#### [XML]

W3C Recommendation, "Extensible Markup Language (XML) 1.0 (Fourth Edition)". ~~September 2006. Second Edition)~~, ~~October 2000.-~~

<http://www.w3.org/TR/REC-xml/>

#### [XML-ns]

W3C Recommendation, "Namespaces in XML," 14 January 1999.-

<http://www.w3.org/TR/1999/REC-xml-names-19990114/>

#### [XML-Schema Part1]

W3C Recommendation, "XML Schema Part 1: Structures," ~~October 2004~~ ~~May 2001~~.

1551 <http://www.w3.org/TR/xmlschema-1/>  
1552 **[XML-Schema Part2]**  
1553 W3C Recommendation, "XML Schema Part 2: Datatypes," ~~October 2004~~ [May 2001](#).  
1554 <http://www.w3.org/TR/xmlschema-2/>  
1555 **[XPath 1.0]**  
1556 W3C Recommendation, "XML Path Language (XPath) Version 1.0," 16 November 1999.  
1557 <http://www.w3.org/TR/xpath>  
1558 **[WSDL 1.1]**  
1559 W3C Note, "Web Services Description Language (WSDL 1.1)," 15 March 2001.  
1560 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>  
1561 **[WS-Addressing]**  
1562 W3C Recommendation, "Web Services Addressing 1.0 - Core", May 2006.  
1563 <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>  
1564 W3C Recommendation, "Web Services Addressing 1.0 – SOAP Binding", May 2006.  
1565 <http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/>

## 1566 **7.2 Non-Normative**

1567 **[BSP 1.0]**  
1568 WS-I Working Group Draft. "Basic Security Profile Version 1.0," ~~August~~ [March](#) 2006  
1569 <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>  
1570 **[RDDL 2.0]**  
1571 Jonathan Borden, Tim Bray, eds. "Resource Directory Description Language (RDDL) 2.0," January 2004  
1572 <http://www.openhealth.org/RDDL/20040118/rddl-20040118.html>  
1573 **[RFC 2617]**  
1574 J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Loutonen, L. Stewart, "HTTP  
1575 Authentication: Basic and Digest Access Authentication," June 1999.  
1576 <http://www.ietf.org/rfc/rfc2617.txt>  
1577 **[RFC 4346]**  
1578 T. Dierks, E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," April 2006.  
1579 <http://www.ietf.org/rfc/rfc4346.txt>  
1580 **[WS-Policy]**  
1581 W3C Member Submission, "Web Services Policy Framework (WS-Policy)," April 2006.  
1582 <http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/>  
1583 **[WS-PolicyAttachment]**  
1584 W3C Member Submission, "Web Services Policy Attachment (WS-PolicyAttachment)," April 2006.

1585 [http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-](http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/)  
1586 [20060425/](http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/)

1587 **[WS-Security]**

1588 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "OASIS Web Services Security:  
1589 SOAP Message Security 1.0 (WS-Security 2004)", OASIS Standard 200401, March 2004.

1590 <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

1591 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "OASIS Web Services Security:  
1592 SOAP Message Security 1.1 (WS-Security 2004)", OASIS Standard 200602, February 2006.

1593 <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>

1594 **[RTTM]**

1595 V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", RFC 1323, May  
1596 1992.

1597 <http://www.rfc-editor.org/rfc/rfc1323.txt>

1598 **[SecurityPolicy]**

1599 G. Della-Libra, et. al. "Web Services Security Policy Language (WS-SecurityPolicy)", July 2005

1600 <http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf>

1601 **[SecureConversation]**

1602 S. Anderson, et al, "Web Services Secure Conversation Language (WS-SecureConversation)," February  
1603 2005.

1604 <http://schemas.xmlsoap.org/ws/2004/04/sc/>

1605 **[Trust]**

1606 S. Anderson, et al, "Web Services Trust Language (WS-Trust)," February 2005.

1607 <http://schemas.xmlsoap.org/ws/2005/02/trust>

## Appendix A. Schema

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

<http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-schema-200702608/wsrn-1.1-schema-200608.xsd>

The following copy is provided for reference.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
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 2002-2007. 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 does 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.
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200702"
targetNamespace="http://docs.oasis-open.org/ws-rx/wsrn/200702"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/2005/08/addressing"
schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
  <!-- Protocol Elements -->
  <xs:complexType name="SequenceType">
    <xs:sequence>
      <xs:element ref="wsrm:Identifier"/>
      <xs:element name="MessageNumber" type="wsrm:MessageNumberType"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>

```



```

1664     </xs:sequence>
1665     <xs:anyAttribute namespace="##other" processContents="lax"/>
1666   </xs:complexType>
1667   <xs:element name="Sequence" type="wsrm:SequenceType"/>
1668   <xs:element name="SequenceAcknowledgement">
1669     <xs:complexType>
1670       <xs:sequence>
1671         <xs:element ref="wsrm:Identifier"/>
1672         <xs:choice>
1673           <xs:sequence>
1674             <xs:choice>
1675               <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
1676                 <xs:complexType>
1677                   <xs:sequence/>
1678                   <xs:attribute name="Upper" type="xs:unsignedLong"
1679 use="required"/>
1680                   <xs:attribute name="Lower" type="xs:unsignedLong"
1681 use="required"/>
1682                 <xs:anyAttribute namespace="##other" processContents="lax"/>
1683               </xs:complexType>
1684             </xs:element>
1685             <xs:element name="None">
1686               <xs:complexType>
1687                 <xs:sequence/>
1688               </xs:complexType>
1689             </xs:element>
1690           </xs:choice>
1691           <xs:element name="Final" minOccurs="0">
1692             <xs:complexType>
1693               <xs:sequence/>
1694             </xs:complexType>
1695           </xs:element>
1696         </xs:sequence>
1697         <xs:element name="Nack" type="xs:unsignedLong"
1698 maxOccurs="unbounded"/>
1699       </xs:choice>
1700       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1701 maxOccurs="unbounded"/>
1702     </xs:sequence>
1703     <xs:anyAttribute namespace="##other" processContents="lax"/>
1704   </xs:complexType>
1705 </xs:element>
1706 <xs:complexType name="AckRequestedType">
1707   <xs:sequence>
1708     <xs:element ref="wsrm:Identifier"/>
1709     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1710 maxOccurs="unbounded"/>
1711   </xs:sequence>
1712   <xs:anyAttribute namespace="##other" processContents="lax"/>
1713 </xs:complexType>
1714 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1715 <xs:element name="Identifier">
1716   <xs:complexType>
1717     <xs:annotation>
1718       <xs:documentation>
1719         This type is for elements whose [children] is an anyURI and can have
1720 arbitrary attributes.
1721       </xs:documentation>
1722     </xs:annotation>
1723     <xs:simpleContent>
1724       <xs:extension base="xs:anyURI">
1725         <xs:anyAttribute namespace="##other" processContents="lax"/>
1726       </xs:extension>

```

```

1727     </xs:simpleContent>
1728   </xs:complexType>
1729 </xs:element>
1730 <xs:element name="Address">
1731   <xs:complexType>
1732     <xs:simpleContent>
1733       <xs:extension base="xs:anyURI">
1734         <xs:anyAttribute namespace="##other" processContents="lax"/>
1735       </xs:extension>
1736     </xs:simpleContent>
1737   </xs:complexType>
1738 </xs:element>
1739 <xs:simpleType name="MessageNumberType">
1740   <xs:restriction base="xs:unsignedLong">
1741     <xs:minInclusive value="1"/>
1742     <xs:maxInclusive value="9223372036854775807"/>
1743   </xs:restriction>
1744 </xs:simpleType>
1745 <!-- Fault Container and Codes -->
1746 <xs:simpleType name="FaultCodes">
1747   <xs:restriction base="xs:QName">
1748     <xs:enumeration value="wsrm:SequenceTerminated"/>
1749     <xs:enumeration value="wsrm:UnknownSequence"/>
1750     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1751     <xs:enumeration value="wsrm:MessageNumberRollover"/>
1752     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1753     <xs:enumeration value="wsrm:SequenceClosed"/>
1754     <xs:enumeration value="wsrm:WSRMRequired"/>
1755     <xs:enumeration value="wsrm:UnsupportedSelection"/>
1756   </xs:restriction>
1757 </xs:simpleType>
1758 <xs:complexType name="SequenceFaultType">
1759   <xs:sequence>
1760     <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1761     <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
1762     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1763 maxOccurs="unbounded"/>
1764   </xs:sequence>
1765   <xs:anyAttribute namespace="##other" processContents="lax"/>
1766 </xs:complexType>
1767 <xs:complexType name="DetailType">
1768   <xs:sequence>
1769     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1770 maxOccurs="unbounded"/>
1771   </xs:sequence>
1772   <xs:anyAttribute namespace="##other" processContents="lax"/>
1773 </xs:complexType>
1774 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1775 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1776 <xs:element name="CreateSequenceResponse"
1777 type="wsrm:CreateSequenceResponseType"/>
1778 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1779 <xs:element name="CloseSequenceResponse"
1780 type="wsrm:CloseSequenceResponseType"/>
1781 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1782 <xs:element name="TerminateSequenceResponse"
1783 type="wsrm:TerminateSequenceResponseType"/>
1784 <xs:complexType name="CreateSequenceType">
1785   <xs:sequence>
1786     <xs:element ref="wsrm:AcksTo"/>
1787     <xs:element ref="wsrm:Expires" minOccurs="0"/>
1788     <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1789     <xs:any namespace="##other" processContents="lax" minOccurs="0"

```

```

1790 maxOccurs="unbounded">
1791   <xs:annotation>
1792     <xs:documentation>
1793       It is the authors intent that this extensibility be used to
1794       transfer a Security Token Reference as defined in WS-Security.
1795     </xs:documentation>
1796   </xs:annotation>
1797 </xs:any>
1798 </xs:sequence>
1799 <xs:anyAttribute namespace="##other" processContents="lax"/>
1800 </xs:complexType>
1801 <xs:complexType name="CreateSequenceResponseType">
1802   <xs:sequence>
1803     <xs:element ref="wsrm:Identifier"/>
1804     <xs:element ref="wsrm:Expires" minOccurs="0"/>
1805     <xs:element name="IncompleteSequenceBehavior"
1806       type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
1807     <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1808     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1809       maxOccurs="unbounded"/>
1810   </xs:sequence>
1811   <xs:anyAttribute namespace="##other" processContents="lax"/>
1812 </xs:complexType>
1813 <xs:complexType name="CloseSequenceType">
1814   <xs:sequence>
1815     <xs:element ref="wsrm:Identifier"/>
1816 Rights Reserved.
1817 This document and translations of it may be copied and furnished to others,
1818 and derivative works that comment on or otherwise explain it or assist in its
1819 implementation may be prepared, copied, published and distributed, in whole or
1820 in part, without restriction of any kind, provided that the above copyright
1821 notice and this paragraph are included on all such copies and derivative
1822 works. However, this document itself does not be modified in any way, such as
1823 by removing the copyright notice or references to OASIS, except as needed for
1824 the purpose of developing OASIS specifications, in which case the procedures
1825 for copyrights defined in the OASIS Intellectual Property Rights document must
1826 be followed, or as required to translate it into languages other than English.
1827 The limited permissions granted above are perpetual and will not be revoked by
1828 OASIS or its successors or assigns.
1829 This document and the information contained herein is provided on an "AS IS"
1830 basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
1831 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT
1832 INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS
1833 FOR A PARTICULAR PURPOSE.
1834 →
1835 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1836   xmlns:wsa="http://www.w3.org/2005/08/addressing"
1837   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200608"
1838   targetNamespace="http://docs.oasis-open.org/ws-rx/wsr/200608"
1839   elementFormDefault="qualified" attributeFormDefault="unqualified">
1840   <xs:import namespace="http://www.w3.org/2005/08/addressing"
1841     schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
1842   <!-- Protocol Elements -->
1843   <xs:complexType name="SequenceType">
1844     <xs:sequence>
1845       <xs:element ref="wsrm:Identifier"/>
1846       <xs:element name="MessageNumber" type="wsrm:MessageNumberType"/>
1847       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1848         maxOccurs="unbounded"/>
1849     </xs:sequence>
1850   <xs:anyAttribute namespace="##other" processContents="lax"/>
1851 </xs:complexType>
1852 <xs:element name="Sequence" type="wsrm:SequenceType"/>

```

```

1853 <xs:element name="SequenceAcknowledgement">
1854 <xs:complexType>
1855 <xs:sequence>
1856 <xs:element ref="wsrm:Identifier"/>
1857 <xs:choice>
1858 <xs:sequence>
1859 <xs:choice>
1860 <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
1861 <xs:complexType>
1862 <xs:sequence/>
1863 <xs:attribute name="Upper" type="xs:unsignedLong"
1864 use="required"/>
1865 <xs:attribute name="Lower" type="xs:unsignedLong"
1866 use="required"/>
1867 <xs:anyAttribute namespace="##other" processContents="lax"/>
1868 </xs:complexType>
1869 </xs:element>
1870 <xs:element name="None">
1871 <xs:complexType>
1872 <xs:sequence/>
1873 </xs:complexType>
1874 </xs:element>
1875 </xs:choice>
1876 <xs:element name="Final" minOccurs="0">
1877 <xs:complexType>
1878 <xs:sequence/>
1879 </xs:complexType>
1880 </xs:element>
1881 </xs:sequence>
1882 <xs:element name="Nack" type="xs:unsignedLong"
1883 maxOccurs="unbounded"/>
1884 </xs:choice>
1885 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1886 maxOccurs="unbounded"/>
1887 </xs:sequence>
1888 <xs:anyAttribute namespace="##other" processContents="lax"/>
1889 </xs:complexType>
1890 </xs:element>
1891 <xs:complexType name="AckRequestedType">
1892 <xs:sequence>
1893 <xs:element ref="wsrm:Identifier"/>
1894 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1895 maxOccurs="unbounded"/>
1896 </xs:sequence>
1897 <xs:anyAttribute namespace="##other" processContents="lax"/>
1898 </xs:complexType>
1899 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1900 <xs:complexType name="MessagePendingType">
1901 <xs:sequence>
1902 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1903 maxOccurs="unbounded"/>
1904 </xs:sequence>
1905 <xs:attribute name="pending" type="xs:boolean"/>
1906 <xs:anyAttribute namespace="##other" processContents="lax"/>
1907 </xs:complexType>
1908 <xs:element name="MessagePending" type="wsrm:MessagePendingType"/>
1909 <xs:element name="Identifier">
1910 <xs:complexType>
1911 <xs:annotation>
1912 <xs:documentation>
1913 This type is for elements whose [children] is an anyURI and can have
1914 arbitrary attributes.
1915 </xs:documentation>

```

```

1916 </xs:annotation>
1917 <xs:simpleContent>
1918 <xs:extension base="xs:anyURI">
1919 <xs:anyAttribute namespace="##other" processContents="lax"/>
1920 </xs:extension>
1921 </xs:simpleContent>
1922 </xs:complexType>
1923 </xs:element>
1924 <xs:element name="Address">
1925 <xs:complexType>
1926 <xs:simpleContent>
1927 <xs:extension base="xs:anyURI">
1928 <xs:anyAttribute namespace="##other" processContents="lax"/>
1929 </xs:extension>
1930 </xs:simpleContent>
1931 </xs:complexType>
1932 </xs:element>
1933 <xs:complexType name="MakeConnectionType">
1934 <xs:sequence>
1935 <xs:element ref="wsrm:Identifier" minOccurs="0" maxOccurs="1"/>
1936 <xs:element ref="wsrm:Address" minOccurs="0" maxOccurs="1"/>
1937 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1938 maxOccurs="unbounded"/>
1939 </xs:sequence>
1940 <xs:anyAttribute namespace="##other" processContents="lax"/>
1941 </xs:complexType>
1942 <xs:element name="MakeConnection" type="wsrm:MakeConnectionType"/>
1943 <xs:simpleType name="MessageNumberType">
1944 <xs:restriction base="xs:unsignedLong">
1945 <xs:minInclusive value="1"/>
1946 <xs:maxInclusive value="9223372036854775807"/>
1947 </xs:restriction>
1948 </xs:simpleType>
1949 <!-- Fault Container and Codes -->
1950 <xs:simpleType name="FaultCodes">
1951 <xs:restriction base="xs:QName">
1952 <xs:enumeration value="wsrm:SequenceTerminated"/>
1953 <xs:enumeration value="wsrm:UnknownSequence"/>
1954 <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1955 <xs:enumeration value="wsrm:MessageNumberRollover"/>
1956 <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1957 <xs:enumeration value="wsrm:SequenceClosed"/>
1958 <xs:enumeration value="wsrm:WSRMRequired"/>
1959 <xs:enumeration value="wsrm:UnsupportedSelection"/>
1960 </xs:restriction>
1961 </xs:simpleType>
1962 <xs:complexType name="SequenceFaultType">
1963 <xs:sequence>
1964 <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1965 <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
1966 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1967 maxOccurs="unbounded"/>
1968 </xs:sequence>
1969 <xs:anyAttribute namespace="##other" processContents="lax"/>
1970 </xs:complexType>
1971 <xs:complexType name="DetailType">
1972 <xs:sequence>
1973 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1974 maxOccurs="unbounded"/>
1975 </xs:sequence>
1976 <xs:anyAttribute namespace="##other" processContents="lax"/>
1977 </xs:complexType>
1978 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>

```

```

1799 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1800 <xs:element name="CreateSequenceResponse"
1801 type="wsrm:CreateSequenceResponseType"/>
1802 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1803 <xs:element name="CloseSequenceResponse"
1804 type="wsrm:CloseSequenceResponseType"/>
1805 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1806 <xs:element name="TerminateSequenceResponse"
1807 type="wsrm:TerminateSequenceResponseType"/>
1808 <xs:complexType name="CreateSequenceType">
1809 <xs:sequence>
1810 <xs:element ref="wsrm:AcksTo"/>
1811 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1812 <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1813 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1814 maxOccurs="unbounded"/>
1815 <xs:annotation>
1816 <xs:documentation>
1817 It is the authors intent that this extensibility be used to
1818 transfer a Security Token Reference as defined in WS-Security.
1819 </xs:documentation>
1820 </xs:annotation>
1821 </xs:any>
1822 </xs:sequence>
1823 <xs:anyAttribute namespace="##other" processContents="lax"/>
1824 </xs:complexType>
1825 <xs:complexType name="CreateSequenceResponseType">
1826 <xs:sequence>
1827 <xs:element ref="wsrm:Identifier"/>
1828 <xs:element ref="wsrm:Expires" minOccurs="0"/>
1829 <xs:element name="IncompleteSequenceBehavior"
1830 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
1831 <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1832 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1833 maxOccurs="unbounded"/>
1834 </xs:sequence>
1835 <xs:anyAttribute namespace="##other" processContents="lax"/>
1836 </xs:complexType>
1837 <xs:complexType name="CloseSequenceType">
1838 <xs:sequence>
1839 <xs:element ref="wsrm:Identifier"/>
1840 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1841 maxOccurs="unbounded"/>
1842 </xs:sequence>
1843 <xs:anyAttribute namespace="##other" processContents="lax"/>
1844 </xs:complexType>
1845 <xs:complexType name="CloseSequenceResponseType">
1846 <xs:sequence>
1847 <xs:element ref="wsrm:Identifier"/>
1848 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1849 maxOccurs="unbounded"/>
1850 </xs:sequence>
1851 <xs:anyAttribute namespace="##other" processContents="lax"/>
1852 </xs:complexType>
1853 <xs:complexType name="TerminateSequenceType">
1854 <xs:sequence>
1855 <xs:element ref="wsrm:Identifier"/>
1856 <xs:any namespace="##other" processContents="lax" minOccurs="0"
1857 maxOccurs="unbounded"/>
1858 </xs:sequence>
1859 <xs:anyAttribute namespace="##other" processContents="lax"/>
1860 </xs:complexType>
1861 <xs:complexType name="TerminateSequenceResponseType">

```



```

2042 <xs:sequence>
2043 <xs:element ref="wsrm:Identifier"/>
2044 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2045 maxOccurs="unbounded"/>
2046 </xs:sequence>
2047 <xs:anyAttribute namespace="##other" processContents="lax"/>
2048 </xs:complexType>
2049 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
2050 <xs:complexType name="OfferType">
2051 <xs:sequence>
2052 <xs:element ref="wsrm:Identifier"/>
2053 <xs:element name="Endpoint" type="wsa:EndpointReferenceType"/>
2054 <xs:element ref="wsrm:Expires" minOccurs="0"/>
2055 <xs:element name="IncompleteSequenceBehavior"
2056 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
2057 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2058 maxOccurs="unbounded"/>
2059 </xs:sequence>
2060 <xs:anyAttribute namespace="##other" processContents="lax"/>
2061 </xs:complexType>
2062 <xs:complexType name="AcceptType">
2063 <xs:sequence>
2064 <xs:element ref="wsrm:AcksTo"/>
2065 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2066 maxOccurs="unbounded"/>
2067 </xs:sequence>
2068 <xs:anyAttribute namespace="##other" processContents="lax"/>
2069 </xs:complexType>
2070 <xs:element name="Expires">
2071 <xs:complexType>
2072 <xs:simpleContent>
2073 <xs:extension base="xs:duration">
2074 <xs:anyAttribute namespace="##other" processContents="lax"/>
2075 </xs:extension>
2076 </xs:simpleContent>
2077 </xs:complexType>
2078 </xs:element>
2079 <xs:simpleType name="IncompleteSequenceBehaviorType">
2080 <xs:restriction base="xs:string">
2081 <xs:enumeration value="DiscardEntireSequence"/>
2082 <xs:enumeration value="DiscardFollowingFirstGap"/>
2083 <xs:enumeration value="NoDiscard"/>
2084 </xs:restriction>
2085 </xs:simpleType>
2086 <xs:element ref="wsrm:MessageNumberType"/>
2087 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2088 maxOccurs="unbounded"/>
2089 </xs:sequence>
2090 <xs:anyAttribute namespace="##other" processContents="lax"/>
2091 </xs:complexType>
2092 <xs:complexType name="CloseSequenceResponseType">
2093 <xs:sequence>
2094 <xs:element ref="wsrm:Identifier"/>
2095 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2096 maxOccurs="unbounded"/>
2097 </xs:sequence>
2098 <xs:anyAttribute namespace="##other" processContents="lax"/>
2099 </xs:complexType>
2100 <xs:complexType name="TerminateSequenceType">
2101 <xs:sequence>
2102 <xs:element ref="wsrm:Identifier"/>
2103 <xs:element ref="wsrm:MessageNumberType" minOccurs="0"/>
2104 <xs:any namespace="##other" processContents="lax" minOccurs="0"

```



```

2105 maxOccurs="unbounded"/>
2106 </xs:sequence>
2107 <xs:anyAttribute namespace="##other" processContents="lax"/>
2108 </xs:complexType>
2109 <xs:complexType name="TerminateSequenceResponseType">
2110 <xs:sequence>
2111 <xs:element ref="wsrm:Identifier"/>
2112 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2113 maxOccurs="unbounded"/>
2114 </xs:sequence>
2115 <xs:anyAttribute namespace="##other" processContents="lax"/>
2116 </xs:complexType>
2117 <xs:element name="AcksTo" type="wsa:EndpointReferenceType"/>
2118 <xs:complexType name="OfferType">
2119 <xs:sequence>
2120 <xs:element ref="wsrm:Identifier"/>
2121 <xs:element name="Endpoint" type="wsa:EndpointReferenceType"/>
2122 <xs:element ref="wsrm:Expires" minOccurs="0"/>
2123 <xs:element name="IncompleteSequenceBehavior"
2124 type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
2125 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2126 maxOccurs="unbounded"/>
2127 </xs:sequence>
2128 <xs:anyAttribute namespace="##other" processContents="lax"/>
2129 </xs:complexType>
2130 <xs:complexType name="AcceptType">
2131 <xs:sequence>
2132 <xs:element ref="wsrm:AcksTo"/>
2133 <xs:any namespace="##other" processContents="lax" minOccurs="0"
2134 maxOccurs="unbounded"/>
2135 </xs:sequence>
2136 <xs:anyAttribute namespace="##other" processContents="lax"/>
2137 </xs:complexType>
2138 <xs:element name="Expires">
2139 <xs:complexType>
2140 <xs:simpleContent>
2141 <xs:extension base="xs:duration">
2142 <xs:anyAttribute namespace="##other" processContents="lax"/>
2143 </xs:extension>
2144 </xs:simpleContent>
2145 </xs:complexType>
2146 </xs:element>
2147 <xs:simpleType name="IncompleteSequenceBehaviorType">
2148 <xs:restriction base="xs:string">
2149 <xs:enumeration value="DiscardEntireSequence"/>
2150 <xs:enumeration value="DiscardFollowingFirstGap"/>
2151 <xs:enumeration value="NoDiscard"/>
2152 </xs:restriction>
2153 </xs:simpleType>
2154 <xs:element name="UsesSequenceSTR">
2155 <xs:complexTypesequence/>
2156 <xs:sequence<xs:anyAttribute namespace="##other" processContents="lax"/>
2157 <xs:anyAttribute namespace="##other" processContents="lax"/>
2158 </xs:complexType>
2159 </xs:element>
2160 <xs:element name="UsesSequenceSSL">
2161 <xs:complexTypesequence/>
2162 <xs:sequence<xs:anyAttribute namespace="##other" processContents="lax"/>
2163 <xs:anyAttribute namespace="##other" processContents="lax"/>
2164 </xs:complexType>
2165 </xs:element>
2166 <xs:element name="UnsupportedElement">
2167 <xs:simpleType>

```

```
2168     <xs:restriction base="xs:QName"/>
2169   </xs:simpleType>
2170 </xs:element>
2171 </xs:schema>
```

## Appendix B. WSDL

This WSDL describes the WS-RM protocol from the point of view of an RM Destination. In the case where an endpoint acts both as an RM Destination and an RM Source, note that additional messages may be present in exchanges with that endpoint.

Also note that this WSDL is intended to describe the internal structure of the WS-RM protocol, and will not generally appear in a description of a WS-RM-capable Web service. See WS-RM Policy [WS-RM Policy] for a higher-level mechanism to indicate that WS-RM is engaged.

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

<http://docs.oasis-open.org/ws-rx/wsrml-1.1-wsdl-200702608/wsd/wsrml-1.1-wsdl-200608.wsdl>

The following non-normative copy is provided for reference.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
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 2002-2007. 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 does 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.
-->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
open.org/ws-rx/wsrml-1.1-wsdl-200702" xmlns:tns="http://docs.oasis-open.org/ws-
rx/wsrml-1.1-wsdl-200702" targetNamespace="http://docs.oasis-open.org/ws-
rx/wsrml-1.1-wsdl-200702">
```

```

2225 <wsdl:types>
2226 <xs:schema>
2227 <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsrn/200702"
2228 schemaLocation="http://docs.oasis-open.org/ws-rx/wsrn/200702/wsrn-1.1-schema-
2229 200702.xsd"/>
2230 </xs:schema>
2231 </wsdl:types>

2232 <wsdl:message name="CreateSequence">
2233 <wsdl:part name="create" element="rm:CreateSequence"/>
2234 </wsdl:message>
2235 <wsdl:message name="CreateSequenceResponse">
2236 <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
2237 </wsdl:message>
2238 <wsdl:message name="CloseSequence">
2239 <wsdl:part name="close" element="rm:CloseSequence"/>
2240 </wsdl:message>
2241 <wsdl:message name="CloseSequenceResponse">
2242 <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
2243 </wsdl:message>
2244 <wsdl:message name="TerminateSequence">
2245 <wsdl:part name="terminate" element="rm:TerminateSequence"/>
2246 </wsdl:message>
2247 <wsdl:message name="TerminateSequenceResponse">
2248 <wsdl:part name="terminateResponse"
2249 element="rm:TerminateSequenceResponse"/>
2250 </wsdl:message>

2251 <wsdl:portType name="SequenceAbstractPortType">
2252 <wsdl:operation name="CreateSequence">
2253 <wsdl:input message="tns:CreateSequence" wsaw:Action="http://docs.oasis-
2254 open.org/ws-rx/wsrn/200702/CreateSequence"/>
2255 <wsdl:output message="tns:CreateSequenceResponse"
2256 wsaw:Action="http://docs.oasis-open.org/ws-
2257 rx/wsrn/200702/CreateSequenceResponse"/>
2258 </wsdl:operation>
2259 <wsdl:operation name="CloseSequence">
2260 <wsdl:input message="tns:CloseSequence" wsaw:Action="http://docs.oasis-
2261 open.org/ws-rx/wsrn/200702/CloseSequence"/>
2262 <wsdl:output message="tns:CloseSequenceResponse"
2263 wsaw:Action="http://docs.oasis-open.org/ws-
2264 rx/wsrn/200702/CloseSequenceResponse"/>
2265 </wsdl:operation>
2266 <wsdl:operation name="TerminateSequence">
2267 <wsdl:input message="tns:TerminateSequence"
2268 wsaw:Action="http://docs.oasis-open.org/ws-rx/wsrn/200702/TerminateSequence"/>
2269 <wsdl:output message="tns:TerminateSequenceResponse"
2270 wsaw:Action="http://docs.oasis-open.org/ws-
2271 rx/wsrn/200702/TerminateSequenceResponse"/>
2272 This document and translations of it may be copied and furnished to others,
2273 and derivative works that comment on or otherwise explain it or assist in its
2274 implementation may be prepared, copied, published and distributed, in whole or
2275 in part, without restriction of any kind, provided that the above copyright
2276 notice and this paragraph are included on all such copies and derivative
2277 works. However, this document itself does not be modified in any way, such as
2278 by removing the copyright notice or references to OASIS, except as needed for
2279 the purpose of developing OASIS specifications, in which case the procedures
2280 for copyrights defined in the OASIS Intellectual Property Rights document must
2281 be followed, or as required to translate it into languages other than English.
2282 The limited permissions granted above are perpetual and will not be revoked by
2283 OASIS or its successors or assigns.
2284 This document and the information contained herein is provided on an "AS IS"
2285 basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT

```

```

2286 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT
2287 INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS
2288 FOR A PARTICULAR PURPOSE.
2289 →
2290 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
2291 xmlns:xs="http://www.w3.org/2001/XMLSchema"
2292 xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
2293 open.org/ws-rx/wsrn/200608" xmlns:tns="http://docs.oasis-open.org/ws-
2294 rx/wsrn/200608/wsdl" targetNamespace="http://docs.oasis-open.org/ws-
2295 rx/wsrn/200608/wsdl">
2296   <wsdl:types>
2297     <xs:schema>
2298       <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsrn/200608"
2299       schemaLocation="http://docs.oasis-open.org/ws-rx/wsrn/200608/wsrn-1.1-schema-
2300       200608.xsd"/>
2301     </xs:schema>
2302   </wsdl:types>
2303   <wsdl:message name="CreateSequence">
2304     <wsdl:part name="create" element="rm:CreateSequence"/>
2305   </wsdl:message>
2306   <wsdl:message name="CreateSequenceResponse">
2307     <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
2308   </wsdl:message>
2309   <wsdl:message name="CloseSequence">
2310     <wsdl:part name="close" element="rm:CloseSequence"/>
2311   </wsdl:message>
2312   <wsdl:message name="CloseSequenceResponse">
2313     <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
2314   </wsdl:message>
2315   <wsdl:message name="TerminateSequence">
2316     <wsdl:part name="terminate" element="rm:TerminateSequence"/>
2317   </wsdl:message>
2318   <wsdl:message name="TerminateSequenceResponse">
2319     <wsdl:part name="terminateResponse"
2320     element="rm:TerminateSequenceResponse"/>
2321   </wsdl:message>
2322   <wsdl:message name="MakeConnection">
2323     <wsdl:part name="makeConnection" element="rm:MakeConnection"/>
2324   </wsdl:message>
2325   <wsdl:portType name="SequenceAbstractPortType">
2326     <wsdl:operation name="CreateSequence">
2327       <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
2328       open.org/ws-rx/wsrn/200608/CreateSequence"/>
2329       <wsdl:output message="tns:CreateSequenceResponse"
2330       wsa:Action="http://docs.oasis-open.org/ws-
2331       rx/wsrn/200608/CreateSequenceResponse"/>
2332     </wsdl:operation>
2333     <wsdl:operation name="CloseSequence">
2334       <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
2335       open.org/ws-rx/wsrn/200608/CloseSequence"/>
2336       <wsdl:output message="tns:CloseSequenceResponse"
2337       wsa:Action="http://docs.oasis-open.org/ws-
2338       rx/wsrn/200608/CloseSequenceResponse"/>
2339     </wsdl:operation>
2340     <wsdl:operation name="TerminateSequence">
2341       <wsdl:input message="tns:TerminateSequence"
2342       wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200608/TerminateSequence"/>
2343       <wsdl:output message="tns:TerminateSequenceResponse"
2344       wsa:Action="http://docs.oasis-open.org/ws-
2345       rx/wsrn/200608/TerminateSequenceResponse"/>

```

```
2346 </wsdl:operation>
2347 <wsdl:operation name="MakeConnection">
2348 <wsdl:input message="tns:MakeConnection" wsaw:Action="http://docs.oasis-
2349 open.org/ws-rx/wsrn/200608/MakeConnection"/>
2350 </wsdl:operation>
2351 </wsdl:portType>
2352 </wsdl:definitions>
```

## Appendix C. Message Examples

### Appendix C.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:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/20070208"
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/wsmr/20070208/CreateSequence</wsa:Action>
    <wsa:ReplyTo>
      <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
    </wsa:ReplyTo>
  </S:Header>
  <S:Body>
    <wsmr:CreateSequence>
      <wsmr:AcksTo>
        <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
      </wsmr:AcksTo>
    </wsmr: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:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/20070208"
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/wsmr/20070208/CreateSequenceResponse
    </wsa:Action>
  </S:Header>
  <S:Body>
    <wsmr:CreateSequenceResponse>
      <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
    </wsmr:CreateSequenceResponse>
  </S:Body>
</S:Envelope>
```

### Appendix C.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:

#### Message 1



```

2404 <?xml version="1.0" encoding="UTF-8"?>
2405 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2406 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702608"
2407 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2408   <S:Header>
2409     <wsa:MessageID>
2410       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
2411     </wsa:MessageID>
2412     <wsa:To>http://example.com/serviceB/123</wsa:To>
2413     <wsa:From>
2414       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2415     </wsa:From>
2416     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2417     <wsmr:Sequence>
2418       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2419       <wsmr:MessageNumber>1</wsmr:MessageNumber>
2420     </wsmr:Sequence>
2421   </S:Header>
2422   <S:Body>
2423     <!-- Some Application Data -->
2424   </S:Body>
2425 </S:Envelope>

```

## 2426 Message 2

```

2427 <?xml version="1.0" encoding="UTF-8"?>
2428 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2429 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702608"
2430 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2431   <S:Header>
2432     <wsa:MessageID>
2433       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
2434     </wsa:MessageID>
2435     <wsa:To>http://example.com/serviceB/123</wsa:To>
2436     <wsa:From>
2437       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2438     </wsa:From>
2439     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2440     <wsmr:Sequence>
2441       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2442       <wsmr:MessageNumber>2</wsmr:MessageNumber>
2443     </wsmr:Sequence>
2444   </S:Header>
2445   <S:Body>
2446     <!-- Some Application Data -->
2447   </S:Body>
2448 </S:Envelope>

```

## 2449 Message 3

```

2450 <?xml version="1.0" encoding="UTF-8"?>
2451 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2452 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702608"
2453 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2454   <S:Header>
2455     <wsa:MessageID>
2456       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
2457     </wsa:MessageID>
2458     <wsa:To>http://example.com/serviceB/123</wsa:To>
2459     <wsa:From>
2460       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2461     </wsa:From>
2462     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2463     <wsmr:Sequence>
2464       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>

```

```

2465     <wsrm:MessageNumber>3</wsrm:MessageNumber>
2466   </wsrm:Sequence>
2467   <wsrm:AckRequested>
2468     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2469   </wsrm:AckRequested>
2470 </S:Header>
2471 <S:Body>
2472   <!-- Some Application Data -->
2473 </S:Body>
2474 </S:Envelope>

```

## 2475 Appendix C.3 First Acknowledgement

2476 Message number 2 has not been accepted by the RM Destination due to some transmission error so it  
 2477 responds with an Acknowledgement for messages 1 and 3:

```

2478 <?xml version="1.0" encoding="UTF-8"?>
2479 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2480 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200702698"
2481 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2482   <S:Header>
2483     <wsa:MessageID>
2484       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
2485     </wsa:MessageID>
2486     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
2487     <wsa:From>
2488       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
2489     </wsa:From>
2490     <wsa:Action>
2491       http://docs.oasis-open.org/ws-rx/wsrn/200702698/SequenceAcknowledgement
2492     </wsa:Action>
2493     <wsrm:SequenceAcknowledgement>
2494       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2495       <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
2496       <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
2497     </wsrm:SequenceAcknowledgement>
2498   </S:Header>
2499   <S:Body/>
2500 </S:Envelope>

```

## 2501 Appendix C.4 Retransmission

2502 The RM Sourcediscovers that message number 2 was not accepted so it resends the message and  
 2503 requests an Acknowledgement:

```

2504 <?xml version="1.0" encoding="UTF-8"?>
2505 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2506 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200702698"
2507 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2508   <S:Header>
2509     <wsa:MessageID>
2510       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
2511     </wsa:MessageID>
2512     <wsa:To>http://example.com/serviceB/123</wsa:To>
2513     <wsa:From>
2514       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2515     </wsa:From>
2516     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
2517     <wsrm:Sequence>
2518       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2519       <wsrm:MessageNumber>2</wsrm:MessageNumber>
2520     </wsrm:Sequence>

```

```

2521 <wsrm:AckRequested>
2522 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2523 </wsrm:AckRequested>
2524 </S:Header>
2525 <S:Body>
2526 <!-- Some Application Data -->
2527 </S:Body>
2528 </S:Envelope>

```

## 2529 Appendix C.5 Termination

2530 The RM Destination now responds with an Acknowledgement for the complete Sequence which can then  
 2531 be terminated:

```

2532 <?xml version="1.0" encoding="UTF-8"?>
2533 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2534 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200702698"
2535 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2536 <S:Header>
2537 <wsa:MessageID>
2538 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
2539 </wsa:MessageID>
2540 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
2541 <wsa:From>
2542 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
2543 </wsa:From>
2544 <wsa:Action>
2545 http://docs.oasis-open.org/ws-rx/wsrn/200702698/SequenceAcknowledgement
2546 </wsa:Action>
2547 <wsrm:SequenceAcknowledgement>
2548 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2549 <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
2550 </wsrm:SequenceAcknowledgement>
2551 </S:Header>
2552 <S:Body/>
2553 </S:Envelope>

```

### 2554 Terminate Sequence

```

2555 <?xml version="1.0" encoding="UTF-8"?>
2556 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2557 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200702698"
2558 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2559 <S:Header>
2560 <wsa:MessageID>
2561 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
2562 </wsa:MessageID>
2563 <wsa:To>http://example.com/serviceB/123</wsa:To>
2564 <wsa:Action>
2565 http://docs.oasis-open.org/ws-rx/wsrn/200702698/TerminateSequence
2566 </wsa:Action>
2567 <wsa:From>
2568 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2569 </wsa:From>
2570 </S:Header>
2571 <S:Body>
2572 <wsrm:TerminateSequence>
2573 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
2574 <wsrm>LastMsgNumber> 3 </wsrm>LastMsgNumber>
2575 </wsrm:TerminateSequence>
2576 </S:Body>
2577 </S:Envelope>

```

### 2578 Terminate Sequence Response

```

2579 <?xml version="1.0" encoding="UTF-8"?>
2580 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2581 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702608"
2582 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2583   <S:Header>
2584     <wsa:MessageID>
2585       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
2586     </wsa:MessageID>
2587     <wsa:To>http://example.com/serviceA/789</wsa:To>
2588     <wsa:Action>
2589       http://docs.oasis-open.org/ws-rx/wsmr/200702608/TerminateSequenceResponse
2590     </wsa:Action>
2591     <wsa:RelatesTo>
2592       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
2593     </wsa:RelatesTo>
2594     <wsa:From>
2595       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
2596     </wsa:From>
2597   </S:Header>
2598   <S:Body>
2599     <wsmr:TerminateSequenceResponse>
2600       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
2601     </wsmr:TerminateSequenceResponse>
2602   </S:Body>
2603 </S:Envelope>

```

## 2604 Appendix C.6 MakeConnection

2605 To illustrate how a MakeConnection message exchange can be used to deliver messages to an  
 2606 Endpoint that is not addressable, consider the case of a pub/sub scenario in which the Endpoint to which  
 2607 notifications are to be delivered (the "event consumer") is not addressable by the notification sending  
 2608 Endpoint (the "event producer"). In this scenario the event consumer must initiate the connections in order  
 2609 for the notifications to be delivered. One possible set of message exchanges (using HTTP) that  
 2610 demonstrate how this can be achieved using MakeConnection is shown below:

2611 **Step 1**—During a "subscribe" operation, the event consumer's EPR specifies the RM anonymous URI  
 2612 and the RM Policy Assertion to indicate whether or not RM is required:

```

2613 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2614 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200608"
2615 xmlns:wsmrp="http://docs.oasis-open.org/ws-rx/wsmrp/200608"
2616 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2617   <S:Header>
2618     <wsa:To> http://example.org/subscriptionService </wsa:To>
2619     <wsa:MessageID> http://client456.org/id-a6d8-a7c2eb546813 </wsa:MessageID>
2620     <wsa:ReplyTo>
2621       <wsa:To> http://client456.org/response </wsa:To>
2622     </wsa:ReplyTo>
2623   </S:Header>
2624   <S:Body>
2625     <sub:Subscribe xmlns:sub="http://example.org/subscriptionService">
2626       <!-- subscription service specific data -->
2627       <targetEPR>
2628         <wsa:Address>http://docs.oasis-open.org/ws-
2629 rx/wsmr/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:Address>
2630       <wsa:Metadata>
2631         <wsp:Policy wsu:Id="MyPolicy">
2632           <wsmrp:RMAssertion/>
2633         </wsp:Policy>
2634       </wsa:Metadata>
2635     </targetEPR>
2636   </sub:Subscribe>

```

```
2637 </S:Body>
2638 </S:Envelope>
```

2639 In this example the `subscribe` and `targetEPR` elements are simply examples of what a subscription request message might contain. Note: the `wsa:Address` element contains the RM anonymous URI indicating that the notification producer needs to queue the messages until they are requested using the `MakeConnection` message exchange. The EPR also contains the RM Policy Assertion indicating the RM must be used when notifications related to this subscription are sent.

2644 **Step 2**—Once the subscription is established, the event consumer checks for a pending message:

```
2645 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2646 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200608"
2647 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2648 <S:Header>
2649 <wsa:Action>http://docs.oasis-open.org/ws-
2650 rx/wsm/200608/MakeConnection</wsa:Action>
2651 <wsa:To>http://example.org/subscriptionService</wsa:To>
2652 </S:Header>
2653 <S:Body>
2654 <wsm:MakeConnection>
2655 <wsm:Address>http://docs.oasis-open.org/ws-
2656 rx/wsm/200608/anonymous?id=550e8400-e29b-11d4-a716-
2657 446655440000</wsm:Address>
2658 </wsm:MakeConnection>
2659 </S:Body>
2660 </S:Envelope>
```

2661 **Step 3**—If there are messages waiting to be delivered then a message will be returned back to the event consumer. However, because WS-RM is being used to deliver the messages, the first message returned is a `CreateSequence`:

```
2664 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
2665 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200608"
2666 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2667 <S:Header>
2668 <wsa:Action>http://docs.oasis-open.org/ws-
2669 rx/wsm/200608/CreateSequence</wsa:Action>
2670 <wsa:To>http://docs.oasis-open.org/ws-
2671 rx/wsm/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:To>
2672 <wsa:ReplyTo>http://example.org/subscriptionService</wsa:ReplyTo>
2673 <wsa:MessageID>http://example.org/id-123-456</wsa:MessageID>
2674 </S:Header>
2675 <S:Body>
2676 <wsm:CreateSequence>
2677 <wsm:AcksTo>
2678 <wsa:Address>http://example.org/subscriptionService</wsa:Address>
2679 </wsm:AcksTo>
2680 </wsm:CreateSequence>
2681 </S:Body>
2682 </S:Envelope>
```

2683 Notice from the perspective of how the RM Source on the event producer interacts with the RM Destination of those messages, nothing new is introduced by the use of the `MakeConnection`, the use of RM protocol is the same as the case where the event consumer is addressable.

2686 **Step 4**—The event consumer will respond with a `CreateSequenceResponse` message per normal WS-Addressing rules:

```

2688 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"-
2689 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200608"-
2690 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2691   <S:Header>
2692     <wsa:Action>http://docs.oasis-open.org/ws-
2693 rx/wsm/200608/CreateSequenceResponse</wsa:Action>
2694     <wsa:To> http://example.org/subscriptionService </wsa:To>
2695     <wsa:RelatesTo> http://example.org/id-123-456 </wsa:RelatesTo>
2696   </S:Header>
2697   <S:Body>
2698     <wsm:CreateSequenceResponse>
2699       <wsm:Identifier> http://example.org/rmid-456 </wsm:Identifier>
2700     </wsm:CreateSequenceResponse>
2701   </S:Body>
2702 </S:Envelope>

```

Note, this message is carried on an HTTP request directed to the `wsa:ReplyTo` EPR, and the HTTP response will be an HTTP 202.

**Step 5**—The event consumer checks for another message pending:

```

2706 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"-
2707 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200608"-
2708 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2709   <S:Header>
2710     <wsa:Action>http://docs.oasis-open.org/ws-
2711 rx/wsm/200608/MakeConnection</wsa:Action>
2712     <wsa:To> http://example.org/subscriptionService </wsa:To>
2713   </S:Header>
2714   <S:Body>
2715     <wsm:MakeConnection>
2716       <wsm:Address>http://docs.oasis-open.org/ws-
2717 rx/wsm/200608/anonymous?id=550e8400-e29b-11d4-a716-
2718 446655440000</wsm:Address>
2719     </wsm:MakeConnection>
2720   </S:Body>
2721 </S:Envelope>

```

Notice this is the same message as the one sent in step 2.

**Step 6**—If there is a message pending for this destination then it is returned on the HTTP response:

```

2724 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"-
2725 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200608"-
2726 xmlns:wsa="http://www.w3.org/2005/08/addressing">
2727   <S:Header>
2728     <wsa:Action> http://example.org/eventType1 </wsa:Action>
2729     <wsa:To>http://docs.oasis-open.org/ws-
2730 rx/wsm/200608/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:To>
2731     <wsm:Sequence>
2732       <wsm:Identifier> http://example.org/rmid-456 </wsm:Identifier>
2733     </wsm:Sequence>
2734     <wsm:MessagePending pending="true"/>
2735   </S:Header>
2736   <S:Body>
2737     <!-- event specific data -->
2738   </S:Body>
2739 </S:Envelope>

```

As noted in step 3, the use of the RM protocol does not change when using `MakeConnection`. The format of the messages, the order of the messages sent and the timing of when to send it remains the same.



**Step 7**— At some later interval, or immediately due to the `MessagePending` header's "pending" attribute being set to "true", the event consumer will poll again:

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200608"
xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:Action>http://docs.oasis-open.org/ws-
rx/wsm/200608/MakeConnection</wsa:Action>
    <wsa:To>http://example.org/subscriptionService</wsa:To>
  </S:Header>
  <S:Body>
    <wsm:MakeConnection>
      <wsm:Address>http://docs.oasis-open.org/ws-
rx/wsm/200608/anonymous?id=550e8400-e29b-11d4-a716-
446655440000</wsm:Address>
    </wsm:MakeConnection>
  </S:Body>
</S:Envelope>
```

Notice this is the same message as the one sent in steps 2 and 5. As in steps 3 and 6, the response to the `MakeConnection` can be any message destined to the specified Endpoint. This allows the event producer to send not only application messages but RM protocol messages (e.g. `CloseSequence`, `TerminateSequence` or even additional `CreateSequences`) as needed.

**Step 8**— If at any point in time there are no messages pending, in response to a `MakeConnection` the event producer returns an HTTP-202 back to the event consumer. The process then repeats (back to step 7) until the subscription ends.



## Appendix D. State Tables

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

The state tables describe the lifetime of a sequence in both the RM Source and the RM Destination

Legend:

The first column of these tables contains the motivating event and has the following format:

Event
<i>Event name</i> [source] {ref}

Where:

- Event Name: indicates the name of the event. Event Names surrounded by "<>" are optional as described by the specification.
- [source]: indicates the source of the event; one of:
  - [msg] a Received message
  - [int]: an internal event such as the firing of a timer
  - [app]: the application
  - [unspec]: the source is unspecified

Each event / state combination cell in the tables in this appendix has the following format:

State Name
<i>Action to take</i> [next state] {ref}

Where:

- action to take: indicates that the state machine performs the following action. Actions surrounded by "<>" are optional as described by the specification. "Xmit" is used as a short form for the word "Transmit"
- [next state]: indicates the state to which the state machine will advance upon the performance of the action. For ease of reading the next state "same" indicates that the state does not change.
- {ref} is a reference to the document section describing the behavior in this cell

"N/A" in a cell indicates a state / event combination self-inconsistent with the state machine; should these conditions occur, it would indicate an implementation error. A blank cell indicates that the behavior is not described in this specification and does not indicate normal protocol operation. Implementations MAY generate a Sequence Terminated fault (see section 4.2) in these circumstances. Robust implementations MUST be able to operate in a stable manner despite the occurrence of unspecified event / state combinations.

2795 Table 1 RM Source Sequence State Transition Table

<b>Events</b>	<b>Sequence States</b>					
	<b>None</b>	<b>Creating</b>	<b>Created</b>	<b>Closing</b>	<b>Closed</b>	<b>Terminating</b>
<b>Create Sequence</b> [unspec] {3.4}	Xmit Create Sequence [Creating] {3.4}	N/A	N/A	N/A	N/A	N/A
<b>Create Sequence Response</b> [msg] {3.4}		Process Create Sequence Response [Created] {3.4}				
<b>Create Sequence Refused Fault</b> [msg] {3.4}		No action [None] {4.6}				
<b>Send message</b> [app] {2.1}	N/A	N/A	Xmit message [Same] {2}	No action [Same] {2}	N/A	N/A
<b>Retransmit of un-ack'd message</b> [int] {2.3}	N/A	N/A	Xmit message [Same] {2.3}	Xmit message [Same] {2.3}	N/A	N/A
<b>SeqAck (non-final)</b> [msg] {3.9}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Process Ack ranges [Same] {3.9}	Process Ack ranges [Same] {3.9}	Process Ack ranges [Same] {3.9}	Process Ack ranges [Same] {3.9}
<b>Nack</b> [msg] {3.9}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	<Xmit message (s)> [Same] {3.9}	<Xmit message (s)> [Same] {3.9}	No action [Same]	No action [Same]
<b>Message Number Rollover Fault</b> [msg] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	No action [Rollover]	No action [Same]	No action [Same]	No action [Same]
<b>CloseSequence</b> [msg] {3.5}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Xmit CloseSequence Response [Closed] {3.5}	Xmit CloseSequence Response [Closed] {3.5}	Xmit CloseSequence Response [Closed] {3.5}	Generate Unknown Sequence Fault [Same] {4.3}
<b>&lt;Close Sequence&gt;</b> [int] {3.5}	N/A		Xmit Close Sequence [Closing] {3.5}	N/A	N/A	N/A
<b>Close Sequence Response</b> [msg] {3.5}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}		No action [Closed] {3.5}	No action [Same] {3.5}	No action [Same] {3.5}

Events	Sequence States					
	None	Creating	Created	Closing	Closed	Terminating
<b>SeqAck (final)</b> [msg] {3.9}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Process Ack ranges [Closed] {3.9}	Process Ack ranges [Closed] {3.9}	Process Ack ranges [Same]	Process Ack ranges [Same]
<b>Sequence Closed Fault</b> [msg] {4.7}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	No action [Closed] {4.7}	No action [Closed] {4.7}	No action [Same]	No action [Same]
<b>Unknown Sequence Fault</b> [msg] {4.3}			Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}
<b>Sequence Terminated Fault</b> [msg] {4.2}	N/A		Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.2}
<b>TerminateSequence</b> [msg] {3.6}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Xmit Terminate Sequence Response [None] {3.6}	Xmit Terminate Sequence Response [None] {3.6}	Xmit Terminate Sequence Response [None] {3.6}	Generate Unknown Sequence Fault [Same] {4.3}
<b>Terminate Sequence</b> [int]	N/A	No action [None] {unspec}	Xmit Terminate Sequence [Terminating]	Xmit Terminate Sequence [Terminating]	Xmit Terminate Sequence [Terminating]	N/A
<b>Terminate Sequence Response</b> [msg]	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}				Terminate Sequence [None] {3.6}
<b>Expires exceeded</b> [int]	N/A	Terminate Sequence [None] {3.7}	Terminate Sequence [None] {3.7}	Terminate Sequence [None] {3.7}	Terminate Sequence [None] {3.7}	Terminate Sequence [None] {3.7}
<b>Invalid Acknowledgement</b> [msg] {4.4}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Unknown Sequence Fault [Same] {4.3}	Generate Invalid Acknowledgement Fault [Same] {4.4}	Generate Invalid Acknowledgement Fault [Same] {4.4}	Generate Invalid Acknowledgement Fault [Same] {4.4}	Generate Invalid Acknowledgement Fault [Same] {4.4}
Events	Sequence States					
	None	Creating	Created	Closing	Closed	Terminating
<b>Create Sequence</b> [unspec] {3.1}	Xmit Create Sequence [Creating] {3.1}	N/A	N/A	N/A	N/A	N/A

Events	Sequence States					
	None	Creating	Created	Closing	Closed	Terminating
<b>Create Sequence Response</b> {msg} {3-1}	-	Process-Create Sequence Response {Created} {3-1}	-	-	-	-
<b>Create Sequence Refused Fault</b> {msg} {3-1}	-	No-action {None} {4-6}	-	-	-	-
<b>Send message-</b> {app} {2-1}	N/A	N/A	Xmit-message {Same} {2}	No-action {Same} {2}	N/A	N/A
<b>Retransmit of un-ack'd message</b> {int}	N/A	N/A	Xmit-message {Same} {2-4}	Xmit-message {Same} {2-4}	N/A	N/A
<b>SeqAck (non-final)</b> {msg} {3-6}	Generate Unknown Sequence-Fault {Same} {4-3}	Generate Unknown Sequence-Fault {Same} {4-3}	Process-Ack ranges {Same} {3-6}	Process-Ack ranges {Same} {3-6}	Process-Ack ranges {Same} {3-6}	Process-Ack ranges {Same} {3-6}
<b>Nack</b> {msg} {3-6}	Generate Unknown Sequence-Fault {Same} {4-3}	Generate Unknown Sequence-Fault {Same} {4-3}	<Xmit-message (s)> {Same} {3-6}	<Xmit-message (s)> {Same} {3-6}	No-action {Same}	No-action {Same}
<b>Message Number Rollover Fault</b> {msg}	Generate Unknown Sequence-Fault {Same} {4-3}	Generate Unknown Sequence-Fault {Same} {4-3}	No-action {Rollover}	No-action {Same}	No-action {Same}	No-action {Same}
<b>&lt;Close Sequence&gt;</b> {int} {3-2}	N/A	-	Xmit-Close Sequence {Closing} {3-2}	N/A	N/A	N/A
<b>Close Sequence Response</b> {msg} {3-2}	Generate Unknown Sequence-Fault {Same} {4-3}	Generate Unknown Sequence-Fault {Same} {4-3}	-	No-action {Closed} {3-2}	No-action {Same} {3-2}	No-action {Same} {3-2}
<b>SeqAck (final)</b> {msg} {3-6}	Generate Unknown Sequence-Fault {Same} {4-3}	Generate Unknown Sequence-Fault {Same} {4-3}	Process-Ack ranges {Closed} {3-6}	Process-Ack ranges {Closed} {3-6}	Process-Ack ranges {Same}	Process-Ack ranges {Same}
<b>Sequence Closed Fault</b> {msg} {4-7}	Generate Unknown Sequence-Fault {Same} {4-3}	Generate Unknown Sequence-Fault {Same} {4-3}	No-action {Closed} {4-7}	No-action {Closed} {4-7}	No-action {Same}	No-action {Same}

Events	Sequence States					
	None	Creating	Created	Closing	Closed	Terminating
<b>Unknown Sequence Fault</b> {msg} {4.3}	-	-	Terminate Sequence {None} {4.3}	Terminate Sequence {None} {4.3}	Terminate Sequence {None} {4.3}	Terminate Sequence {None} {4.3}
<b>Sequence Terminated Fault</b> {msg} {4.2}	N/A	-	Terminate Sequence {None} {4.2}	Terminate Sequence {None} {4.2}	Terminate Sequence {None} {4.2}	Terminate Sequence {None} {4.2}
<b>Terminate Sequence</b> {int}	N/A	No-action {None} {unspee}-	Xmit Terminate Sequence {Terminating}	Xmit Terminate Sequence {Terminating}	Xmit Terminate Sequence {Terminating}	N/A
<b>Terminate Sequence Response</b> {msg}	Generate Unknown Sequence-Fault {Same} {4.3}	Generate Unknown Sequence-Fault {Same} {4.3}	-	-	-	Terminate Sequence {None} {3.3}
<b>Expires exceeded</b> {int}	N/A	Terminate Sequence {None} {3.4}	Terminate Sequence {None} {3.4}	Terminate Sequence {None} {3.4}	Terminate Sequence {None} {3.4}	Terminate Sequence {None} {3.4}
<b>Invalid Acknowledgement</b> {msg} {4.4}	Generate Unknown Sequence-Fault {Same} {4.3}	Generate Unknown Sequence-Fault {Same} {4.3}	Generate Invalid Acknowledgement Fault {Same} {4.4}	Generate Invalid Acknowledgement Fault {Same} {4.4}	Generate Invalid Acknowledgement Fault {Same} {4.4}	Generate Invalid Acknowledgement Fault {Same} {4.4}

2796 Table 2 RM Destination Sequence State Transition Table

Events	Sequence States			
	None	Created	Closed	Terminating
<b>CreateSequence (successful)</b> {msg/int} {3.4}	Xmit Create Sequence Response {Created} {3.4}	N/A	N/A	
<b>CreateSequence (unsuccessful)</b> {msg/int} {3.4}	Generate Create Sequence Refused Fault {None} {3.4}	N/A	N/A	
<b>Message (with message number within range)</b> {msg}	Generate Unknown Sequence Fault {Same} {4.3}	Accept Message: <Xmit SeqAck> {Same}	Generate Sequence Closed Fault (with SeqAck+Final) {Same} {3.5}	Generate Sequence Terminated Fault {Same} {4.2}
<b>Message (with message number outside of range)</b> {msg}	Generate Unknown Sequence Fault {Same} {4.3}	Xmit Message Number Rollover Fault {Same} {3.7}{4.5}	Generate Sequence Closed Fault (with SeqAck+Final) {Same} {3.5}	Generate Sequence Terminated Fault {Same} {4.2}
<b>&lt;AckRequested&gt;</b> {msg} {3.8}	Generate Unknown Seq Fault {Same} {4.3}	Xmit SeqAck {Same} {3.8}	Xmit SeqAck+Final {Same} {3.9}	Generate Sequence Terminated Fault {Same} {4.2}

Events	Sequence States			
	None	Created	Closed	Terminating
<b>CloseSequence</b> [msg] {3.5}	Generate Unknown Sequence Fault [Same] {4.3}	Xmit CloseSequence Response with SeqAck+Final [Closed] {3.5}	Xmit CloseSequence Response with SeqAck+Final [Closed] {3.5}	Generate Sequence Terminated Fault [Same] {4.2}
<b>&lt;CloseSequence autonomously&gt;</b> [int]		Xmit CloseSequence with SeqAck+Final [Closed] {3.5}	Xmit CloseSequence with SeqAck+Final [Same] {3.5}	
<b>CloseSequenceResponse</b> [msg] {3.5}	Generate Unknown Sequence Fault [Same] {4.3}		No Action [Closed] {3.5}	Generate Sequence Terminated Fault [Same] {4.2}
<b>TerminateSequence</b> [msg] {3.6}	Generate Unknown Sequence Fault [Same] {4.3}	Xmit Terminate Sequence Response [None] {3.6}	Xmit Terminate Sequence Response [None] {3.6}	Xmit Terminate Sequence Response [None] {3.6}
<b>&lt;TerminateSequence autonomously&gt;</b> [int]		Xmit TerminateSequence with SeqAck+Final [Terminating] {3.6}	Xmit TerminateSequence with SeqAck+Final [Terminating] {3.6}	Xmit TerminateSequence with SeqAck+Final [Terminating] {3.6}
<b>TerminateSequenceResponse</b> [msg]	Generate Unknown Sequence Fault [Same] {4.3}			Terminate Sequence [None]
<b>UnknownSequence Fault</b> [msg] {4.3}		Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}	Terminate Sequence [None] {4.3}
<b>SequenceTerminated Fault</b> [msg] {4.2}		Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.2}	Terminate Sequence [None] {4.3}
<b>Invalid Acknowledgement Fault</b> [msg] {4.4}	N/A			
<b>Expires exceeded</b> [int]	N/A	Terminate Sequence [None] {3.4}	Terminate Sequence [None] {3.4}	
<b>&lt;Seq Acknowledgement autonomously&gt;</b> [int] {3.9}	N/A	Xmit SeqAck [Same] {3.9}	Xmit SeqAck+Final [Same] {3.9}	
<b>Non WSRM message when WSRM required</b> [msg] {4.8}	Generate WSRMRequired Fault [Same] {4.8}	Generate WSRMRequired Fault [Same] {4.8}	Generate WSRMRequired Fault [Same] {4.8}	
Events	Sequence States			
	None	Created	Closed	
<b>CreateSequence- (successful)</b> [msg/int] {3.1}	Xmit Create-Sequence Response [Created] {3.1}	N/A	N/A	

Events	Sequence States		
	None	Created	Closed
<b>CreateSequence- (unsuccessful)</b> {msg/int} {3.1}	Generate-Create-Sequence Refused-Fault {None} {3.1}	N/A	N/A
<b>Message (with message number within range)</b> {msg}	Generate-Unknown-Sequence Fault {Same} {4.3}	Accept-Message; <Xmit-SeqAck> {Same}	Generate-Sequence-Closed Fault-(with-SeqAck+Final) {Same} {3.2}
<b>Message (with message number outside of range)</b> {msg}	Generate-Unknown-Sequence Fault {Same} {4.3}	Xmit-Message-Number Rollover-Fault {Same} {3.4}{4.5}	Generate-Sequence-Closed Fault-(with-SeqAck+Final) {Same} {3.2}
<b>&lt;AckRequested&gt;</b> {msg} {3.5}	Generate-Unknown-Seq-Fault {Same} {4.3}	Xmit-SeqAck {Same} {3.5}	Xmit-SeqAck+Final {Same} {3.6}
<b>CloseSequence</b> {msg} {3.2}	Generate-Unknown-Sequence Fault {Same} {4.3}	Xmit-CloseSequence Response-with-SeqAck+Final {Closed} {3.2}	Generate-Sequence-Closed Fault- {Same} {4.7}
<b>&lt;CloseSequence autonomously&gt;</b> {int}	N/A	No-Action {Closed}	N/A
<b>TerminateSequence</b> {msg} {3.3}	Generate-Unknown-Sequence Fault {Same} {4.3}	Xmit-Terminate-Sequence Response {None} {3.3}	Xmit-Terminate-Sequence Response {None} {3.3}
<b>UnknownSequence-Fault</b> {msg} {4.3}	-	Terminate-Sequence {None} {4.3}	Terminate-Sequence {None} {4.3}
<b>SequenceTerminated-Fault</b> {msg} {4.2}	-	Terminate-Sequence {None} {4.2}	Terminate-Sequence {None} {4.2}
<b>Invalid-Acknowledgement Fault</b> {msg} {4.4}	N/A	-	-
<b>Expires-exceeded</b> {int}	N/A	Terminate-Sequence {None} {3.4}	Terminate-Sequence {None} {3.4}
<b>&lt;Seq-Acknowledgement autonomously&gt;</b> {int} {3.6}	N/A	Xmit-SeqAck {Same} {3.6}	Xmit-SeqAck+Final {Same} {3.6}
<b>Non-WSRM-message-when WSRM-required</b> {msg} {4.8}	Generate-WSRMRequired Fault {Same} {4.8}	Generate-WSRMRequired Fault {Same} {4.8}	Generate-WSRMRequired Fault {Same} {4.8}



2797 The following two tables apply only if the ~~MakeConnection~~ mechanism is utilized.

2798 Table 3 Sending Endpoint Message Transfer Engine

Event	None	Queued n=1	Queued, n>1
Message destined to anon Endpoint when channel unavailable {int} {3.7}	Queue message {Queued n=1}	Queue message {Queued n>1}	Queue message {Queued n>1}
MakeConnection {msg} {3.7}	-	Send message {none}	Xmit message with MessagePending {if n=2 then (Queued n=1) else (Queued n>1)}

2799 Table 4 Receiving Endpoint Message Transfer Engine

Event	None	Polling
Expectation of unreceived message {int, unspecified}	No Action {Polling}	No Action {Same}
Polling trigger {int, unspecified}	-	Xmit MakeConnection {Polling} {3.7}

## Appendix 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-Editor(BM), Tom Freund(IBM), Mary Ann Hondo(IBM), John Ibbotson(IBM), Lei Jin(BEA), Chris Kaler(Microsoft), David Langworthy-Editor(Microsoft), 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), Rebecca Bergersen(Iona), 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), David Ingham(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:

Abbie Barbir(Nortel), Charlton Barreto(Adobe), Stefan Batres(Microsoft), Hamid Ben Malek(Fujitsu), Andreas Bjarlestam(Ericsson), Toufic Boubez(Layer 7), Doug Bunting(Sun), Lloyd Burch(Novell), Steve Carter(Novell), Martin Chapman(Oracle), Dave Chappell(Sonic), Paul Cotton(Microsoft), Glen Daniels(Sonic), Doug Davis(IBM), Blake Dournaee(Intel), Jacques Durand(Fujitsu), Colleen Evans(Microsoft), Christopher Ferris(IBM), Paul Fremantle(WSO2), Robert Freund(Hitachi), Peter Furniss(Erebor), Marc Goodner(Microsoft), Alastair Green(Choreology), Mike Grogan(Sun), Ondrej Hrebicek(Microsoft), Kazunori Iwasa(Fujitsu), Chamikara Jayalath(WSO2), Lei Jin(BEA), Ian Jones(BT plc), Anish Karmarkar(Oracle), Paul Knight(Nortel), Dan Leshchiner(Tibco), Mark Little(JBoss), Lily Liu(webMethods), Matt Lovett(IBM), Ashok Malhotra(Oracle), Jonathan Marsh(Microsoft), Daniel Millwood(IBM), Jeff Mischkinsky(Oracle), Nilo Mitra(Ericsson), Peter Niblett(IBM), Duane Nickull(Adobe), Eisaku Nishiyama(Hitachi), Dave Orchard(BEA), Chouthri Palanisamy(NEC), Sanjay Patil(SAP), Gilbert Pilz(BEA), Martin Raeppe(SAP), Eric Rajkovic(Oracle), Stefan Rossmanith(SAP), Tom Rutt(Fujitsu), Rich Salz(IBM), Shivajee Samdarshi(Tibco), Vladimir Videlov(SAP), Claus von Riegen(SAP), Pete Wenzel(Sun), Steve Winkler(SAP), Ümit Yalçınalp(SAP), Nobuyuki Yamamoto(Hitachi).

## Appendix F. Revision History

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

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

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

Rev	Date	By Whom	What
<a href="#">wd-11</a>	<a href="#">2006-03-08</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 100 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-03-20</a>	<a href="#">Doug Davis</a>	<a href="#">Added space in "SOAP1.x" – PaulCotton</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 007 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 090 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 098 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 099 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 101 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 103 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 104 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 105 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 107 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 109 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 110 applied</a>
<a href="#">wd-12</a>	<a href="#">2006-04-12</a>	<a href="#">Doug Davis</a>	<a href="#">Used "generated" instead of "issue" or "send" when talking about faults.</a>
<a href="#">wd-12</a>	<a href="#">2006-04-24</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Update references to WS-Addressing to the Proposed Recommendations; update WS-RM namespace to "200604".</a>
<a href="#">wd-13</a>	<a href="#">2006-05-08</a>	<a href="#">Gilbert Pilz</a>	<a href="#">i093 part 1; more work needed</a>
<a href="#">wd-13</a>	<a href="#">2006-05-10</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 096 applied</a>
<a href="#">wd-13</a>	<a href="#">2006-05-26</a>	<a href="#">Gilbert Pilz</a>	<a href="#">i093 part 2; reflects decisions from 2006-05-25 meeting</a>
<a href="#">wd-13</a>	<a href="#">2006-05-28</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Issue 106 applied</a>
<a href="#">wd-13</a>	<a href="#">2006-05-29</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Issue 118 applied</a>
<a href="#">wd-13</a>	<a href="#">2006-05-29</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Issue 120 applied</a>
<a href="#">wd-13</a>	<a href="#">2006-05-30</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Issue 114 applied</a>
<a href="#">wd-13</a>	<a href="#">2006-05-30</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Issue 116 applied</a>
<a href="#">wd-14</a>	<a href="#">2006-06-05</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Accept all changes; bump WD number</a>
<a href="#">wd-14</a>	<a href="#">2006-06-07</a>	<a href="#">Doug Davis</a>	<a href="#">Applied lots of minor edits from Marc Goodner</a>
<a href="#">wd-14</a>	<a href="#">2006-06-07</a>	<a href="#">Doug Davis</a>	<a href="#">Change a couple of period/sp/sp to period/sp</a>
<a href="#">wd-14</a>	<a href="#">2006-06-07</a>	<a href="#">Doug Davis</a>	<a href="#">Added a space in "URI]of" – per Marc Goodner</a>
<a href="#">wd-14</a>	<a href="#">2006-06-07</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 131 applied</a>
<a href="#">wd-14</a>	<a href="#">2006-06-07</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 132 applied</a>
<a href="#">wd-14</a>	<a href="#">2006-06-07</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 119 applied</a>
<a href="#">wd-14</a>	<a href="#">2006-06-07</a>	<a href="#">Doug Davis</a>	<a href="#">Applied lots of minor edits from Doug Davis</a>
<a href="#">wd-14</a>	<a href="#">2006-06-07</a>	<a href="#">Doug Davis</a>	<a href="#">s/"none"/"full-uri"/ - per Marc Goodner</a>
<a href="#">wd-14</a>	<a href="#">2006-06-12</a>	<a href="#">Doug Davis</a>	<a href="#">Complete i106</a>
<a href="#">wd-14</a>	<a href="#">2006-06-12</a>	<a href="#">Doug Davis</a>	<a href="#">Issues 089 applied</a>
<a href="#">wd-14</a>	<a href="#">2006-06-12</a>	<a href="#">Doug Davis</a>	<a href="#">Fix for several RFC2119 keywords – per Anish</a>
<a href="#">wd-15</a>	<a href="#">2006-06-12</a>	<a href="#">Doug Davis</a>	<a href="#">Accept all changed, dump WD number</a>
<a href="#">wd-15</a>	<a href="#">2006-06-12</a>	<a href="#">Doug Davis</a>	<a href="#">Move WSDL after Schema</a>

Rev	Date	By Whom	What
<a href="#">wd-15</a>	<a href="#">2006-06-12</a>	<a href="#">Doug Davis</a>	<a href="#">Nits – remove tabs, extra [yyy]'s ...</a>
<a href="#">wd-15</a>	<a href="#">2006-06-14</a>	<a href="#">Doug Davis</a>	<a href="#">Remove extra "OPTIONAL"s – Matt Lovett</a>
<a href="#">wd-15</a>	<a href="#">2006-06-14</a>	<a href="#">Doug Davis</a>	<a href="#">Remove blank rows/columns from state table. Fix italics in state table</a>
<a href="#">wd-15</a>	<a href="#">2006-06-15</a>	<a href="#">Doug Davis</a>	<a href="#">Typo – section D was empty</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 125 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 126 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 127 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 133 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 136 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 138 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-16</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 135 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-20</a>	<a href="#">Doug Davis</a>	<a href="#">Added all TC members to the ack list</a>
<a href="#">wd-15</a>	<a href="#">2006-06-22</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 129 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-22</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 130 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-22</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 137 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-26</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 111 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-26</a>	<a href="#">Doug Davis</a>	<a href="#">Missed a part of issue 129</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Fixed a typo in schema</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 141 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 142 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 148 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 149 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-06-30</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 150 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-06</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 121 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 139 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 144 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 147 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">Issues 122-124 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-27</a>	<a href="#">Doug Davis</a>	<a href="#">Updated list of oasis TC members (i134)</a>
<a href="#">wd-15</a>	<a href="#">2006-07-27</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 140 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-27</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 145 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-27</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 143 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-07-28</a>	<a href="#">Doug Davis</a>	<a href="#">Lots of minor typos found by Matt L.</a>
<a href="#">wd-15</a>	<a href="#">2006-07-28</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 113 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Update old namespaces – found by PaulC</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 150 applied</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Minor typos – found by PeterN</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Verify all [refs]</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Change namespace to 2006/08</a>
<a href="#">wd-15</a>	<a href="#">2006-08-04</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 148 applied</a>



Rev	Date	By Whom	What
<a href="#">wd-15</a>	<a href="#">2006-08-07</a>	<a href="#">Doug Davis</a>	<a href="#">Add some new glossary terms – per GilP</a>
<a href="#">cd-04</a>	<a href="#">2006-08-10</a>	<a href="#">Gilbert Pilz</a>	<a href="#">Formatting changes for better HTML rendering.</a>
<a href="#">cd-04</a>	<a href="#">2006-08-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 158 applied</a>
<a href="#">cd-04</a>	<a href="#">2006-08-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 153 applied</a>
<a href="#">cd-04</a>	<a href="#">2006-08-11</a>	<a href="#">Doug Davis</a>	<a href="#">Issue 156 applied</a>
<a href="#">cd-04</a>	<a href="#">2006-08-15</a>	<a href="#">Gilbert Pilz</a>	<a href="#">More formatting changes for better HTML rendering.</a>
<a href="#">wd-16</a>	<a href="#">2006-10-25</a>	<a href="#">Doug Davis</a>	<a href="#">Accept all changes, update to wd16</a>
<a href="#">wd-16</a>	<a href="#">2006-10-26</a>	<a href="#">Doug Davis</a>	<a href="#">PR002 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-10-26</a>	<a href="#">Doug Davis</a>	<a href="#">PR003 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-10-26</a>	<a href="#">Doug Davis</a>	<a href="#">PR004 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-10-27</a>	<a href="#">Doug Davis</a>	<a href="#">PR005 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-10-27</a>	<a href="#">Doug Davis</a>	<a href="#">PR006 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-10-27</a>	<a href="#">Doug Davis</a>	<a href="#">PR024 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-11-13</a>	<a href="#">Doug Davis</a>	<a href="#">PR010 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-11-13</a>	<a href="#">Doug Davis</a>	<a href="#">PR011 applied (technically as part of PR004)</a>
<a href="#">wd-16</a>	<a href="#">2006-11-13</a>	<a href="#">Doug Davis</a>	<a href="#">PR016 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-11-13</a>	<a href="#">Doug Davis</a>	<a href="#">PR032 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-11-20</a>	<a href="#">Doug Davis</a>	<a href="#">PR025 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-11-20</a>	<a href="#">Doug Davis</a>	<a href="#">PR023 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-12-03</a>	<a href="#">Doug Davis</a>	<a href="#">PR036 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-12-03</a>	<a href="#">Doug Davis</a>	<a href="#">PR017 applied</a>
<a href="#">wd-16</a>	<a href="#">2006-12-11</a>	<a href="#">Doug Davis</a>	<a href="#">PR012 applied (and PR013)</a>
<a href="#">wd-16</a>	<a href="#">2006-12-14</a>	<a href="#">Doug Davis</a>	<a href="#">PR033 applied – changed a 'return' to 'generate' when talking about a fault</a>
<a href="#">wd-16</a>	<a href="#">2007-01-04</a>	<a href="#">Doug Davis</a>	<a href="#">PR018 applied</a>
<a href="#">wd-16</a>	<a href="#">2007-01-05</a>	<a href="#">Doug Davis</a>	<a href="#">Moved MakeConnection to new spec</a>
<a href="#">wd-16</a>	<a href="#">2007-01-17</a>	<a href="#">Doug Davis</a>	<a href="#">PR026 applied</a>
<a href="#">wd-16</a>	<a href="#">2007-01-18</a>	<a href="#">Doug Davis</a>	<a href="#">PR021 applied</a>
<a href="#">wd-16</a>	<a href="#">2007-01-18</a>	<a href="#">Doug Davis</a>	<a href="#">PR022 applied</a>
<a href="#">wd-16</a>	<a href="#">2007-01-18</a>	<a href="#">Doug Davis</a>	<a href="#">Fixed a few typos (Doug,Gil)</a>
<a href="#">wd-16</a>	<a href="#">2007-01-18</a>	<a href="#">Gilbert Pilz</a>	<a href="#">PR007 applied</a>
<a href="#">wd-16</a>	<a href="#">2007-01-25</a>	<a href="#">Doug Davis</a>	<a href="#">PR039 applied</a>
<a href="#">wd-17</a>	<a href="#">2007-01-31</a>	<a href="#">Doug Davis</a>	<a href="#">Lots of typos from MarcG Updated WD number and date</a>
<a href="#">wd-17</a>	<a href="#">2007-02-01</a>	<a href="#">Doug Davis</a>	<a href="#">PR038 applied</a>
<a href="#">wd-17</a>	<a href="#">2007-02-01</a>	<a href="#">Doug Davis</a>	<a href="#">PR035 (009.020 dups) applied Fixed typo in state table</a>
Rev	Date	By Whom	What
<a href="#">wd-01</a>	<a href="#">2005-07-07</a>	<a href="#">Christopher Ferris</a>	<a href="#">Initial version created based on submission by the authors.–</a>
<a href="#">ws-02</a>	<a href="#">2005-07-21</a>	<a href="#">Doug Davis</a>	<a href="#">I011 (PT0S) added</a>

Rev	Date	By Whom	What
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 'optional'/'required' to 'OPTIONAL'/'REQUIRED') added
wd-05	2005-09-30	Anish Karmarkar	i017 (Change NS to <a href="http://docs.oasis-open.org/wsrn/200510/">http://docs.oasis-open.org/wsrn/200510/</a> )
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
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

Rev	Date	By Whom	What
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 TOG. 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 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.

Rev	Date	By Whom	What
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 RDDL doc; added non-normative reference to RDDL 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
wd-12	2006-03-20	Doug Davis	Added space in "SOAP1.x"—Paul Cotton
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

Rev	Date	By Whom	What
wd-12	2006-04-11	Doug Davis	Issue 103 applied
wd-12	2006-04-11	Doug Davis	Issue 104 applied
wd-12	2006-04-11	Doug Davis	Issue 105 applied
wd-12	2006-04-11	Doug Davis	Issue 107 applied
wd-12	2006-04-11	Doug Davis	Issue 109 applied
wd-12	2006-04-11	Doug Davis	Issue 110 applied
wd-12	2006-04-12	Doug Davis	Used "generated" instead of "issue" or "send" when talking about faults.
wd-12	2006-04-24	Gilbert Pilz	Update references to WS-Addressing to the Proposed Recommendations; update WS-RM namespace to "200604".
wd-13	2006-05-08	Gilbert Pilz	i093-part 1; more work needed
wd-13	2006-05-10	Doug Davis	Issue 096 applied
wd-13	2006-05-26	Gilbert Pilz	i093-part 2; reflects decisions from 2006-05-25 meeting
wd-13	2006-05-28	Gilbert Pilz	Issue 106 applied
wd-13	2006-05-29	Gilbert Pilz	Issue 118 applied
wd-13	2006-05-29	Gilbert Pilz	Issue 120 applied
wd-13	2006-05-30	Gilbert Pilz	Issue 114 applied
wd-13	2006-05-30	Gilbert Pilz	Issue 116 applied
wd-14	2006-06-05	Gilbert Pilz	Accept all changes; bump WD number
wd-14	2006-06-07	Doug Davis	Applied lots of minor edits from Marc Goodner
wd-14	2006-06-07	Doug Davis	Change a couple of period/sp/sp to period/sp
wd-14	2006-06-07	Doug Davis	Added a space in "URI}of"—per Marc Goodner
wd-14	2006-06-07	Doug Davis	Issue 131 applied
wd-14	2006-06-07	Doug Davis	Issue 132 applied
wd-14	2006-06-07	Doug Davis	Issue 119 applied
wd-14	2006-06-07	Doug Davis	Applied lots of minor edits from Doug Davis
wd-14	2006-06-07	Doug Davis	s/"none"/"full-uri"/—per Marc Goodner
wd-14	2006-06-12	Doug Davis	Complete i106
wd-14	2006-06-12	Doug Davis	Issues 089 applied
wd-14	2006-06-12	Doug Davis	Fix for several RFC2119 keywords—per Anish
wd-15	2006-06-12	Doug Davis	Accept all changed; dump WD number
wd-15	2006-06-12	Doug Davis	Move WSDL after Schema
wd-15	2006-06-12	Doug Davis	Nits—remove tabs, extra [yyy]'s ...
wd-15	2006-06-14	Doug Davis	Remove extra "OPTIONAL"s—Matt Lovett
wd-15	2006-06-14	Doug Davis	Remove blank rows/columns from state table. Fix italics in state table
wd-15	2006-06-15	Doug Davis	Type—section D was empty
wd-15	2006-06-16	Doug Davis	Issue 125 applied
wd-15	2006-06-16	Doug Davis	Issue 126 applied

Rev	Date	By Whom	What
wd-15	2006-06-16	Doug Davis	Issue 127 applied
wd-15	2006-06-16	Doug Davis	Issue 133 applied
wd-15	2006-06-16	Doug Davis	Issue 136 applied
wd-15	2006-06-16	Doug Davis	Issue 138 applied
wd-15	2006-06-16	Doug Davis	Issue 135 applied
wd-15	2006-06-20	Doug Davis	Added all TC members to the ack list
wd-15	2006-06-22	Doug Davis	Issue 129 applied
wd-15	2006-06-22	Doug Davis	Issue 130 applied
wd-15	2006-06-22	Doug Davis	Issue 137 applied
wd-15	2006-06-26	Doug Davis	Issue 111 applied
wd-15	2006-06-26	Doug Davis	Missed a part of issue 129
wd-15	2006-06-30	Doug Davis	Fixed a typo in schema
wd-15	2006-06-30	Doug Davis	Issue 141 applied
wd-15	2006-06-30	Doug Davis	Issue 142 applied
wd-15	2006-06-30	Doug Davis	Issue 148 applied
wd-15	2006-06-30	Doug Davis	Issue 149 applied
wd-15	2006-06-30	Doug Davis	Issue 150 applied
wd-15	2006-07-06	Doug Davis	Issue 121 applied
wd-15	2006-07-21	Doug Davis	Issue 139 applied
wd-15	2006-07-21	Doug Davis	Issue 144 applied
wd-15	2006-07-21	Doug Davis	Issue 147 applied
wd-15	2006-07-21	Doug Davis	Issues 122-124 applied
wd-15	2006-07-27	Doug Davis	Updated list of oasis TC members (i134)
wd-15	2006-07-27	Doug Davis	Issue 140 applied
wd-15	2006-07-27	Doug Davis	Issue 145 applied
wd-15	2006-07-27	Doug Davis	Issue 143 applied
wd-15	2006-07-28	Doug Davis	Lots of minor typos found by Matt L.
wd-15	2006-07-28	Doug Davis	Issue 113 applied
wd-15	2006-08-04	Doug Davis	Update old namespaces — found by PaulG
wd-15	2006-08-04	Doug Davis	Issue 150 applied
wd-15	2006-08-04	Doug Davis	Minor typos — found by PeterN
wd-15	2006-08-04	Doug Davis	Verify all [refs]
wd-15	2006-08-04	Doug Davis	Change namespace to 2006/08
wd-15	2006-08-04	Doug Davis	Issue 148 applied
wd-15	2006-08-07	Doug Davis	Add some new glossary terms — per GilP
cd-04	2006-08-10	Gilbert Pilz	Formatting changes for better HTML rendering.
cd-04	2006-08-11	Doug Davis	Issue 158 applied
cd-04	2006-08-11	Doug Davis	Issue 153 applied
cd-04	2006-08-11	Doug Davis	Issue 156 applied
cd-04	2006-08-15	Gilbert Pilz	More formatting changes for better HTML rendering.





## Appendix 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 (2007~~6~~). 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.