



# 1 Web Services Reliable Messaging 2 (WS-Reliable-Messaging)

3 Working Draft 1~~3~~2, May 30, 2006

## 4 Document identifier:

5 wsrn-1.1-spec-wd-1~~3~~2

## 6 Location:

## 7 Editors:

8 Doug Davis, IBM <dug@us.ibm.com>  
9 Anish Karmarkar, Oracle <Anish.Karmarkar@oracle.com>  
10 Gilbert Pilz, BEA <gpilz@bea.com>  
11 Steve Winkler, SAP <steve.winkler@sap.com>  
12 Ümit Yalçınalp, SAP <umit.yalcinalp@sap.com>

## 13 Contributors:

14 TBD

## 15 Abstract:

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

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

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

## 30 Status:

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

## Table of Contents

33	1 Introduction.....	4
34	1.1 Goals and Requirements.....	4
35	1.1.1 Requirements.....	4
36	1.2 Notational Conventions.....	4
37	1.3 Namespace.....	4
38	1.4 Compliance.....	5
39	2 Reliable Messaging Model.....	6
40	2.1 Glossary.....	6
41	2.2 Protocol Preconditions.....	7
42	2.3 Protocol Invariants.....	7
43	2.4 Example Message Exchange.....	7
44	3 RM Protocol Elements.....	10
45	3.1 Sequence Creation.....	10
46	3.2 Closing A Sequence.....	13
47	3.3 Sequence Termination.....	15
48	3.4 Sequences.....	16
49	3.5 Request Acknowledgement.....	17
50	3.6 Sequence Acknowledgement.....	18
51	4 Faults.....	21
52	4.1 SequenceFault Element.....	22
53	4.2 Sequence Terminated.....	23
54	4.3 Unknown Sequence.....	23
55	4.4 Invalid Acknowledgement.....	24
56	4.5 Message Number Rollover.....	24
57	4.6 Create Sequence Refused.....	24
58	4.7 Sequence Closed.....	25
59	4.8 WSRM Required.....	25
60	5 Security Considerations.....	26
61	6 References.....	28
62	6.1 Normative.....	28
63	6.2 Non-Normative.....	28
64	A. Schema.....	30
65	B. Message Examples.....	35
66	B.1 Create Sequence.....	35
67	B.2 Initial Transmission.....	35
68	B.3 First Acknowledgement.....	37
69	B.4 Retransmission.....	37

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

# 1 Introduction

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

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

## 1.1 Goals and Requirements

### 1.1.1 Requirements

## 1.2 Notational Conventions

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

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

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

## 1.3 Namespace

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

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

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

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

114 The following namespaces are used in this document:

115 *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>
wsm	<a href="http://docs.oasis-open.org/ws-rx/wsm/200604">http://docs.oasis-open.org/ws-rx/wsm/200604</a>
wsa	<a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

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

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

118 All sections explicitly noted as examples are informational and are not to be considered normative.~~If an~~  
119 ~~action IRI is used, and one is not already defined per the rules of the WS-Addressing specification [WS-~~  
120 ~~Addressing], then the action IRI MUST consist of the WS-RM namespace URI concatenated with a '/',~~  
121 ~~followed by the message element name. For example:~~

122 ~~<http://docs.oasis-open.org/ws-rx/wsm/200604/SequenceAcknowledgement>~~

## 123 1.4 Compliance

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

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

## 2 Reliable Messaging Model

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

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

The protocol supports reliability features that enable ordered delivery, duplicate elimination, and guaranteed receipt for the RMD. It is expected that the AD and RMD will implement as many of these or as few of these characteristics as necessary to implement the AD. Regardless of which of the reliability features are employed, the wire protocol does not change.

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

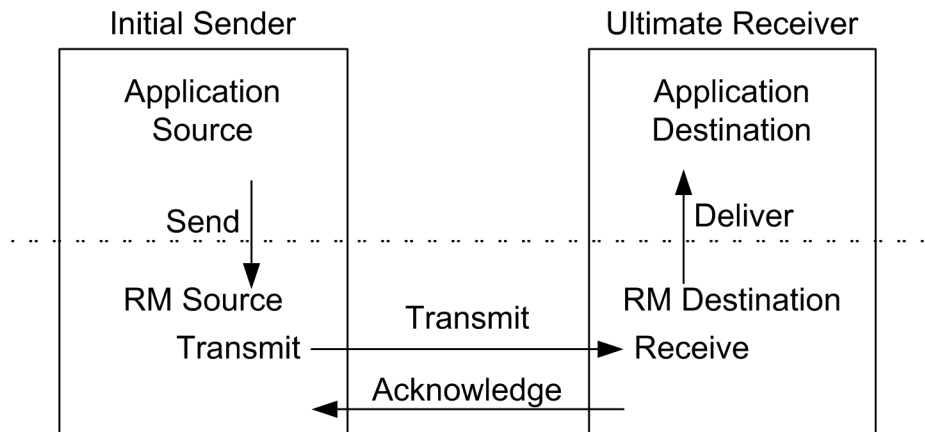


Figure 1: Reliable Messaging Model

### 2.1 Glossary

The following definitions are used throughout this specification:

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

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

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

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

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

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

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

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

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

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

130 **2.2 Protocol Preconditions**

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

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

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

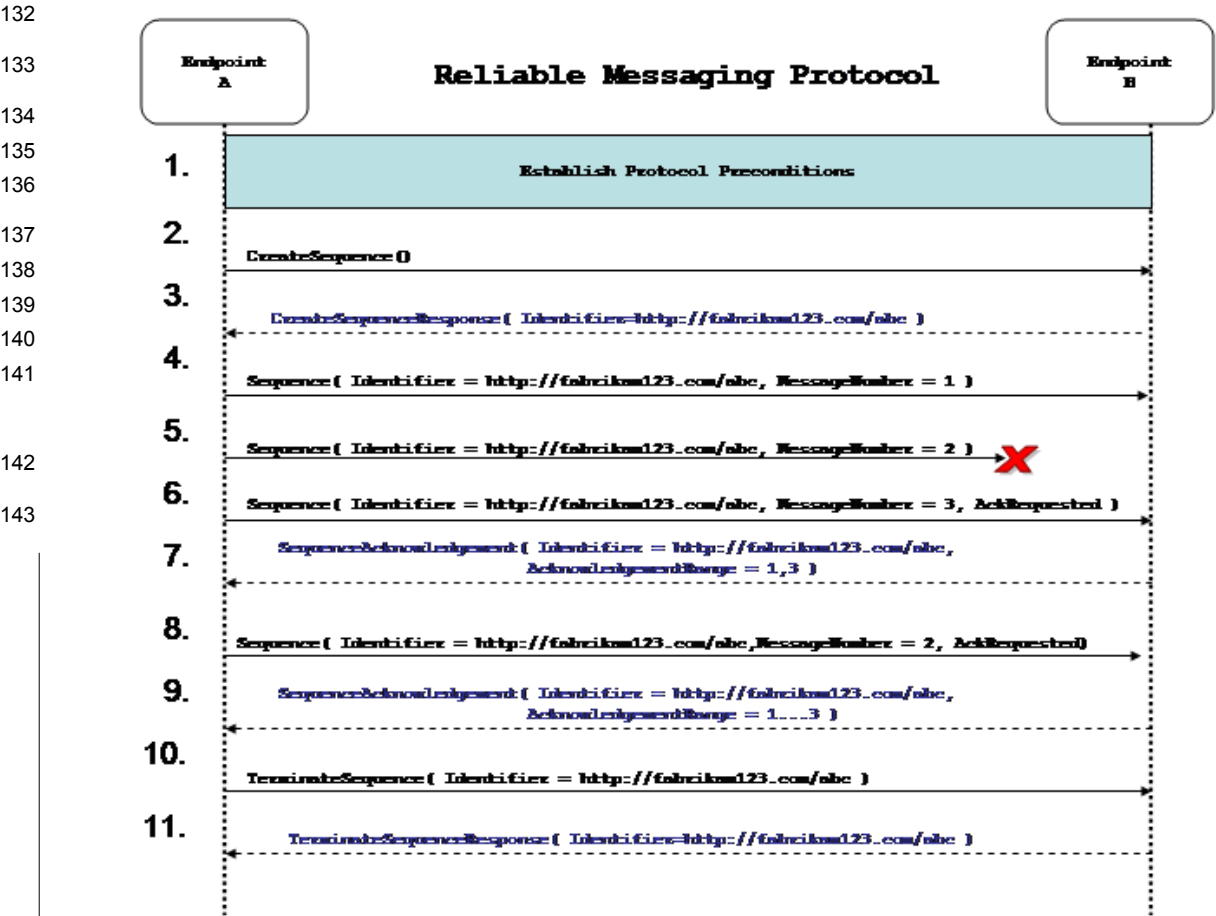


Figure 2: The WS-ReliableMessaging Protocol

## Figure 2: The WS-ReliableMessaging Protocol

1. The protocol preconditions are established. These include policy exchange, endpoint resolution, and establishing trust.
2. The RM Source requests creation of a new Sequence.
3. The RM Destination creates a new Sequence and returns its globally unique identifier.
4. The RM Source begins transmittingsending messages in the Sequence beginning with MessageNumber 1. In the figure above, the RM Source sends 3 messages in the Sequence.
5. The 2<sup>nd</sup> message in the Sequence is lost in transit.
6. The 3<sup>rd</sup> message is the last in this Sequence and the RM Source includes a  
<wsrm:AckRequested> header to ensure that it gets a timely  
<wsrm:SequenceAcknowledgement> for the Sequence. Since the 3rd message is the last in this  
exchange, the RM Source includes a <wsrm:AckRequested> Header.
7. The RM Destination acknowledges receipt of message numbers 1 and 3 as a result of receiving the  
RM Source's <wsrm:AckRequested> header. The 2nd message is lost in transit.
8. The RM Source retransmits the unacknowledged message with MessageNumber 2. This is a new  
message from the perspective of the underlying transport, but it has the same Sequence Identifier  
and MessageNumber so the RM Destination can recognize it as a duplicate of the earlier message.  
in case the original and retransmitted messages are both received. The RM Source includes an  
<wsrm:AckRequested> header in the retransmitted message so the RM Destination will expedite  
an acknowledgement. The RM Destination acknowledges receipt of message numbers 1 and 3 as a  
result of receiving the RM Source's <wsrm:AckRequested> Header.
9. The RM Source retransmits the 2nd message. This is a new message on the underlying transport,  
but it has the same Sequence identifier and message number so the RM Destination can recognize  
it as equivalent to the earlier message, in case both are received.
10. The RM Source includes an <wsrm:AckRequested> element so the RM Destination will expedite  
an acknowledgement.
11. The RM Destination receives the second transmission of the message with MessageNumber 2 and acknowledges receipt of message numbers 1, 2, and 3.
12. The RM Source receives this acknowledgement and sends a TerminateSequence message to the RM Destination indicating that the Sequence is completed and reclaims any resources associated with the Sequence.
13. The RM Destination receives the TerminateSequence message indicating that the RM Source will not be sending any more messages. The RM Destination sends a TerminateSequenceResponse message to the RM Source and reclaims any resources associated with the Sequence.

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

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



## 3 RM Protocol Elements

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

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

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

### 3.1 Sequence Creation

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

The SOAP version used for the CreateSequence message SHOULD be used for all subsequent messages in or for that Sequence, sent by either the RMS or the RMD.-

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

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

/wsrm:CreateSequence

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

/wsrm:CreateSequence/wsrm:AcksTo

The RM Source MUST include this element in any CreateSequence message it sends. This element is This REQUIRED element, of type wsa:EndpointReferenceType (as specified by WS-Addressing [WS-Addressing]). It specifies the endpoint reference to which messages containing all-

<wsrm:SequenceAcknowledgement> header block messages and faults related to the created Sequence are to be sent, unless otherwise noted in this specification (for example, see Section 3.2).

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

237 /wsrm:CreateSequence/wsrm:Expires

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

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

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

245 /wsrm:CreateSequence/wsrm:Offer

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

248 /wsrm:CreateSequence/wsrm:Offer/wsrm:Identifier

249 ~~The RM Source MUST set the value of this element to~~~~This REQUIRED element MUST contain~~ an  
250 absolute URI ~~(conformant with RFC3986 [URI][URI])~~ that ~~will~~ uniquely identify~~ies~~ the offered Sequence.

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

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

254 /wsrm:CreateSequence/wsrm:Offer/wsrm:Endpoint

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

259 /wsrm:CreateSequence/wsrm:Offer/wsrm:Expires

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

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

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

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

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

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

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

272 /wsrm:CreateSequence/{any}

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

275 /wsrm:CreateSequence/@{any}

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

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

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

```
283 <wsrm:CreateSequenceResponse ...>
284   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
285   <wsrm:Expires> xs:duration </wsrm:Expires> ?
286   <wsrm:AcknowledgementInterval Milliseconds="xs:unsignedLong" ... /> ?
287   <wsrm:IncompleteSequenceBehavior> wsrm:IncompleteSequenceBehaviorType
288   </wsrm:IncompleteSequenceBehavior> ?
289   <wsrm:Accept ...>
290     <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
291     ...
292   </wsrm:Accept> ?
293   ...
294 </wsrm:CreateSequenceResponse>
```

295 /wsrm:CreateSequenceResponse

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

299 /wsrm:CreateSequenceResponse/wsrm:Identifier

300 The RM Destination MUST include this element within any CreateSequenceResponse message it sends.  
301 The RM Destination MUST set the value of this element to the ~~This REQUIRED element MUST contain an~~  
302 absolute URI (conformant with RFC3986 [URI]) of the Sequence that has been created by the RM  
303 Destination.

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

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

307 /wsrm:CreateSequenceResponse/wsrm:Expires

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

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

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

316 /wsrm:CreateSequenceResponse/wsrm:AcknowledgementInterval

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

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

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

324 /wsrm:CreateSequenceResponse/wsrm:IncompleteSequenceBehavior  
 325 This optional element, if present, specifies the behavior that the RM Destination will exhibit upon the  
 326 closure of an incomplete sequence.

327 A value of "DiscardEntireSequence" indicates that the entire sequence will be discarded by the RM  
 328 Destination if the sequence is closed when there are one or more gaps in the  
 329 SequenceAcknowledgement/Final.

330 A value of "DiscardFollowingFirstGap" indicates that messages in the sequence beyond the first gap will  
 331 be discarded by the RM Destination when there are one or more gaps in the  
 332 SequenceAcknowledgement/Final.

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

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

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

341 /wsrm:CreateSequenceResponse/wsrm:Accept/wsrm:AcksTo  
 342 The RM Destination MUST include this element~~This REQUIRED element~~, of type  
 343 wsa:EndpointReferenceType (as specified by WS-Addressing [**WS-Addressing**]), The RM Source  
 344 SHOULD send~~specifies the endpoint reference to which messages with~~  
 345 <wsrm:SequenceAcknowledgement> header block~~messages~~ related to the accepted Sequence to  
 346 the referenced endpoint~~are to be sent~~.

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

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

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

356 /wsrm:CreateSequenceResponse/@{any}

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

## 359 3.2 Closing A Sequence

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

365 If the RM Source wishes to close the Sequence, then it sends a `<wsrm:CloseSequence>` element, in  
366 the body of a message, to the RM Destination. This message indicates that the RM Destination MUST  
367 NOT receive any new messages for the specified Sequence, other than those already received at the time  
368 the `<wsrm:CloseSequence>` element is interpreted by the RMD. Upon receipt of this message, or  
369 subsequent to the RM Destination closing the Sequence of its own volition, the RM Destination MUST  
370 include a final `<wsrm:SequenceAcknowledgement>` (within which the RM Destination that MUST  
371 include the `<wsrm:Final>` element) header block on any messages associated with the Sequence  
372 destined to the RM Source, including the CloseSequenceResponse message or on any Sequence Fault  
373 transmitted to the RMS.-

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

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

382 The following exemplar defines the CloseSequence syntax:

```
383 <wsrm:CloseSequence ...>  
384   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
385   ...  
386 </wsrm:CloseSequence>
```

387 /wsrm:CloseSequence

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

391 /wsrm:CloseSequence/wsrm:Identifier

392 The RM Source MUST include this element in any CloseSequence messages it sends. The RM Source  
393 MUST set the value of this element to the~~This-REQUIRED-element MUST contain an~~ absolute URI  
394 ~~(conformant with RFC3986 [URI])~~ of the Sequence that is being closed.-

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

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

398 /wsrm:CloseSequence/{any}

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

401 /wsrm:CloseSequence@{any}

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

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

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

```
408 <wsrm:CloseSequenceResponse ...>  
409   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
410   ...  
411 </wsrm:CloseSequenceResponse>
```

412 /wsrm:CloseSequenceResponse

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

416 /wsrm:CloseSequenceResponse/wsrm:Identifier

417 The RM Destination MUST include this element in any CloseSequenceResponse message it sends. The  
418 RM Destination MUST set the value of this element to the~~This REQUIRED element MUST contain an~~  
419 absolute URI (conformant with RFC3986 [URI]) of the Sequence that is being closedterminated.-

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

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

423 /wsrm:CloseSequenceResponse/{any}

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

426 /wsrm:CloseSequenceResponse@{any}

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

### 429 3.3 Sequence Termination

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

437 The following exemplar defines the TerminateSequence syntax:



```

438 <wsrm:TerminateSequence ...>
439   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
440   ...
441 </wsrm:TerminateSequence>

```

442 /wsrm:TerminateSequence

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

448 /wsrm:TerminateSequence/wsrm:Identifier

449 The RM Source MUST include this element in any TerminateSequence message it sends. The RM  
450 Source MUST set the value of this element to the This REQUIRED element ~~MUST contain an~~ absolute  
451 URI (conformant with RFC3986 [URI]) of the Sequence that is being terminated.-

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

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

455 /wsrm:TerminateSequence/{any}

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

458 /wsrm:TerminateSequence/@{any}

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

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

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

```

465 <wsrm:TerminateSequenceResponse ...>
466   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
467   ...
468 </wsrm:TerminateSequenceResponse>

```

469 /wsrm:TerminateSequenceResponse

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

474 /wsrm:TerminateSequenceResponse/wsrm:Identifier

475 The RMD Destination MUST include this element in any TerminateSequenceResponse message it sends.  
476 The RMD Destination MUST set the value of this element to the This REQUIRED element ~~MUST contain-~~  
477 ~~an~~ absolute URI (conformant with RFC3986 [URI]) of the Sequence that is being terminated.

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

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

481 /wsrm:TerminateSequenceResponse/{any}

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

484 /wsrm:TerminateSequenceResponse/@{any}

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

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

### 489 3.4 Sequences

490 The RM protocol uses a <wsrm:Sequence> header block to track and manage the reliable delivery of  
491 messages. The RM Source MUST include a <wsrm:Sequence> header block in all  
492 messages Messages for which a reliable delivery is required ~~MUST contain a <wsrm:Sequence>~~  
493 ~~header block. The RM Source MUST identify Sequences with Each Sequence MUST have a~~  
494 unique <wsrm:Identifier> elements and the RM Source MUST assign each message within a  
495 Sequence ~~MUST have a~~ <wsrm:MessageNumber> element that increments by 1 from an initial value of  
496 1. These values are contained within a <wsrm:Sequence> header block accompanying each message  
497 being delivered in the context of a Sequence.

498 The RM Source MUST NOT include ~~There MUST be no~~ more than one <wsrm:Sequence> header block  
499 in any message.

500 A following exemplar defines its syntax:

```
501 <wsrm:Sequence ...>  
502   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
503   <wsrm:MessageNumber> wsrm:MessageNumberType </wsrm:MessageNumber>  
504   ...  
505 </wsrm:Sequence>
```

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

507 /wsrm:Sequence

508 This protocol element associates the message in which it is contained with a previously established RM  
509 Sequence. It contains the Sequence's unique identifier and the containing message's ordinal position  
510 within that Sequence. The RM Destination MUST understand the ~~The~~ <wsrm:Sequence> header  
511 ~~block~~ element MUST be understood by the RM Destination. The RM Source MUST assign ~~The~~  
512 ~~<wsrm:Sequence> element MUST have~~ a mustUnderstand attribute with a value 1/true (from the  
513 namespace corresponding to the version of SOAP to which the <wsrm:Sequence> SOAP header block  
514 is bound) to the <wsrm:Sequence> header block element.

515 /wsrm:Sequence/wsrm:Identifier

516 An RM Source that includes a <wsrm:Sequence> header block in a SOAP envelope MUST include  
517 this element in that header block. The RM Source MUST set the value of this element to the ~~This~~  
518 ~~REQUIRED element MUST contain an~~ absolute URI (conformant with RFC3986 [URI]) that uniquely  
519 identifies the Sequence.-

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



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

523 /wsrm:Sequence/wsrm:MessageNumber

524 ~~The RM Source MUST include this element within any Sequence headers it creates. This REQUIRED-~~  
525 ~~element is of type~~wsrm:MessageNumberType. ~~It represents~~representing the ordinal  
526 position of the message within a Sequence. Sequence ~~message numbers~~MessageNumbers start at 1 and  
527 monotonically increase throughout the Sequence. If the message number exceeds the internal limitations  
528 of an RM Source or RM Destination or reaches the maximum value of 9,223,372,036,854,775,807 the RM  
529 Source or Destination MUST generate a MessageNumberRollover fault.

530 /wsrm:Sequence/{any}

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

533 /wsrm:Sequence/@{any}

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

536 The following example illustrates a Sequence header block.

```
537 <wsrm:Sequence>  
538   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>  
539   <wsrm:MessageNumber>10</wsrm:MessageNumber>  
540 </wsrm:Sequence>
```

## 541 3.5 Request Acknowledgement

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

544 The RM Source MAY request an acknowledgement message from the RM Destination at any time by  
545 including an <wsrm:AckRequested> header block in any message targeted to the RM Destination. An  
546 RM Destination that receives a message that contains an <wsrm:AckRequested> header block MUST  
547 send a message containing a <wsrm:SequenceAcknowledgement> header block to the wsrm:AcksTo  
548 endpoint reference (see Section 3.1). If a non-mustUnderstand fault occurs when processing an RM  
549 Header that was piggy-backed on another message, a fault MUST be generated, but the processing of  
550 the original message MUST NOT be affected. It is RECOMMENDED that the RMD return a  
551 <wsrm:AcknowledgementRange> or <wsrm:None> element instead of a <wsrm:Nack> element (see  
552 below).

553 The following exemplar defines its syntax:

```
554 <wsrm:AckRequested ...>  
555   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>  
  
556   ...  
557 </wsrm:AckRequested>
```

558 /wsrm:AckRequested

559 This element requests an acknowledgement for the identified Sequence.

560 /wsrm:AckRequested/wsrm:Identifier

561 ~~An RM Source that includes a <wsrm:AckRequested> header block in a SOAP envelope MUST include~~  
562 ~~this element in that header block. The RM Source MUST set the value of this element to the~~This-

563 ~~REQUIRED element MUST contain an~~ absolute URI, (conformant with RFC3986 [\[URI\]](#)), that uniquely  
564 identifies the Sequence to which the request applies.-

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

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

568 /wsrm:AckRequested/{any}

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

571 /wsrm:AckRequested/@{any}

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

### 574 3.6 Sequence Acknowledgement

575 The RM Destination informs the RM Source of successful message receipt using a  
576 <wsrm:SequenceAcknowledgement> header block. ~~The RM Destination MAY transmit the~~  
577 <wsrm:SequenceAcknowledgement> header block ~~MAY be transmitted~~ independently or ~~it MAY~~  
578 ~~include the <wsrm:SequenceAcknowledgement> header block~~ included on any message targeted to  
579 the AcksTo EPR. The RM Destination MAY send a <wsrm:SequenceAcknowledgement> header block  
580 at any point during which the Sequence is valid. Acknowledgements can be explicitly requested using the  
581 <wsrm:AckRequested> directive (see Section [Request Acknowledgement](#)). If a non-mustUnderstand  
582 fault occurs when processing an RM Header that was piggy-backed on another message, a fault MUST  
583 be generated, but the processing of the original message MUST NOT be affected.

584 A RMD MAY include a wsrm:SequenceAcknowledgement header block on any SOAP envelope targetted  
585 to the endpoint referenced by the wsrm:AcksTo EPR. This concept is often referred to as "piggy-backing"  
586 Sequence acknowledgements.

587 ~~During creation of a Sequence the RM Source~~ A wsrm:AcksTo EPR MAY specify the WS-Addressing  
588 anonymous IRI as ~~the address of the <wsrm:AcksTo> EPR for that Sequence~~ its address. When the ~~RM~~  
589 ~~Source~~ wsrm:AcksTo EPR specifies the WS-Addressing anonymous IRI as ~~the address of the~~  
590 ~~<wsrm:AcksTo> EPR~~ its address, the RM Destination MUST transmit any  
591 ~~<wsrm:SequenceAcknowledgement>~~ headers for the created Sequence in a SOAP envelope to be  
592 transmitted on the protocol binding-specific channel. Such a channel is provided by the context of a  
593 received message containing a SOAP envelope that contains a ~~<wsrm:Sequence>~~ header block and/or  
594 a ~~<wsrm:AckRequested>~~ header block for that same Sequence identifier.

595 The following exemplar defines its syntax:

```
596 <wsrm:SequenceAcknowledgement ...>
597   <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
598   [ [ [ <wsrm:AcknowledgementRange ...
599     Upper="wsrm:MessageNumberType"
600     Lower="wsrm:MessageNumberType"/> +
601
602     | <wsrm:None/> ]
603     <wsrm:Final/> ? ]
604   | <wsrm:Nack> wsrm:MessageNumberType </wsrm:Nack> + ]
605   ...
606 </wsrm:SequenceAcknowledgement>
```

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

608 `/wsrm:SequenceAcknowledgement`

609 This element contains the Sequence acknowledgement information.

610 `/wsrm:SequenceAcknowledgement/wsrm:Identifier`

611 An RM Destination that includes a `<wsrm:SequenceAcknowledgement>` header block in a SOAP  
612 envelope MUST include this REQUIRED element in that header block. The RM Destination MUST set  
613 the value of this element to the~~an~~contain absolute URI (conformant with RFC3986 [URI]) that uniquely  
614 identifies the Sequence. The RM Destination A message MUST NOT ~~include~~contain multiple  
615 `<wsrm:SequenceAcknowledgement>` header blocks that share the same value for  
616 `<wsrm:Identifier>` within the same SOAP envelope.-

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

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

620 `/wsrm:SequenceAcknowledgement/wsrm:AcknowledgementRange`

621 The RM Destination MAY include one or more instances of this element within a  
622 `<wsrm:SequenceAcknowledgement>` header block.~~This OPTIONAL element, if present, can occur 1~~  
623 ~~or more times.~~ It contains a range of Sequence MessageNumbers successfully received by the RM  
624 Destination. The ranges SHOULD NOT overlap. The RM Destination MUST NOT include t~~This element~~  
625 ~~MUST NOT be present~~ if a sibling `<wsrm:Nack>` or `<wsrm:None>` element is also present as a child of  
626 `<wsrm:SequenceAcknowledgement>`.

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

628 The RM Destination MUST set the value of this attribute~~This REQUIRED attribute tainson a~~  
629 ~~`wsrm:MessageNumberType` equal to the message number~~representing the `<wsrm:MessageNumber>`  
630 of the highest contiguous message in a Sequence range received by the RM Destination.

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

632 The RM Destination MUST set the value of this attribute equal to the message number~~This REQUIRED~~  
633 ~~attribute contains a `wsrm:MessageNumberType` representing the `<wsrm:MessageNumber>`~~ of the  
634 lowest contiguous message in a Sequence range received by the RM Destination.

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

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

638 `/wsrm:SequenceAcknowledgement/wsrm:Final`

639 The RM Destination MAY include this element within a `<wsrm:SequenceAcknowledgement>` header  
640 block. This ~~OPTIONAL~~ element, ~~if present~~, indicates that the RM Destination is not receiving new  
641 messages for the specified Sequence. The RM Source can be assured that the ranges of messages  
642 acknowledged by this SequenceAcknowledgement header block will not change in the future. The RM  
643 Destination~~This element~~ MUST include this element~~be present~~ when the Sequence is closed. Note: the  
644 RM Destination~~this element~~ MUST NOT include this element~~be used~~ when sending a Nack; it can only be  
645 used when sending `<wsrm:AcknowledgementRange>`~~AcknowledgementRange~~s or `<wsrm:None>`.

646 `/wsrm:SequenceAcknowledgement/wsrm:Nack`

647 ~~The RM Destination MAY include this element within a <wsrm:SequenceAcknowledgement> header~~  
648 ~~block. If used, the RM Destination MUST set the value of this OPTIONAL element, if present, MUST~~  
649 ~~contain to~~ a wsrm:MessageNumberType representing the <wsrm:MessageNumber> of an unreceived  
650 message in a Sequence. ~~The RM Destination MUST NOT include a~~ The <wsrm:Nack> element ~~MUST~~  
651 ~~NOT be present~~ if a sibling <wsrm:AcknowledgementRange> or <wsrm:None> element is also  
652 present as a child of <wsrm:SequenceAcknowledgement>. Upon the receipt of a Nack, an RM Source  
653 SHOULD retransmit the message identified by the Nack. The RM Destination MUST NOT issue a  
654 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that it has previously  
655 acknowledged within a <wsrm:AcknowledgementRange>. The RM Source SHOULD ignore a  
656 <wsrm:SequenceAcknowledgement> containing a <wsrm:Nack> for a message that has previously  
657 been acknowledged within a <wsrm:AcknowledgementRange>.

658 /wsrm:SequenceAcknowledgement/wsrm:None

659 ~~The RM Destination MUST include this element within a <wsrm:SequenceAcknowledgement> header~~  
660 ~~block if this OPTIONAL element, if present, MUST be used when~~ the RM Destination has not received  
661 any messages for the specified Sequence. The ~~RM Destination~~ <wsrm:None> element MUST NOT  
662 ~~include this element~~ be present if a sibling <wsrm:AcknowledgementRange> or <wsrm:Nack> element  
663 is also present as a child of the <wsrm:SequenceAcknowledgement>.

664 /wsrm:SequenceAcknowledgement/{any}

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

667 /wsrm:SequenceAcknowledgement/@{any}

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

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

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

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

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

```
678 <wsrm:SequenceAcknowledgement>
679   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
680   <wsrm:AcknowledgementRange Upper="2" Lower="1"/>
681   <wsrm:AcknowledgementRange Upper="6" Lower="4"/>
682   <wsrm:AcknowledgementRange Upper="10" Lower="8"/>
683 </wsrm:SequenceAcknowledgement>
```

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

```
685 <wsrm:SequenceAcknowledgement>
686   <wsrm:Identifier>http://example.com/abc</wsrm:Identifier>
687   <wsrm:Nack>3</wsrm:Nack>
688 </wsrm:SequenceAcknowledgement>
```

## 4 Faults

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

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

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

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

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

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 MUST be either "Sender" or "Receiver". These properties are serialized into text XML as follows:

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

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

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

```

729     <S:Code>
730       <S:Value> [Code] </S:Value>
731       <S:Subcode>
732         <S:Value> [Subcode] </S:Value>
733       </S:Subcode>
734     </S:Code>
735     <S:Reason>
736       <S:Text xml:lang="en"> [Reason] </S:Text>
737     </S:Reason>
738     <S:Detail>
739       [Detail]
740       ...
741     </S:Detail>
742   </S:Fault>
743 </S:Body>
744 </S:Envelope>

```

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

```

747 <S11:Envelope>
748   <S11:Header>
749     <wsrm:SequenceFault>
750       <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
751       <wsrm:Detail> [Detail] </wsrm:Detail>
752       ...
753     </wsrm:SequenceFault>
754     <!-- Headers elided for clarity. -->
755   </S11:Header>
756   <S11:Body>
757     <S11:Fault>
758       <faultcode> [Code] </faultcode>
759       <faultstring> [Reason] </faultstring>
760     </S11:Fault>
761   </S11:Body>
762 </S11:Envelope>

```

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

```

765 <S11:Envelope>
766   <S11:Body>
767     <S11:Fault>
768       <faultcode> [Subcode] </faultcode>
769       <faultstring> [Reason] </faultstring>
770     </S11:Fault>
771   </S11:Body>
772 </S11:Envelope>

```

## 773 4.1 SequenceFault Element

774 The purpose of the <wsrm:SequenceFault> element is to carry the specific details of a fault generated  
775 during the reliable messaging specific processing of a message belonging to a Sequence. WS-  
776 ReliableMessaging nodes MUST use the <wsrm:SequenceFault> container MUST only be used-  
777 only in conjunction with the SOAP 1.1 fault mechanism. -WS-ReliableMessaging nodes# MUST NOT use  
778 the <wsrm:SequenceFault> containerbe-used in conjunction with the SOAP 1.2 binding.

779 The following exemplar defines its syntax:

```

780 <wsrm:SequenceFault ...>

```

```

781 <wsrm:FaultCode> wsrm:FaultCodes </wsrm:FaultCode>
782 <wsrm:Detail> ... </wsrm:Detail> ?
783 ...
784 </wsrm:SequenceFault>

```

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

786 `/wsrm:SequenceFault`

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

788 `/wsrm:SequenceFault/wsrm:FaultCode`

789 ~~WS-ReliableMessaging nodes that generate a <wsrm:SequenceFault> MUST set the value of this~~  
790 ~~element, if present, MUST contain to~~ a qualified name from the set of fault [Subcodes] defined below.-

791 `/wsrm:SequenceFault/wsrm:Detail`

792 This optional element is intended for carrying application specific error information related to the fault  
793 being described.

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

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

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

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

798 `/wsrm:SequenceFault/{any}`

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

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

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

## 804 4.2 Sequence Terminated

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

809 Receipt of SequenceTerminated by either the RMD or the RMS shall terminate the Sequence if it is not  
810 otherwise terminated.

811 Properties:

812 [Code] Sender or Receiver

813 [Subcode] `wsrm:SequenceTerminated`

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

815 [Detail]

```

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

```



### 4.3 Unknown Sequence

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

Properties:

[Code] Sender

[Subcode] wsrn:UnknownSequence

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

[Detail]

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

### 4.4 Invalid Acknowledgement

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

[Code] Sender

[Subcode] wsrn:InvalidAcknowledgement

[Reason] The SequenceAcknowledgement violates the cumulative acknowledgement invariant.

[Detail]

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

### 4.5 Message Number Rollover

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

Properties:

[Code] Sender

[Subcode] wsrn:MessageNumberRollover

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

[Detail]

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

### 4.6 Create Sequence Refused

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

Properties:

[Code] Sender

[Subcode] wsrn:CreateSequenceRefused

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



851 [Detail]

852 `xs:any`

## 853 **4.7 Sequence Closed**

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

855 This fault MUST be generated when an RM Destination is asked to receive a message for a Sequence  
856 that is closed.

857 Properties:

858 [Code] Sender

859 [Subcode] wsrn:SequenceClosed

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

861 [Detail]

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

## 863 **4.8 WSRM Required**

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

866 Properties:

867 [Code] Sender

868 [Subcode] wsrn:WSRMRequired

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

870 [Detail]

871 `xs:any`

## 5 Security Considerations

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

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

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

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

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

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

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

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

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

## 6 References

### 6.1 Normative

#### [KEYWORDS]

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

#### [SOAP 1.1]

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

#### [SOAP 1.2]

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

#### [URI]

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

#### [XML]

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

#### [XML-ns]

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

#### [XML-Schema Part1]

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

#### [XML-Schema Part2]

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

#### [WSDL 1.1]

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

#### [WS-Addressing]

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

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

### 6.2 Non-Normative

#### [RDDL 2.0]

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

#### [WS-Policy]

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

#### [WS-PolicyAttachment]

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

#### [WS-Security]

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

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

965 **[RTTM]**

966 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May  
967 1992.

968 **[SecurityPolicy]**

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

970 **[SecureConversation]**

971 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," February  
972 2005.

973 **[Trust]**

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

## 975 **A. Schema**

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

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

979 The following copy is provided for reference.

```

980 <?xml version="1.0" encoding="UTF-8"?>
981 <!--
982 OASIS takes no position regarding the validity or scope of any intellectual
983 property or other rights that might be claimed to pertain to the
984 implementation or use of the technology described in this document or the
985 extent to which any license under such rights might or might not be available;
986 neither does it represent that it has made any effort to identify any such
987 rights. Information on OASIS's procedures with respect to rights in OASIS
988 specifications can be found at the OASIS website. Copies of claims of rights
989 made available for publication and any assurances of licenses to be made
990 available, or the result of an attempt made to obtain a general license or
991 permission for the use of such proprietary rights by implementors or users of
992 this specification, can be obtained from the OASIS Executive Director.
993 OASIS invites any interested party to bring to its attention any copyrights,
994 patents or patent applications, or other proprietary rights which may cover
995 technology that may be required to implement this specification. Please
996 address the information to the OASIS Executive Director.
997 Copyright © OASIS Open 2002-2006. All Rights Reserved.
998 This document and translations of it may be copied and furnished to others,
999 and derivative works that comment on or otherwise explain it or assist in its
1000 implementation may be prepared, copied, published and distributed, in whole or
1001 in part, without restriction of any kind, provided that the above copyright
1002 notice and this paragraph are included on all such copies and derivative
1003 works. However, this document itself does not be modified in any way, such as
1004 by removing the copyright notice or references to OASIS, except as needed for
1005 the purpose of developing OASIS specifications, in which case the procedures
1006 for copyrights defined in the OASIS Intellectual Property Rights document must
1007 be followed, or as required to translate it into languages other than English.
1008 The limited permissions granted above are perpetual and will not be revoked by
1009 OASIS or its successors or assigns.
1010 This document and the information contained herein is provided on an "AS
1011 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
1012 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
1013 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
1014 FITNESS FOR A PARTICULAR PURPOSE.
1015 -->
1016 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1017 xmlns:wsa="http://www.w3.org/2005/08/addressing"
1018 xmlns:wsm="http://docs.oasis-open.org/ws-rx/wsm/200604"
1019 targetNamespace="http://docs.oasis-open.org/ws-rx/wsm/200604"
1020 elementFormDefault="qualified" attributeFormDefault="unqualified">
1021   <xs:import namespace="http://www.w3.org/2005/08/addressing"
1022   schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
1023   <!-- Protocol Elements -->
1024   <xs:complexType name="SequenceType">
1025     <xs:sequence>
1026       <xs:element ref="wsm:Identifier"/>
1027       <xs:element name="MessageNumber" type="wsm:MessageNumberType"/>
1028       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1029 maxOccurs="unbounded"/>
1030     </xs:sequence>
1031     <xs:anyAttribute namespace="##other" processContents="lax"/>
1032   </xs:complexType>
1033   <xs:element name="Sequence" type="wsm:SequenceType"/>
1034   <xs:element name="SequenceAcknowledgement">
1035     <xs:complexType>
1036       <xs:sequence>
1037         <xs:element ref="wsm:Identifier"/>
1038         <xs:choice>
1039           <xs:sequence>
1040             <xs:choice>
1041               <xs:element name="AcknowledgementRange" maxOccurs="unbounded">
1042                 <xs:complexType>

```

```

1043         <xs:sequence/>
1044         <xs:attribute name="Upper" type="xs:unsignedLong"
1045 use="required"/>
1046         <xs:attribute name="Lower" type="xs:unsignedLong"
1047 use="required"/>
1048         <xs:anyAttribute namespace="##other" processContents="lax"/>
1049     </xs:complexType>
1050 </xs:element>
1051     <xs:element name="None" minOccurs="0">
1052         <xs:complexType>
1053             <xs:sequence/>
1054         </xs:complexType>
1055     </xs:element>
1056 </xs:choice>
1057     <xs:element name="Final" minOccurs="0">
1058         <xs:complexType>
1059             <xs:sequence/>
1060         </xs:complexType>
1061     </xs:element>
1062 </xs:sequence>
1063     <xs:element name="Nack" type="xs:unsignedLong"
1064 maxOccurs="unbounded"/>
1065 </xs:choice>
1066     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1067 maxOccurs="unbounded"/>
1068 </xs:sequence>
1069     <xs:anyAttribute namespace="##other" processContents="lax"/>
1070 </xs:complexType>
1071 </xs:element>
1072 <xs:complexType name="AckRequestedType">
1073     <xs:sequence>
1074         <xs:element ref="wsrm:Identifier"/>
1075         <xs:any namespace="##other" processContents="lax" minOccurs="0"
1076 maxOccurs="unbounded"/>
1077     </xs:sequence>
1078     <xs:anyAttribute namespace="##other" processContents="lax"/>
1079 </xs:complexType>
1080 <xs:element name="AckRequested" type="wsrm:AckRequestedType"/>
1081 <xs:element name="Identifier">
1082     <xs:complexType>
1083         <xs:annotation>
1084             <xs:documentation>
1085                 This type is for elements whose [children] is an anyURI and can have
1086 arbitrary attributes.
1087             </xs:documentation>
1088         </xs:annotation>
1089         <xs:simpleContent>
1090             <xs:extension base="xs:anyURI">
1091                 <xs:anyAttribute namespace="##other" processContents="lax"/>
1092             </xs:extension>
1093         </xs:simpleContent>
1094     </xs:complexType>
1095 </xs:element>
1096 <xs:simpleType name="MessageNumberType">
1097     <xs:restriction base="xs:unsignedLong">
1098         <xs:minInclusive value="1"/>
1099         <xs:maxInclusive value="9223372036854775807"/>
1100     </xs:restriction>
1101 </xs:simpleType>
1102 <!-- Fault Container and Codes -->
1103 <xs:simpleType name="FaultCodes">
1104     <xs:restriction base="xs:QName">
1105         <xs:enumeration value="wsrm:SequenceTerminated"/>

```



```

1106     <xs:enumeration value="wsrm:UnknownSequence"/>
1107     <xs:enumeration value="wsrm:InvalidAcknowledgement"/>
1108     <xs:enumeration value="wsrm:MessageNumberRollover"/>
1109     <xs:enumeration value="wsrm:CreateSequenceRefused"/>
1110     <xs:enumeration value="wsrm:SequenceClosed"/>
1111     <xs:enumeration value="wsrm:WSRMRequired"/>
1112   </xs:restriction>
1113 </xs:simpleType>
1114 <xs:complexType name="SequenceFaultType">
1115   <xs:sequence>
1116     <xs:element name="FaultCode" type="wsrm:FaultCodes"/>
1117     <xs:element name="Detail" type="wsrm:DetailType" minOccurs="0"/>
1118     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1119 maxOccurs="unbounded"/>
1120   </xs:sequence>
1121   <xs:anyAttribute namespace="##other" processContents="lax"/>
1122 </xs:complexType>
1123 <xs:complexType name="DetailType">
1124   <xs:sequence>
1125     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1126 maxOccurs="unbounded"/>
1127   </xs:sequence>
1128   <xs:anyAttribute namespace="##other" processContents="lax"/>
1129 </xs:complexType>
1130 <xs:element name="SequenceFault" type="wsrm:SequenceFaultType"/>
1131 <xs:element name="CreateSequence" type="wsrm:CreateSequenceType"/>
1132 <xs:element name="CreateSequenceResponse"
1133 type="wsrm:CreateSequenceResponseType"/>
1134 <xs:element name="CloseSequence" type="wsrm:CloseSequenceType"/>
1135 <xs:element name="CloseSequenceResponse"
1136 type="wsrm:CloseSequenceResponseType"/>
1137 <xs:element name="TerminateSequence" type="wsrm:TerminateSequenceType"/>
1138 <xs:element name="TerminateSequenceResponse"
1139 type="wsrm:TerminateSequenceResponseType"/>
1140 <xs:complexType name="CreateSequenceType">
1141   <xs:sequence>
1142     <xs:element ref="wsrm:AcksTo"/>
1143     <xs:element ref="wsrm:Expires" minOccurs="0"/>
1144     <xs:element name="Offer" type="wsrm:OfferType" minOccurs="0"/>
1145     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1146 maxOccurs="unbounded">
1147       <xs:annotation>
1148         <xs:documentation>
1149           It is the authors intent that this extensibility be used to
1150 transfer a Security Token Reference as defined in WS-Security.
1151         </xs:documentation>
1152       </xs:annotation>
1153     </xs:any>
1154   </xs:sequence>
1155   <xs:anyAttribute namespace="##other" processContents="lax"/>
1156 </xs:complexType>
1157 <xs:complexType name="CreateSequenceResponseType">
1158   <xs:sequence>
1159     <xs:element ref="wsrm:Identifier"/>
1160     <xs:element ref="wsrm:Expires" minOccurs="0"/>
1161     <xs:element ref="wsrm:AcknowledgementInterval" minOccurs="0"/>

```

```

1162     <xs:element name="IncompleteSequenceBehaviour"
1163     type="wsrm:IncompleteSequenceBehaviorType" minOccurs="0"/>
1164     <xs:element name="Accept" type="wsrm:AcceptType" minOccurs="0"/>
1165     <xs:any namespace="##other" processContents="lax" minOccurs="0"
1166     maxOccurs="unbounded"/>
1167     </xs:sequence>
1168     <xs:anyAttribute namespace="##other" processContents="lax"/>
1169   </xs:complexType>
1170   <xs:complexType name="CloseSequenceType">
1171     <xs:sequence>
1172       <xs:element ref="wsrm:Identifier"/>
1173       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1174       maxOccurs="unbounded"/>
1175     </xs:sequence>
1176     <xs:anyAttribute namespace="##other" processContents="lax"/>
1177   </xs:complexType>
1178   <xs:complexType name="CloseSequenceResponseType">
1179     <xs:sequence>
1180       <xs:element ref="wsrm:Identifier"/>
1181       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1182       maxOccurs="unbounded"/>
1183     </xs:sequence>
1184     <xs:anyAttribute namespace="##other" processContents="lax"/>
1185   </xs:complexType>
1186   <xs:complexType name="TerminateSequenceType">
1187     <xs:sequence>
1188       <xs:element ref="wsrm:Identifier"/>
1189       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1190       maxOccurs="unbounded"/>
1191     </xs:sequence>
1192     <xs:anyAttribute namespace="##other" processContents="lax"/>
1193   </xs:complexType>
1194   <xs:complexType name="TerminateSequenceResponseType">
1195     <xs:sequence>
1196       <xs:element ref="wsrm:Identifier"/>
1197       <xs:any namespace="##other" processContents="lax" minOccurs="0"
1198       maxOccurs="unbounded"/>
1199     </xs:sequence>
1200     <xs:anyAttribute namespace="##other" processContents="lax"/>
1201   </xs:complexType>
1202   <xs:element name="AcksTo"

```

```

1203     type="wsa:EndpointReferenceType"/>
1204     <xs:complexType name="OfferType">
1205         <xs:sequence>
1206             <xs:element ref="wsrm:Identifier"/>
1207             <xs:element ref="wsrm:Expires" minOccurs="0"/>
1208             <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
1209             <xs:any namespace="##other" processContents="lax" minOccurs="0"
1210 maxOccurs="unbounded"/>
1211         </xs:sequence>
1212         <xs:anyAttribute namespace="##other" processContents="lax"/>
1213     </xs:complexType>
1214     <xs:complexType name="AcceptType">
1215         <xs:sequence>
1216             <xs:element ref="wsrm:AcksTo"/>
1217             <xs:any namespace="##other" processContents="lax" minOccurs="0"
1218 maxOccurs="unbounded"/>
1219         </xs:sequence>
1220         <xs:anyAttribute namespace="##other" processContents="lax"/>
1221     </xs:complexType>
1222     <xs:element name="Expires">
1223         <xs:complexType>
1224             <xs:simpleContent>
1225                 <xs:extension base="xs:duration">
1226                     <xs:anyAttribute namespace="##other" processContents="lax"/>
1227                 </xs:extension>
1228             </xs:simpleContent>
1229         </xs:complexType>
1230     </xs:element>
1231     <xs:element name="AcknowledgementInterval">
1232         <xs:complexType>
1233             <xs:sequence/>
1234             <xs:attribute name="Milliseconds" type="xs:unsignedLong"
1235 use="required"/>
1236             <xs:anyAttribute namespace="##other" processContents="lax"/>
1237         </xs:complexType>
1238     </xs:element>
1239     <xs:simpleType name="IncompleteSequenceBehaviorType">
1240         <xs:restriction base="xs:string">
1241             <xs:enumeration value="DiscardEntireSequence"/>
1242             <xs:enumeration value="DiscardFollowingFirstGap"/>
1243             <xs:enumeration value="NoDiscard"/>
1244         </xs:restriction>
1245     </xs:simpleType>
1246 </xs:schema>

```

## B. Message Examples

### B.1 Create Sequence

#### Create Sequence

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

#### Create Sequence Response

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

### B.2 Initial Transmission

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

## 1297 Message 1

```
1298 <?xml version="1.0" encoding="UTF-8"?>
1299 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1300 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
1301 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1302   <S:Header>
1303     <wsa:MessageID>
1304       http://Business456.com/guid/71e0654e-5ce8-477b-bb9d-34f05cfc9e
1305     </wsa:MessageID>
1306     <wsa:To>http://example.com/serviceB/123</wsa:To>
1307     <wsa:From>
1308       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1309     </wsa:From>
1310     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1311     <wsmr:Sequence>
1312       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1313       <wsmr:MessageNumber>1</wsmr:MessageNumber>
1314     </wsmr:Sequence>
1315   </S:Header>
1316   <S:Body>
1317     <!-- Some Application Data -->
1318   </S:Body>
1319 </S:Envelope>
```

## 1320 Message 2

```
1321 <?xml version="1.0" encoding="UTF-8"?>
1322 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1323 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
1324 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1325   <S:Header>
1326     <wsa:MessageID>
1327       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1328     </wsa:MessageID>
1329     <wsa:To>http://example.com/serviceB/123</wsa:To>
1330     <wsa:From>
1331       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1332     </wsa:From>
1333     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1334     <wsmr:Sequence>
1335       <wsmr:Identifier>http://Business456.com/RM/ABC</wsmr:Identifier>
1336       <wsmr:MessageNumber>2</wsmr:MessageNumber>
1337     </wsmr:Sequence>
1338   </S:Header>
1339   <S:Body>
1340     <!-- Some Application Data -->
1341   </S:Body>
1342 </S:Envelope>
```

## 1343 Message 3

```
1344 <?xml version="1.0" encoding="UTF-8"?>
1345 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1346 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200604"
1347 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1348   <S:Header>
1349     <wsa:MessageID>
1350       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546819
1351     </wsa:MessageID>
1352     <wsa:To>http://example.com/serviceB/123</wsa:To>
1353     <wsa:From>
1354       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
```

```

1355     </wsa:From>
1356     <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1357     <wsrm:Sequence>
1358       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1359       <wsrm:MessageNumber>3</wsrm:MessageNumber>
1360     </wsrm:Sequence>
1361     <wsrm:AckRequested>
1362       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1363     </wsrm:AckRequested>
1364   </S:Header>
1365   <S:Body>
1366     <!-- Some Application Data -->
1367   </S:Body>
1368 </S:Envelope>

```

### 1369 B.3 First Acknowledgement

1370 Message number 2 has not been received by the RM Destination due to some transmission error so it  
 1371 responds with an acknowledgement for messages 1 and 3:

```

1372 <?xml version="1.0" encoding="UTF-8"?>
1373 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1374   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
1375   xmlns:wsa="http://www.w3.org/2005/08/addressing">
1376   <S:Header>
1377     <wsa:MessageID>
1378       http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546810
1379     </wsa:MessageID>
1380     <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1381     <wsa:From>
1382       <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1383     </wsa:From>
1384     <wsa:Action>
1385       http://docs.oasis-open.org/ws-rx/wsrn/200604/SequenceAcknowledgement
1386     </wsa:Action>
1387     <wsrm:SequenceAcknowledgement>
1388       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1389       <wsrm:AcknowledgementRange Upper="1" Lower="1"/>
1390       <wsrm:AcknowledgementRange Upper="3" Lower="3"/>
1391     </wsrm:SequenceAcknowledgement>
1392   </S:Header>
1393   <S:Body/>
1394 </S:Envelope>

```

### 1395 B.4 Retransmission

1396 The RM Sourcediscovers that message number 2 was not received so it resends the message and  
 1397 requests an acknowledgement:

```

1398 <?xml version="1.0" encoding="UTF-8"?>
1399 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1400   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
1401   xmlns:wsa="http://www.w3.org/2005/08/addressing">
1402   <S:Header>
1403     <wsa:MessageID>
1404       http://Business456.com/guid/daa7d0b2-c8e0-476e-a9a4-d164154e38de
1405     </wsa:MessageID>
1406     <wsa:To>http://example.com/serviceB/123</wsa:To>
1407     <wsa:From>
1408       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1409     </wsa:From>

```

```

1410 <wsa:Action>http://example.com/serviceB/123/request</wsa:Action>
1411 <wsrm:Sequence>
1412 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1413 <wsrm:MessageNumber>2</wsrm:MessageNumber>
1414 </wsrm:Sequence>
1415 <wsrm:AckRequested>
1416 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1417 </wsrm:AckRequested>
1418 </S:Header>
1419 <S:Body>
1420 <!-- Some Application Data -->
1421 </S:Body>
1422 </S:Envelope>

```

## 1423 B.5 Termination

1424 The RM Destination now responds with an acknowledgement for the complete Sequence which can then  
 1425 be terminated:

```

1426 <?xml version="1.0" encoding="UTF-8"?>
1427 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1428 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
1429 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1430 <S:Header>
1431 <wsa:MessageID>
1432 http://example.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546811
1433 </wsa:MessageID>
1434 <wsa:To>http://Business456.com/serviceA/789</wsa:To>
1435 <wsa:From>
1436 <wsa:Address>http://example.com/serviceB/123</wsa:Address>
1437 </wsa:From>
1438 <wsa:Action>
1439 http://docs.oasis-open.org/ws-rx/wsr/200604/SequenceAcknowledgement
1440 </wsa:Action>
1441 <wsrm:SequenceAcknowledgement>
1442 <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1443 <wsrm:AcknowledgementRange Upper="3" Lower="1"/>
1444 </wsrm:SequenceAcknowledgement>
1445 </S:Header>
1446 <S:Body/>
1447 </S:Envelope>

```

## 1448 Terminate Sequence

```

1449 <?xml version="1.0" encoding="UTF-8"?>
1450 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1451 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsr/200604"
1452 xmlns:wsa="http://www.w3.org/2005/08/addressing">
1453 <S:Header>
1454 <wsa:MessageID>
1455 http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1456 </wsa:MessageID>
1457 <wsa:To>http://example.com/serviceB/123</wsa:To>
1458 <wsa:Action>
1459 http://docs.oasis-open.org/ws-rx/wsr/200604/TerminateSequence
1460 </wsa:Action>
1461 <wsa:From>
1462 <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1463 </wsa:From>
1464 </S:Header>
1465 <S:Body>
1466 <wsrm:TerminateSequence>

```

```
1467     <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1468   </wsrm:TerminateSequence>
1469 </S:Body>
1470 </S:Envelope>
```

#### 1471 Terminate Sequence Response

```
1472 <?xml version="1.0" encoding="UTF-8"?>
1473 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
1474   xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrn/200604"
1475   xmlns:wsa="http://www.w3.org/2005/08/addressing">
1476   <S:Header>
1477     <wsa:MessageID>
1478       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546813
1479     </wsa:MessageID>
1480     <wsa:To>http://example.com/serviceA/789</wsa:To>
1481     <wsa:Action>
1482       http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequenceResponse
1483     </wsa:Action>
1484     <wsa:RelatesTo>
1485       http://Business456.com/guid/0baaf88d-483b-4ecf-a6d8-a7c2eb546812
1486     </wsa:RelatesTo>
1487     <wsa:From>
1488       <wsa:Address>http://Business456.com/serviceA/789</wsa:Address>
1489     </wsa:From>
1490   </S:Header>
1491   <S:Body>
1492     <wsrm:TerminateSequenceResponse>
1493       <wsrm:Identifier>http://Business456.com/RM/ABC</wsrm:Identifier>
1494     </wsrm:TerminateSequenceResponse>
1495   </S:Body>
1496 </S:Envelope>
```



## 1497 C. WSDL

1498 The non-normative WSDL 1.1 definition for WS-ReliableMessaging is located at:

1499 <http://docs.oasis-open.org/ws-rx/wsrn/200604/wSDL/wsrn-1.1-wsdl-200604.wsd>

1500 The following non-normative copy is provided for reference.

```

1501 <?xml version="1.0" encoding="utf-8"?>
1502 <!--
1503 OASIS takes no position regarding the validity or scope of any intellectual
1504 property or other rights that might be claimed to pertain to the
1505 implementation or use of the technology described in this document or the
1506 extent to which any license under such rights might or might not be available;
1507 neither does it represent that it has made any effort to identify any such
1508 rights. Information on OASIS's procedures with respect to rights in OASIS
1509 specifications can be found at the OASIS website. Copies of claims of rights
1510 made available for publication and any assurances of licenses to be made
1511 available, or the result of an attempt made to obtain a general license or
1512 permission for the use of such proprietary rights by implementors or users of
1513 this specification, can be obtained from the OASIS Executive Director.
1514 OASIS invites any interested party to bring to its attention any copyrights,
1515 patents or patent applications, or other proprietary rights which may cover
1516 technology that may be required to implement this specification. Please
1517 address the information to the OASIS Executive Director.
1518 Copyright (c) OASIS Open 2002-2006. All Rights Reserved.
1519 This document and translations of it may be copied and furnished to others,
1520 and derivative works that comment on or otherwise explain it or assist in its
1521 implementation may be prepared, copied, published and distributed, in whole or
1522 in part, without restriction of any kind, provided that the above copyright
1523 notice and this paragraph are included on all such copies and derivative
1524 works. However, this document itself does not be modified in any way, such as
1525 by removing the copyright notice or references to OASIS, except as needed for
1526 the purpose of developing OASIS specifications, in which case the procedures
1527 for copyrights defined in the OASIS Intellectual Property Rights document must
1528 be followed, or as required to translate it into languages other than English.
1529 The limited permissions granted above are perpetual and will not be revoked by
1530 OASIS or its successors or assigns.
1531 This document and the information contained herein is provided on an "AS
1532 IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
1533 BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL
1534 NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
1535 FITNESS FOR A PARTICULAR PURPOSE.
1536 -->
1537 <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
1538 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1539 xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:rm="http://docs.oasis-
1540 open.org/ws-rx/wsr/200604" xmlns:tns="http://docs.oasis-open.org/ws-
1541 rx/wsr/200604/wsdl" targetNamespace="http://docs.oasis-open.org/ws-
1542 rx/wsr/200604/wsdl">
1543
1544   <wsdl:types>
1545     <xs:schema
1546       <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsr/200604"
1547       schemaLocation="http://docs.oasis-open.org/ws-rx/wsr/200604/wsr-1.1-schema-
1548       200604.xsd"/>
1549     </xs:schema>
1550   </wsdl:types>
1551
1552   <wsdl:message name="CreateSequence">
1553     <wsdl:part name="create" element="rm:CreateSequence"/>
1554   </wsdl:message>
1555   <wsdl:message name="CreateSequenceResponse">
1556     <wsdl:part name="createResponse" element="rm:CreateSequenceResponse"/>
1557   </wsdl:message>
1558   <wsdl:message name="CloseSequence">
1559     <wsdl:part name="close" element="rm:CloseSequence"/>
1560   </wsdl:message>
1561   <wsdl:message name="CloseSequenceResponse">
1562     <wsdl:part name="closeResponse" element="rm:CloseSequenceResponse"/>
1563   </wsdl:message>

```

```

1562     <wsdl:message name="TerminateSequence">
1563         <wsdl:part name="terminate" element="rm:TerminateSequence"/>
1564     </wsdl:message>
1565     <wsdl:message name="TerminateSequenceResponse">
1566         <wsdl:part name="terminateResponse"
1567 element="rm:TerminateSequenceResponse"/>
1568     </wsdl:message>

1569     <wsdl:portType name="SequenceAbstractPortType">
1570         <wsdl:operation name="CreateSequence">
1571             <wsdl:input message="tns:CreateSequence" wsa:Action="http://docs.oasis-
1572 open.org/ws-rx/wsrn/200604/CreateSequence"/>
1573             <wsdl:output message="tns:CreateSequenceResponse"
1574 wsa:Action="http://docs.oasis-open.org/ws-
1575 rx/wsrn/200604/CreateSequenceResponse"/>
1576         </wsdl:operation>
1577         <wsdl:operation name="CloseSequence">
1578             <wsdl:input message="tns:CloseSequence" wsa:Action="http://docs.oasis-
1579 open.org/ws-rx/wsrn/200604/CloseSequence"/>
1580             <wsdl:output message="tns:CloseSequenceResponse"
1581 wsa:Action="http://docs.oasis-open.org/ws-
1582 rx/wsrn/200604/CloseSequenceResponse"/>
1583         </wsdl:operation>
1584         <wsdl:operation name="TerminateSequence">
1585             <wsdl:input message="tns:TerminateSequence"
1586 wsa:Action="http://docs.oasis-open.org/ws-rx/wsrn/200604/TerminateSequence"/>
1587             <wsdl:output message="tns:TerminateSequenceResponse"
1588 wsa:Action="http://docs.oasis-open.org/ws-
1589 rx/wsrn/200604/TerminateSequenceResponse"/>
1590         </wsdl:operation>
1591     </wsdl:portType>

1592 </wsdl:definitions>

```

## D. State Tables

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

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

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

Table 2 RM Source State Transition Table

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

Events	States							
	None	Connecting	Connected	Rollover	Closing	Closed	Terminating	Terminated
number			Rollover	Rollover				
Message Number Rollover Fault	N/A	N/A	<a href="#">No action</a> Rollover	<a href="#">No action</a> Rollover	<a href="#">No action</a> N/A <a href="#">Closing</a>	<a href="#">No action</a> Closed?	<a href="#">No action</a> Ignore? <a href="#">Terminating</a>	Transmit Unknown Sequence Fault Terminated
Close Sequence	N/A	N/A	Transmit Close Sequence Closing	Transmit Close Sequence Closing	Transmit Close Sequence Closing	<a href="#">No action</a> Transmit Close Sequence Closed	<a href="#">No action</a> N/A? <a href="#">Terminating</a>	N/A
Close Sequence Response	N/A	N/A	N/A	N/A	<a href="#">No action</a> Closed	<a href="#">No action</a> Closed	<a href="#">No action</a> Ignore? <a href="#">Terminating</a>	Transmit Unknown Sequence Fault Terminated
SeqAck (final)	N/A	N/A	<a href="#">Process Ack/Nack ranges</a> Closed?	<a href="#">Process Ack/Nack ranges</a> Closed?	<a href="#">Process Ack/Nack ranges</a> Closed?	<a href="#">Process Ack/Nack ranges</a> Closed?	<a href="#">Process Ack/Nack ranges</a> Ignore? <a href="#">Terminating</a>	Transmit Unknown Sequence fault Terminated
Sequence Closed Fault	N/A	N/A	<a href="#">No action?</a> <a href="#">Closed</a>	<a href="#">No action?</a> <a href="#">Closed</a>	<a href="#">No action?</a> <a href="#">Closed</a>	<a href="#">No action?</a> <a href="#">Closed</a>	<a href="#">No action</a> Ignore? <a href="#">Terminating</a>	Transmit Unknown Sequence Fault Terminated
Unknown Sequence Fault	N/A	N/A	<a href="#">No action</a> Terminated	<a href="#">No action</a> Terminated	<a href="#">No action</a> Terminated	<a href="#">No action</a> Terminated	<a href="#">No action</a> Terminated	<a href="#">No action</a> Ignore Terminated
Sequence Terminated Fault	N/A	<a href="#">N/A</a> <a href="#">Terminated</a>	<a href="#">No action</a> Terminated	<a href="#">No action</a> Terminated	<a href="#">No action</a> Terminated	<a href="#">No action</a> Terminated	<a href="#">No action</a> Terminated	<a href="#">No action</a> Ignore Terminated
Terminate Sequence	N/A	N/A	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	Transmit Terminate Sequence Terminating	N/A
Terminate Sequence Response	N/A	N/A	N/A	N/A	N/A	N/A	<a href="#">No action</a> Terminated	<a href="#">No action</a> Terminated
Elapse Expires duration	N/A	N/A	<a href="#">Send SequenceTerminated Fault</a> Terminated	<a href="#">Send SequenceTerminated Fault</a> Terminated	<a href="#">Send SequenceTerminated Fault</a> Terminated	<a href="#">Send SequenceTerminated Fault</a> Terminated	<a href="#">Send SequenceTerminated Fault</a> Terminated?	N/A

1597 In Table 2 above, the rows consists of events that occur at the RM Source throughout the lifetime of an  
1598 RM Sequence and the columns consists of various RM Source states. Each cell in the table above lists  
1599 the action that the RM Source takes on occurrence of a particular event and the next state that it  
1600 transitions.

1601 Table 3 RM Destination State Transition Table

Events	States						
	None	Connecting	Connected	Rollover	Rollover-Closed	Closed	Terminated
Creation request not satisfied	N/A	Send Create Sequence Refused Fault Terminated	N/A	N/A	N/A	N/A	
Unrecoverable error on creation	N/A	Send-Sequence-Terminated-Fault? Terminated	N/A	N/A	N/A	N/A	
New Message (with message number within range)	N/A	N/A	No action Send-SequenceAck Connected	Send-Message-Number-Rollover-Fault Rollover	Send-Message-Number-Rollover-or-Sequence-Closed-Fault?(with SeqAck+Final) Rollover-Closed	Send Sequence Closed Fault (with SeqAck+Final) Closed	Send Unknown Seq Fault? Terminated
Retransmitted message	N/A	N/A	Send-SequenceAck Connected	Send-SequenceAck Rollover	Send-SeqAck+Final Rollover-Closed	Send-SeqAck+Final Closed	Send-Unknown-Seq-Fault Terminated
Ack requested	N/A	N/A	Send SequenceAck Connected	Send-SequenceAck Rollover	Send-SeqAck+Final Rollover-Closed	Send SeqAck+Final Closed	Send Unknown Seq Fault Terminated
Message (with Reach-max-message number outside of range)	N/A	N/A	Send Message-Number-Rollover-Fault ConnectedRollover	Rollover	Rollover-Closed	N/A	N/A
Message-Number-Rollover-Fault	N/A	N/A	Rollover	Rollover	Rollover-Closed	Closed?	Send-Unknown-Sequence-Fault Terminated

Events	States						
	None	Connecting	Connected	Rollover	Rollover-Closed	Closed	Terminated
Close Sequence	N/A	N/A	Send CloseSequenceResponse with SequenceAck (Final) Closed	Send-CloseSequenceResponse-with-SequenceAck-Final Rollover-Closed	Send-Close-Sequence-Response-with-SeqAck+Final Rollover-Closed	Send Close Sequence Response with SeqAck+Final Closed	Send Unknown Sequence Fault Terminated
Close Sequence itself	N/A	N/A	Closed	Rollover-Closed	Rollover-Closed	Send Sequence Closed Fault Closed	N/A
Terminate Sequence	N/A	N/A	Send Terminate Sequence Response Terminated	Terminated	Terminated	Send Terminate Sequence Response Terminated	Send Unknown Sequence Fault Terminated
Unknown Sequence Fault	N/A	N/A	No action Terminated	Terminated	Terminated	No action Terminated	No action/ignore Terminated
Sequence Terminated Fault	N/A	N/A	No action Terminated	Terminated	Terminated	No action Terminated	No action/ignore Terminated
Terminate-Sequence	N/A	N/A	Terminated	Terminated	Terminated	Terminated	N/A
Elapse Expires duration	N/A	N/A	Send Sequence Terminated Fault Terminated	Terminated	Terminated	Send Sequence Terminated Fault Terminated	N/A

1602 In Table 3 above, the rows consists of events that occur at the RM Destination throughout the lifetime of  
 1603 an RM Sequence and the columns consists of various RM Destination states. Each cell in the table above  
 1604 lists the action that the RM Destination takes on occurrence of a particular event and the next state that it  
 1605 transitions.

## E. Acknowledgments

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

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

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

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

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

*TBD*



## F. Revision History

Rev	Date	By Whom	What
wd-01	2005-07-07	Christopher Ferris	Initial version created based on submission by the authors.
ws-02	2005-07-21	Doug Davis	i011 (PT0S) added
wd-02	2005-08-16	Anish Karmarkar	Trivial editorial changes
ws-03	2005-09-15	Doug Davis	i019 and i028 (CloseSeq) added
wd-05	2005-09-26	Gilbert Pilz	i005 (Source resend of nacks messages when ack already received) added.
wd-05	2005-09-27	Doug Davis	i027 (InOrder delivery assurance spanning multiple sequences) added
wd-05	2005-09-27	Doug Davis	i020 (Semantics of "At most once" Delivery Assurance) added
wd-05	2005-09-27	Doug Davis	i034 (Fault while processing a piggy-backed RM header) added
wd-05	2005-09-27	Doug Davis	i033 (Processing model of NACKs) added
wd-05	2005-09-27	Doug Davis	i031 (AckRequested schema inconsistency) added
wd-05	2005-09-27	Doug Davis	i025 (SeqAck/None) added
wd-05	2005-09-27	Doug Davis	i029 (Remove dependency on WS-Security) added
wd-05	2005-09-27	Doug Davis	i039 (What does 'have a mU attribute' mean) added
wd-05	2005-09-27	Doug Davis	i040 (Change 'optiona'/'required' to 'OPTIONAL'/'REQUIRED') added
wd-05	2005-09-30	Anish Karmarkar	i017 (Change NS to <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

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

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

Rev	Date	By Whom	What
wd-12	2006-03-20	Doug Davis	Added space in "SOAP1.x" – PaulCotton
wd-12	2006-04-11	Doug Davis	Issue 007 applied
wd-12	2006-04-11	Doug Davis	Issue 090 applied
wd-12	2006-04-11	Doug Davis	Issue 098 applied
wd-12	2006-04-11	Doug Davis	Issue 099 applied
wd-12	2006-04-11	Doug Davis	Issue 101 applied
wd-12	2006-04-11	Doug Davis	Issue 103 applied
wd-12	2006-04-11	Doug Davis	Issue 104 applied
wd-12	2006-04-11	Doug Davis	Issue 105 applied
wd-12	2006-04-11	Doug Davis	Issue 107 applied
wd-12	2006-04-11	Doug Davis	Issue 109 applied
wd-12	2006-04-11	Doug Davis	Issue 110 applied
wd-12	2006-04-12	Doug Davis	Used "generated" instead of "issue" or "send" when talking about faults.
wd-12	2006-04-24	Gilbert Pilz	Update references to WS-Addressing to the Proposed Recommendations; update WS-RM namespace to "200604".
<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>

## G. Notices

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

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

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

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

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

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