



# Web Services Reliable Messaging TC WS-Reliability

## Working Draft 0.992, 10 March 2004

### Document identifier:

wd-web services reliable messaging tc-ws-reliability-0.992

### Location:

<http://www.oasis-open.org/committees/wsrn/documents/specs/1.1/ws-reliability1.1.pdf>

### Editor:

Kazunori Iwasa, Fujitsu Limited <[kiwasa@jp.fujitsu.com](mailto:kiwasa@jp.fujitsu.com)>

### Abstract:

Web Services Reliability (WS-Reliability) is a SOAP-based protocol for exchanging SOAP messages with guaranteed delivery, no duplicates, and guaranteed message ordering. WS-Reliability is defined as SOAP header extensions, and is independent of the underlying protocol. This specification contains a binding to HTTP.

### Status:

This document is updated aperiodically on no particular schedule.

Committee members should send comments on this specification to the [wsrn@lists.oasis-open.org](mailto:wsrn@lists.oasis-open.org) list. Others should subscribe to and send comments to the [wsrn-comment@lists.oasis-open.org](mailto:wsrn-comment@lists.oasis-open.org) list. To subscribe, send an email message to [wsrn-comment-request@lists.oasis-open.org](mailto:wsrn-comment-request@lists.oasis-open.org) with the word "subscribe" as the body of the message.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Web Services Reliable Messaging TC web page (<http://www.oasis-open.org/committees/wsrn/>).

The errata page for this specification is at <http://www.oasis-open.org/committees/wsrn/documents/errata/1.1/index.html>.

---

## 29 Table of Contents

30	1 Introduction.....	4
31	1.1 Purpose of WS-Reliability.....	4
32	1.2 Scope and Definition of Reliable Messaging.....	4
33	1.3 Notational Conventions.....	5
34	1.4 Relation to Other Specifications .....	5
35	1.5 Examples of Messages Compliant with WS-Reliability.....	6
36	1.6 Terminology.....	9
37	1.7 The Reliability agreement.....	11
38	2 Messaging Model.....	14
39	2.1 Messaging Context .....	14
40	2.2 Message Reply Patterns.....	14
41	2.3 Message Identification and Grouping.....	15
42	3 Reliability Features.....	16
43	3.1 Guaranteed Delivery.....	16
44	3.2 Duplicate Elimination.....	16
45	3.3 Guaranteed Message Ordering.....	17
46	3.4 Sequence Number.....	18
47	4 Message Format.....	19
48	4.1 Structure.....	19
49	4.2 Request Element.....	21
50	4.3 PollRequest Element.....	27
51	4.4 Response Element.....	30
52	4.5 Fault Codes For Reliable Messaging Failures.....	34
53	5 Operational Aspects and Semantics.....	38
54	5.1 Message Group Life Cycle.....	38
55	5.2 WSDL Operation Type.....	42
56	5.3 Poll Reply Pattern Semantics and Usage.....	43
57	5.4 Attachments.....	43
58	6 HTTP Binding.....	44
59	6.1 Reliable Messaging with “Response” binding pattern.....	44
60	6.2 Reliable Messaging with “Callback” binding pattern.....	46
61	6.3 Reliable Messaging with “Poll” binding pattern.....	48
62	7 Conformance .....	51
63	8 References.....	52

64	8.1 Normative References.....	52
65	8.2 Non-normative References.....	53
66	Appendix A. WS-Reliability Features, Properties and Compositor (Normative and Optional).....	54
67	Appendix B. Acknowledgments.....	64
68	Appendix C. Revision History.....	66
69	Appendix D. Futures List.....	76
70	Appendix E. Notices.....	77
71		

---

## 72 1 Introduction

### 73 1.1 Purpose of WS-Reliability

74 The purpose of WS-Reliability is to address reliable messaging requirements, which become  
75 critical, for example, when using Web Services in B2B applications. SOAP [SOAP1.2] over HTTP  
76 [RFC2616] is not sufficient when an application-level messaging protocol must also address  
77 reliability and security. This specification is intended as an initial proposal for defining reliability in  
78 the context of current Web Services standards. The specification borrows from previous work in  
79 messaging and transport protocols, e.g., SOAP, and the ebXML Message Service [ebMS].

### 80 1.2 Scope and Definition of Reliable Messaging

81 The focus of this specification is on the SOAP layer and envelope. In the current specification, we  
82 will define reliable messaging as the mechanism supporting any of the following requirements:

- 83 • Guaranteed message delivery, or At-Least-Once delivery semantics.
- 84 • Guaranteed message duplicate elimination, or At-Most-Once delivery semantics.
- 85 • Guaranteed message delivery and duplicate elimination, or Exactly-Once delivery  
86 semantics.
- 87 • Guaranteed message ordering for delivery, within a context delimited using a group ID.

88 Within the scope of this specification, the following features are investigated:

- 89 • Asynchronous messaging at the application level.
- 90 • Three reliability features: Guaranteed Delivery, Duplicate Elimination, and Guaranteed  
91 Message Ordering.

92 Some messaging features are not mentioned in this specification. They are considered out of  
93 scope, yet the design of this specification is preserving compatibility with some of them. They are:

- 94 • Application level synchronous messaging. Synchronous messaging applications that  
95 require immediate knowledge of the error status instead of waiting for the messaging  
96 layer to resend the message when an error is returned.
- 97 • Routing features. This specification addresses end-to-end reliability, and is not  
98 concerned with intermediaries. The mechanisms described are orthogonal to routing  
99 techniques, and can be used in combination with these.

100 The OASIS WS-RM TC does not attempt to cover all aspects of Reliable Messaging. Several  
101 fundamental questions on reliability need to be addressed in subsequent work, and are only  
102 partially addressed in this specification:

- 103 • Given that some reliability objectives cannot always be guaranteed or attainable,  
104 should a reliability contract include advanced quality of service elements (which may  
105 translate into specifying quantitative thresholds, e.g., Rate of delivery success, scope  
106 of a duplicate check, size of a message archive)? How could these quantitative  
107 parameters adjust to resource availability - memory, storage, computing - which  
108 depends on the communication system (mobile device, messaging hub, etc.)?

- 109           • Beyond the specified qualities of message delivery (Guaranteed Delivery, Duplicate  
 110           Elimination, and Guaranteed Message Ordering), how much of the synchronization  
 111           between sender and receiver applications can and should be supported (i.e., the  
 112           degree to which both sender and receiver parties share the same understanding about  
 113           the outcome of a reliable exchange)?

### 114 1.3 Notational Conventions

115 This document occasionally uses terms that appear in capital letters. When the terms "MUST",  
 116 "REQUIRED", "SHALL", "SHOULD", "RECOMMENDED", "MAY", "OPTIONAL", "MUST NOT",  
 117 "NOT REQUIRED", "SHALL NOT", and "SHOULD NOT" appear capitalized, they are being used  
 118 to indicate particular requirements of this specification. An interpretation of the meanings of these  
 119 terms appears in [RFC2119].

120 Section 4 includes tables to explain each element. The meaning of labels in the table are follows:

- 121           • **Cardinality** : A constraint on the number of instances of an item type which may be  
 122           present in an enclosing item. (e.g. "Cardinality = 0 or 1" means the message may not  
 123           include the element, or it may include the element only once.)
- 124           • **Value** : A type or format for a value of the element.
- 125           • **Attributes** : Attribute names for the element. And type or format for its value is also  
 126           included in parentheses.
- 127           • **Child elements**: Child element for the element.

128 This specification uses the following namespace prefixes:

<i>Prefix</i>	<i>Namespace</i>
soap	http://schemas.xmlsoap.org/soap/envelope/
wsrn	http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1

129

130 *The choice of any namespace prefix is arbitrary and not semantically significant.*

### 131 1.4 Relation to Other Specifications

- 132           • **W3C SOAP1.1/1.2**: SOAP1.1 [SOAP1.1] and SOAP1.2 [SOAP1.2] are the base  
 133           protocols for this specification. This specification defines reliable messaging protocol  
 134           embedded in the SOAP Header.
- 135           • **OASIS ebXML Message Service Specification 2.0**: The reliable message  
 136           mechanism defined in the ebXML Message Service Specification 2.0 [ebMS] is  
 137           implemented in a number of products and open source efforts, many of which have  
 138           undergone interoperability testing. WS-Reliability borrows from this technology.
- 139           • **OASIS WS-Security**: This specification defines reliability independently from security,  
 140           each of these features mapping to different SOAP header extensions. Although both  
 141           features can be used in combination, the specification does not attempt to compose  
 142           them in a more intricate way, nor does it attempt to profile their combination. This  
 143           specification can be used with WS-Security [WSS] when that effort is completed in  
 144           OASIS.

- 145       • **WS-I Basic Profile 1.0:** This specification is compliant with WS-I Basic Profile 1.0a  
146       [WS-I BP1.0] for use of other technologies including SOAP, WSDL [WSDL1.1], and  
147       XML schema [XML Schema].

148

## 149 **1.5 Examples of Messages Compliant with WS-Reliability**

### 150 **Example 1 Reliable Message embedded in HTTP Request**

```
151 POST /abc/servlet/wsrListener HTTP/1.0
152 Content-Type: text/xml; charset=utf-8
153 Host: 192.168.183.100
154 SOAPAction: ""
155 Content-Length: 1214
156
157 <?xml version="1.0" encoding="UTF-8"?>
158 <soap:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
159   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
160   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
161   xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
162   <soap:Header>
163     <Request
164       xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"
165       soap:mustUnderstand="1">
166       <MessageId groupId="mid://20040202.103832@oasis-open.org">
167         <SequenceNum number="0" status="Start"
168           groupExpiryTime="2005-02-02T03:00:33-31:00" />
169       </MessageId>
170       <ExpiryTime>2004-09-07T03:01:03-03:50</ExpiryTime>
171       <ReplyPattern>Poll</ReplyPattern>
172       <AckRequested/>
173       <DuplicateElimination/>
174       <MessageOrder/>
175     </Request>
176   </soap:Header>
177   <soap:Body>
178     <Request xmlns="http://wsr-example.org/">Request Message</Request>
179   </soap:Body>
180 </soap:Envelope>
```

181 The message above uses the Request reliability element, which specifies among other things,  
182 that all three features should be used: Guaranteed delivery ("AckRequested" element), No  
183 Duplicate Delivery ("DuplicateElimination" element) and Ordered Delivery ("MessageOrder"  
184 element).

185

### 186 **Example 2 PollRequest Message embedded in HTTP Request**

```
187 POST /abc/servlet/wsrListener HTTP/1.0
188 Content-Type: text/xml; charset=utf-8
189 Host: 192.168.183.100
190 SOAPAction: ""
191 Content-Length: 1021
192
193 <?xml version="1.0" encoding="UTF-8"?>
194 <soap:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
195   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
196   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
197   xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
198   <soap:Header>
199     <PollRequest
200       xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"
201       soap:mustUnderstand="1">
202       <RefToMessageIds groupId="mid://20040202.103832@oasis-open.org">
203         <SequenceNumberRange from="0" to="20"/>
204       </RefToMessageIds>
205     </PollRequest>
206   </soap:Header>
207   <soap:Body />
208 </soap:Envelope>
```

209 The message above uses the PollRequest reliability element, which is polling the receiver for the  
210 status of messages within the range of sequence numbers 0 to 20 of a particular group. The  
211 expected response will tell which of these messages have been delivered (Acknowledged).

212

### 213 **Example 3 Acknowledgment Message embedded in HTTP Response**

```
214 HTTP/1.0 200 OK
215 Server: WS-ReliabilityServer
216 Date: Mon, 02 Feb 2004 10:38:32 GMT
217 Content-Language: en
218 Content-Type: text/xml; charset=utf-8
```

```

219 Content-Length: 924
220
221 <?xml version="1.0" encoding="UTF-8"?>
222 <soap:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
223   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
224   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
225   xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
226   <soap:Header>
227     <Response
228       xmlns="http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1"
229       soap:mustUnderstand="1" replyPattern="Poll">
230       <NonSequenceReply groupId="mid://20040202.103811@oasis-open.org">
231       <SequenceReplies groupId="mid://20040202.103832@oasis-open.org">
232         <ReplyRange from="0" to="14"/>
233         <ReplyRange from="16" to="20"/>
234       </SequenceReplies>
235     </Response>
236   </soap:Header>
237   <soap:Body />
238 </soap:Envelope>

```

239 The message above uses the Response reliability element, which in this case is carrying the  
240 response of a previous PollRequest element. The response acknowledges messages for a  
241 particular group within the ranges of sequence numbers 0 to 14 and 16 to 20 (meaning that 15  
242 has not been delivered yet, possibly because it was not received.)

243

#### 244 **Example 4 Fault Message embedded in HTTP Response**

```

245 HTTP/1.0 200 OK
246 Server: WS-ReliabilityServer
247 Date: Mon, 02 Feb 2004 10:38:32 GMT
248 Content-Language: en
249 Content-Type: text/xml; charset=utf-8
250 Content-Length: 624
251
252 <?xml version="1.0" encoding="UTF-8"?>
253 <soap:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
254   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
255   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```



```
256 xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
257 <soap:Header>
258 <Response
259 xmlns="http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1"
260 soap:mustUnderstand="1" replyPattern="Poll" >
261 <SequenceReplies groupId="mid://20040202.103832@oasis-open.org">
262 <ReplyRange from="15" to="15" fault="InvalidRequest"/>
263 </SequenceReplies>
264 </Response>
265 </soap:Header>
266 <soap:Body />
267 </soap:Envelope>
```

268 The message above uses the Response reliability element, which in this case is carrying the resp  
269 onse of a previous PollRequest element. The response is reporting a reliability Fault for message  
270 with sequence number 15 within a particular group.

## 271 **1.6 Terminology**

### 272 **Reliable Messaging:**

273 The set of mechanisms and procedures required to send messages reliably. This includes the  
274 processing of Acknowledgment messages, re-sending of messages, duplicate message  
275 elimination, and message ordering.

276

### 277 **Reliable Messaging Processor (RMP):**

278 A module capable of processing and enforcing Reliable Messaging as described in this  
279 specification. With regard to the transmission of a message from one RMP to another, the former  
280 will be act in the role of "sender" and the latter in the role of "receiver".

281

### 282 **Deliver:**

283 An abstract operation the Receiving RMP may invoke per Reliable Message (e.g, a request to the  
284 application layer to take responsibility for the Reliable message).

285

### 286 **Submit:**

287 An abstract operation the Sending RMP supports, invoked per Reliable message (e.g., a request  
288 to the sending RMP to take responsibility for the reliable message. The time at which this  
289 operation is invoked must be clearly identifiable so that the RMP can always establish in which  
290 order two submissions are made.

291

### 292 **Notify:**

293 An abstract operation the Sending RMP may invoke per Reliable Message (e.g, a notification that  
294 the Sending RMP cannot insure that the Requested Reliability feature were realized).

295

### 296 **Message Identifier:**

297 A Message Identifier is a value or a combination of values in the message header, that uniquely  
298 identifies reliable messages. This identifier is only meaningful to the reliability features described  
299 here.

300

### 301 **Message Delivery:**

302 Message delivery is the action of invoking the deliver operation for a Reliable Message. This  
303 action marks the end of the RMP processing for this message. The time at which this action  
304 occurs must be clearly identifiable so that the next message processor (application) can always  
305 establish in which order two deliveries are made.

306 Examples of message delivery are:

- 307 • pushing the message in a queue accessible by an application,
- 308 • calling back an application component,
- 309 • storing the message in a database where it is accessible by the next processor.

310

### 311 **Reliable Message:**

312 A message for which the sender requires some level of reliable delivery, typically requiring  
313 acknowledgment for notification of delivery.

314

### 315 **PollRequest Message:**

316 A polling message for Acknowledgment message(s). A sender RMP may send a PollRequest  
317 Message for polling of Acknowledgment message(s) regardless of RM-Reply Pattern of the  
318 original Reliable Message. E.g., Sender RMP may send PollRequest Message to retrieve  
319 Acknowledgment message for a message originally sent with Callback ReplyPattern.

320

### 321 **Acknowledgment Indication:**

322 An indication which refers to a previous message delivered by the Receiving RMP. An  
323 Acknowledgment signals that the acknowledged message has been successfully delivered,  
324 meaning that it has satisfied all the reliability requirements placed on it for delivery.

325

### 326 **Reliable Messaging Fault Indication:**

327 An indication which refers to a previous message which encountered a Reliable Messaging fault  
328 condition at the Receiving RMP. It signals to the sender of the referred message that there was a  
329 failure to receive or process the message.

330

331

332 **Duplicate Message:**

333 A message is duplicate of another message if it has same message identifier.

334

335 **Reliable Messaging Reply (RM-Reply):**

336 An indication referring to a previous message, that is either an Acknowledgment Indication or a  
337 Reliable Messaging Fault Indication.

338

339 **Response RM-Reply Pattern:**

340 The Response RM-Reply pattern is used if the outbound Reliable Message is sent in a request of  
341 the underlying protocol and the RM-Reply is sent in the response message of the underlying  
342 protocol that corresponds to the request.

343

344 **Callback RM-Reply Pattern:**

345 The Callback RM-Reply pattern is used if the RM-Reply of a previous message is contained in an  
346 underlying protocol request of a second request/response exchange (or a second one-way  
347 message).

348

349 **Polling RM-Reply Pattern:**

350 The Polling RM-Reply pattern is used if a second underlying protocol request is issued to the  
351 receiver of a previous message, in order to obtain a RM-Reply. The RM-Reply can be either  
352 contained in the underlying protocol response to this request or in a separate underlying request  
353 from the receiver to the sender. This polling pattern is generally expected to be used in situations  
354 where it is inappropriate for the sender of reliable messages to receive underlying protocol  
355 requests (behind the firewall cases) or to avoid resending bulk messages often.

356

357 **1.7 The Reliability agreement**

358 **1.7.1 Definition**

359 A Reliability agreement for messaging, or RM Agreement, describes an agreed contract between  
360 a sender RMP and a receiver RMP regarding:

- 361
- The nature, content and occurrence of exchanged messages.
  - The timing, content and occurrence of the submit, deliver, notify operations on these RMPs.
- 362  
363

364 In so far as the submit, notify and deliver operations are interpreted as implementing  
365 communication between an RMP and an application, the above contract can be seen as a  
366 contract between the application layer, the sender and receiver RMPs.

367 The way such a contract is established or communicated to each party is out of scope, although  
368 the assumption is that only the sender RMP needs to initially have knowledge of the RM  
369 Agreement. No prior communication of the contract to the receiving party (RMP and its  
370 application) is required. I.e., the Receiver RMP does not need other input than the header of

371 received messages to get knowledge of the reliability requirements to which these messages are  
372 subject.

373

## 374 **1.7.2 RM Agreement Items**

375 An RM Agreement is a list of Agreement Items. An RMP implementation MUST be capable of:

376 (1) taking knowledge of a set of values that represent the RM Agreement Items described in this  
377 specification,

378 E.g., via configuration, or

379 via an API call, or

380 via a message, or

381 via the result of an algorithm.

382 (2) processing them according to the semantics described in this specification.

383 Some of these items will appear in the message protocol (i.e., map to some message header  
384 field), and some will not.

385 The following list of Agreement Items is considered by this specification. Each item is listed with  
386 its possible values:

387 • GuaranteedDelivery (enabled/disabled): for setting Guaranteed Delivery. (See Section  
388 3.1 for details)

389 • NoDuplicateDelivery (enabled/disabled): for setting message delivery without  
390 duplicates, or Duplicate Elimination. (See Section 3.2 for details)

391 • OrderedDelivery (enabled/disabled): for setting Guaranteed Message Ordering. (See  
392 Section 3.3 for details)

393 • GroupMaxIdleDuration (number of seconds): For setting the elapsed time limit from  
394 the last message sent or received in a group, after which the group can be terminated.  
395 The value MUST NOT be zero or smaller.

396 • GroupExpiryTime (number of seconds): For setting the date and time after which the  
397 group can be terminated. The value MUST NOT be zero or smaller.

398 • ExpiryTime (number of seconds): For setting the date and time after which a message  
399 must not be delivered to the receiving application.

400 • RetryMaxTimes (integer number): For setting the maximum number of times a  
401 message must be resent if not acknowledged. The value MUST be zero or larger.

402 • RetryTimeInterval (number of seconds): For setting the minimal elapsed time between  
403 two re-sending of the same message. The value MUST NOT be zero or smaller.

404 • ReplyPattern ("Response", "Callback", "Poll") For setting the mode of response for  
405 Acknowledgments or Faults.

406

### 407 **1.7.3 Messaging Scope of Agreement Items**

408 The messaging scope of these agreement items may vary, as messages may be associated with  
409 a group. There are three scopes to consider:

- 410 • (s1) All messages sent over a connection between a Sender RMP and a Receiver  
411 RMP (default).
- 412 • (s2) All messages sent within a group.
- 413 • (s3) A single message, standalone (singleton) or within a group of several messages  
414 (non-singleton group).

415 Some agreement items obviously relate to a particular scope, e.g. ExpiryTime is affecting each  
416 message separately, while GroupExpiryTime is an agreement item about groups.

417 The smallest required scope for each RM Agreement item is:

418 Message scope (s3):

- 419 • ExpiryTime
- 420 • RetryMaxTimes
- 421 • RetryTimeInterval
- 422 • ReplyPattern

423 Group scope (s2):

- 424 • GuaranteedDelivery
- 425 • NoDuplicateDelivery
- 426 • OrderedDelivery
- 427 • GroupExpiryTime
- 428 • GroupMaxIdleDuration

429 NOTE: Although a RMP must support each agreement item at the scope level shown, the RMP  
430 implementation may also provide a way to assign a broader scope to these items.

431 Example: a RMP implementation may decide to provide a way to specify the same ExpiryTime  
432 value for all messages of a group.

433

### 434 **1.7.4 Rules about Agreement Items**

435 When defining an RM Agreement instance, there are some dependencies between the items of  
436 the agreement that must be respected:

- 437 • If GuaranteedOrdering is enabled for a messaging scope, then GuaranteedDelivery  
438 and NoDuplicateDelivery MUST also be enabled for that messaging scope.
- 439 • If GroupExpiryTime is enabled for a messaging scope, then the item  
440 GroupMaxIdleTime MUST NOT be enabled, and vice versa.

441

---

## 442 2 Messaging Model

443 The following sections provide an overview of the WS-Reliability Messaging Model.

### 444 2.1 Messaging Context

445 The Reliable Messaging Model described in this document makes the following assumptions:

- 446 • Reliability is a contract between two messaging nodes, with respective roles of sender  
447 and receiver: (1) the sender RMP on which the submit message operation is invoked,  
448 and (2) the receiver RMP which invokes the deliver message operation. Intermediaries  
449 are transparent to this specification. Signal messages resulting from a reliable  
450 exchange, such as Acknowledgment message or Reliable Messaging Fault message  
451 are sent from the receiving RMP to the sender RMP.
- 452 • The underlying protocol is a request-response protocol. In other words, this  
453 specification assumes the underlying protocol distinguishes two kinds of messages:  
454 requests and responses. Under normal conditions, a response is always sent back for  
455 each request. This assumption is not essential to the reliable features described here:  
456 these could be reformulated without this assumption.

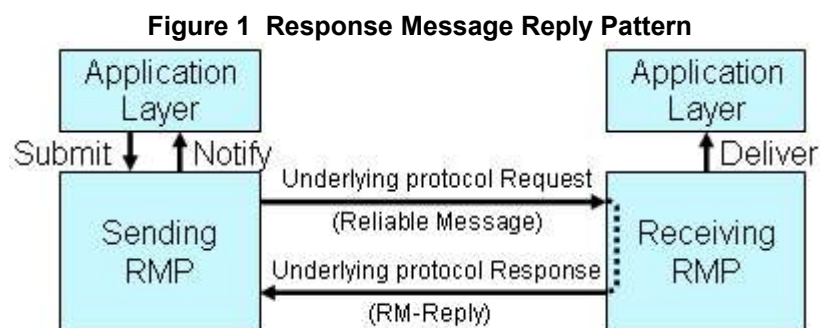
### 457 2.2 Message Reply Patterns

458 There are three ways to send back an Acknowledgment message or a Fault message as  
459 described as follows:

#### 460 (1) Response Message Reply Pattern

461 With this message reply pattern, the outbound Reliable Message is sent in the underlying protocol  
462 request and the RM-Reply is contained in the underlying protocol response message  
463 corresponding to the original request. The figure 1 shows this reply pattern.

464

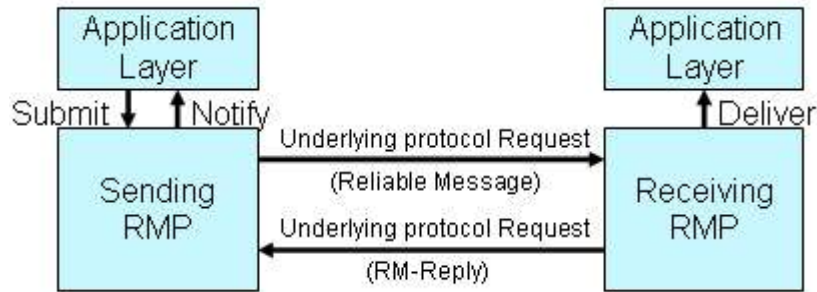


#### 466 (2) Callback Message Reply Pattern

467 With this message reply pattern, the RM-Reply is contained in an underlying protocol request of a  
468 second request/response exchange (or a second one-way message), operating in the opposite  
469 direction to the message containing the outbound Reliable Message. The figure 2 shows this reply  
470 pattern.

471

**Figure 2 Callback Message Reply Pattern**

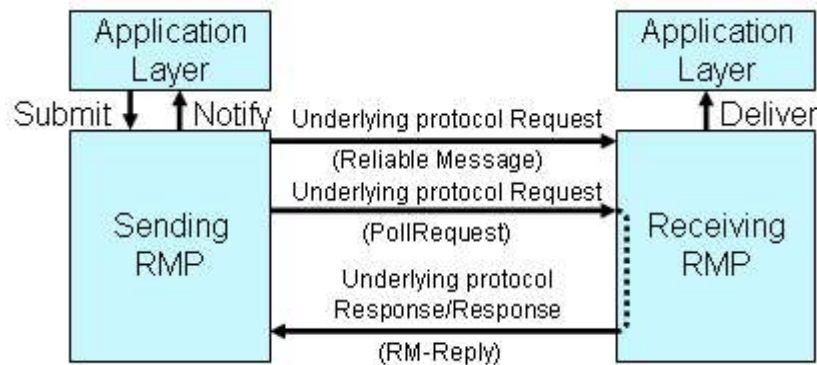


473 **(3) Poll Message Reply Pattern**

474 With this message reply pattern, a second underlying protocol request is issued in the same  
 475 direction as the one containing the outbound Reliable Message to act as a request for  
 476 acknowledgment. The RM-Reply is contained in the underlying protocol response to this request.  
 477 This reply pattern may be used in situations where it is inappropriate for the sender of reliable  
 478 messages to receive underlying protocol requests. The figure 3 shows this reply pattern.

479

**Figure 3 Poll Message Reply Pattern**



481 **2.3 Message Identification and Grouping**

482 Every Reliable Message MUST contain a globally unique Message Identifier. This Message  
 483 Identifier relies on the notion of group. A message always belongs to a group. A group of  
 484 messages is sent from the sender RMP to the receiver RMP as a sequence of individual  
 485 messages. The Message Identifier is a combination of a group ID and of an optional sequence  
 486 number which is an integer, and which is unique within a group. More precisely, a message is  
 487 identified as follows:

488 (1) In case there is only one message in the group (singleton): the group ID, which is a globally  
 489 unique group identifier, may be used alone as Message Identifier. No sequence number is  
 490 required, although allowed.

491 (2) In case the message belongs to a group of several messages: the message is identified by the  
 492 group ID and a sequence number. The group is submitted to the sender RMP as a sequence of  
 493 messages, each sequence number value MUST be numbered with consecutive values starting  
 494 with 0, in the submission order, and MUST be sent in the same order.

495

496

---

## 497 3 Reliability Features

### 498 3.1 Guaranteed Delivery

499 When a business payload is submitted to the sender RMP, the GuaranteedDelivery agreement  
500 item requires that either: (1) the payload is successfully delivered by the receiver RMP, or (2) the  
501 Sender RMP notifies a delivery failure.

502 The guaranteed delivery mechanism will however do its best to get the message delivered, e.g.  
503 resend a message in case of previous failure. In order for the mechanism described here to  
504 operate reliably, it is assumed that the underlying transport protocol prevents message corruption.

505 If the RMP sending a Reliable Message does not receive an Acknowledgment or Fault for a sent  
506 message that has not yet expired, it MUST resend the same message with same MessageId to  
507 the receiver RMP until either one of the following occurs (whichever occurs first):

- 508 • The sender gets a RM-Reply for the message from the receiver.
- 509 • The number of resending attempts specified by the RetryMaxTimes agreement item is  
510 exhausted.
- 511 • The message expires (ExpiryTime is past).

512 The time interval between two retries is specified by the RetryTimeInterval agreement item. If the  
513 sender RMP cannot guarantee that the message has been successfully delivered by the receiving  
514 RMP, the sender RMP MUST notify a delivery error.

515 The sending RMP MUST NOT send retries with a MessageId, for which it received an RM-Reply  
516 with one of the following Fault types:

- 517 • An Invalid Message Format fault code (Table 16)
- 518 • A NonSupportedFeature fault code
- 519 • A PermanentProcessingFailure fault code

520 The RMP MUST NOT return an Reliable Messaging Fault for a delivered MessageId. The RMP  
521 MUST NOT deliver a message which encounters an Reliable Messaging Fault.

522 Guaranteed Delivery assumes also that the RMP functions are operational.

523 Example 1). A PC Server may use a HDD for it's persistent Storage, and those messages  
524 persisted in the HDD are reliably maintained even if the the system software crashes and the  
525 system is rebooted. However, if the HDD itself crashes, it is neither possible to deliver the  
526 message on the receiver side, nor to notify failure on the sender side.

527 Example 2) . A message persisted in a sending mobile phone may be lost when it's battery is  
528 detached. In this case, neither successful message transmission and delivery, nor failure  
529 notification will be possible.

### 530 3.2 Duplicate Elimination

531 When an RMP delivers a received business payload, the NoDuplicateDelivery agreement item  
532 requires that no future business payload from a message with same identity as the message  
533 containing the first payload will ever be delivered.



534 A number of conditions may result in reception of duplicate message(s), e.g., temporary downtime  
 535 of the sender or receiver, a routing problem between the sender and receiver, etc. In order to  
 536 provide Duplicate Elimination (At-Most-Once) semantics, the receiver RMP MUST NOT deliver a  
 537 message that is a duplicate of a previously delivered message.

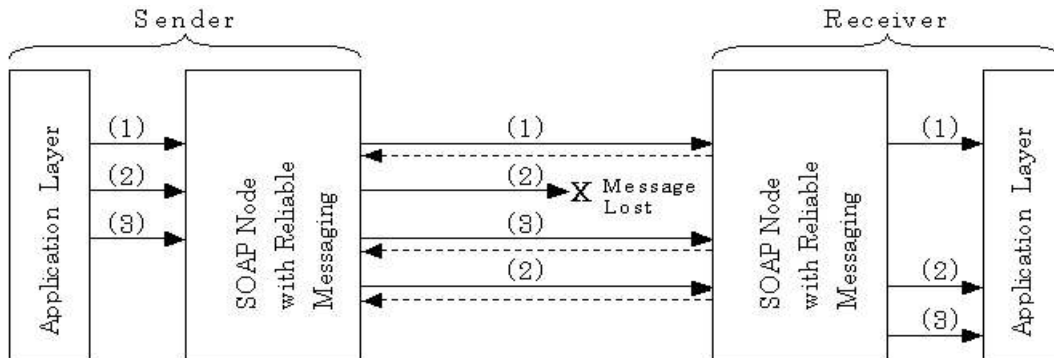
### 538 3.3 Guaranteed Message Ordering

539 When an ordered sequence of business payloads is submitted to a sender RMP, the  
 540 OrderedDelivery agreement requires that when the receiver RMP delivers one of these business  
 541 payloads, all previous payloads in the sequence have already been delivered.

542 Some applications will expect to receive a sequence of messages from the same sender in the  
 543 same order these messages were sent. Although there are often means to enforce this at the  
 544 application layer, this is not always possible or practical. In such cases, the messaging layer is  
 545 required to guarantee the message order. Guaranteed Message Ordering provides this function.  
 546 Figure 4 illustrates how Guaranteed Message Ordering works.

547 In the example illustrated by Figure 4, when the sender application submits three messages (1),  
 548 (2), and (3) with Guaranteed Message Ordering, the receiver's RMP delivers these messages in  
 549 the same order. The receiver RMP received message (1) and (3). The receiver RMP delivers the  
 550 message (1), but it persists message (3) until message (2) is received. When message (2) is  
 551 received, the RMP delivers message (2) and (3) in order.

552 **Figure 4 Guaranteed Message Ordering**



553

554 This behavior can be subject to variants and additional rules to deal with specific failure use  
 555 cases, such as when a node cannot deliver the proper-sequence of messages due to a message  
 556 being lost or expired.

#### 557 **Failure Case:**

558 In case a message is missing in the sequence and if either one of the two following conditions is  
 559 verified:

- 560 • A previously received and not yet delivered out-of-order message has expired.
- 561 • Restoring an ordered delivery would require too much effort from an implementation  
 562 (e.g. The number of out-of-order received messages is too large for the available  
 563 storage space).

564 Then the receiver RMP MUST abort the ordered delivery. i.e., It MUST NOT deliver any message  
 565 for the group, beyond the last message delivered in order.

566

### 567 **3.4 Sequence Number**

568 A sequence number mechanism is used to track and enforce the order of a sequence of  
569 messages within the same group. Such a mechanism has been widely used in the past. In the  
570 Figure 4 above, messages (1), (2), and (3) will be respectively assigned sequence numbers 1, 2,  
571 and 3. If the message (2) was not properly received for any reason, the sender will resend the  
572 message. Sequence numbering allows the receiver RMP to easily detect a missing message in a  
573 sequence, that is (2), as soon as receiving (3). This condition is recognized by the receiver when  
574 the sequence numbers of the messages it receives are not contiguous (e.g., 1, 3, 2).

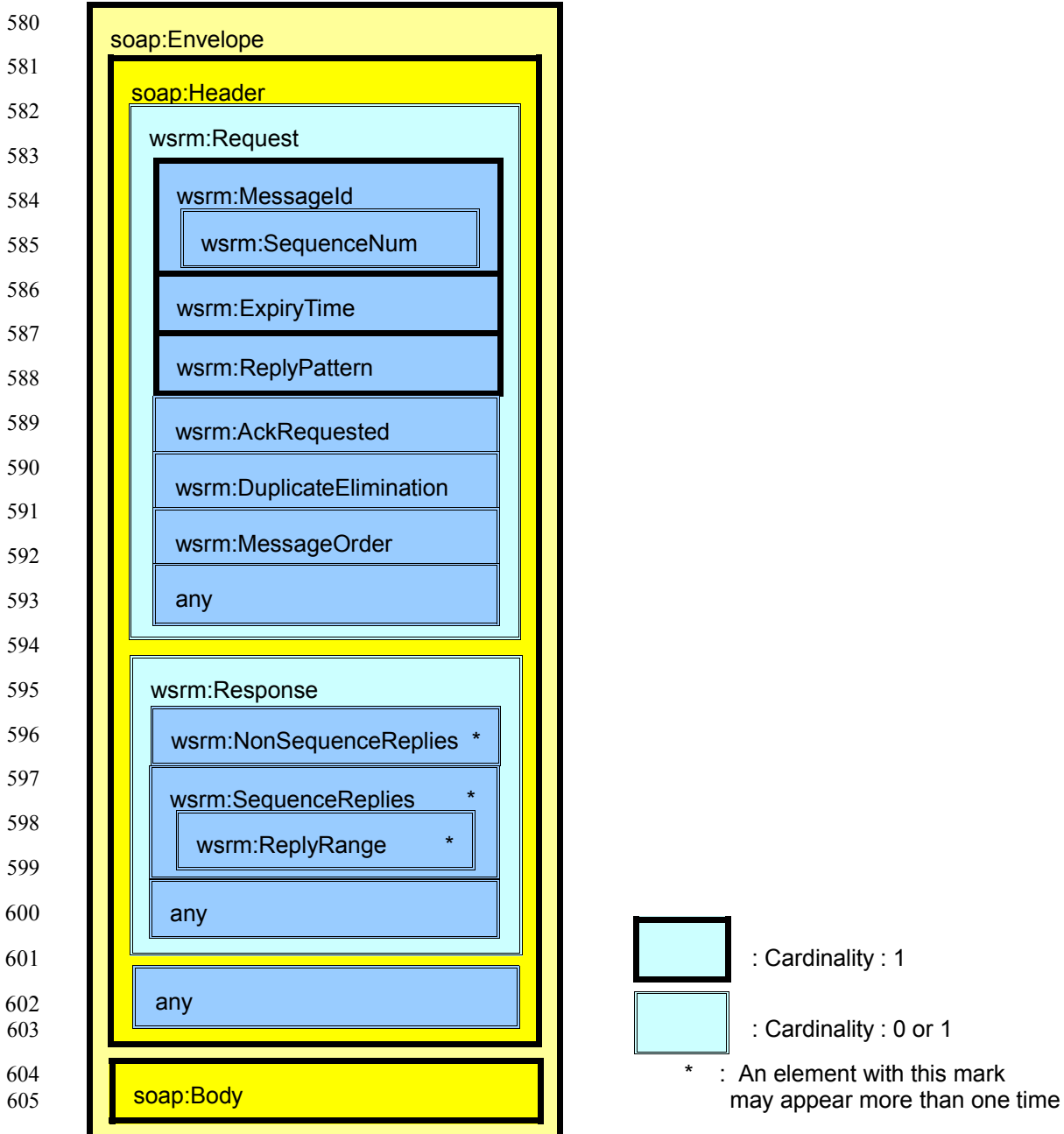
575

## 576 4 Message Format

### 577 4.1 Structure

578 Figure 5 shows the structure of WS-Reliability elements embedded in the SOAP Envelope.

579 **Figure 5 Structure of WS-Reliability elements**



606 Figure 6 shows the structure of PollRequest message embedded in the SOAP Envelope.

607

**Figure 6 Structure of PollRequest message elements**

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

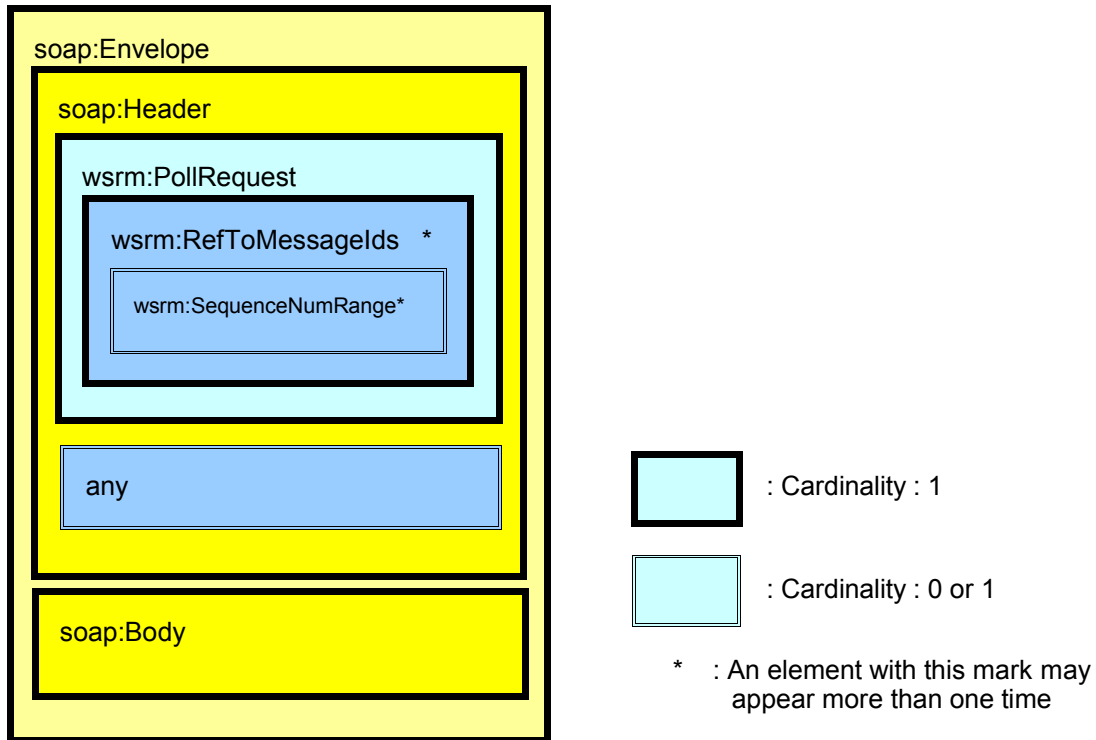
633

634

635

636

637



The namespaces [XML Namespaces] for reliable messaging defined in this specification are:

<http://www.oasis-open.org/committees/wsm/schema/1.1/SOAP1.1> for SOAP1.1 and

<http://www.oasis-open.org/committees/wsm/schema/1.1/SOAP1.2> for SOAP1.2

If there are additional elements that are not described in this specification present in a message, the Reliable Messaging Processor MUST ignore those elements.

Any of the following three elements can be direct child element of the SOAP Header:

- **Request** element
- **PollRequest** element
- **Response** element

## 638 4.2 Request Element

639 A sending RMP MUST include a Request element in a Reliable Message. The Request element  
640 includes specific information to be used for a reliable message. All messages in a group MUST  
641 have the same values for the three Reliable Messaging Quality of Service parameters  
642 (AckRequested, DuplicateElimination and MessageOrder) in their Request element. This element  
643 includes the following attribute and child elements:

- 644 • SOAP **mustUnderstand** attribute with a value of "1"
- 645 • **MessageId** element
- 646 • **ExpiryTime** element
- 647 • **ReplyPattern** element
- 648 • **AckRequested** element
- 649 • **DuplicateElimination** element
- 650 • **MessageOrder** element

651

Table 1 Request Element

Cardinality	1
Value	None
Attributes	MustUnderstand (boolean)
Child elements	MessageId ExpiryTime ReplyPattern AckRequested DuplicateElimination MessageOrder

652

653 Example 5 shows an example of a Request element.

654

655

### Example 5 Request Element

```
656 <Request  
657   xmlns="http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1"  
658   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
659   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
660   soap:mustUnderstand="1">  
661   <MessageId groupId="mid://20040202.103832@oasis-open.org/">  
662     <SequenceNum number="0" status="Start"  
663     groupExpiryTime="2005-02-02T03:00:33-31:00" />
```

```

664 </MessageId>
665 <ExpiryTime>2004-09-07T03:01:03-03:50</ExpiryTime>
666 <ReplyPattern>Response</ReplyPattern>
667 <AckRequested/>
668 <DuplicateElimination/>
669 <MessageOrder/>
670 </Request>

```

## 671 4.2.1 MessageId Element

672 The sending RMP MUST include the MessageId element for a Reliable Message.

673 This element includes the following attribute:

- 674 • a **groupId** attribute

675 Table 2 MessageId Element

Cardinality	1
Value	None
Attributes	groupId (RFC2396 *See 3.1.1 for details)
Child elements	SequenceNum

676

### 677 (1) groupId attribute

678 The RMP MUST include this attribute in the MessageId element. This attribute is to identify a  
679 sequence of messages, where each sequence is of length 1 or more. The sending RMP MUST  
680 use a distinct globally unique groupId for any distinct group of messages. Any group of messages  
681 will have a common groupId value. The syntax of this identification is URI, as defined in  
682 [RFC2396]. It is RECOMMENDED to use the Message-ID schema, as defined in [RFC2392].

### 683 4.2.1.1 SequenceNum Element

684 The sender MUST include the SequenceNum element for a Group with more than one message.

685 When a message includes a MessageOrder element, the SequenceNum element is used for  
686 guaranteeing the message order within the group of messages specified by the same groupId  
687 value. When the MessageOrder element is present, the Message Ordering semantics as  
688 described in Section 3.3 applies.

689 When the sender requests Guaranteed Message Ordering, the sender MUST use Guaranteed  
690 Message Delivery and Duplicate Elimination for that message as well. In other words, an  
691 AckRequested element and a DuplicateElimination element MUST be present when the  
692 MessageOrder element is present.

693 This element includes the following attributes:

- 694 • a **groupExpiryTime** attribute

- 695 • a **groupMaxIdleDuration** attribute
- 696 • a **number** attribute
- 697 • a **status** attribute

698 In a request message, the sender MAY include either a groupExpiryTime attribute or a  
 699 groupMaxIdleDuration attribute corresponding to the group termination parameters specified in  
 700 Section 5.1.2:

701 If the MessageOrder element appears in the message sent, the receiver of the message MUST  
 702 make messages available to the application layer only after all messages with the same groupId  
 703 value and a lower number value have been made available to the application. Example 6  
 704 illustrates some message fragments with SequenceNum element:

### 705 **Example 6 SequenceNum Element**

707 1) First message

```
708 <MessageId groupId="mid://20040202.103832@oasis-open.org/">
709   <SequenceNum number="0" status="Start"
710     groupExpiryTime="2005-02-02T03:00:33-31:00" />
711 </MessageId>
```

712 2) Second message

```
713 <MessageId groupId="mid://20040202.103832@oasis-open.org/">
714   <SequenceNum number="1" status="Continue"
715     groupExpiryTime="2005-02-02T03:00:33-31:00" />
716 </MessageId>
```

717 3) Third message

```
718 <MessageId groupId="mid://20040202.103832@oasis-open.org/">
719   <SequenceNum number="2" status="Continue"
720     groupExpiryTime="2005-02-02T03:00:33-31:00" />
721 </MessageId>
```

722 Table 3 SequenceNum Element

Cardinality	0 or 1 *See 3.1.2 for details
Value	None
Attributes	groupExpiryTime (dateTime) groupMaxIdleDuration (duration) number (unsignedLong) status (string)
Child elements	None

723

## 724 (1) groupExpiryTime attribute

725 A sender MAY include this attribute when groupMaxIdleDuration attribute is not present. This  
726 attribute is used to specify the the date and time at which the sender wishes the sequence group  
727 to terminate. The groupExpiryTime MUST be expressed as UTC and MUST conform to a [XML  
728 Schema] dateTime.

## 729 (2) groupMaxIdleDuration attribute

730 A sender MAY include this attribute when groupExpiryTime attribute is not present. This attribute  
731 is used to specify the maximum idle time. On the receiver side, if the time interval since the last  
732 message was received exceeds the groupMaxIdleDuration, then the sequence group may be  
733 terminated. On the sender side, the same condition applies to the time since the last message  
734 was sent. The groupMaxIdleDuration MUST conform to a [XML Schema] duration.

## 735 (3) number attribute

736 The value of this attribute MUST be unique within the same groupId, and the combination of  
737 groupId and SequenceNum MUST be globally unique to be used for Message Identifier.

738 When a sender node communicates with a receiver node across several groupId values, the  
739 sender MUST maintain an independent counter of the value of number attribute for each groupId.

740 When sending a message containing a MessageOrder element with a new groupId, the sender  
741 MUST start with "0" for the number attribute in the groupId.

742 The value of number attribute MUST conform to [XMLSchema] unsignedLong. For the initial  
743 message with a specific groupId that is sent to the receiver, the number value MUST be "0". After  
744 the initial message has been sent to the receiver, the sender MUST increment the value by one  
745 for each message sent. When the value of a number reaches the maximum value, the sender  
746 MUST generate a new groupId for any following messages. This begins a new sequence that  
747 could overlap with the old in rare circumstances. From the receiver's perspective, no link exists  
748 between the two sequences. To improve the chances that the message ordering is maintained  
749 across this change, the sender SHOULD wait until all Acknowledgment messages have been  
750 received for the old groupId before starting the new sequence.

## 751 (4) status attribute

752 This attribute is used to specify status of the group of messages. The first message in a group  
753 MUST include this attribute, and the last message in a group MAY include this attribute. When  
754 this attribute is present, its value MUST be one of the following three:

755 • **Start:** Indicating the message is the first message for a group of messages.

756 • **Continue:** Indicating the message is in the middle of a group of messages.

757 • **End:** Indicating the message is the last message for a group of messages.

758 The sender node MUST send a very first message, to guarantee the message order, with "Start"  
759 for this attribute. Also, the sender MUST send subsequent messages for the same series of  
760 messages with "Continue", until the message sent is the last one for the series of messages, for  
761 which case the value MUST be "End". When omitted, the default value for this attribute is  
762 "Continue."

763 When an application is receiving messages from an RMP, the actual order of delivered payloads  
764 may be affected by subsequent operations after the "deliver" operation has been invoked. For  
765 example, the actual order of delivery to the application may be affected by queuing taking place  
766 between the RMP and the application - and by the way the application reads such a queue - which  
767 would be out of scope of this specification.



768

### 769 4.2.2 ExpiryTime Element

770 The ExpiryTime element is used to indicate the ultimate time after which the receiver RMP MUST  
 771 NOT invoke the deliver operation for the received message. An RMP MUST include this element  
 772 in a Request element. After a message has been sent for the first time, the value of the  
 773 ExpiryTime in a message MUST NOT be modified in any manner by the Sending RMP, when  
 774 resending the message: two messages with same Message Identifier (duplicates) MUST have the  
 775 same value for ExpiryTime. When a message expires on the Sender side before being  
 776 successfully sent, a Sender RMP MUST NOT send it or resend it, and MUST communicate a  
 777 delivery failure to the Sender application. The time MUST be expressed as UTC and MUST  
 778 conform to a [XML Schema] dateTime. The message is considered expired if the current time, in  
 779 UTC, is greater than the value of the ExpiryTime element.

780 NOTES: Given the above definition of ExpiryTime, in case Duplicate Elimination is required,  
 781 when a received message is processed, it is sufficient to only check for its duplicates among  
 782 MessageIds of past messages that have not expired yet at the time of the duplicate check.

783

Table 4 ExpiryTime Element

Cardinality	1
Value	dateTime
Attributes	None
Child elements	None

784

### 785 4.2.3 ReplyPattern Element

786 The ReplyPattern element is used for a sender to indicate what reply pattern is requested. A RMP  
 787 MUST include the ReplyPattern element in a Request element. This element is used to specify  
 788 whether the Acknowledgment message (or Fault message) should be sent back directly in the  
 789 reply to the reliable message, in a separate callback request, or in the response to a separate poll  
 790 request. This element MUST have one of the following three values:

- 791 • **Response** : A RM-Reply MUST be sent back directly in the response to the Reliable  
 792 Message. This pattern is not applicable for one-way application level MEP.
- 793 • **Callback**: A RM-Reply MUST be sent as a callback request, using the address in the  
 794 replyTo attribute. This pattern is not applicable for request-response application level  
 795 MEP.
- 796 • **Poll**: A RM-Reply MUST be sent as a response to a poll request. This pattern is not  
 797 applicable for request-response application level MEP.

798 The ReplyPattern element contains the following attribute:

- 799 • a **replyTo** attribute

800

801

Table 5 ReplyPattern Element

Cardinality	1
Value	String : Response, Callback, or Poll
Attributes	replyTo (URI)
Child elements	None

802

### 803 (1) replyTo attribute

804 A sender MUST include this attribute for a message with “Callback” value for ReplyPattern  
805 element. The sender MUST NOT include this attribute for a message with “Response” or “Poll”  
806 value for ReplyPattern element. It is to specify the initial sender’s endpoint to receive a callback  
807 Acknowledgment message or Fault message.

808 If present, the replyTo attribute MUST be URI as defined in [RFC 2396].

809

### 810 4.2.4 AckRequested Element

811 A sender MUST include the AckRequested element for Guaranteed Delivery and Guaranteed  
812 Message Ordering. This element is used by a sender to request the receiver to send back an  
813 Acknowledgment if the message sent was delivered, or else a Fault message. If a receiver  
814 receives a message with AckRequested element, the receiver MUST send an Acknowledgment  
815 message even when the message is a duplicate, and if it has already been previously delivered.  
816 (Refer to “Section 3.1 Guaranteed Delivery” for details)

817 The pattern used to send the Acknowledgment or Fault message is based on the value of the  
818 ReplyPattern element.

819

Table 6 AckRequested Element

Cardinality	0 or 1
Value	None
Attributes	None
Child elements	None

820

### 821 4.2.5 DuplicateElimination Element

822 The DuplicateElimination element is used to request the receiver RMP to identify duplicate  
823 messages it has received and process them accordingly (Refer to “Section 3.2 Duplicate  
824 Elimination” for details).

825

Table 7 DuplicateElimination Element

Cardinality	0 or 1
Value	None

Cardinality	0 or 1
Attributes	None
Child elements	None

826

## 827 4.2.6 MessageOrder Element

828 This element is used to request the receiver RMP to invoke delivery operation with the same order  
829 that the sender has submitted. When a sender submits multiple messages with Guaranteed  
830 Message Ordering, the sender **MUST** include the MessageOrder element in every message. All  
831 messages to be delivered in order **MUST** have the same groupId and **MUST** have sequence  
832 number as a value of SequenceNum element in order of the message to be delivered to receiver's  
833 application.

834

Table 8 MessageOrder Element

Cardinality	0 or 1
Value	None
Attributes	None
Child elements	None

835

## 836 4.3 PollRequest Element

837 A sender **MUST** include the PollRequest element only in the PollRequest message as shown in  
838 the Figure6. The PollRequest message contains the PollRequest element. The PollRequest  
839 message is used to query RM-Reply for specific message. Typically, the PollRequest message is  
840 to receive RM-Reply for a message sent with Polling RM-Reply Pattern. However PollRequest  
841 message also can be used to receive RM-Reply for a message that was originally sent with  
842 Response RM-Reply Pattern or Callback RM-Reply Pattern. The response to a PollRequest  
843 message includes RM-Reply information about prior messages. In addition to its use for receiving  
844 replies for requests using the poll RM-Reply pattern, a Sending RMP may use it as a general  
845 query to determine non-expired messages which have been delivered. If a Receiving RMP does  
846 not support this general query, it **MAY** return a notSupportedFeature fault.

847 RM-Reply **MUST** be contained in the underlying response of the Poll request if the replyTo  
848 attribute doesn't exist and should be sent in an underlying request to the endpoint identified by this  
849 attribute if exists.

850 This element includes the following attributes and child element:

- 851 • SOAP **mustUnderstand** attribute with a value of "1"
- 852 • a **replyTo** attribute
- 853 • a **RefToMessageIds** element

854

Table 9 PollRequest Element

Cardinality	0 or 1
Value	None

Cardinality	0 or 1
Attributes	MustUnderstand (boolean) replyTo (URI)
Child elements	RefToMessagelds

855

856

### Example 7 PollRequest Element

857

<PollRequest

858

xmlns="http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1"

859

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

860

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

861

soap:mustUnderstand="1">

862

<RefToMessagelds groupId="mid://20040202.103832@oasis-open.org/">

863

<SequenceNumRange from="0" to="5"/>

864

<SequenceNumRange from="15" to="20"/>

865

</RefToMessagelds>

866

<RefToMessagelds groupId="mid://20040202.103811@oasis-open.org/" />

867

<RefToMessagelds groupId="mid://20040202.103807@oasis-open.org/">

868

<SequenceNumRange from="713" to="6150"/>

869

</RefToMessagelds>

870

</PollRequest>

871

### 872 (1) replyTo attribute

873

This attribute, of type URI, MAY be included by the sending RMP. If present, then the receiver

874

MUST send the RM-Reply in an underlying request to the value of the URI. If not present, the RM-

875

Reply MUST be sent back in the underlying response of the Poll request itself.

### 876 4.3.1 RefToMessagelds Element

877

A sender MUST include the RefToMessagelds element for PollRequest message. This element is

878

to be used to specify RM-Reply to be returned. This element MUST have one groupId attribute

879

and MAY contain zero or more SequenceNumRange element as follows:

880

- a **groupId** attribute

881

- zero or more **SequenceNumRange** element

882

Table 10 RefToMessagelds Element

Cardinality	1 or more
Value	None

Cardinality	1 or more
Attributes	groupId (URI)
Child elements	SequenceNumRange

883 When this RefToMessageIds element has a groupId attribute, but doesn't have  
884 SequenceNumRange element, the receiver MUST send back all RM-Replies for the messages  
885 received in the MessageId. When the RefToMessageIds element has a groupId attribute and  
886 SequenceNumRange element(s), the receiver MUST return RM-Reply for messages received that  
887 were specified by the combination of groupId of RefToMessageIds and SequenceNumRange  
888 element(s). When sender RMP requests multiple RM-Replies with different groupId value in one  
889 PollRequest Message, it MUST include RefToMessageIds element for each groupId.

## 890 (1) groupId attribute

891 The RefToMessageIds element MUST include one or more groupId attribute(s). The groupId  
892 attribute is to be used to specify the groupId for Acknowledgment message to be returned. The  
893 syntax of this attribute is URI, as defined in [RFC2396].

### 894 4.3.1.1 SequenceNumRange element

895 The sender MUST include the SequenceNumRange element when it specifies messages in a  
896 group to be acknowledged. If present, attributes of this element MUST contain the value of the  
897 SequenceNum of the message. This element MUST contain the following two attributes:

- 898 • a **from** attribute
- 899 • a **to** attribute

900

Table 11 SequenceNumRange Element

Cardinality	0 or more
Value	None
Attributes	from (unsignedLong) to (unsignedLong)
Child elements	None

## 901 (1) from attribute

902 A sender MUST include the from attribute in the SequenceNumRange element. This attribute is to  
903 be used to specify the smallest SequenceNum of the message range. The value of this attribute  
904 MUST be equal or smaller than the value of to attribute. It MUST be the same with the value of  
905 the to attribute to specify only one message. The value of this attribute is unsignedLong.

## 906 (2) to attribute

907 A sender MUST include the to attribute in the SequenceNumRange element. This attribute is to  
908 be used to specify the largest SequenceNum of the message range. The value of this attribute  
909 MUST be equal or larger than the value of from attribute. It MUST be the same with the value of  
910 the from attribute to specify only one message. The value of this attribute is unsignedLong.

## 911 4.4 Response Element

912 A receiver MUST include the Response element to indicate Acknowledgment Message for  
913 Reliable Messages and indications of Reliable Messaging Fault Messages. This element includes  
914 the following attributes:

- 915 • SOAP **mustUnderstand** attribute with a value of "1"
- 916 • a **ReplyPattern** attribute, which defaults to the value "Response"

917 Response element MUST include at least one of the following child elements:

- 918 • zero or more **NonSequenceReply** element
- 919 • zero or more **SequenceReplies** element

920 When the response is using the callback reply pattern, if the reply and the new request share a  
921 common destination URI, a Response element can coexist with a Request element, enabling the  
922 combination of an Acknowledgment message with the business response to the original  
923 message. This coexistence also enables a receiver sending another independent message to the  
924 sender with an Acknowledgment message (e.g., to reduce network traffic).

925 Table 12 Response Element

Cardinality	0 or 1
Value	None
Attributes	MustUnderstand (boolean) replyPattern (string)
Child elements	NonSequenceReply SequenceReplies

926

927 Example 8 shows an example of the Response element.

928

### 929 Example 8 Response Element

```
930 <Response  
931   xmlns="http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1"  
932   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
933   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
934   soap:mustUnderstand="1" replyPattern="Callback">  
935     <NonSequenceReply groupId="mid://20040202.103832@oasis-open.org" />  
936     <NonSequenceReply groupId="mid://20040202.103811@oasis-open.org"  
937       fault="wsrm:PermanentProcessingFailure" />  
938     <SequenceReplies groupId="mid://20040202.103807@oasis-open.org/">  
939       <ReplyRange from="1" to="4" />  
940       <ReplyRange from="5" to="5" fault="wsrm:InvalidRequest" />
```

```

941     <ReplyRange from="6" to="42" />
942   </SequenceReplies>
943 </Response>
944

```

### 945 (1) replyPattern attribute

946 If the response is being returned as a result of a Poll Message Reply Pattern, this attribute must  
 947 have the value "Poll".

948 If the response is being returned using the Callback Reply Pattern, this attribute must have the  
 949 value "Callback".

950 If the response is being returned using the Response Reply Pattern, this attribute indicate the  
 951 "Response" value. In the case of a response returned using the Response Reply Pattern, the  
 952 following restrictions apply:

- 953 • If the group does not use sequence numbers, the first element of the response must  
 954 be a NonSequenceReply element containing the groupId which is the globally unique  
 955 message identifier for the Reliable Messaging Request.
- 956 • If the group uses sequence numbering, the first element of the response must be a  
 957 SequenceReplies element, with its groupId equal to that of the request, and with its  
 958 first Range element having its from and to attributes both equal to the sequence  
 959 number in the request.

## 960 4.4.1 NonSequenceReply Element

961 An acknowledgment or an Reliable Messaging Fault indication for a message which does not  
 962 have a sequence number in its MessageId element MUST include a NonSequenceReply element.

963 This element MUST contain the value of the groupId attribute for the message the reply pertains  
 964 to. If the reply is an acknowledgment of delivery, the Receiving RMP MUST NOT include the fault  
 965 attribute. If the reply is an indication of an Reliable Messaging Fault, the Receiving RMP MUST  
 966 include the fault attribute, and its value denotes the fault condition which was encountered.

967 Table 13 NonSequenceReply Element

Cardinality	0 or more
Value	RFC2396
Attributes	groupId (URI) fault (Cardinality 0 or 1)
Child elements	None

968

### 969 (1) groupId attribute

970 This groupId attribute is to be used to specify the groupId of message (which did not have a  
 971 sequence number in its MessageId) to be acknowledged, or to have a fault indicated. The syntax  
 972 of this attribute is URI, as defined in [RFC2396].

## 973 4.4.2 SequenceReplies Element

974 A receiver MUST include the SequenceReplies element to Acknowledgment message or to  
975 indicate Reliable Messaging Faults, for messages which include a SequenceNum element in their  
976 MessageId element. This element MUST contain the values of the original MessageIds of the  
977 messages delivered for a group, and for each Fault Code being reported, the MessageIds of  
978 messages which encountered the particular Fault Code.

979

Table 14 MessageReplies Element

Cardinality	0 or more
Value	RFC2396
Attributes	groupId (URI)
Child elements	ReplyRange

### 980 (1) groupId attribute

981 This groupId attribute is to be used to specify the group of message(s) to be acknowledged, or to  
982 have their faults indicated. The syntax of this attribute is URI, as defined in [RFC2396].

## 983 4.4.2.1 ReplyRange Element

984 A receiver MUST include the ReplyRange element in a SequenceReplies element to indicate  
985 sequence numbers which either are being acknowledged (in which case receiving RMP MUST  
986 NOT include the fault attribute) or have encountered a particular fault condition (in which case the  
987 receiving RMP MUST include the fault attribute with that particular RM fault code encountered).

988

Table 15 ReplyRange Element

Cardinality	None
Value	None
Attributes	from (unsigned Long) to (unsigned Long) fault (QName)
Child elements	None

989

### 990 (1) from attribute

991 A receiver MUST include the from attribute in the ReplyRange element. This attribute is to be  
992 used to specify the smallest SequenceNum of the message range. The value of this attribute  
993 MUST be equal or smaller than the value of to attribute. It MUST be the same with the value of  
994 the to attribute to specify only one message. The value of this attribute is unsignedLong.

### 995 (2) to attribute

996 A receiver MUST include the to attribute in the ReplyRange element. This attribute is to be used to  
997 specify the largest SequenceNum of the message range. The value of this attribute MUST be



998 equal or larger than the value of from attribute. It MUST be the same with the value of the from  
999 attribute to specify only one message. The value of this attribute is unsignedLong.

### 1000 **(3) fault attribute**

1001 This attribute is used to indicate a Reliable Messaging Fault code which was encountered while  
1002 processing all of the messages indicated by sequence numbers in the range. The receiving RMP  
1003 MUST NOT include this attribute for a ReplyRange element used for Acknowledgments.

1004

1005

## 1006 **4.5 Fault Codes For Reliable Messaging Failures**

1007 This section describes the protocol specific fault codes that are needed to better describe the  
1008 reason for WS-Reliability protocol processing failures.

1009 We categorize the faults into 2 categories based on whether the fault was generated because  
1010 Reliable Messaging Headers are malformed or invalid due to some runtime processing errors  
1011 encountered by the RMP. The former category is called Invalid Message Format fault set and the  
1012 latter is called Request Processing fault set. They are explained in detail in the following sections.

1013 These protocol specific fault codes are returned by the receiving RMP within the response header  
1014 element. The WS-Reliability protocol does not directly map our Reliable Messaging Faults to the  
1015 SOAP Fault model.

1016 The SOAP Fault model is used for reporting faults due to the request payload, which fits the  
1017 SOAP fault model better. Thus a response may have a SOAP Fault message, but the reason for  
1018 the SOAP fault would be due to problems associated with the WSDL operation message payload.  
1019 (E.g., A problem with the soap:body of a request message or the inability of the receiving RMP to  
1020 return the WSDL response in the soap:body of when using the Response RM-Reply pattern).

### 1021 **Example case 1:**

1022 For WSDL Request/Response operation types, a SOAP Fault can occur for a reliable request  
1023 which was delivered, but then encountered an application level Fault due to something wrong in  
1024 the payload (SOAP Body of request which is not under control of Sending RMP) or application  
1025 processing space outside the realm of the receiving RMP.

1026 That means a Acknowledgment can be delivered on a SOAP Fault.

### 1027 **Example case 2:**

1028 For the Response Reply Pattern, used with WSDL two way operation type, the return message  
1029 could conceivably carry an indication of an RM Fault, which is not itself carried on a SOAP Fault.  
1030 The exact behavior in such a case might be an implementation matter.

1031 A message with an RM Fault indication MUST NOT be delivered by the receiving RMP. If the  
1032 message cannot be delivered due, say an request fault, then there would be no meaningful data  
1033 for the responder to put into the SOAP Body for the WSDL response.

1034 When using the Response RM-Reply pattern, a WSDL operation reply will not always be available  
1035 for the receiving RMP to return with the RM-Response. This will occur when there is a Reliable  
1036 Messaging Fault for the message in the request, or when the message in the request is a  
1037 duplicate of a prior delivered message with Duplicate Elimination in use.

1038 When a receiving RMP cannot return the WSDL operation response for a request using the  
1039 Response Reply Pattern, it MUST return the RM Response in a SOAP Fault message. If the RM  
1040 Fault encountered was due to a problem with the request header element, a SOAP client fault  
1041 MUST be returned. If the RM Fault encountered was due to a problem with processing by the  
1042 receiving RMP (including the inability to return a response due to Duplicate Elimination), a  
1043 soap:server fault must be returned.

1044 The following Fault codes may be carried in a Response element associated with a MessageId.

## 1045 **4.5.1 Invalid Message Format Fault**

1046 These faults are thrown by the receiving RMP when the message format of the Reliable  
1047 Messaging Headers are either invalid or wrong.

1048

**Table 16 Invalid Message Format Fault Code Values**

<b>Local part name</b>	<b>Description and Cause(s)</b>
InvalidRequest	<p>This fault is sent when the Request element is wrong or invalid. Examples are:</p> <ol style="list-style-type: none"> <li>1. When any of the mandatory elements such as MessageId, ExpiryTime, ReplyPattern are missing</li> <li>2. When AckRequested, DuplicateElimination or MessageOrder elements appear twice</li> <li>3. soap:mustUnderstand attribute is missing</li> </ol>
InvalidPollRequest	<p>This fault is sent when the PollRequest element is wrong or invalid. Examples are:</p> <ol style="list-style-type: none"> <li>1. When the required RefToMessageId element is missing</li> <li>2. soap:mustUnderstand attribute is missing</li> </ol>
InvalidMessageId	<p>This fault is sent in any of the following cases:</p> <ol style="list-style-type: none"> <li>1. If groupId attribute (for MessageId or RefToMessageIds ) doesn't exist, or if exists, and the value is wrong or invalid.</li> <li>2. If number attribute in SequenceNum element doesn't exist, or if exist, the value is invalid or wrong.</li> <li>3. Attributes (from and to) of SequenceNumRange doesn't exist, or if exists, the values are invalid or wrong.</li> </ol>
InvalidMessageParameters	<p>This fault is sent for any of these cases:</p> <ol style="list-style-type: none"> <li>1. groupExpiryTime is wrong or invalid</li> <li>2. groupMaxIdleDuration is wrong or invalid</li> <li>3. when both group parameters are present</li> <li>4. when groupExpiryTime decreases for a subsequent messages. in an ordered group</li> <li>5. If the status attribute of SequenceNum element exist and is not one of allowed {begin continue end} value.</li> </ol>
InvalidReplyPattern	<p>This fault is sent if the ReplyPattern format is wrong or invalid or when the replyTo attribute is missing for the Callback pattern.</p>

InvalidExpiryTime	This fault is sent if the ExpiryTime format is wrong or invalid.
-------------------	--

1050

## 1051 4.5.2 Message Processing Failure Faults

1052 These faults are thrown by the receiving RMP when there is an error processing a valid Reliable  
1053 Messaging message.

1054 **Table 17 Messaging Processing Failure Fault Code Values**

Local part name	Description and Cause(s)
NonSupportedFeature	This fault is thrown by the receiving RMP when it receives a message with a RM feature that it doesn't support. An example is a RM message with MessageOrder element to a receiving RMP that doesn't support Guaranteed Message Ordering
PermanentProcessingFailure	This fault is sent for permanent/fatal processing failures such as: <ul style="list-style-type: none"> <li>1. Persistence Storage failures</li> <li>2. Message Delivery failures</li> </ul> A PermanentProcessingFailure fault indicates that the failure is fatal and subsequent retries of the same message will also fail.
MessageProcessingFailure	This fault is sent for transient failures such as: <ul style="list-style-type: none"> <li>1. Maximum number of buffered requests exceeded the limit.</li> <li>2. Maximum number of threads reached the limit etc.</li> </ul> A transient fault unlike a permanent fault is a temporary one and MAY succeed in subsequent retries.

1055

1056 **Example 9 Fault Message for Reliable Messaging**

```
1057 <?xml version="1.0" encoding="UTF-8"?>
1058 <soap:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
1059   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
1060   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1061   xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
1062   <soap:Header>
1063     <Response
1064       xmlns="http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1"
1065       soap:mustUnderstand="1" replyPattern="Callback">
1066       <SequenceReplies groupId="mid://20040202.103832@oasis-open.org">
1067         <ReplyRange from="1" to="1" fault="InvalidRequest" />
1068       </SequenceReplies>
1069     </Response>
1070   </soap:Header>
1071   <soap:Body />
1072 </soap:Envelope>
```

1073

1074

1075

---

## 1076 5 Operational Aspects and Semantics

### 1077 5.1 Message Group Life Cycle

#### 1078 5.1.1 Group Termination

1079 Being able to know when a group may be terminated, is essential for efficient management of the  
1080 persistent store of an RMP. As groups may last a long time and their state requires persistence, it  
1081 is important to know when their persistent image can be reclaimed. The termination cases  
1082 described in this section may seem multiple and complex. This plurality results from the flexibility  
1083 given to users in specifying various ways a group can be terminated, which in turn depends on  
1084 application needs. However, in spite of this plurality, the termination logic is straightforward to  
1085 implement and shares the same basic mechanisms across termination cases.

1086 Termination of a group in the sender RMP and in the receiver RMP are two distinct events not  
1087 synchronized by any special message, but instead occurring as the result of rules applying  
1088 separately to the sender and to the receiver. As a consequence, the termination of a group may  
1089 occur at quite different times on the sender and receiver RMPs. However, this lack of  
1090 synchronization allowed by these termination rules is not consequential.

1091 More precisely, there are two distinct steps that an RMP must perform when terminating a group,  
1092 and these may also occur at different times, especially on the receiver side:

- 1093 • **Group Closing:** When a group is closed in the Sender RMP, no new message is  
1094 expected to be sent by the RMP for this group. When a group closes in the receiver  
1095 RMP, no new message is expected to be received for this group anymore. After a  
1096 group is closed, all subsequent messages sent with same group ID would be handled  
1097 as belonging to a new group, unless they are duplicates of previous messages in the  
1098 group, in which case they are treated as duplicates within this group. If a message is  
1099 received after the closing of a group, with same group ID as the closed group, it may  
1100 be considered by the Receiver as belonging to a new group (the Receiver is not  
1101 required to verify that a new group ID value has not already been used in a previous  
1102 group, as this test is impractical).
- 1103 • **Group State Removal:** The state of a group includes group-specific attributes such  
1104 as group status (e.g. "active", "closed"), group ID, current sequence number, as well  
1105 as all received Message Identifiers for the group (e.g. sequence number intervals).  
1106 The state of a group also includes the persistent image of messages of this group  
1107 being currently processed, although the removal of the persistence of messages  
1108 follows its own rules. E.g. The resending mechanism for guaranteed delivery will take  
1109 care of removal of messages on the sender side, once they are acknowledged. State  
1110 removal occurs at the time or after the group is closed. When the state of a group is  
1111 removed, all group attributes are removed, including the past message IDs on receiver  
1112 side. Therefore not duplicate check may be done over past messages of this group.

1113 In all termination cases (t1, t2, t3, t4, t5) described in this section, it is not necessary to remember  
1114 the ExpiryTime of all messages of a group, as only the max(ExpiryTime) of messages received  
1115 for the group is needed. These termination rules apply to both ordered and unordered groups.  
1116 However, these rules do NOT apply to singleton groups, which contain a single message with no  
1117 sequence number.

1118 Assuming the last message of a group is marked with "end" status value, a group is defined as  
1119 being "complete" on the receiver RMP when all the messages from 'start' to 'end' are received.

## 1120 5.1.2 Group Termination Parameters

- 1121 There are two RM Agreement items - GroupExpiryTime and GroupMaxIdleDuration that can be  
1122 used to determine when a group can be terminated. These two items can be considered as  
1123 controlling the persistence of group data.  
1124 To each of these agreement items, correspond respectively the message header attributes:  
1125 groupExpiryTime and groupMaxIdleDuration. The following requirements pertain to these header  
1126 attributes:
- 1127 a) the First message in a group (the one with status=start) MUST be used by the sender to  
1128 indicate that time based group persistence control is in use for the group.
- 1129 • If the first message in the sequence of a group has neither group persistence  
1130 parameter present, the group will be terminated according to condition t4 or t5.
  - 1131 • If the first message has either one of the two group persistence parameter present  
1132 (either groupExpiryTime, or groupMaxIdleDuration) then the group will be subject to  
1133 termination rules t1, t2 or t3 described below.
  - 1134 • A fault MUST be returned if both group persistence parameters are present in any  
1135 request message. An InvalidGroupParameter fault shall be sent in this case..
  - 1136 • If groupExpiryTime is in use, the sender must not send a message in that group with  
1137 an ExpiryTime greater than the groupExpiryTime.
- 1138 b) The group termination parameter which was sent on the first message in the group MUST be  
1139 used on all subsequent messages in that group, and MUST be assigned a value.
- 1140 c) The receiver MUST use the value from the message with the highest sequence number  
1141 received for the group.
- 1142 d) In any subsequent message the parameter which was sent in the first message can be  
1143 changed by sending a new value. A new value for groupMaxIdleDuration can either be increased  
1144 or decreased. The protocol allows change (up or down) of groupExpiryTime, as long as it is never  
1145 less than max(ExpiryTime) of messages received so far for the group.
- 1146 An InvalidMessageParameters Fault MUST be returned if the value of groupExpiryTime is  
1147 decreased to be less than the max(ExpiryTime) of messages received for the group.
- 1148 The receiving RMP MUST update its Group Termination Criteria using parameter values from a  
1149 Reliable Message, even if that request encounters a Reliable Messaging Fault.

## 1150 5.1.3 Termination Rules

### 1151 (1) Termination (t1):

#### 1152 Context:

1153 The group had groupExpiryTime specified.

#### 1154 Receiver side:

1155 Triggering event: groupExpiryTime is over.

1156 The RMP MUST NOT accept any new message for this group and MUST close the group. It is  
1157 RECOMMENDED that its state be removed as soon as possible after this. No duplicate check  
1158 needs to be done against that group ever. If a "late duplicate" arrives, it would never be delivered  
1159 to the next layer, as its ExpiryTime, which is always earlier than groupExpiryTime, would have  
1160 expired.

- 1161 **Sender side:**
- 1162 Triggering event: groupExpiryTime is over.
- 1163 The group MUST be closed, and its state removed from the RMP.
- 1164 **(2) Termination (t2):**
- 1165 **Context:**
- 1166 The group had groupMaxIdleDuration specified.
- 1167 **Receiver side:**
- 1168 Triggering event: groupMaxIdleDuration is over.
- 1169 The group MUST be closed. But unlike (t1), some of its past messages may not have expired yet,  
1170 and therefore their ID still be needed for duplicate checks. If we define max(ExpiryTime) as the  
1171 max of all ExpiryTimes of messages received for a group, an RMP MUST persist the state of a  
1172 group even after closing of the group, at least until max(ExpiryTime) is reached, in case Duplicate  
1173 Elimination is required.
- 1174 **Sender side:**
- 1175 Triggering event: groupMaxIdleDuration is over.
- 1176 The group MUST be closed, and its state removed from the RMP when the time elapsed since the  
1177 last sent message (including retries) exceeds groupMaxIdleDuration.
- 1178 **(3) Termination (t3):**
- 1179 **Subcase t3.1: The group is complete on receiver side.**
- 1180 **Context:**
- 1181 The group had either groupExpiryTime or groupMaxIdleDuration specified. All received messages  
1182 for the group have been delivered (no missing message).
- 1183 **Receiver side:**
- 1184 Triggering event: The RMP receives a status="end" message.
- 1185 The group MUST be closed. However, its state is removed according to (t1) or (t2), depending  
1186 which termination criterion was specified for the group.
- 1187 **Sender side:**
- 1188 Triggering event: The RMP sends a status="end" message.
- 1189 All messages of the group have been sent. If Guaranteed Delivery was required, the group MUST  
1190 be closed and state is removed once all sent messages have either been acknowledged, or their  
1191 delivery failure notified. If no Guaranteed Delivery was required, the group MUST be closed and  
1192 its state may be removed immediately.
- 1193 **Subcase t3.2: The group is not complete on receiver side.**
- 1194 **Context:**



1195 The group had either groupExpiryTime or groupMaxIdleDuration specified. Not all received  
1196 messages for the group have been delivered (some message is missing).

1197 **Receiver Side:**

1198 Triggering event: the RMP receives a status="end" message.

1199 In this case, the group is not yet closed. Indeed, an "end" status only tells that "no greater  
1200 sequence number will ever be received after", but late messages may still arrive for this group.  
1201 Then the Receiver RMP MUST apply termination rules of (t1) or (t2), depending which one of the  
1202 two group termination parameters (i.e. groupExpiryTime or groupMaxIdleDuration) was specified.

1203 **Sender Side:**

1204 Triggering event: the RMP sends a status="end" message.

1205 As all messages for the group have been sent, same rules apply as in t3.1.

## 1206 **(4) Termination (t4):**

### 1207 **Subcase t4.1: The group was complete on receiver side.**

1208 **Context:**

1209 The group had neither groupExpiryTime nor groupMaxIdleDuration specified. All received  
1210 messages for the group have been delivered (no missing message).

1211 **Receiver side:**

1212 Triggering event: the RMP receives a status="end" message.

1213 The group MUST be closed. The time of removal of its state is determined by the max  
1214 (ExpiryTime) received of the Group.

1215 **Sender side:**

1216 Triggering event: the RMP sends a status="end" message.

1217 Same rule applies as in t3.1.

### 1218 **Subcase t4.2: The group was not complete on receiver side.**

1219 **Context:**

1220 The group had neither groupExpiryTime nor groupMaxIdleDuration specified. Not all received  
1221 messages for the group have been delivered (some message is missing).

1222 **Receiver side:**

1223 Triggering event: The RMP receives a status="end" message.

1224 In this subcase, the RMP should keep the group processing active: this event, by itself, does not  
1225 cause the closing of the group.

1226

1227 **Sender side:**

1228 Triggering event: the RMP sends a status="end" message.

1229 Same rule applies as in t3.1.

## 1230 (5) Termination (t5):

### 1231 Context:

1232 The group is under Guaranteed Message Ordering reliability requirement. Not all received  
1233 messages for the group have been delivered (some message is missing).

### 1234 Receiving side:

1235 Triggering event: In an ordered group, a message expires before delivery in out-of-order  
1236 sequence.

1237 The group MUST be closed.

1238 State is removed according to the following rule:

- 1239 • If the group does not have termination parameter, then it will be removed when the  
1240 max(ExpiryTime) received of the group passes.
- 1241 • If the group uses groupExpiryTime, then removal criteria as defined in t1 will apply.
- 1242 • If the group uses groupMaxIdleDuration, then removal criteria as defined in t2 will  
1243 apply.

### 1244 Sender Side:

1245 Triggering event: In an ordered group, a non-acknowledged message expires.

1246 State is removed according to the following rule:

- 1247 • If the group does not have termination parameter, then it will be removed when the  
1248 max(ExpiryTime) sent for the group passes.
- 1249 • If the group uses groupExpiryTime, then removal criteria as defined in t1 will apply.
- 1250 • If the group uses groupMaxIdleDuration, then removal criteria as defined in t2 will  
1251 apply.

## 1252 5.2 WSDL Operation Type

1253 This specification supports Reliable Messaging capabilities for WSDL 1.1 [WSDL 1.1] One-way  
1254 and Request-response operation types only. While a Request-reponse operation can use any of  
1255 the three RM-Reply patterns to receive acknowledgments or faults, an One-way operation can  
1256 only use either Callback or Poll RM-Reply pattern. See the table below for a complete support  
1257 matrix:

1258

**Table 18 WSDL operation types**

<b><i>WSDL operation type</i></b>	<b><i>Response RM-Reply pattern</i></b>	<b><i>Callback RM-Reply pattern</i></b>	<b><i>Poll RM-Reply pattern</i></b>
Request/Response WSDL operation type*	Supported	Supported	Supported
One-way WSDL operation type	Disallowed **	Supported	Supported

1259 \* The current version of the WS-Reliability protocol does not support reliability of WSDL response  
1260 messages (the "output" messages in WSDL operations). It only supports reliability of the WSDL  
1261 request ("input" messages).

1262 \*\* WS-I BP 1.0 disallows sending a SOAP envelope in HTTP response, so an RMP is not required  
1263 to support this. However, this specification does not require an RMP to enforce this restriction (i.e.  
1264 WS-I BP compliance). The receiver can do whatever the header asks for.

1265 While the specification doesn't prohibit using Callback or Poll RM-Reply patterns to receive  
1266 acknowledgments or faults for a Request-response operation, it is encouraged to use Response  
1267 RM-Reply pattern for such operations as the acknowledgment or the fault can be sent on the  
1268 same response itself thus saving extra round trips.

## 1269 **5.3 Poll Reply Pattern Semantics and Usage**

1270 Guaranteed Delivery will be most commonly used for a one-way message as the Sender know the  
1271 status of the message delivery otherwise.

1272 So the most common use case is to use AckRequested with Callback RM-Reply pattern so that  
1273 the Sender can receive the acknowledgment or a fault on a listener at its end. However this  
1274 pattern doesn't help when the Sender is within a firewall, as one cannot receive requests without  
1275 opening a firewall, thus causing security lapses.

1276 An alternate solution is for the Sender to ask the Receiver for the receiving status of the message  
1277 it has sent earlier on a different channel. Such a pattern is called the poll RM-Reply pattern. The  
1278 Sender sends a Poll request for a message it is to inquire and the Receiver sends a Poll response  
1279 with the acknowledgment. The Sender can also batch multiple Poll requests for an efficient use.  
1280 Receiver in such case will send acknowledgments for those messages it has received. If a Poll  
1281 request is partially or completely invalid or wrong, then the Receiver sends either a  
1282 InvalidPollRequest or InvalidMessageFault back.

1283 Also, a RM Poll response MUST NOT be piggybacked on a different RM Poll request.

## 1284 **5.4 Attachments**

1285 When this spec is used with W3C note SOAP messages with Attachments specification [SOAP  
1286 with Attachments], the following rules MUST be met:

- 1287 1) The first MIME part MUST include whole SOAP envelope with WS-Reliability header elements.
- 1288 2) The charset of the Content-Header of the first MIME part MUST be either UTF-8 or UTF-16.
- 1289 3) Zero or more additional MIME parts MAY be included in a reliable message.
- 1290 4) The receiver RMP MUST deliver all MIME parts in a Reliable Message to the receiving  
1291 application.

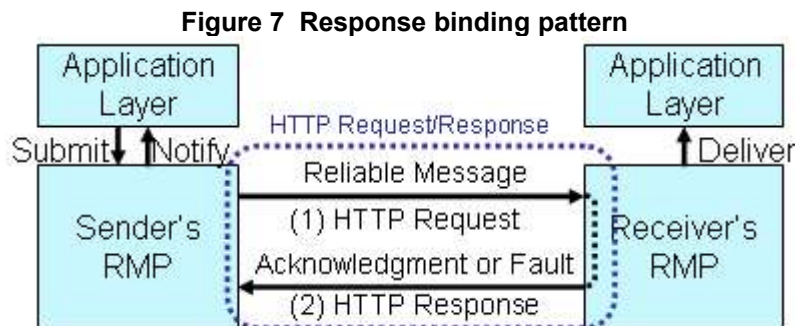
## 1292 6 HTTP Binding

1293 This section describes the three binding pattern - "Response", "Callback", and "Poll" binding  
1294 pattern for HTTP. These binding pattern is identified by the value of ReplyPattern element (See  
1295 Section 4.2.3 for detail). This specification expects that the transport layer will not deliver a  
1296 corrupted message to the reliability layer. When a request message contains AckRequested  
1297 element, upon receipt of a reliable message, the receiver's RMP MUST send a reply. This reply  
1298 MUST be either an Acknowledgment message or a Fault message. This reply MUST be sent by  
1299 specified binding pattern in the ReplyPattern element of the request message.

### 1300 6.1 Reliable Messaging with "Response" binding pattern

1301 The Reliable Messaging Acknowledgment or Fault message MUST be sent back on the same  
1302 HTTP connection with the HTTP Request that the sender initiated to send the Message. This is  
1303 illustrated in Figure 7. Both Acknowledgment Message and Fault message MUST be sent back to  
1304 the sender on the same HTTP connection the sender sent a message.

1305



- 1307 1) The sender initiates an HTTP connection, and sends a Message using the HTTP POST  
1308 Request. Example 10 is an example of such a message.
- 1309 2) The receiver sends back an Acknowledgment message to the sender on the same connection,  
1310 with the HTTP response.

#### 1311 Example 10 Request Message with Response binding pattern

```
1312 POST /abc/servlet/wsrListener HTTP/1.0
1313 Content-Type: text/xml; charset=utf-8
1314 Host: 192.168.183.100
1315 SOAPAction: ""
1316 Content-Length: 1214
1317
1318 <?xml version="1.0" encoding="UTF-8"?>
1319 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1320   <soap:Header>
1321     <Request
1322       xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"
```

```

1323     soap:mustUnderstand="1">
1324         <MessageId groupId="mid://20040202.103832@oasis-open.org/">
1325             <SequenceNum number="0" status="Start"
1326                 groupExpiryTime="2005-02-02T03:00:33-31:00" />
1327         </MessageId>
1328         <ExpiryTime>2004-09-07T03:01:03-03:50</ExpiryTime>
1329         <ReplyPattern replyTo="http://www.oasis-open.org/">Response</ReplyPattern>
1330         <AckRequested/>
1331         <DuplicateElimination/>
1332         <MessageOrder/>
1333     </Request>
1334 </soap:Header>
1335 <soap:Body>
1336     <Request xmlns="http://wsr-example.org/">Request Message</Request>
1337 </soap:Body>
1338 </soap:Envelope>

```

1339

#### Example 11 Acknowledgment Message with Response binding pattern

```

1340
1341 HTTP/1.0 200 OK
1342 Server: WS-ReliabilityServer
1343 Date: Mon, 02 Feb 2004 10:38:32 GMT
1344 Content-Language: en
1345 Content-Type: text/xml; charset=utf-8
1346 Content-Length: 924
1347
1348 <?xml version="1.0" encoding="UTF-8"?>
1349 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1350     <soap:Header>
1351         <Response
1352             xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"
1353             soap:mustUnderstand="1" replyPattern="Response">
1354             <SequenceReplies groupId="mid://20040202.103832@oasis-open.org/">
1355                 <ReplyRange from="0" to="0"/>
1356             </SequenceReplies>
1357         </Response>
1358     </soap:Header>

```

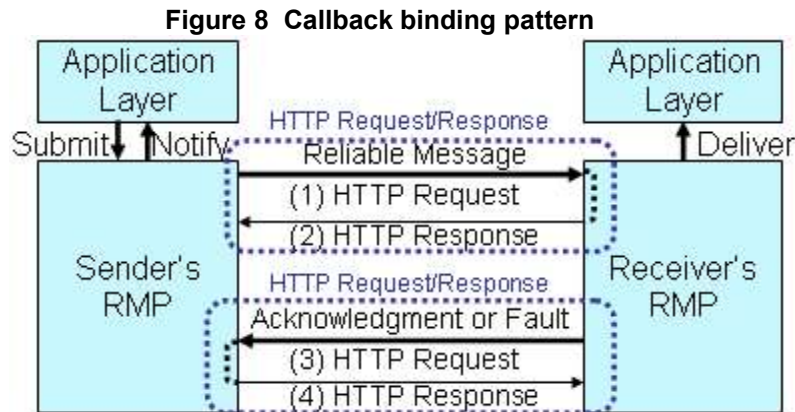
1359 <soap:Body />  
 1360 </soap:Envelope>

1361

## 1362 6.2 Reliable Messaging with “Callback” binding pattern

1363 The Reliable Messaging Acknowledgment or Fault message MUST be sent back on a different  
 1364 HTTP connection from the HTTP connection that the sender initiated to send the message. The  
 1365 direction of the HTTP connection that receiver initiates is from the receiver to the sender. This is  
 1366 illustrated in Figure 8.

1367



1369 (1) The sender initiates a HTTP connection, and sends a Message using HTTP POST Request.  
 1370 Example 12 is an example of this message.

1371 (2) The HTTP response to the (1) has no content. Example 13 is an example of this HTTP  
 1372 response.

1373 (3) The Acknowledgment Message is sent with another HTTP connection from the receiver to the  
 1374 sender. Example 14 is an example of this message.

1375 (4) The HTTP response for (3) has no content. Example 13 is an example for this HTTP  
 1376 Response.

1377

### 1378 Example 12 Request Message with Callback binding pattern

```
1379 POST /abc/servlet/wsrListener HTTP/1.0
1380 Content-Type: text/xml; charset=utf-8
1381 Host: 192.168.183.100
1382 SOAPAction: ""
1383 Content-Length: 1214
1384
1385 <?xml version="1.0" encoding="UTF-8"?>
1386 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1387   <soap:Header>
1388     <Request
```

```

1389     xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"
1390     soap:mustUnderstand="1">
1391         <MessageId groupId="mid://20040202.103832@oasis-open.org/">
1392             <SequenceNum number="0" status="Start"
1393                 groupExpiryTime="2005-02-02T03:00:33-31:00" />
1394         </MessageId>
1395         <ExpiryTime>2004-09-07T03:01:03-03:50</ExpiryTime>
1396         <ReplyPattern replyTo="http://www.oasis-open.org/">Callback</ReplyPattern>
1397         <AckRequested/>
1398         <DuplicateElimination/>
1399         <MessageOrder/>
1400     </Request>
1401 </soap:Header>
1402 <soap:Body>
1403     <Request xmlns="http://wsr-example.org/">Request Message</Request>
1404 </soap:Body>
1405 </soap:Envelope>

```

1406

1407 **Example 13 HTTP response with no content**

```

1408 HTTP/1.0 200 OK
1409 Server: WS-ReliabilityServer
1410 Date: Mon, 02 Feb 2004 10:38:32 GMT
1411 Content-Language: en
1412 Content-Type: text/xml; charset=utf-8
1413 Content-Length: 184

```

1414

1415 **Example 14 Acknowledgment Message with Callback binding pattern**

```

1416 POST /xyz/servlet/wsrListener HTTP/1.0
1417 Content-Type: text/xml; charset=utf-8
1418 Host: 192.168.183.200
1419 SOAPAction: ""
1420 Content-Length: 1024
1421
1422 <?xml version="1.0" encoding="UTF-8"?>
1423 <soap:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
1424     xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

```

```

1425 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1426 xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
1427 <soap:Header>
1428   <Response
1429     xmlns="http://www.oasis-open.org/committees/wsrm/schema/1.1/SOAP1.1"
1430     soap:mustUnderstand="1" replyPattern="Callback">
1431     <SequenceReplies groupId="mid://20040202.103832@oasis-open.org/">
1432       <ReplyRange from="0" to="0"/>
1433     </SequenceReplies >
1434   </Response>
1435 </soap:Header>
1436 <soap:Body />
1437 </soap:Envelope>

```

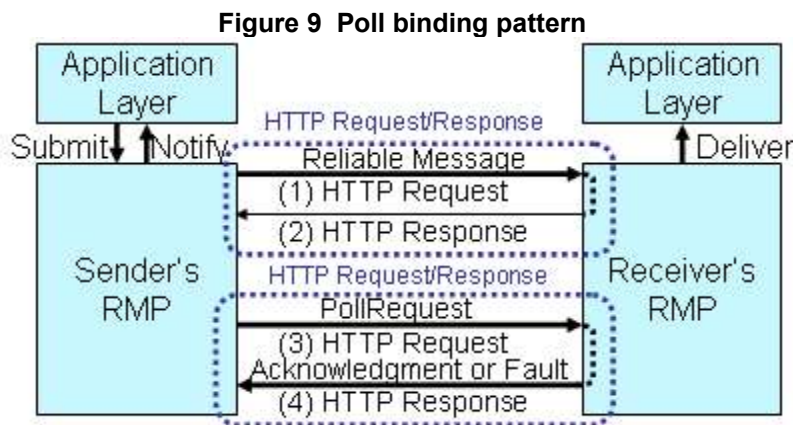
1438

### 1439 6.3 Reliable Messaging with "Poll" binding pattern

1440 The Reliable Messaging Acknowledgment message MAY also be sent back on a different HTTP  
 1441 connection from the HTTP connection used to send the message being acknowledged. This is  
 1442 illustrated in Figure 9.

1443

1444



1446 (1) The sender initiates a HTTP connection, and sends a Message using HTTP POST Request.

1447 (2) The HTTP response to the (1) has no content. Example 13 is an example of this HTTP  
 1448 response.

1449 (3) The sender initiates a HTTP connection, and sends a PollRequest message with HTTP POST  
 1450 Request. Example 15 is an example of this message.

1451 (4) The HTTP response for (3) includes Acknowledgment message and/or Reliable Messaging  
 1452 Fault. Example 16 is an example for this message.



1453

1454 **Example 15 PollRequest message with Poll binding pattern**

1455 POST /abc/servlet/wsrListener HTTP/1.0

1456 Content-Type: text/xml; charset=utf-8

1457 Host: 192.168.183.100

1458 SOAPAction: ""

1459 Content-Length: 1021

1460

1461 <?xml version="1.0" encoding="UTF-8"?>

1462 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >

1463 <soap:Header>

1464 <PollRequest

1465 xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"

1466 soap:mustUnderstand="1">

1467 <RefToMessageIds groupId="mid://20040202.103832@oasis-open.org/">

1468 <SequenceNumberRange from="0" to="20"/>

1469 </RefToMessageIds>

1470 </PollRequest>

1471 </soap:Header>

1472 <soap:Body />

1473 </soap:Envelope>

1474

1475 **Example 16 Acknowledgment message with Poll binding pattern**

1476 HTTP/1.0 200 OK

1477 Server: WS-ReliabilityServer

1478 Date: Mon, 02 Feb 2004 10:38:32 GMT

1479 Content-Language: en

1480 Content-Type: text/xml; charset=utf-8

1481 Content-Length: 924

1482

1483 <?xml version="1.0" encoding="UTF-8"?>

1484 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >

1485 <soap:Header>

1486 <Response

1487 xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"

1488 soap:mustUnderstand="1" replyPattern="Poll" >

1489 <SequenceReplies groupId="mid://20040202.103832@oasis-open.org/">

```
1490     <ReplyRange from="0" to="14"/>
1491     <ReplyRange from="16" to="20"/>
1492   </SequenceReplies>
1493 </Response>
1494 </soap:Header>
1495 <soap:Body />
1496 </soap:Envelope>
```

1497

1498

1499

1500

---

## 1501 7 Conformance

1502 In order to be conform to this specification, an implementation must satisfy all the following  
1503 conditions:

- 1504 • It complies with the following interpretation of the keywords OPTIONAL and MAY:  
1505 When these keywords apply to the behavior of the implementation, the implementation  
1506 is free to support these behaviors or not, as stated in [RFC2119].
- 1507 • If it has implemented optional features and/or behavior defined in this specification, it  
1508 MUST be capable of interoperating with another implementation that has not  
1509 implemented the optional syntax, features and/or behavior. It MUST be capable of  
1510 processing the prescribed failure mechanism for those optional features it has chosen  
1511 to implement.
- 1512 • If it has chosen NOT to implement optional features, it is capable of interoperating with  
1513 another implementation that has chosen to implement these. It MUST be capable of  
1514 generating the prescribed failure mechanism for those optional features it has chosen  
1515 to NOT implement.

1516

---

## 1517 8 References

### 1518 8.1 Normative References

1519 [RFC1738] "Uniform Resource Locators (URL)", T. Berners-Lee et al, RFC 1738, IESG and IETF,  
1520 December 1994. Available at

1521 <http://www.ietf.org/rfc/rfc1738.txt>

1522

1523 [RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119,  
1524 Bradner, S., IESG and IETF, March 1997. Available at

1525 <http://www.ietf.org/rfc/rfc2119.txt>

1526

1527 [RFC2392] "Content-ID and Message-ID Uniform Resource Locators", RFC2392, E. Levinson,  
1528 IESG and IETF, August 1998. Available at

1529 <http://www.ietf.org/rfc/rfc2392.txt>

1530

1531 [RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, Tim Berners-Lee et  
1532 al, IESG and IETF, August 1998. Available at

1533 <http://www.ietf.org/rfc/rfc2396.txt>

1534

1535 [RFC2616] "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, R. Fielding et al, IESG and  
1536 IETF, June 1999. Available at

1537 <http://www.ietf.org/rfc/rfc2616.txt>

1538

1539 [RFC2822] "Internet Message Format", RFC 2822, P. Resnick, Editor, IESG and IETF, April 2001.  
1540 Available at

1541 <http://www.ietf.org/rfc/rfc2822.txt>

1542

1543 [SOAP 1.2] "SOAP 1.2 Part 1: Messaging Framework", Martin Gudgin, Marc Hadley, Noah  
1544 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, eds., W3C Recommendation, 24  
1545 June 2003. Available at

1546 <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>

1547

1548 [XML] "Extensible Markup Language (XML) 1.0, Second Edition", Tim Bray et al, eds., W3C  
1549 Recommendation, 6 October 2000. Available at

1550 <http://www.w3.org/TR/2000/REC-xml-20001006/>

1551

1552 [XML Namespaces] "Namespaces in XML", Tim Bray et al., eds., W3C Recommendation, 14  
1553 January 1999. Available at  
1554 <http://www.w3.org/TR/1999/REC-xml-names-19990114/>  
1555  
1556 [XML Schema] "XML Schema Part 2: Datatypes", Paul V. Biron and Ashok Malhotra, eds. W3C  
1557 Recommendation, 2 May 2001. Available at  
1558 <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>  
1559  
1560 [WS-I BP1.0] "Basic Profile Version 1.0a", Keith Ballinger, David Ehnebuske, Martin Gudgin, Mark  
1561 Nottingham, Prasad Yendluri, eds., WS-I specification, 8 August 2003. Available at  
1562 <http://www.ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.html>  
1563

## 1564 **8.2 Non-normative References**

1565 [ebMS] "Message Service Specification Version 2.0", OASIS ebXML Messaging Services  
1566 Technical Committee, OASIS Standard, 1 April 2002. Available at  
1567 <http://www.ebxml.org/specs/ebMS2.pdf>  
1568  
1569 [SOAP 1.1] "Simple Object Access Protocol (SOAP) 1.1", Don Box et al, W3C Note, 8 May, 2000.  
1570 Available at  
1571 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>  
1572  
1573 [WSDL1.1] "Web Services Description Language (WSDL) 1.1", Erik Christensen, Francisco  
1574 Curbera, Greg Meredith, Sanjiva Weerawarana, eds., W3C Note, 15 March 2001. Available at  
1575 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>  
1576  
1577 [WSS] "OASIS Web Services Security TC", documentation in progress, OASIS Technical  
1578 Committee. Committee home page at  
1579 <http://www.oasis-open.org/committees/wss/>  
1580  
1581 [XML Schema - Part 1] "XML Schema Part 1: Structures", Henry S. Thompson, David Beech,  
1582 Murray Maloney, Noah Mendelsohn, eds., W3C Recommendation, 2 May 2001. Available at  
1583 <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>  
1584  
1585 [SOAP with Attachments] "SOAP Messages with Attachments", John J. Barton, Satish Thatte,  
1586 Henrik Frystyk Nielsen, W3C Note, 11 December 2000, Available at  
1587 <http://www.w3.org/TR/SOAP-attachments>  
1588

---

1589 **Appendix A. WS-Reliability Features, Properties**  
1590 **and Compositor (Normative and Optional)**

1591 **I. Introduction**

1592 Users of a Web Service will need to be aware of the reliability capabilities (RM capabilities) that are  
1593 supported/required by the service. One practical location to advertise these capabilities is in the  
1594 service description (WSDL document), which allows for publishing both abstract service  
1595 definitions as well as concrete protocol details (bindings). This allows clients (or other Web  
1596 services) to easily obtain information about specific capabilities such as guaranteed delivery,  
1597 duplicate elimination, message ordering, and various reply patterns of a specific Web service,  
1598 before calling the service. While bundling reliability capabilities with the service description may  
1599 not be desirable in all cases, it is expected that this convenient approach will often be appropriate.  
1600 The WSDL annotation mechanism described here is a flexible way to add such capability  
1601 assertions.

1602 WS-Reliability uses the WSDL 1.1 extensibility points to define an extensible framework  
1603 consisting of features, properties and compositors to address the needs of a reliable Web service  
1604 to advertise its capabilities, and composability of those capabilities.

1605 The following extensibility elements relevant to RM capabilities are used:

- 1606 • feature - abstract RM capability or assertion associated with WSDL elements.
- 1607 • property - an assertion or constraint on an atomic RM capability and its value(s)  
1608 associated with WSDL elements.
- 1609 • compositor - specify how features and properties are combined.

1610 An annotation composed with the above extensibility elements will specify the reliability features  
1611 and properties associated with specific WSDL constructs. Features and properties represent  
1612 reliability capabilities and compositors specify how these capabilities are composed.

1613 This would allow, for example, a Web service description to advertise the fact that clients invoking  
1614 the service must use duplicate elimination or message ordering.

1615 **I.A Notational Convention**

1616 This specification uses the following namespace prefixes:

<i>Prefix</i>	<i>Namespace</i>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
wsdl11	<a href="http://schemas.xmlsoap.org/wsdl/">http://schemas.xmlsoap.org/wsdl/</a>
fnp	<a href="http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/">http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/</a>
wsrn	<a href="http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/">http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/</a>

1617

1618 The choice of any namespace prefix is arbitrary and not semantically significant.

1619

## 1620 I.B Conformance

1621 Implementations of WS-Reliability are expected, though not required, to understand the WSDL  
1622 extensibility points defined in this section.

1623 Understanding of these extensibility points promotes interoperability. When a WSDL document  
1624 contains these extensibility points, it is through these extensibility points that a service advertises  
1625 its supported and required features. Therefore it is RECOMMENDED that implementations  
1626 recognize, understand and support these extensibility points.

1627 It is also possible for services to advertise features through other channels (such as UDDI) in  
1628 addition to these extensibility point.

1629

## 1630 II. WSDL Extensibility Elements

### 1631 II.A Compositor

1632 The compositor semantics describe how features and properties are composed for the enclosing  
1633 component (or WSDL 1.1 element). The compositor's semantics determine whether the usage of  
1634 composed elements by a client to the service, is required or optional. The RM capabilities  
1635 represented by these elements must all be supported by the Service. A compositor element can  
1636 occur as a child element of wsdl11:portType, wsdl11:operation (which may itself be a child of  
1637 wsdl11:portType or wsdl11:binding), wsdl11:binding, wsdl11:service and wsdl11:port. The  
1638 compositor element utilizes the extensibility defined by WSDL 1.1. A compositor element specifies  
1639 the semantics for combining its children elements. These children elements can be additional  
1640 compositor, features, properties, or extensibility element(s).

1641 A compositor element is expressed by the following pseudo-syntax:

```
1642 <fnp:compositor uri="..." name="NCName"?>  
1643   [fnp:feature/> | <fnp:property/> | <fnp:compositor/> |  
1644   <extensibility-element/>]+  
1645 </fnp:compositor>
```

1646 The uri attribute of the compositor specifies its semantics. Four different compositors (URIs) and  
1647 their capability-related semantics are described below. It is possible to provide additional  
1648 compositors by using other URIs. The ability to define additional compositors and the existence of  
1649 extensibility points (represented by "<extensibility-element>") make the framework extensible. The  
1650 optional name attribute identifies the compositor. An element built with such compositors  
1651 represents an RM capability.

1652       • **all:** this compositor specifies that a service invocation MUST comply with all the  
1653       children elements (representing RM capability assertions). This compositor is identified  
1654       by using the URI:

1655       "<http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositors/all>"

1656       • **choice:** this compositor specifies that a service invocation MUST comply with exactly  
1657       one of the possibly many children elements (representing RM capability assertions).  
1658       This compositor is identified by using the URI:

1659       "<http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositors/choice>"

1660           • **one-or-more:** this compositor specifies that a service invocation MUST comply with at  
1661           least one of the possibly many children elements (representing RM capability  
1662           assertions). This compositor is identified by using the URI:

1663           "<http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositors/one-or-more>"

1664           • **zero-or-more:** this compositor specifies that a service invocation MAY comply with  
1665           one or more of the children elements (representing RM capability assertions). This  
1666           compositor is identified by using the URI:

1667           "<http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositors/zero-or-more>"

1668   Examples for each compositor are provided in Section VII below.

1669   Compositors specified at different WSDL components are implicitly aggregated using the 'all'  
1670   compositor at the dependent WSDL component. Consider the example below,

```
1671 <wsdl11:definitions>
1672   ...
1673   <wsdl11:portType name="myPortType">
1674     <fnp:compositor uri="..." name="A">
1675       ...
1676     </fnp:compositor>
1677     ...
1678   </wsdl11:portType>
1679
1680   <wsdl11:binding name="myBinding" type="myPortType">
1681     <fnp:compositor uri="..." name="B">
1682       ...
1683     </fnp:compositor>
1684     ...
1685   </wsdl11:binding>
1686   <wsdl11:service name="myService">
1687     <wsdl11:port name="myPort" binding="myBinding">
1688       ...
1689     </wsdl11:port>
1690   </wsdl11:service>
1691 </wsdl11:definitions>
```

1692   Compositor specified at the wsdl11:portType "myPortType" and compositor specified at  
1693   wsdl11:binding "myBinding" are aggregated at the dependent wsdl11:port "myPort" using the 'all'  
1694   compositor. I.e., the equivalent compositor at "myPort" is,

```
1695 <fnp:compositor
1696 uri="http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositor/all">
1697   <fnp:compositor uri="..." name="A">
1698     </fnp:compositor>
```



1699 <fnp:compositor uri="..." name="B">

1700 ...

1701 </fnp:compositor>

1702 </fnp:compositor>

1703

## 1704 **II.B Feature**

1705 A feature describes an abstract RM capability or assertion associated with a WSDL element. A  
1706 feature can occur only as a child of a compositor.

1707 Whether the usage of a feature is required or not is defined by the enclosing compositor(s). A  
1708 feature is identified by a URI. Recognizing the URI of a feature is considered to be equivalent to  
1709 understanding the feature identified by that URI.

1710 A feature element is expressed by the following pseudo-syntax:

1711 <fnp:feature uri="...">

1712 [**<fnp:compositor/>** | **<extensibility-element/>**]\*

1713 </fnp:feature>

1714

## 1715 **II.C Property**

1716 A property is identified by a QName. A property is an assertion or constraint on a specific RM  
1717 capability and its value(s) associated with WSDL elements.

1718 Typically properties are associated with a feature (but are not required to) and are described in a  
1719 feature specification. The QName identifier of a property uniquely identifies the property.

1720 Recognizing the property QName identifier is considered to be equivalent to understanding the  
1721 semantics associated with that property. The property QName identifier typically points a global  
1722 XML Schema element declaration. A property specification typically specifies the schema that  
1723 contains this global element declaration. A constraint on the set of values that a property can have  
1724 is specified by a QName that identifies a XML Schema type.

1725 <fnp:property name="xs:QName">

1726 [**<fnp:value>**xs:anyType**</fnp:value>** |

1727 **<fnp:constraint>**xs:QName**</fnp:constraint>**]

1728 [**<extensibility-element/>**]\*

1729 </fnp:property>

1730

## 1731 **III. WS-Reliability Feature**

1732 The WS-Reliability feature is identified by the URI

1733 "<http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/>"

1734 This feature URI identifies the WS-Reliability specification. Understanding this URI implies  
1735 understanding the WS-Reliability specification.

1736

## 1737 **IV. WS-Reliability Properties**

1738 This section identifies properties for the WS-Reliability specification. Typically these properties  
1739 would be scoped within the feature identified by the URI

1740 "http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/"

1741

### 1742 **IV.A. Guaranteed Delivery Property**

1743 This property is identified by the QName "wsrm:GuaranteedDelivery" and corresponds to the  
1744 semantics specified by the WS-Reliability guaranteed delivery semantics. The type of this property  
1745 is "xs:boolean".

1746

### 1747 **IV.B. Duplicate Elimination Property**

1748 This property is identified by the QName "wsrm:NoDuplicateDelivery" and corresponds to the  
1749 semantics specified by the WS-Reliability duplicate elimination semantics. The type of this  
1750 property is "xs:boolean".

1751

### 1752 **IV.C. Message Ordering Property**

1753 This property is identified by the QName "wsrm:OrderedDelivery" and corresponds to the  
1754 semantics specified by the WS-Reliability message ordering semantics. The type of this property  
1755 is "xs:boolean".

1756

### 1757 **IV.D. Reply Pattern Property**

1758 This property is identified by the QName "wsrm:ReplyPattern" and corresponds to the semantics  
1759 specified by the WS-Reliability reply pattern options. The type of this property is "xs:String".  
1760 (values: Response, Poll, Callback)

1761

## 1762 **V. Other Reliability Properties**

1763 In addition to the properties defined in section III, there are WS-Reliability properties that are used  
1764 on the Sender side (usually the client side and therefore do not occur in the WSDL document).

1765 This section identifies such properties. These properties MUST NOT be specified in the WSDL  
1766 document. How the properties are specified and/or represented does not affect interoperability as  
1767 these properties are client-side only properties. They are defined here for convenience only.

1768

## 1769 **V.A. Group Expiry Time**

1770 This property is identified by the QName "wsrm:GroupExpiryTime" and corresponds to the  
1771 semantics specified by the WS-Reliability group expiration time. The type of this property is  
1772 xs:duration.

1773 Note: The expiry time is calculated at the time a message is sent, but adding this duration to the  
1774 time the message is sent.

1775

## 1776 **V.B. Group Maximum Idle Duration**

1777 This property is identified by the QName "wsrm:GroupMaxIdleDuration" and corresponds to the  
1778 semantics specified by the WS-Reliability group maximum idle duration. The type of this property  
1779 is xs:duration.

1780

## 1781 **V.C. Message Expiration Time**

1782 This property is identified by the QName "wsrm:ExpiryTime" and corresponds to the semantics  
1783 specified by the WS-Reliability message expiration time. The type of this property is xs:duration.

1784 Note: The expiry time is calculated at the time a message is sent, but adding this duration to the  
1785 time the message is sent.

1786

## 1787 **V.D. Retry Maximum Time**

1788 This property is identified by the QName "wsrm:RetryMaxTimes" and corresponds to the  
1789 semantics specified by the WS-Reliability maximum retry times. The type of this property is xs:int.

1790

## 1791 **V.E. Retry Time Interval**

1792 This property is identified by the QName "wsrm:RetryTimeInterval" and corresponds to the  
1793 semantics specified by the WS-Reliability retry time interval. The type of this property is  
1794 xs:duration.

1795

## 1796 **V.F. ReplyTo URI**

1797 This property is identified by the QName "wsrm:ReplyTo" and corresponds to the semantics  
1798 specified by the WS-Reliability reply-to. The type of this property is xs:anyURI.

1799

## 1800 **VI. Schema**

1801 <?xml version="1.0" encoding="UTF-8" ?>

1802 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

```

1803   xmlns:wsm="http://www.oasis-open.org/committees/wsm/schema/1.1/feature/rel/"
1804   targetNamespace="http://www.oasis-open.org/committees/wsm/schema/1.1/feature/rel/"
1805   elementFormDefault="qualified" >
1806
1807   <!-- properties to be used in WSDL -->
1808   <xs:element name="GuaranteedDelivery" type="xs:boolean"/>
1809   <xs:element name="NoDuplicateDelivery" type="xs:boolean"/>
1810   <xs:element name="OrderedDelivery" type="xs:boolean"/>
1811   <xs:element name="ReplyPattern" type="xs:string"/>
1812
1813   <!-- properties to be used on the client side -->
1814   <xs:element name="GroupExpiryTime" type="xs:duration"/>
1815   <xs:element name="GroupMaxIdleDuration" type="xs:duration"/>
1816   <xs:element name="ExpiryTime" type="xs:duration"/>
1817   <xs:element name="RetryMaxTimes" type="xs:int"/>
1818   <xs:element name="RetryTimeInterval" type="xs:duration"/>
1819   <xs:element name="ReplyTo" type="xs:anyURI"/>
1820 </xs:schema>

```

1821

## 1822 **VII. Examples**

### 1823 **VII.A Example for the "all" compositor**

```

1824 <wsdl11:portType name="Example-1">
1825   <fnp:compositor uri="http://www.oasis-
1826   open.org/committees/wsm/schema/1.1/fnp/compositor/all">
1827     <fnp:feature uri="http://www.oasis-open.org/committees/wsm/schema/1.1/feature/rel/"
1828     <fnp:compositor uri="http://www.oasis-
1829   open.org/committees/wsm/schema/1.1/fnp/compositor/all">
1830     <fnp:property name="wsm:DuplicateElimination">
1831       <fnp:value>true</fnp:value>
1832     </fnp:property>
1833     <fnp:property name="wsm:OrderedDelivery">
1834       <fnp:value>true</fnp:value>
1835     </fnp:property>
1836     <fnp:property name="wsm:GuaranteedDelivery">
1837       <fnp:value>true</fnp:value>
1838     </fnp:property>

```

```
1839     </fnp:compositor>
1840 </fnp:feature>
1841 </fnp:compositor>
1842 ...
1843 </wsdl11:portType>
```

1844 In the example above, the reliability feature identified by URI "http://www.oasis-  
1845 open.org/committees/wsrn/schema/1.1/feature/rel/" is required by the portType. This feature  
1846 consists of three properties, all of which are required because of the semantics of the 'all'  
1847 compositor that composes the three properties.

1848

## 1849 **VII.B Example for the "choice" compositor:**

```
1850 <wsdl11:binding name="Example-2">
1851   <fnp:compositor uri="http://www.oasis-
1852   open.org/committees/wsrn/schema/1.1/fnp/compositor/all">
1853     <fnp:feature uri="http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/"
1854     <fnp:compositor uri="http://www.oasis-
1855     open.org/committees/wsrn/schema/1.1/fnp/compositors/choice">
1856       <fnp:property name="wsrm:ReplyPattern">
1857         <value>Response</value>
1858       </fnp:property>
1859       <fnp:property name="wsrm:ReplyPattern">
1860         <value>Callback</value>
1861       </fnp:property>
1862       <fnp:property name="wsrm:ReplyPattern">
1863         <value>Poll</value>
1864       </fnp:property>
1865     </fnp:compositor>
1866   </fnp:feature>
1867 </fnp:compositor>
1868 ...
1869 </wsdl11:binding>
```

1870 In the example above, the reliability feature identified by URI "http://www.oasis-  
1871 open.org/committees/wsrn/schema/1.1/feature/rel/" is required by the portType. This feature  
1872 consists of three properties, of which the client must choose one.

1873

## 1874 **VII.C Example for the "one-or-more" compositor:**

```
1875 <wsdl11:portType name="Example-3">
```

```

1876 <fnp:compositor uri="http://www.oasis-
1877 open.org/committees/wsrn/schema/1.1/fnp/compositor/all">
1878 <fnp:feature uri="http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/"
1879 <fnp:compositor uri="http://www.oasis-
1880 open.org/committees/wsrn/schema/1.1/fnp/compositor/one-or-more">
1881 <fnp:property name="wsrm:DuplicateElimination">
1882 <fnp:value>true</fnp:value>
1883 </fnp:property>
1884 <fnp:property name="wsrm:OrderedDelivery">
1885 <fnp:value>true</fnp:value>
1886 </fnp:property>
1887 <fnp:property name="wsrm:GuaranteedDelivery">
1888 <fnp:value>true</fnp:value>
1889 </fnp:property>
1890 </fnp:compositor>
1891 </fnp:feature>
1892 </fnp:compositor>
1893 ...
1894 </wsdl11:portType>
1895

```

## 1896 VII.D Example for the "zero-or-more" compositor:

```

1897 <wsdl11:portType name="Example-4">
1898 <fnp:compositor uri="http://www.oasis-
1899 open.org/committees/wsrn/schema/1.1/fnp/compositor/all">
1900 <fnp:feature uri="http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/"
1901 <fnp:compositor uri="http://www.oasis-
1902 open.org/committees/wsrn/schema/1.1/fnp/compositor/zero-or-more">
1903 <fnp:property name="wsrm:DuplicateElimination">
1904 <fnp:value>true</fnp:value>
1905 </fnp:property>
1906 <fnp:property name="wsrm:OrderedDelivery">
1907 <fnp:value>true</fnp:value>
1908 </fnp:property>
1909 <fnp:property name="wsrm:GuaranteedDelivery">
1910 <fnp:value>true</fnp:value>
1911 </fnp:property>
1912 </fnp:compositor>
1913 </fnp:feature>

```

```
1914 </fnp:compositor>
1915 ...
1916 </wsdl11:portType>
```

---

## 1917 **Appendix B. Acknowledgments**

- 1918 The following individuals were members of the committee during the development of this  
1919 specification:
- 1920 David Ingham, Arjuna Technologies Limited
- 1921 Joseph Chiusano, Booz Allen Hamilton
- 1922 Peter Furniss, Choreology Ltd
- 1923 Jeff Turpin, Cyclone Commerce
- 1924 Pramila Mullan, France Telecom
- 1925 Jacques Durand, Fujitsu
- 1926 Kazunori Iwasa (Secretary), Fujitsu
- 1927 Tom Rutt (Chair), Fujitsu
- 1928 Jishnu Mukerji, Hewlett-Packard
- 1929 Robert Freund, Hitachi
- 1930 Eisaku Nishiyama, Hitachi
- 1931 Nobuyuki Yamamoto, Hitachi
- 1932 Ben Bloch, Individual
- 1933 Mark Hansen, Individual
- 1934 Paolo Romano, Individual
- 1935 Dock Allen, Mitre Corporation
- 1936 Junichi Tatemura, NEC Corporation
- 1937 Alan Weissberger, NEC Corporation
- 1938 Magdolna Gerendai, Nokia
- 1939 Szabolcs Payrits, Nokia
- 1940 Mark Peel, Novell
- 1941 Sunil Kunisetty (Secretary), Oracle
- 1942 Anish Karmarkar, Oracle
- 1943 Jeff Mischkinsky, Oracle
- 1944 Marc Goodner (Secretary), SAP
- 1945 Pete Wenzel, SeeBeyond Technology Corporation
- 1946 Doug Bunting (Secretary), Sun Microsystems



- 1947 Tony Graham, Sun Microsystems
- 1948 Chi-Yuen Ng, University of Hong Kong
- 1949 Patrick Yee, University of Hong Kong
- 1950 Prasad Yendluri, webMethods, Inc.
- 1951 Scott Werden, WRQ, Inc.

1952

1953 And the following people made conditions to produce Ver1.0 of this specification:

- 1954 Colleen Evans, Sonic Software Corporation
- 1955 Dave Chappell, Sonic Software Corporation
- 1956 Doug Bunting, Sun Microsystems, Inc.
- 1957 George Tharakan, Sun Microsystems, Inc.
- 1958 Hisashi Shimamura, NEC Corporation
- 1959 Jacques Durand, Fujitsu Software Corporation
- 1960 Jeff Mischkinsky, Oracle Corporation
- 1961 Katsutoshi Nihei, NEC Corporation
- 1962 Kazunori Iwasa, Fujitsu Limited
- 1963 Martin Chapman, Oracle Corporation
- 1964 Masayoshi Shimamura, Fujitsu Limited
- 1965 Nicholas Kassem, Sun Microsystems, Inc.
- 1966 Nobuyuki Yamamoto, Hitachi Limited
- 1967 Sunil Kunisetty, Oracle Corporation
- 1968 Tetsuya Hashimoto, Hitachi Limited
- 1969 Tom Rutt, Fujitsu Software Corporation
- 1970 Yoshihide Nomura, Fujitsu Limited
- 1971 Akira Ochi, Fujitsu Limited
- 1972 Hirotaka Hara, Fujitsu Limited
- 1973 Hiroyuki Tomisawa, Hitachi Limited
- 1974 Katsuhisa Nakazato, Fujitsu Limited
- 1975 Masahiko Narita, Fujitsu Limited
- 1976 Nobuyuki Saji, NEC Corporation
- 1977 Shuichi Imabayashi, Fujitsu Limited

1978 **Appendix C. Revision History**

1979 *[This appendix is optional, but helpful. It should be removed for specifications that are at OASIS*  
 1980 *Standard level.]*

<b>Rev</b>	<b>Date</b>	<b>By Whom</b>	<b>What</b>
WD-0.5	2003-09-04	Kazunori Iwasa	Initial version
WD-0.51		Kazunori Iwasa	Editorial update
WD-0.52		Kazunori Iwasa	Editorial update
WD-0.54	-2003-10-23	Kazunori Iwasa	Issue Rel-38 : Section 3.1.3 Timestamp Issue Rel-98 : Section 3.1.2 and 3.2.3 Issue Rel-40 : Section 3.1.4 Issue Rel-88 : Section 3.1.1 Issue Rel-16 : Section 3.2.1 to 3.2.3 Issue Rel-14 : Appendix C Editorial update
WD-0.60	-2003-10-28	Kazunori Iwasa	Editorial update at F2F in South SF.
WD-0.70	-2003-10-30	Kazunori Iwasa	Section2: Messaging models Section3: Message Format, and others Section4: PollRequest Section5: Binding patterns Editorial update

<b>Rev</b>	<b>Date</b>	<b>By Whom</b>	<b>What</b>
WD-0.83	-2003-11-18	Kazunori Iwasa	<p>Section2.6: Added description of Figure3</p> <p>Section3: Added tables for each element</p> <p>Rel-31: Section2.5</p> <p>Rel-38: Timestamp was removed from Section 3</p> <p>Rel-100: Added Section2.9 Attachments</p> <p>Rel-32: Added definitions to Section1.8</p> <p>Rel-94: Figure5 and Section 3.3 (Needs additional descriptions and examples in Section3.3)</p> <p>Editorial updates, especially for : <a href="http://lists.oasis-open.org/archives/wsrn/200310/msg00054.html">http://lists.oasis-open.org/archives/wsrn/200310/msg00054.html</a></p> <p>All editorial comments above are incorporated except one, which is a comment for line 357, to keep consistency with other sections.</p>
WD-0.84	-2003-12-15	Kazunori Iwasa	<p>Rel-33:Section 1.8: Update on Message Delivery and Acknowledgment Message</p> <p>Rel-50:Section 3.1.3 ExpiryTime</p> <p>Editorial updates</p>
WD-0.85	-2004-01-06	Kazunori Iwasa	<p>Section2.4, Section2.5, and Section 3.1.1 are updated to incorporate resolutions for Rel-52, Rel-57, and Rel-82.</p>

<b>Rev</b>	<b>Date</b>	<b>By Whom</b>	<b>What</b>
WD-0.86	-2004-01-14	Kazunori Iwasa	<p>Updated for comments at :  <a href="http://www.oasis-open.org/archives/wsrn/200401/msg0010.html">http://www.oasis-open.org/archives/wsrn/200401/msg0010.html</a></p> <p>except for:</p> <ul style="list-style-type: none"> <li>- More faults for Tables1  (Need to list up all faults)</li> <li>- Section2.4 Line#259 in Spec20040106 (Ver0.85): It should read "after the message has been processed and delivered to the "next processing layer".  (Need to confirm with TC for this change, since the current text was approved one.)</li> <li>- Figure1,2,and3 "New processor Entity"  (Want to confirm with TC member)</li> <li>-New terminologies for "Group Termination", "Removal", "Complete", and others.  (Needs definitions)</li> </ul> <p>--</p> <p>And other editorial updates.</p>

<b>Rev</b>	<b>Date</b>	<b>By Whom</b>	<b>What</b>
WD-0.87	-2004-01-15	Kazunori Iwasa	<p>Updated for:</p> <p>Prelim Wed minutes on 1/14/2004 at:  <a href="http://www.oasis-open.org/archives/wsrn/200401/msg00038.html">http://www.oasis-open.org/archives/wsrn/200401/msg00038.html</a>.</p> <p>It includes:</p> <p>Rel33: New definitions in 1.8(deliver, submit, notify)</p> <p>Rel37: editorial change in 3.1.1</p> <p>Rel40: editorial change in 3.1.3</p> <p>Rel44: updates for 3.1.1</p> <p>Rel51: change definition for Message Acknowledgment</p> <p>Rel52: Moved some of 3.1.1(line546-571) to 2.5.1</p> <p>Rel98: removed informative notes in 2.4</p> <p>Tables: Changed "Required" to "Cardinality" (Yes-1, No-0)</p> <p>The following resolutions are not updated yet:</p> <p>Rel 83-86 and 56:</p> <p>Change of element names and location (Eg. GroupId -&gt; MessageId)</p>

<b>Rev</b>	<b>Date</b>	<b>By Whom</b>	<b>What</b>
WD-0.88	-2004-01-16	Kazunori Iwasa	<p>Updated for:</p> <p>1) Prelim minutes for Thursday on 1/15/2004 at:  <a href="http://www.oasis-open.org/apps/org/workgroup/wsrn/email/archives/200401/msg00053.html">http://www.oasis-open.org/apps/org/workgroup/wsrn/email/archives/200401/msg00053.html</a></p> <p>2) Remaining items including:            Rel36: Message ID -&gt; Message Identifier            Rel37: Reference for RFC2392            Changed element names and location (Eg. GroupId -&gt; MessageId)            And others.</p> <p>The following items are still in progress:            Rel22: usage for MUST, MAY, Should, Optional</p>

<b>Rev</b>	<b>Date</b>	<b>By Whom</b>	<b>What</b>
WD-0.90	-2004-01-26	Kazunori Iwasa	<p>Updated for remaining action items at:  <a href="http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5089/Minutes-Jan04f2f.htm">http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5089/Minutes-Jan04f2f.htm</a></p> <p>This includes :</p> <p>1) 2.4: Message Identifier -&gt; MessageId  Sequence Number -&gt; SequenceNum  Included "Next processing layer"</p> <p>2) 2.11: Chart is updated (Now2.9)</p> <p>3) 3.1.1 and 3.1.2: two group attributes were moved from MessageId to SequenceNum</p> <p>4) 3.1.4 Response : Some sentences are added to restrict sending back previous Acknowledgment message in the other Response.</p> <p>5) 3.3, 3.3.1 and 3.4.1:  MessageId -&gt; RefToMessageIds  value -&gt; groupId</p> <p>6) Section4: Replaced with new text</p> <p>7) Appendix A: Replaced with new schema</p> <p>Other changes includes:</p> <p>Cover page: location of the spec and errata were added.</p> <p>Section 1 and 2: Editorial review</p> <p>1.9: New section was added (RM agreement)</p> <p>1.6: Description for WS-I BP1.0 was included.</p> <p>6.2: Added non-normative Reference for SOAP messages with Attachments</p>

<b>Rev</b>	<b>Date</b>	<b>By Whom</b>	<b>What</b>
			<p>Remaining Action items and editorial changes for 2004-01-26(0.90):</p> <ol style="list-style-type: none"> <li>1. Consistency of word: e.g. sender RMP, sending RMP or sender's RMP</li> <li>2. Removing MAY, Optional</li> <li>3. NotSupportedFeaturesFault</li> <li>4. Explanatin for cardinality and others</li> <li>5. SOAP1.2 statement in the new section in 3 on rm:fault element</li> <li>6. Update fault in section3</li> <li>7. Examples</li> </ol>
WD-0.91	-2004-02-02	Kazunori Iwasa	<p>Updated for remaining action items at:</p> <p><a href="http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5089/Minutes-Jan04f2f.htm">http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5089/Minutes-Jan04f2f.htm</a></p> <p>or</p> <p><a href="http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5090/Action%20Item%20List%20from%20Jan%20Face%20To%20Face%20Meeting.htm">http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5090/Action%20Item%20List%20from%20Jan%20Face%20To%20Face%20Meeting.htm</a></p> <p>This includes :</p> <p>AI10. Done. Throughout the spec.</p> <p>AI16. Done. Section 1.5.</p> <p>AI20. Done. Section 3.5 is added.</p> <p>AI22. Done. Section1.5.</p> <p>AI24. Done. Section 3</p> <p>AI28. (Still working)</p> <p>AI8, 9 and 25. Done. Section 2.9.</p> <p>AI4 &amp; 19. Done. Section 2.10 is added.</p> <p>Schema was replaced with ws-reliability-2004-01-27.xsd.</p> <p>Table numbers were maintained sequentially.</p> <p>And other editorial updating.</p>



<b>Rev</b>	<b>Date</b>	<b>By Whom</b>	<b>What</b>
WD-0.92	-2004-02-09	Kazunori Iwasa	<p>This was updated for:</p> <p>AI10: Rel22: Remaining updates are done</p> <p>AI20: Section3.5: Added "SOAP1.2 can't use Fault element."</p> <p>Section3.4.1.1: from and to attribute are included here.</p> <p>Section1.8: Definition of PollRequest was added.</p> <p>Section5: Examples are added.</p> <p>And editorial updates.</p>
WD-0.93	-2004-02-10	Kazunori Iwasa	<p>This was updated for:</p> <p>Section1: Editorial updates.</p>
WD-0.94	-2004-02-12	Kazunori Iwasa	<p>This was updated for:</p> <p>Section2.8: Section number correction</p> <p>Section3.1.1 and 3.1.2: Messageld text</p> <p><a href="http://www.oasis-open.org/archives/wsrn/200402/msg00038.html">http://www.oasis-open.org/archives/wsrn/200402/msg00038.html</a></p> <p><a href="http://www.oasis-open.org/archives/wsrn/200402/msg00068.html">http://www.oasis-open.org/archives/wsrn/200402/msg00068.html</a></p> <p>Rel108: Section1.7 and 3.3: Clarified that PollRequest can be used for any RM-Reply pattern, and a reply to PollRequest only includes successfully delivered messages.</p> <p>Section4: Example11 is added. Numbering of examples are corrected sequentially.</p> <p>Section2: Some editorial comments at:</p> <p><a href="http://www.oasis-open.org/archives/wsrn/200402/msg00080.html">http://www.oasis-open.org/archives/wsrn/200402/msg00080.html</a></p> <p>Rel76: Section 3.3: Agreed text was added.</p> <p>Appendix B: Acknowledgments for Ver1.0 was added.</p> <p>And some other editorial updates.</p>

<b>Rev</b>	<b>Date</b>	<b>By Whom</b>	<b>What</b>
WD-0.98	-2004-02-26	Kazunori Iwasa	<p>This was updated with minutes at:  <a href="http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5630/MinutesWSRMTTC021704.htm">http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5630/MinutesWSRMTTC021704.htm</a> and  <a href="http://www.oasis-open.org/archives/wsrn/200402/msg00223.html">http://www.oasis-open.org/archives/wsrn/200402/msg00223.html</a></p> <p>Rel102: Remove Section2.7  Rel108/115: Remove section 3.5                    Updates on Section 3.4                    Updates on Section 4</p> <p>Rel113: Section 2.4                    Section 2.8.1</p> <p>New issue: removing MessageHeader                    throughout the spec</p> <p>Editorial reshuffle with:  <a href="http://www.oasis-open.org/archives/wsrn/200402/msg00161.html">http://www.oasis-open.org/archives/wsrn/200402/msg00161.html</a></p> <p>Appendix A: Schema  Appendix B: Acknowledgments  And minor editorial updates</p>
WD-0.99	-2004-03-03	Kazunori Iwasa	<p>This was updated with minutes at:  <a href="http://www.oasis-open.org/apps/org/workgroup/wsrn/email/archives/200403/msg00035.html">http://www.oasis-open.org/apps/org/workgroup/wsrn/email/archives/200403/msg00035.html</a></p> <p>except for Minutes 4.4 Rel119, which  requires discussion.</p> <p>And also minor editorial updates were  done.</p>
WD-0.991	-2004-03-04	Kazunori Iwasa	<p>This was updated with :</p> <p>Editorial updates : Bullet list consistency</p> <p>Appendix A: Two new members are  added in the Acknowledgments.</p>

<i>Rev</i>	<i>Date</i>	<i>By Whom</i>	<i>What</i>
WD-0.992	-2004-03-10	Kazunori Iwasa	This was updated with a minutes at: <a href="http://www.oasis-open.org/archives/wsrn/200403/msg00084.html">http://www.oasis-open.org/archives/wsrn/200403/msg00084.html</a> And some editorial updates.

1981

1982

---

1983 **Appendix D. Futures List**

1984 The features and issues in the table below are listed as forward-looking statements regarding  
1985 possible enhancements or the evolution of this specification.

1986

	<b>Category</b>	<b>Details</b>
1	WSDL	Define WSDL extensions profiling the use of RM SOAP extensions.

1987

---

## 1988 Appendix E. Notices

1989 OASIS takes no position regarding the validity or scope of any intellectual property or other rights  
1990 that might be claimed to pertain to the implementation or use of the technology described in this  
1991 document or the extent to which any license under such rights might or might not be available;  
1992 neither does it represent that it has made any effort to identify any such rights. Information on  
1993 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS  
1994 website. Copies of claims of rights made available for publication and any assurances of licenses  
1995 to be made available, or the result of an attempt made to obtain a general license or permission  
1996 for the use of such proprietary rights by implementors or users of this specification, can be  
1997 obtained from the OASIS Executive Director.

1998 OASIS invites any interested party to bring to its attention any copyrights, patents or patent  
1999 applications, or other proprietary rights which may cover technology that may be required to  
2000 implement this specification. Please address the information to the OASIS Executive Director.

2001 **Copyright © OASIS Open 2003-2004. All Rights Reserved.**

2002 This document and translations of it may be copied and furnished to others, and derivative works  
2003 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,  
2004 published and distributed, in whole or in part, without restriction of any kind, provided that the  
2005 above copyright notice and this paragraph are included on all such copies and derivative works.  
2006 However, this document itself does not be modified in any way, such as by removing the copyright  
2007 notice or references to OASIS, except as needed for the purpose of developing OASIS  
2008 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual  
2009 Property Rights document must be followed, or as required to translate it into languages other  
2010 than English.

2011 The limited permissions granted above are perpetual and will not be revoked by OASIS or its  
2012 successors or assigns.

2013 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
2014 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO  
2015 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY  
2016 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
2017 PARTICULAR PURPOSE.