



Web Services Reliable Messaging TC WS-Reliability

Working Draft 0.993, 19 April 2004

Document identifier:

wd-web services reliable messaging tc-ws-reliability-0.993

Location:

[http://www.oasis-open.org/committees/wsrn/documents/specs/\(TBD\)](http://www.oasis-open.org/committees/wsrn/documents/specs/(TBD))

Editor:

Kazunori Iwasa, Fujitsu Limited <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 [wsrm@lists.oasis-open.org](mailto:wsrn@lists.oasis-open.org) list. Others should subscribe to and send comments to the [wsrm-comment@lists.oasis-open.org](mailto:wsrn-comment@lists.oasis-open.org) list. To subscribe, send an email message to [wsrm-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..... | 17 |
| 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..... | 29 |
| 52 | 4.5 Fault Codes For Reliable Messaging Failures..... | 34 |
| 53 | 5 Operational Aspects and Semantics..... | 39 |
| 54 | 5.1 Message Group Life Cycle..... | 39 |
| 55 | 5.2 WSDL Operation Type..... | 43 |
| 56 | 5.3 Poll Reply Pattern Semantics and Usage..... | 44 |
| 57 | 5.4 Attachments..... | 44 |
| 58 | 6 HTTP Binding..... | 45 |
| 59 | 6.1 Reliable Messaging with “Response” binding pattern..... | 45 |
| 60 | 6.2 Reliable Messaging with “Callback” binding pattern..... | 47 |
| 61 | 6.3 Reliable Messaging with “Poll” binding pattern..... | 49 |
| 62 | 7 Conformance | 52 |
| 63 | 8 References..... | 53 |

| | | |
|----|--|----|
| 64 | 8.1 Normative References..... | 53 |
| 65 | 8.2 Non-normative References..... | 54 |
| 66 | Appendix A. WS-Reliability Features, Properties and Compositor (Normative and Optional)..... | 55 |
| 67 | Appendix B. Acknowledgments..... | 65 |
| 68 | Appendix C. Revision History..... | 67 |
| 69 | Appendix D. Futures List..... | 77 |
| 70 | Appendix E. Notices..... | 78 |
| 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.)?

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

1.3 Notational Conventions

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

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

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

This specification uses the following namespace prefixes:

| Prefix | Namespace |
|---------------|--|
| 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.*

1.4 Relation to Other Specifications

- **W3C SOAP1.1/1.2**: SOAP1.1 [SOAP1.1] and SOAP1.2 [SOAP1.2] are the base protocols for this specification. This specification defines reliable messaging protocol embedded in the SOAP Header.
- **OASIS ebXML Message Service Specification 2.0**: The reliable message mechanism defined in the ebXML Message Service Specification 2.0 [ebMS] is implemented in a number of products and open source efforts, many of which have undergone interoperability testing. WS-Reliability borrows from this technology.
- **OASIS WS-Security**: This specification defines reliability independently from security, each of these features mapping to different SOAP header extensions. Although both features can be used in combination, the specification does not attempt to compose them in a more intricate way, nor does it attempt to profile their combination. This specification can be used with WS-Security [WSS] when that effort is completed in 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"
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 Receiver 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 Sender RMP supports, invoked per Reliable message (e.g., a request to
288 the Sender RMP to take responsibility for the reliable message. The time at which this operation is
289 invoked must be clearly identifiable so that the RMP can always establish in which order two
290 submissions are made.

291

292 **Notify:**

293 An abstract operation the Sender RMP may invoke per Reliable Message (e.g, a notification that
294 the Sender 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 Receiver 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 Receiver 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 Receiver 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.

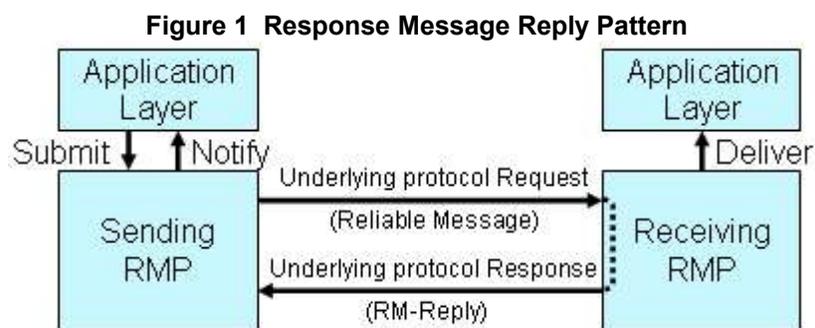
457 2.2 Message Reply Patterns

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

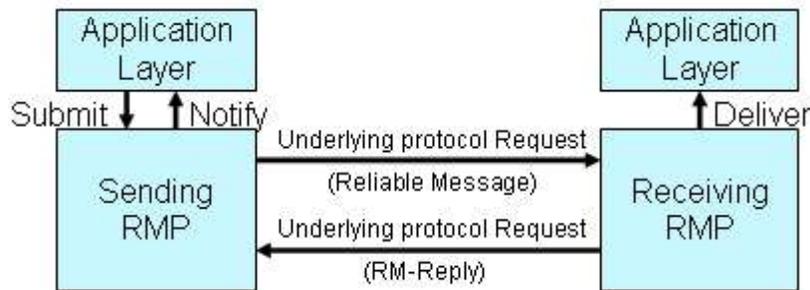


466 (2) Callback Message Reply Pattern

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

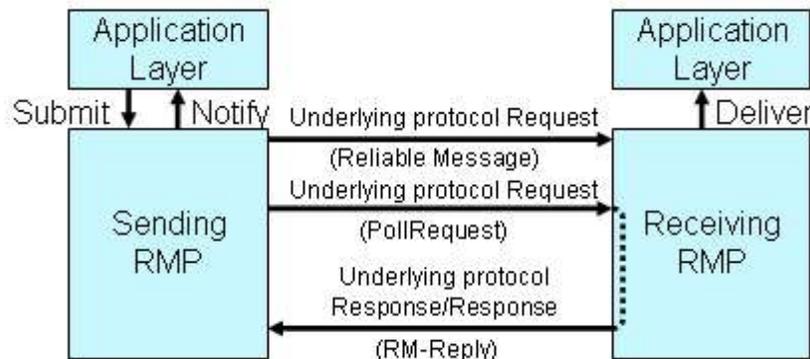


473 **(3) Poll Message Reply Pattern**

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, then the Sender MAY either Poll the Receiver for the status of
507 that message or MAY resend the same message with same MessageId as long as it is not
508 expired (i.e., ExpiryTime is not passed).

509 The time interval between two retries is specified by the RetryTimeInterval agreement item. If the
510 sender RMP cannot guarantee that the message has been successfully delivered by the Receiver
511 RMP, the sender RMP MUST notify a delivery error.

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

- 514 • An Invalid Message Format fault code (Table 16)
- 515 • A NonSupportedFeature fault code
- 516 • A PermanentProcessingFailure fault code

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

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

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

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

527 **3.2 Duplicate Elimination**

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

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

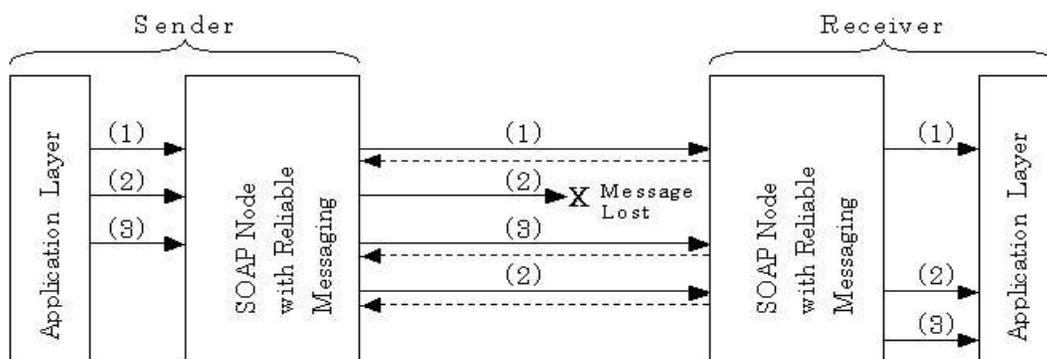
535 3.3 Guaranteed Message Ordering

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

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

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

549 **Figure 4 Guaranteed Message Ordering**



550

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

554 **Failure Case:**

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

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

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

563

564 3.4 Sequence Number

565 A sequence number mechanism is used to track and enforce the order of a sequence of
566 messages within the same group. Such a mechanism has been widely used in the past. In the

567 Figure 4 above, messages (1), (2), and (3) will be respectively assigned sequence numbers 1, 2,
568 and 3. If the message (2) was not properly received for any reason, the sender will resend the
569 message. Sequence numbering allows the receiver RMP to easily detect a missing message in a
570 sequence, that is (2), as soon as receiving (3). This condition is recognized by the receiver when
571 the sequence numbers of the messages it receives are not contiguous (e.g., 1, 3, 2).

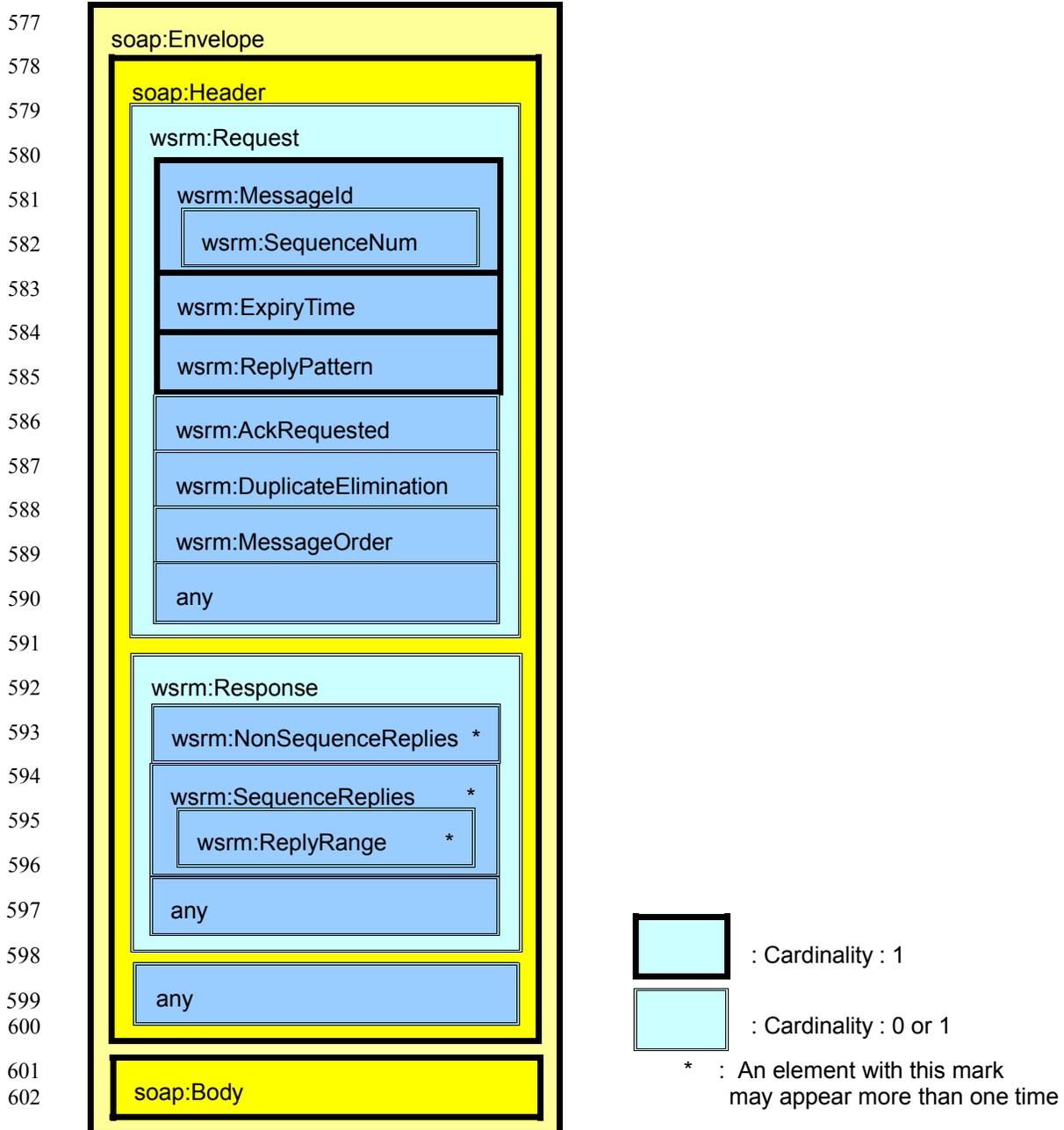
572

573 4 Message Format

574 4.1 Structure

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

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



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

604

Figure 6 Structure of PollRequest message elements

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

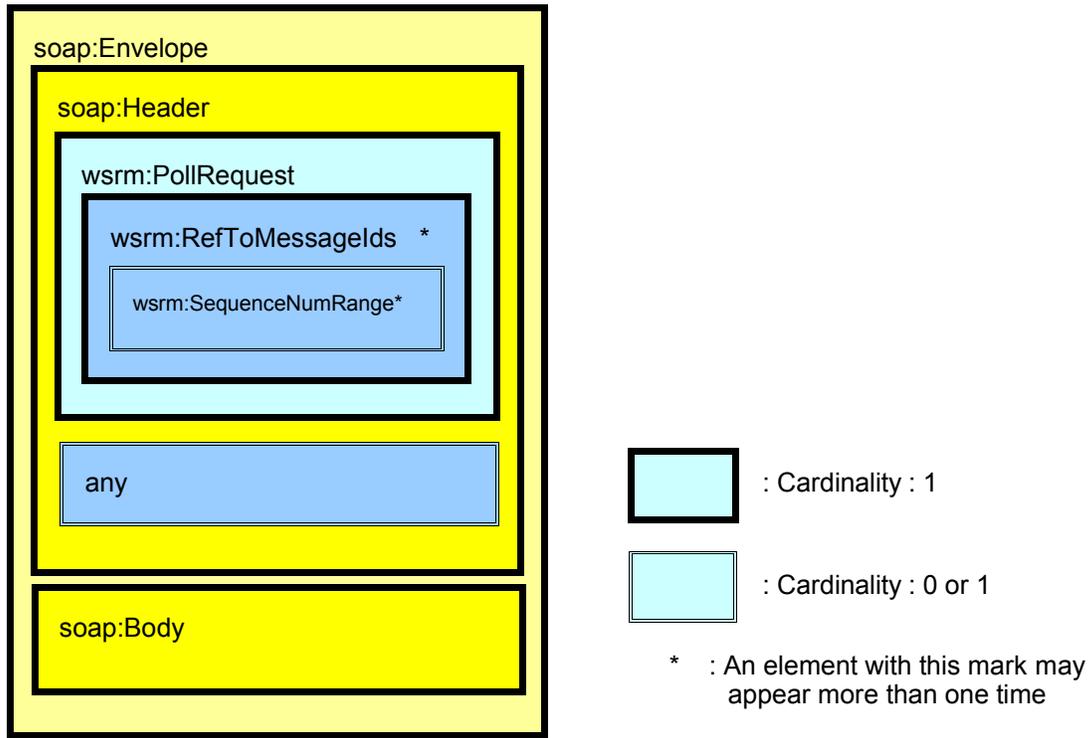
630

631

632

633

634



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

635 4.2 Request Element

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

- 641 • SOAP **mustUnderstand** attribute with a value of "1"
- 642 • **MessageId** element
- 643 • **ExpiryTime** element
- 644 • **ReplyPattern** element
- 645 • **AckRequested** element
- 646 • **DuplicateElimination** element
- 647 • **MessageOrder** element

648

Table 1 Request Element

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

649

650 Example 5 shows an example of a Request element.

651

652

Example 5 Request Element

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

```

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

```

668 4.2.1 MessageId Element

669 The Sender RMP MUST include the MessageId element for a Reliable Message.

670 This element includes the following attribute:

- 671 • a **groupId** attribute

672 Table 2 MessageId Element

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

673

674 (1) groupId attribute

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

680 4.2.1.1 SequenceNum Element

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

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

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

690 This element includes the following attributes:

- 691 • a **groupExpiryTime** attribute

- 692 • a **groupMaxIdleDuration** attribute
- 693 • a **number** attribute
- 694 • a **last** attribute

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

698 If the MessageOrder element appears in the message sent, the receiver RMP MUST NOT deliver
 699 the message until all messages with the same groupId value and a lower number value have
 700 been delivered. Example 6 illustrates some message fragments with SequenceNum element:

701

702

Example 6 SequenceNum Element

703

1) First message

704

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

705

706

707

708

2) Second message

709

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

710

711

712

713

3) Third message

714

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

715

716

717

718

Table 3 SequenceNum Element

| | |
|----------------|--|
| Cardinality | 0 or 1 |
| Value | None |
| Attributes | groupExpiryTime (dateTime) groupMaxIdleDuration (duration) number (unsignedLong) last (boolean) |
| Child elements | None |

719

720 (1) groupExpiryTime attribute

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

725 (2) groupMaxIdleDuration attribute

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

731 (3) number attribute

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

734 The Number attribute MUST have a value between 0 and 18446744073709551615 (maximum
735 value for XMLschema unsignedlong). As the Number value is incremented of 1 for each message
736 submitted to the Sender RMP, once the value reaches the maximum the group is terminated (See
737 Section 5).

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) last attribute

752 This attribute is used to mark the end of a group, when its last message is known from the Sender
753 before the message is sent. When this attribute is present, its boolean value has the following
754 meaning:

- 755 • **False:** Indicating the message is not the last message of the group, or is not known to
756 be the last message of the group. (Default value)
- 757 • **True:** Indicating the message is known to be the last message sent within a group of
758 messages.

759 4.2.2 ExpiryTime Element

760 The ExpiryTime element is used to indicate the ultimate time after which the receiver RMP MUST
761 NOT invoke the deliver operation for the received message. An RMP MUST include this element

762 in a Request element. After a message has been sent for the first time, the value of the
 763 ExpiryTime in a message MUST NOT be modified in any manner by the Sender RMP, when
 764 resending the message: two messages with same Message Identifier (duplicates) MUST have the
 765 same value for ExpiryTime. When a message expires on the Sender side before being
 766 successfully sent, a Sender RMP MUST NOT send it or resend it, and MUST communicate a
 767 delivery failure to the Sender application. The time MUST be expressed as UTC and MUST
 768 conform to a [XML Schema] dateTime. The message is considered expired if the current time, in
 769 UTC, is greater than the value of the ExpiryTime element.

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

773

Table 4 ExpiryTime Element

| | |
|----------------|----------|
| Cardinality | 1 |
| Value | dateTime |
| Attributes | None |
| Child elements | None |

774

775 4.2.3 ReplyPattern Element

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

- 781 • **Response** : A RM-Reply MUST be sent back directly in the response to the Reliable
 782 Message. This pattern is not applicable for one-way application level MEP.
- 783 • **Callback**: A RM-Reply MUST be sent as a callback request, using the address in the
 784 replyTo attribute. This pattern is not applicable for request-response application level
 785 MEP.
- 786 • **Poll**: A RM-Reply MUST be sent as a response to a poll request. This pattern is not
 787 applicable for request-response application level MEP.

788 The ReplyPattern element contains the following attribute:

- 789 • a **replyTo** attribute

790

791

Table 5 ReplyPattern Element

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

792

793 (1) replyTo attribute

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

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

799

800 4.2.4 AckRequested Element

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

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

809

Table 6 AckRequested Element

| | |
|----------------|--------|
| Cardinality | 0 or 1 |
| Value | None |
| Attributes | None |
| Child elements | None |

810

811 4.2.5 DuplicateElimination Element

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

815

Table 7 DuplicateElimination Element

| | |
|----------------|--------|
| Cardinality | 0 or 1 |
| Value | None |
| Attributes | None |
| Child elements | None |

816

817 4.2.6 MessageOrder Element

818 This element is used to request the receiver RMP to invoke delivery operation with the same order
819 that the sender has submitted. When a sender submits multiple messages with Guaranteed
820 Message Ordering, the sender MUST include the MessageOrder element in every message. All
821 messages to be delivered in order MUST have the same groupId and MUST have sequence
822 number as a value of SequenceNum element in order of the message to be delivered to receiver's
823 application.

824

Table 8 MessageOrder Element

| | |
|----------------|--------|
| Cardinality | 0 or 1 |
| Value | None |
| Attributes | None |
| Child elements | None |

825

826 4.3 PollRequest Element

827 A sender MUST include the PollRequest element only in the PollRequest message as shown in
828 the Figure6. The PollRequest message contains the PollRequest element. The PollRequest
829 message is used to query RM-Reply for specific message. Typically, the PollRequest message is
830 to receive RM-Reply for a message sent with Polling RM-Reply Pattern. However PollRequest
831 message also can be used to receive RM-Reply for a message that was originally sent with
832 Response RM-Reply Pattern or Callback RM-Reply Pattern. The response to a PollRequest
833 message includes RM-Reply information about prior messages. In addition to its use for receiving
834 replies for requests using the poll RM-Reply pattern, a Sender RMP may use it as a general query
835 to determine non-expired messages which have been delivered. If a Receiving RMP does not
836 support this general query, it MAY return a NonSupportedFeature fault.

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

840 This element includes the following attributes and child element:

- 841 • SOAP **mustUnderstand** attribute with a value of "1"
- 842 • a **replyTo** attribute
- 843 • a **RefToMessageIds** element

844

Table 9 PollRequest Element

| | |
|----------------|---|
| Cardinality | 0 or 1 |
| Value | None |
| Attributes | MustUnderstand (boolean) replyTo (URI) |
| Child elements | RefToMessageIds |

845

846

Example 7 PollRequest Element

```

847 <PollRequest
848   xmlns="http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1"
849   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
850   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
851   soap:mustUnderstand="1">
852   <RefToMessageIds groupId="mid://20040202.103832@oasis-open.org/">
853     <SequenceNumRange from="0" to="5"/>
854     <SequenceNumRange from="15" to="20"/>
855   </RefToMessageIds>
856   <RefToMessageIds groupId="mid://20040202.103811@oasis-open.org/" />
857   <RefToMessageIds groupId="mid://20040202.103807@oasis-open.org/">
858     <SequenceNumRange from="713" to="6150"/>
859   </RefToMessageIds>
860 </PollRequest>

```

861

862 (1) replyTo attribute

863 This attribute, of type URI, MAY be included by the Sender RMP. If present, then the receiver
 864 MUST send the RM-Reply in an underlying request to the value of the URI. If not present, the RM-
 865 Reply MUST be sent back in the underlying response of the Poll request itself.

866 4.3.1 RefToMessageIds Element

867 A sender MUST include the RefToMessageIds element for PollRequest message. This element is
 868 to be used to specify RM-Reply to be returned. This element MUST have one groupId attribute
 869 and MAY contain zero or more SequenceNumRange element as follows:

- 870 • a **groupId** attribute
- 871 • zero or more **SequenceNumRange** element

872

Table 10 RefToMessageIds Element

| | |
|----------------|------------------|
| Cardinality | 1 or more |
| Value | None |
| Attributes | groupId (URI) |
| Child elements | SequenceNumRange |

873 When this RefToMessageIds element has a groupId attribute, but doesn't have
 874 SequenceNumRange element, the receiver MUST send back RM-Replies for non-expired
 875 messages that were either delivered or faulted in that message range. When the
 876 RefToMessageIds element has a groupId attribute and SequenceNumRange element(s), the
 877 receiver MUST return RM-Replies for non-expired delivered or fault messages for messages
 878 received that were specified by the combination of groupId of RefToMessageIds and
 879 SequenceNumRange element(s). When sender RMP requests multiple RM-Replies with different

880 groupId value in one PollRequest Message, it MUST include RefToMessageIds element for each
881 groupId.

882 (1) groupId attribute

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

886 4.3.1.1 SequenceNumRange element

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

890 • a **from** attribute

891 • a **to** attribute

892 Table 11 SequenceNumRange Element

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

893 (1) from attribute

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

898 (2) to attribute

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

903 4.4 Response Element

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

907 • SOAP **mustUnderstand** attribute with a value of "1"

908 • a **ReplyPattern** attribute, which defaults to the value "Response"

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

910 • zero or more **NonSequenceReply** element

911 • zero or more **SequenceReplies** element

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

917 Table 12 Response Element

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

918

919 Example 8 shows an example of the Response element.

920

921

Example 8 Response Element

922 <Response

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

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

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

926 soap:mustUnderstand="1" replyPattern="Callback">

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

928 <NonSequenceReply groupId="mid://20040202.103811@oasis-open.org"

929 fault="wsrm:PermanentProcessingFailure" />

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

931 <ReplyRange from="1" to="4" />

932 <ReplyRange from="5" to="5" fault="wsrm:InvalidRequest" />

933 <ReplyRange from="6" to="42" />

934 </SequenceReplies>

935 </Response>

936

937 (1) replyPattern attribute

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

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

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

- 945 • If the group does not use sequence numbers, the first element of the response must
946 be a NonSequenceReply element containing the groupId which is the globally unique
947 message identifier for the Reliable Messaging Request.
- 948 • If the group uses sequence numbering, the first element of the response must be a
949 SequenceReplies element, with its groupId equal to that of the request, and with its
950 first Range element having its from and to attributes both equal to the sequence
951 number in the request.

952 4.4.1 NonSequenceReply Element

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

955 This element MUST contain the value of the groupId attribute for the message the reply pertains
956 to. If the reply is an acknowledgment of delivery, the Receiver RMP MUST NOT include the fault
957 attribute. If the reply is an indication of an Reliable Messaging Fault, the Receiver RMP MUST
958 include the fault attribute, and its value denotes the fault condition which was encountered.

959

Table 13 NonSequenceReply Element

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

960

961 (1) groupId attribute

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

965 4.4.2 SequenceReplies Element

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

971

Table 14 MessageReplies Element

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

972 **(1) groupId attribute**

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

975 **4.4.2.1 ReplyRange Element**

976 A receiver MUST include the ReplyRange element in a SequenceReplies element to indicate
 977 sequence numbers which either are being acknowledged (in which case Receiver RMP MUST
 978 NOT include the fault attribute) or have encountered a particular fault condition (in which case the
 979 Receiver RMP MUST include the fault attribute with that particular RM fault code encountered).

980

Table 15 ReplyRange Element

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

981

982 **(1) from attribute**

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

987 **(2) to attribute**

988 A receiver MUST include the to attribute in the ReplyRange element. This attribute is to be used to
 989 specify the largest SequenceNum of the message range. The value of this attribute MUST be
 990 equal or larger than the value of from attribute. It MUST be the same with the value of the from
 991 attribute to specify only one message. The value of this attribute is unsignedLong.

992 **(3) fault attribute**

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

996

997

998 **4.5 Fault Codes For Reliable Messaging Failures**

999 This section describes the protocol specific fault codes that are needed to better describe the
1000 reason for WS-Reliability protocol processing failures.

1001 We categorize the faults into 2 categories based on whether the fault was generated because
1002 Reliable Messaging Headers are malformed or invalid due to some runtime processing errors
1003 encountered by the RMP. The former category is called Invalid Message Format fault set and the
1004 latter is called Request Processing fault set. They are explained in detail in the following sections.

1005 These protocol specific fault codes are returned by the Receiver RMP within the response header
1006 element. The WS-Reliability protocol does not directly map our Reliable Messaging Faults to the
1007 SOAP Fault model.

1008 The SOAP Fault model is used for reporting faults due to the request payload, which fits the
1009 SOAP fault model better. Thus a response may have a SOAP Fault message, but the reason for
1010 the SOAP fault would be due to problems associated with the WSDL operation message payload.
1011 (E.g., A problem with the soap:body of a request message or the inability of the Receiver RMP to
1012 return the WSDL response in the soap:body of when using the Response RM-Reply pattern).

1013 **Example case 1:**

1014 For WSDL Request/Response operation types, a SOAP Fault can occur for a reliable request
1015 which was delivered, but then encountered an application level Fault due to something wrong in
1016 the payload (SOAP Body of request which is not under control of Sender RMP) or application
1017 processing space outside the realm of the Receiver RMP.

1018 That means a Acknowledgment can be delivered on a SOAP Fault.

1019 **Example case 2:**

1020 For the Response Reply Pattern, used with WSDL two way operation type, the return message
1021 could conceivably carry an indication of an RM Fault, which is not itself carried on a SOAP Fault.
1022 The exact behavior in such a case might be an implementation matter.

1023 A message with an RM Fault indication MUST NOT be delivered by the Receiver RMP. If the
1024 message cannot be delivered due, say an request fault, then there would be no meaningful data
1025 for the responder to put into the SOAP Body for the WSDL response.

1026 When using the Response RM-Reply pattern, a WSDL operation reply will not always be available
1027 for the Receiver RMP to return with the RM-Response. This will occur when there is a Reliable
1028 Messaging Fault for the message in the request, or when the message in the request is a
1029 duplicate of a prior delivered message with Duplicate Elimination in use.

1030 When a Receiver RMP cannot return the WSDL operation response for a request using the
1031 Response Reply Pattern, it MUST return the RM Response in a SOAP Fault message. If the RM
1032 Fault encountered was due to a problem with the request header element, a SOAP client fault
1033 MUST be returned. If the RM Fault encountered was due to a problem with processing by the
1034 Receiver RMP (including the inability to return a response due to Duplicate Elimination), a
1035 soap:server fault must be returned.

1036 The following Fault codes may be carried in a Response element associated with a MessageId.

1037 **4.5.1 Invalid Message Format Fault**

1038 These faults are sent by the Receiver RMP when the message format of the Reliable Messaging
1039 Headers are either invalid or wrong.

1040

Table 16 Invalid Message Format Fault Code Values

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

1042

1043 4.5.2 Message Processing Failure Faults

1044 These faults are sent by the Receiver RMP when there is an error processing a valid Reliable
1045 Messaging message.

1046 **Table 17 Messaging Processing Failure Fault Code Values**

| Local part name | Description and Cause(s) |
|----------------------------|---|
| NonSupportedFeature | This fault is sent by the Receiver 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 Receiver RMP that doesn't support Guaranteed Message Ordering |
| PermanentProcessingFailure | This fault is sent for permanent/fatal processing failures such as: <ol style="list-style-type: none">1. Persistence Storage failures2. Message Delivery failures 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: <ol style="list-style-type: none">1. Maximum number of buffered requests exceeded the limit.2. Maximum number of threads reached the limit etc. A transient fault unlike a permanent fault is a temporary one and MAY succeed in subsequent retries. |

1047 Note that there may be cases where in the Receiver RMP is not able to send fault messages with
1048 invalid message headers such as:

- 1049 • The replyTo attribute is missing or invalid when it is required such as for Callback and
1050 asynchronous Poll cases.
- 1051 • The MessageId element is missing for Request element.
- 1052 • The RefToMessageIds is missing for PollRequest element.

1053

Example 9 Fault Message for Reliable Messaging

```
1054 <?xml version="1.0" encoding="UTF-8"?>
1055 <soap:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
1056   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
1057   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1058   xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
1059   <soap:Header>
1060     <Response
1061       xmlns="http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1"
1062       soap:mustUnderstand="1" replyPattern="Callback">
1063       <SequenceReplies groupId="mid://20040202.103832@oasis-open.org">
1064         <ReplyRange from="1" to="1" fault="InvalidRequest" />
1065       </SequenceReplies>
1066     </Response>
1067   </soap:Header>
1068   <soap:Body />
1069 </soap:Envelope>
```

1070

1071 If PollRequest element in Example 7 were missing soap:mustUnderstand attribute, the
1072 InvalidPollRequest fault may be sent as follows.

1073

Example 10 Fault Message for PollRequest message

```
1074 <?xml version="1.0" encoding="UTF-8"?>
1075 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
1076   <soap:Header>
1077     <Response
1078       xmlns="http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1";
1079       soap:mustUnderstand="1" replyPattern="Poll">
1080       <SequenceReplies groupId="mid://20040202.103832@oasis-open.org";>
1081         <ReplyRange from="0" to="5" fault="InvalidPollRequest"/>
1082         <ReplyRange from="15" to="20" fault="InvalidPollRequest"/>
1083       </SequenceReplies>
1084       <NonSequenceReply groupId="mid://20040202.103811@oasis-open.org";
1085         fault="InvalidPollRequest"/>
1086       <SequenceReplies groupId="mid://20040202.103807@oasis-open.org";>
1087         <ReplyRange from="713" to="6150" fault="InvalidPollRequest"/>
1088       </SequenceReplies>
1089     </Response>
```

```
1090 </soap:Header>
1091 <soap:Body />
1092 </soap:Envelope>
```

1093 5 Operational Aspects and Semantics

1094 5.1 Message Group Life Cycle

1095 5.1.1 Group Termination

1096 Being able to know when a group may be terminated, is essential for efficient management of the
1097 persistent store of an RMP. As groups may last a long time and their state requires persistence, it
1098 is important to know when their persistent image can be reclaimed. The termination cases
1099 described in this section may seem numerous. This plurality results from the flexibility given to
1100 users in specifying various ways a group can be terminated, which in turn depends on application
1101 needs. However, in spite of this plurality, the termination logic is straightforward to implement and
1102 shares the same basic mechanisms across termination cases.

1103 Termination of a group in the sender RMP and in the receiver RMP are two distinct events not
1104 synchronized by any special message, but instead occurring as the result of rules applying
1105 separately to the Sender RMP and to the Receiver RMP. As a consequence, the termination of a
1106 group may occur at quite different times on the sender and receiver sides. However, the lack of
1107 synchronization allowed by these termination rules is not consequential.

1108 The states of a group on the sender and the receiver side are defined as follows:

1109 **Group complete:**

- 1110 • A group is considered complete in the Sender RMP, when all its messages have been
1111 sent and the last sent message has an ending marker (`SequenceNum/@last="true"`,
1112 or it has a sequence number with maximum value). Note that completeness occurs
1113 even if not all messages have been either acknowledged or faulted (in case
1114 `GuaranteedDelivery` is enabled.)
- 1115 • A group is considered complete in the Receiver RMP, when a last message with an
1116 ending marker has been received, and all previous messages for this group have also
1117 been received, (no number missing in the sequence) although not necessarily
1118 delivered yet.

1119 **Group closed:**

- 1120 • When a group is closed in the Sender RMP, no new message is expected to be sent
1121 by the RMP for this group. However, messages MAY still be resent in case
1122 `GuaranteedDelivery` is enabled. If a new message is submitted for a closed group, the
1123 Sender RMP MUST notify the submitting application that the group is closed and
1124 MUST NOT send the message.
- 1125 • When a group is closed in the receiver RMP, no new message is expected to be
1126 received for this group anymore. After a group is closed, and before the group is
1127 "removed" (see definition), a Receiver RMP MUST NOT deliver messages received
1128 with this group ID, whether they are duplicates or not of previous messages, and
1129 regardless whether they result from a resending of previously failed messages initiated
1130 before closing on the Sender side (in case `GuaranteedDelivery` is enabled).

1131 **Note:** A group may be closed without being complete, due to timeout. Once complete, a group will
1132 close (see termination rules).

1133 **Group Removed:**

1134 Group removal occurs at the time the group is closed, or after. Intuitively, a group is removed
1135 when a receiver does not need to remember anything about this group, i.e. there is no need to
1136 check for duplicates of its messages, in the future. This is the case when all its messages have
1137 expired.

1138 • When a group is removed in the Sender RMP, the RMP is not required to verify that
1139 future messages that are submitted are not associated with the removed group, and
1140 MAY treat these as belonging to a new group. However, in case the Sender RMP is in
1141 charge of generating group IDs, it MUST NOT reuse the group ID of a removed group,
1142 when initiating a new group.

1143 • When a group is removed in the Receiver RMP, the RMP is no longer supposed to
1144 remember anything about this group. In particular, the group ID is discarded from the
1145 RMP state. When receiving a message with same group ID as a removed group, a
1146 Receiver RMP is not required to verify if this group ID value has already been used.
1147 Such a message MAY be treated as belonging to a new group.

1148 5.1.2 Group Termination Parameters

1149 There are two RM Agreement items - GroupExpiryTime and GroupMaxIdleDuration that can be
1150 used to determine when a group can be terminated.
1151 To each of these agreement items, correspond respectively the message header attributes:
1152 groupExpiryTime and groupMaxIdleDuration. The following requirements pertain to these header
1153 attributes:

1154 a) the First message in a group (the one with Request/MessageId/SequenceNum/@number=0)
1155 MUST be used by the sender to indicate that timeout parameters are in use for the group.

1156 • If the first message in the sequence of a group has neither group timeout parameter
1157 present, the group will be terminated according to condition t3, t4 or t5.

1158 • If the first message has either one of the two group timeout parameters present (either
1159 groupExpiryTime, or groupMaxIdleDuration) then the group will be subject to
1160 termination rules t1 or t2 described below.

1161 • A fault MUST be returned if both group persistence parameters are present in any
1162 request message. An InvalidMessageParameters fault shall be sent in this case.

1163 • If groupExpiryTime is in use, the sender MUST NOT send a message in that group
1164 with an ExpiryTime greater than the groupExpiryTime.

1165 b) The group termination parameter which was sent on the first message in the group MUST be
1166 used on all subsequent messages in that group, and MUST be assigned a value.

1167 c) The Sender RMP MAY modify the value by sending a subsequent message with a new value.
1168 When applying termination rules, the Sender MUST use the value in the message with the
1169 highest sequence number sent for the group. The Receiver RMP MUST use the value from the
1170 message with the highest sequence number received for the group.

1171 d) A new value for groupMaxIdleDuration can either be increased or decreased. The protocol
1172 allows change (up or down) of groupExpiryTime, as long as it is never less than max(ExpiryTime)
1173 of messages received so far for the group.

1174 An InvalidMessageParameters Fault MUST be returned if the value of groupExpiryTime is
1175 decreased to be less than the max(ExpiryTime) of messages received for the group.

1176 **5.1.3 Termination Rules**

1177 Termination is the process by which an RMP discontinues the use of a group, allowing the RMP to
1178 reclaim resources used by the group. Termination typically involves two steps that may not occur
1179 at the same time: closing and removal. Removal of a group may happen some time later after it is
1180 closed, so that it will be possible to filter-out potential duplicate messages. The general rule is that
1181 a group is removed once all its messages have expired. If we define $\max(\text{ExpiryTime})$ as the
1182 maximum date of all ExpiryTime values of messages sent for a group (on the Sender side) or
1183 received for a group (on the Receiver side), then a group will not be removed before \max
1184 (ExpiryTime) occurs.

1185 As a summary, there are two general indicators an RMP will use to terminate a group:

1186 (a) Message marker: Information within a message (either ending marker, or maximum
1187 sequence number) that indicates a last message for the group. This is used by termination rules
1188 T3, T4.

1189 (b) Timing: Either the group lifespan expired, or its idle time exceeds a timeout. This is used by
1190 termination rules T1, T2. Or, due to message expiration, a group with ordering requirement cannot
1191 be delivered. This is used by termination rule T5.

1192 These termination rules apply to both ordered and unordered groups. However, these rules do
1193 NOT apply to singleton groups, which contain a single message with no sequence number.

1194 **(1) Termination by expiration (T1):**

1195 **Context:**

1196 The group had groupExpiryTime specified.

1197 **Receiver side:**

1198 Triggering event: groupExpiryTime is over.

1199 The RMP MUST close and remove the group.

1200 **Sender side:**

1201 Triggering event: groupExpiryTime is over. (Note that in that case, $\max(\text{ExpiryTime})$ is also over.)

1202 The RMP MUST close and remove the group.

1203 **(2) Termination by idle timeout (T2):**

1204 **Context:**

1205 The group had $\text{groupMaxIdleDuration}$ specified.

1206 **Receiver side:**

1207 Triggering event: The time since the last received message for the group is over
1208 $\text{groupMaxIdleDuration}$.

1209 The group MUST be closed. But unlike (T1), some of its past messages may not have expired
1210 yet. In order to make sure all potential duplicates for the group will not be delivered, the group
1211 MUST NOT be removed until $\max(\text{ExpiryTime})$ is reached, in case Duplicate Elimination is
1212 required.

1213 **Sender side:**

1214 Triggering event: The time since the last sent message for the group is over
1215 groupMaxIdleDuration.

1216 The group MUST be closed and removed.

1217 (3) Termination by completeness (T3):

1218 **Context:**

1219 No specific context.

1220 **Receiver side:**

1221 Triggering event: The RMP receives a message marked last
1222 (Request/MessageId/SequenceNum/@last="true"), which closes the group, assuming that all
1223 previous messages for the group have been received. Or, assuming that the message with ending
1224 marker has already been received, the RMP receives the last missing message in the group.

1225 The group MUST be closed. However, its removal is done according to (T1) or (T2), depending
1226 which timeout parameter was specified for the group. If no timeout parameter was specified, the
1227 group is removed once all its messages have expired: i.e. the date max(ExpiryTime) is passed.

1228 **Sender side:**

1229 Triggering event: The RMP sends a message marked last.

1230 All messages of the group have been sent. The group MUST be closed. If Guaranteed Delivery
1231 was required, the group MUST be removed once all sent messages have either been
1232 acknowledged, or their delivery failure notified. If no Guaranteed Delivery was required, the group
1233 MUST be removed immediately.

1234 **Note:** In case a message is received with an ending marker, but not all previous messages have
1235 been received, then the group remains active. No termination process is initiated yet.

1236 (4) Termination by sequence exhaustion (T4):

1237 **Context:**

1238 No specific context.

1239 **Receiver side:**

1240 Triggering event: The RMP receives a message with a sequence number with maximum value,
1241 assuming that all previous messages for the group have been received. Or, assuming that the
1242 message with maximum sequence number has already been received, the RMP receives the last
1243 missing message in the group.

1244 The group closing and removal follows the rules in T3, the message with maximum sequence
1245 number acting as a message with ending mark.

1246 **Sender side:**

1247 Triggering event: The RMP sends a message with a sequence number with maximum value.

1248 The group closing and removal follows the rules in T3, the message with maximum sequence
1249 number acting as a message with ending mark.

1250 **Note:** In case a message is received with with maximum sequence number, but not all previous
1251 messages have been received, then the group remains active. No termination process is initiated
1252 yet.

1253 **(5) Termination by ordering failure (T5):**

1254 **Context:**

1255 The group is under Guaranteed Message Ordering reliability requirement.

1256 **Receiving side:**

1257 Triggering event: In an ordered group, a received message expires before delivery.

1258 The group **MUST** be closed. The group is removed according to rule T3.

1259 **Sender Side:**

1260 Triggering event: In an ordered group, a non-acknowledged message expires.

1261 The group **MUST** be closed. The group is removed according to rule T3.

1262

1263 Summary of group termination rules for a Receiver RMP:

| <i>Group Closing</i> | <i>Group Removal</i> |
|--|-----------------------------|
| GroupExpiryTime is over | GroupExpiryTime is over |
| GroupMaxIdleDuration timeout is over | Max(ExpiryTime) is over |
| Group is complete | Max(ExpiryTime) is over |
| Group is ordered AND a non-delivered message expired | Max(ExpiryTime) is over |

1264

1265 **5.2 WSDL Operation Type**

1266 This specification supports Reliable Messaging capabilities for WSDL 1.1 [WSDL 1.1] One-way
 1267 and Request-response operation types only. While a Request-reponse operation can use any of
 1268 the three RM-Reply patterns to receive acknowledgments or faults, an One-way operation can
 1269 only use either Callback or Poll RM-Reply pattern. See the table below for a complete support
 1270 matrix:

1271

Table 18 WSDL operation types

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

1272 * The current version of the WS-Reliability protocol does not support reliability of WSDL response
 1273 messages (the "output" messages in WSDL operations). It only supports reliability of the WSDL
 1274 request ("input" messages).

1275 ** WS-I BP 1.0 disallows sending a SOAP envelope in HTTP response, so an RMP is not required
1276 to support this. However, this specification does not require an RMP to enforce this restriction (i.e.
1277 WS-I BP compliance). The receiver can do whatever the header asks for.

1278 While the specification doesn't prohibit using Callback or Poll RM-Reply patterns to receive
1279 acknowledgments or faults for a Request-response operation, it is encouraged to use Response
1280 RM-Reply pattern for such operations as the acknowledgment or the fault can be sent on the
1281 same response itself thus saving extra round trips.

1282 **5.3 Poll Reply Pattern Semantics and Usage**

1283 Guaranteed Delivery will be most commonly used for One-way messages as the Sender does not
1284 know the status of the message delivery otherwise.

1285 So the most common use case is to use AckRequested with Callback RM-Reply pattern so that
1286 the Sender can receive the acknowledgment or a fault on a listener at its end. However this
1287 pattern doesn't help when the Sender is within a firewall, as one cannot receive requests without
1288 opening a firewall, thus causing security lapses.

1289 An alternate solution is for the Sender to ask the Receiver for the receiving status of the message
1290 it has sent earlier on a different channel. Such a pattern is called the Poll RM-Reply pattern. The
1291 Sender sends a Poll request for a message it has to inquire and the Receiver sends a Poll
1292 response with the RM-Reply. The Sender can also batch multiple Poll requests for an efficient
1293 use. Receiver in such case will send RM-Replies for those messages it has received. If a Poll
1294 request is partially or completely invalid or wrong, then the Receiver sends either a
1295 InvalidPollRequest or InvalidMessageFault back.

1296 Also, a RM Poll response MUST NOT be piggybacked on a different RM Poll request.

1297 **5.4 Attachments**

1298 When this spec is used with W3C note SOAP messages with Attachments specification [SOAP
1299 with Attachments], the following rules MUST be met:

1300 1) The first MIME part MUST include whole SOAP envelope with WS-Reliability header elements.

1301 2) The charset of the Content-Header of the first MIME part MUST be either UTF-8 or UTF-16.

1302 3) Zero or more additional MIME parts MAY be included in a reliable message.

1303 4) The receiver RMP MUST deliver all MIME parts in a Reliable Message to the receiving
1304 application.

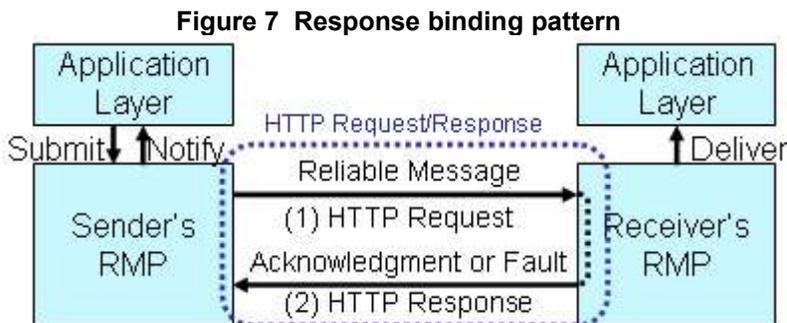
1305 6 HTTP Binding

1306 This section describes the three binding pattern - "Response", "Callback", and "Poll" binding
1307 pattern for HTTP. These binding pattern is identified by the value of ReplyPattern element (See
1308 Section4.2.3 for detail). This specification expects that the transport layer will not deliver a
1309 corrupted message to the reliability layer. When a request message contains AckRequested
1310 element, upon receipt of a reliable message, the Receiver RMP MUST send a reply. This reply
1311 MUST be either an Acknowledgment message or a Fault message. This reply MUST be sent by
1312 specified binding pattern in the ReplyPattern element of the request message.

1313 6.1 Reliable Messaging with "Response" binding pattern

1314 The Reliable Messaging Acknowledgment or Fault message MUST be sent back on the same
1315 HTTP connection with the HTTP Request that the sender initiated to send the Message. This is
1316 illustrated in Figure 7. Both Acknowledgment Message and Fault message MUST be sent back to
1317 the sender on the same HTTP connection the sender sent a message.

1318



- 1320 1) The sender initiates an HTTP connection, and sends a Message using the HTTP POST
1321 Request. Example 11 is an example of such a message.
- 1322 2) The receiver sends back an Acknowledgment message to the sender on the same connection,
1323 with the HTTP response.

1324 Example 11 Request Message with Response binding pattern

```
1325 POST /abc/servlet/wsrListener HTTP/1.0
1326 Content-Type: text/xml; charset=utf-8
1327 Host: 192.168.183.100
1328 SOAPAction: ""
1329 Content-Length: 1214
1330
1331 <?xml version="1.0" encoding="UTF-8"?>
1332 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1333   <soap:Header>
1334     <Request
1335       xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"
```

```
1336     soap:mustUnderstand="1">
1337     <MessageId groupId="mid://20040202.103832@oasis-open.org/">
1338     <SequenceNum number="0"
1339     groupExpiryTime="2005-02-02T03:00:33-31:00" />
1340     </MessageId>
1341     <ExpiryTime>2004-09-07T03:01:03-03:50</ExpiryTime>
1342     <ReplyPattern>Response</ReplyPattern>
1343     <AckRequested/>
1344     <DuplicateElimination/>
1345     <MessageOrder/>
1346     </Request>
1347 </soap:Header>
1348 <soap:Body>
1349     <Request xmlns="http://wsr-example.org/">Request Message</Request>
1350 </soap:Body>
1351 </soap:Envelope>
```

1352

1353 **Example 12 Acknowledgment Message with Response binding pattern**

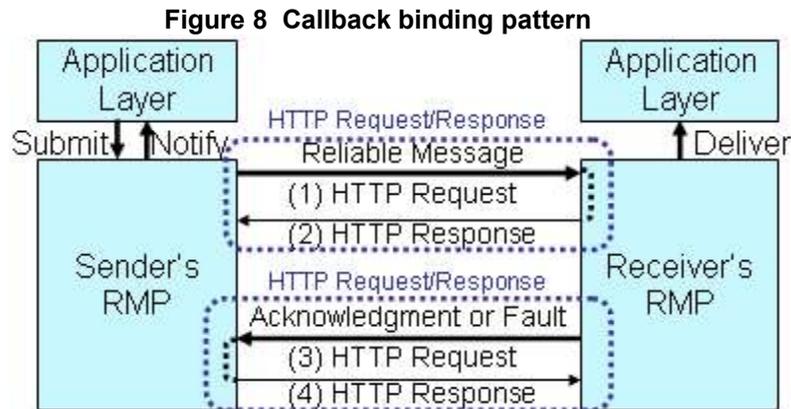
```
1354 HTTP/1.0 200 OK
1355 Server: WS-ReliabilityServer
1356 Date: Mon, 02 Feb 2004 10:38:32 GMT
1357 Content-Language: en
1358 Content-Type: text/xml; charset=utf-8
1359 Content-Length: 924
1360
1361 <?xml version="1.0" encoding="UTF-8"?>
1362 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1363     <soap:Header>
1364         <Response
1365             xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"
1366             soap:mustUnderstand="1" replyPattern="Response">
1367             <SequenceReplies groupId="mid://20040202.103832@oasis-open.org/">
1368                 <ReplyRange from="0" to="0"/>
1369             </SequenceReplies>
1370         </Response>
1371     </soap:Header>
```

1372 <soap:Body />
 1373 </soap:Envelope>
 1374

1375 6.2 Reliable Messaging with “Callback” binding pattern

1376 The Reliable Messaging Acknowledgment or Fault message MUST be sent back on a different
 1377 HTTP connection from the HTTP connection that the sender initiated to send the message. The
 1378 direction of the HTTP connection that receiver initiates is from the receiver to the sender. This is
 1379 illustrated in Figure 8.

1380



1382 (1) The sender initiates a HTTP connection, and sends a Message using HTTP POST Request.
 1383 Example 13 is an example of this message.

1384 (2) The HTTP response to the (1) has no HTTP message body. Example 14 is an example of this
 1385 HTTP response.

1386 (3) The Acknowledgment Message is sent with another HTTP connection from the receiver to the
 1387 sender. Example 15 is an example of this message.

1388 (4) The HTTP response for (3) has no HTTP message body. Example 14 is an example for this
 1389 HTTP Response.

1390

1391 Example 13 Request Message with Callback binding pattern

```

1392 POST /abc/servlet/wsrListener HTTP/1.0
1393 Content-Type: text/xml; charset=utf-8
1394 Host: 192.168.183.100
1395 SOAPAction: ""
1396 Content-Length: 1214
1397
1398 <?xml version="1.0" encoding="UTF-8"?>
1399 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1400   <soap:Header>
1401     <Request
  
```

```

1402     xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"
1403     soap:mustUnderstand="1">
1404         <MessageId groupId="mid://20040202.103832@oasis-open.org/">
1405             <SequenceNum number="0"
1406                 groupExpiryTime="2005-02-02T03:00:33-31:00" />
1407         </MessageId>
1408         <ExpiryTime>2004-09-07T03:01:03-03:50</ExpiryTime>
1409         <ReplyPattern replyTo="http://www.oasis-open.org/">Callback</ReplyPattern>
1410         <AckRequested/>
1411         <DuplicateElimination/>
1412         <MessageOrder/>
1413     </Request>
1414 </soap:Header>
1415 <soap:Body>
1416     <Request xmlns="http://wsr-example.org/">Request Message</Request>
1417 </soap:Body>
1418 </soap:Envelope>

```

1419

Example 14 HTTP response with no content

```

1420
1421 HTTP/1.0 200 OK
1422 Server: WS-ReliabilityServer
1423 Date: Mon, 02 Feb 2004 10:38:32 GMT
1424 Content-Language: en
1425 Content-Type: text/xml; charset=utf-8
1426 Content-Length: 184

```

1427

Example 15 Acknowledgment Message with Callback binding pattern

```

1428
1429 POST /xyz/servlet/wsrListener HTTP/1.0
1430 Content-Type: text/xml; charset=utf-8
1431 Host: 192.168.183.200
1432 SOAPAction: ""
1433 Content-Length: 1024
1434
1435 <?xml version="1.0" encoding="UTF-8"?>
1436 <soap:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
1437     xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

```

```

1438 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1439 xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
1440 <soap:Header>
1441   <Response
1442     xmlns="http://www.oasis-open.org/committees/wsrn/schema/1.1/SOAP1.1"
1443     soap:mustUnderstand="1" replyPattern="Callback">
1444     <SequenceReplies groupId="mid://20040202.103832@oasis-open.org/">
1445       <ReplyRange from="0" to="0"/>
1446     </SequenceReplies >
1447   </Response>
1448 </soap:Header>
1449 <soap:Body />
1450 </soap:Envelope>

```

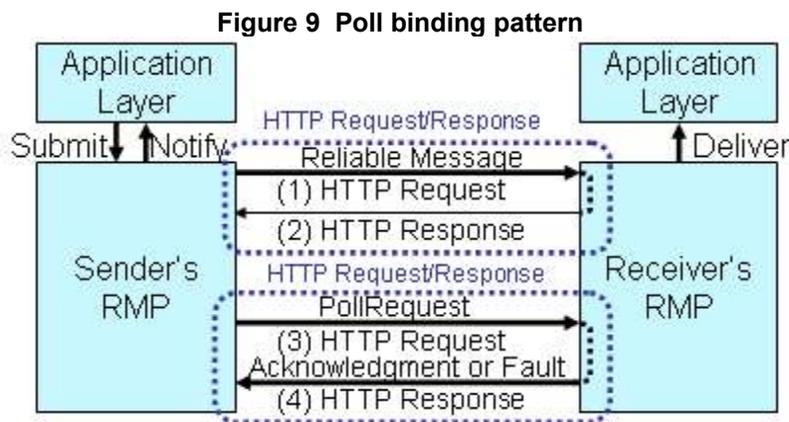
1451

1452 6.3 Reliable Messaging with "Poll" binding pattern

1453 The Reliable Messaging Acknowledgment message MAY also be sent back on a different HTTP
 1454 connection from the HTTP connection used to send the message being acknowledged. This is
 1455 illustrated in Figure 9.

1456

1457



1459 (1) The sender initiates a HTTP connection, and sends a Message using HTTP POST Request.

1460 (2) The HTTP response to the (1) has no HTTP message body. Example 14 is an example of this
 1461 HTTP response.

1462 (3) The sender initiates a HTTP connection, and sends a PollRequest message with HTTP POST
 1463 Request. Example 16 is an example of this message.

1464 (4) The HTTP response for (3) includes Acknowledgment message and/or Reliable Messaging
 1465 Fault. Example 17 is an example for this message.

1466

1467 **Example 16 PollRequest message with Poll binding pattern**

1468 POST /abc/servlet/wsrListener HTTP/1.0

1469 Content-Type: text/xml; charset=utf-8

1470 Host: 192.168.183.100

1471 SOAPAction: ""

1472 Content-Length: 1021

1473

1474 <?xml version="1.0" encoding="UTF-8"?>

1475 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >

1476 <soap:Header>

1477 <PollRequest

1478 xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"

1479 soap:mustUnderstand="1">

1480 <RefToMessageIds groupId="mid://20040202.103832@oasis-open.org/">

1481 <SequenceNumberRange from="0" to="20"/>

1482 </RefToMessageIds>

1483 </PollRequest>

1484 </soap:Header>

1485 <soap:Body />

1486 </soap:Envelope>

1487

1488 **Example 17 Acknowledgment message with Poll binding pattern**

1489 HTTP/1.0 200 OK

1490 Server: WS-ReliabilityServer

1491 Date: Mon, 02 Feb 2004 10:38:32 GMT

1492 Content-Language: en

1493 Content-Type: text/xml; charset=utf-8

1494 Content-Length: 924

1495

1496 <?xml version="1.0" encoding="UTF-8"?>

1497 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >

1498 <soap:Header>

1499 <Response

1500 xmlns="http://www.oasis-open.org/committees/wsr/schema/1.1/SOAP1.1"

1501 soap:mustUnderstand="1" replyPattern="Poll" >

1502 <SequenceReplies groupId="mid://20040202.103832@oasis-open.org/">

```
1503     <ReplyRange from="0" to="14"/>
1504     <ReplyRange from="16" to="20"/>
1505     </SequenceReplies>
1506     </Response>
1507 </soap:Header>
1508 <soap:Body />
1509 </soap:Envelope>
```

1510

1511

1512

1513

1514 7 Conformance

1515 In order to be conform to this specification, an implementation must satisfy all the following
1516 conditions:

- 1517 • It complies with the following interpretation of the keywords OPTIONAL and MAY:
1518 When these keywords apply to the behavior of the implementation, the implementation
1519 is free to support these behaviors or not, as stated in [RFC2119].
- 1520 • If it has implemented optional features and/or behavior defined in this specification, it
1521 MUST be capable of interoperating with another implementation that has not
1522 implemented the optional syntax, features and/or behavior. It MUST be capable of
1523 processing the prescribed failure mechanism for those optional features it has chosen
1524 to implement.
- 1525 • If it has chosen NOT to implement optional features, it is capable of interoperating with
1526 another implementation that has chosen to implement these. It MUST be capable of
1527 generating the prescribed failure mechanism for those optional features it has chosen
1528 to NOT implement.

1529

1530 8 References

1531 8.1 Normative References

1532 [RFC1738] "Uniform Resource Locators (URL)", T. Berners-Lee et al, RFC 1738, IESG and IETF,
1533 December 1994. Available at

1534 <http://www.ietf.org/rfc/rfc1738.txt>

1535

1536 [RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119,
1537 Bradner, S., IESG and IETF, March 1997. Available at

1538 <http://www.ietf.org/rfc/rfc2119.txt>

1539

1540 [RFC2392] "Content-ID and Message-ID Uniform Resource Locators", RFC2392, E. Levinson,
1541 IESG and IETF, August 1998. Available at

1542 <http://www.ietf.org/rfc/rfc2392.txt>

1543

1544 [RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, Tim Berners-Lee et
1545 al, IESG and IETF, August 1998. Available at

1546 <http://www.ietf.org/rfc/rfc2396.txt>

1547

1548 [RFC2616] "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, R. Fielding et al, IESG and
1549 IETF, June 1999. Available at

1550 <http://www.ietf.org/rfc/rfc2616.txt>

1551

1552 [RFC2822] "Internet Message Format", RFC 2822, P. Resnick, Editor, IESG and IETF, April 2001.
1553 Available at

1554 <http://www.ietf.org/rfc/rfc2822.txt>

1555

1556 [SOAP 1.2] "SOAP 1.2 Part 1: Messaging Framework", Martin Gudgin, Marc Hadley, Noah
1557 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, eds., W3C Recommendation, 24
1558 June 2003. Available at

1559 <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>

1560

1561 [XML] "Extensible Markup Language (XML) 1.0, Second Edition", Tim Bray et al, eds., W3C
1562 Recommendation, 6 October 2000. Available at

1563 <http://www.w3.org/TR/2000/REC-xml-20001006/>

1564

1565 [XML Namespaces] "Namespaces in XML", Tim Bray et al., eds., W3C Recommendation, 14
1566 January 1999. Available at
1567 <http://www.w3.org/TR/1999/REC-xml-names-19990114/>
1568
1569 [XML Schema] "XML Schema Part 2: Datatypes", Paul V. Biron and Ashok Malhotra, eds. W3C
1570 Recommendation, 2 May 2001. Available at
1571 <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>
1572
1573 [WS-I BP1.0] "Basic Profile Version 1.0a", Keith Ballinger, David Ehnebuske, Martin Gudgin, Mark
1574 Nottingham, Prasad Yendluri, eds., WS-I specification, 8 August 2003. Available at
1575 <http://www.ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.html>
1576

1577 **8.2 Non-normative References**

1578 [ebMS] "Message Service Specification Version 2.0", OASIS ebXML Messaging Services
1579 Technical Committee, OASIS Standard, 1 April 2002. Available at
1580 <http://www.ebxml.org/specs/ebMS2.pdf>
1581
1582 [SOAP 1.1] "Simple Object Access Protocol (SOAP) 1.1", Don Box et al, W3C Note, 8 May, 2000.
1583 Available at
1584 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
1585
1586 [WSDL1.1] "Web Services Description Language (WSDL) 1.1", Erik Christensen, Francisco
1587 Curbera, Greg Meredith, Sanjiva Weerawarana, eds., W3C Note, 15 March 2001. Available at
1588 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
1589
1590 [WSS] "OASIS Web Services Security TC", documentation in progress, OASIS Technical
1591 Committee. Committee home page at
1592 <http://www.oasis-open.org/committees/wss/>
1593
1594 [XML Schema - Part 1] "XML Schema Part 1: Structures", Henry S. Thompson, David Beech,
1595 Murray Maloney, Noah Mendelsohn, eds., W3C Recommendation, 2 May 2001. Available at
1596 <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>
1597
1598 [SOAP with Attachments] "SOAP Messages with Attachments", John J. Barton, Satish Thatte,
1599 Henrik Frystyk Nielsen, W3C Note, 11 December 2000, Available at
1600 <http://www.w3.org/TR/SOAP-attachments>
1601

1602 Appendix A. WS-Reliability Features, Properties 1603 and Compositor (Normative and Optional)

1604 I. Introduction

1605 Users of a Web Service will need to be aware of the reliability capabilities (RM capabilities) that are
1606 supported/required by the service. One practical location to advertise these capabilities is in the
1607 service description (WSDL document), which allows for publishing both abstract service
1608 definitions as well as concrete protocol details (bindings). This allows clients (or other Web
1609 services) to easily obtain information about specific capabilities such as guaranteed delivery,
1610 duplicate elimination, message ordering, and various reply patterns of a specific Web service,
1611 before calling the service. While bundling reliability capabilities with the service description may
1612 not be desirable in all cases, it is expected that this convenient approach will often be appropriate.
1613 The WSDL annotation mechanism described here is a flexible way to add such capability
1614 assertions.

1615 WS-Reliability uses the WSDL 1.1 extensibility points to define an extensible framework
1616 consisting of features, properties and compositors to address the needs of a reliable Web service
1617 to advertise its capabilities, and composability of those capabilities.

1618 The following extensibility elements relevant to RM capabilities are used:

- 1619 • feature - abstract RM capability or assertion associated with WSDL elements.
- 1620 • property - an assertion or constraint on an atomic RM capability and its value(s)
1621 associated with WSDL elements.
- 1622 • compositor - specify how features and properties are combined.

1623 An annotation composed with the above extensibility elements will specify the reliability features
1624 and properties associated with specific WSDL constructs. Features and properties represent
1625 reliability capabilities and compositors specify how these capabilities are composed.

1626 This would allow, for example, a Web service description to advertise the fact that clients invoking
1627 the service must use duplicate elimination or message ordering.

1628 I.A Notational Convention

1629 This specification uses the following namespace prefixes:

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

1630

1631 The choice of any namespace prefix is arbitrary and not semantically significant.

1632

1633 I.B Conformance

1634 Implementations of WS-Reliability are expected, though not required, to understand the WSDL
1635 extensibility points defined in this section.

1636 Understanding of these extensibility points promotes interoperability. When a WSDL document
1637 contains these extensibility points, it is through these extensibility points that a service advertises
1638 its supported and required features. Therefore it is RECOMMENDED that implementations
1639 recognize, understand and support these extensibility points.

1640 It is also possible for services to advertise features through other channels (such as UDDI) in
1641 addition to these extensibility point.

1642

1643 II. WSDL Extensibility Elements

1644 II.A Compositor

1645 The compositor semantics describe how features and properties are composed for the enclosing
1646 component (or WSDL 1.1 element). The compositor's semantics determine whether the usage of
1647 composed elements by a client to the service, is required or optional. The RM capabilities
1648 represented by these elements must all be supported by the Service. A compositor element can
1649 occur as a child element of wsdl11:portType, wsdl11:operation (which may itself be a child of
1650 wsdl11:portType or wsdl11:binding), wsdl11:binding, wsdl11:service and wsdl11:port. The
1651 compositor element utilizes the extensibility defined by WSDL 1.1. A compositor element specifies
1652 the semantics for combining its children elements. These children elements can be additional
1653 compositor, features, properties, or extensibility element(s).

1654 A compositor element is expressed by the following pseudo-syntax:

```
1655 <fnp:compositor uri="..." name="NCName"?>  
1656   [fnp:feature/> | <fnp:property/> | <fnp:compositor/> |  
1657   <extensibility-element/>]+  
1658 </fnp:compositor>
```

1659 The uri attribute of the compositor specifies its semantics. Four different compositors (URIs) and
1660 their capability-related semantics are described below. It is possible to provide additional
1661 compositors by using other URIs. The ability to define additional compositors and the existence of
1662 extensibility points (represented by "<extensibility-element>") make the framework extensible. The
1663 optional name attribute identifies the compositor. An element built with such compositors
1664 represents an RM capability.

1665 • **all:** this compositor specifies that a service invocation MUST comply with all the
1666 children elements (representing RM capability assertions). This compositor is identified
1667 by using the URI:

1668 ["http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositors/all"](http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositors/all)

1669 • **choice:** this compositor specifies that a service invocation MUST comply with exactly
1670 one of the possibly many children elements (representing RM capability assertions).
1671 This compositor is identified by using the URI:

1672 ["http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositors/choice"](http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositors/choice)

1673 • **one-or-more:** this compositor specifies that a service invocation MUST comply with at
1674 least one of the possibly many children elements (representing RM capability
1675 assertions). This compositor is identified by using the URI:

1676 "<http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositors/one-or-more>"

1677 • **zero-or-more:** this compositor specifies that a service invocation MAY comply with
1678 one or more of the children elements (representing RM capability assertions). This
1679 compositor is identified by using the URI:

1680 "<http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositors/zero-or-more>"

1681 Examples for each compositor are provided in Section VII below.

1682 Compositors specified at different WSDL components are implicitly aggregated using the 'all'
1683 compositor at the dependent WSDL component. Consider the example below,

```
1684 <wsdl11:definitions>
1685   ...
1686   <wsdl11:portType name="myPortType">
1687     <fnp:compositor uri="..." name="A">
1688       ...
1689     </fnp:compositor>
1690   ...
1691 </wsdl11:portType>
1692
1693 <wsdl11:binding name="myBinding" type="myPortType">
1694   <fnp:compositor uri="..." name="B">
1695     ...
1696   </fnp:compositor>
1697   ...
1698 </wsdl11:binding>
1699 <wsdl11:service name="myService">
1700   <wsdl11:port name="myPort" binding="myBinding">
1701     ...
1702   </wsdl11:port>
1703 </wsdl11:service>
1704 </wsdl11:definitions>
```

1705 Compositor specified at the wsdl11:portType "myPortType" and compositor specified at
1706 wsdl11:binding "myBinding" are aggregated at the dependent wsdl11:port "myPort" using the 'all'
1707 compositor. I.e., the equivalent compositor at "myPort" is,

```
1708 <fnp:compositor
1709 uri="http://www.oasis-open.org/committees/wsrn/schema/1.1/fnp/compositor/all">
1710 <fnp:compositor uri="..." name="A">
1711 </fnp:compositor>
```

1712 <fnp:compositor uri="..." name="B">

1713 ...

1714 </fnp:compositor>

1715 </fnp:compositor>

1716

1717 **II.B Feature**

1718 A feature describes an abstract RM capability or assertion associated with a WSDL element. A
1719 feature can occur only as a child of a compositor.

1720 Whether the usage of a feature is required or not is defined by the enclosing compositor(s). A
1721 feature is identified by a URI. Recognizing the URI of a feature is considered to be equivalent to
1722 understanding the feature identified by that URI.

1723 A feature element is expressed by the following pseudo-syntax:

1724 <fnp:feature uri="...">

1725 [**<fnp:compositor/>** | **<extensibility-element/>**]*

1726 </fnp:feature>

1727

1728 **II.C Property**

1729 A property is identified by a QName. A property is an assertion or constraint on a specific RM
1730 capability and its value(s) associated with WSDL elements.

1731 Typically properties are associated with a feature (but are not required to) and are described in a
1732 feature specification. The QName identifier of a property uniquely identifies the property.
1733 Recognizing the property QName identifier is considered to be equivalent to understanding the
1734 semantics associated with that property. The property QName identifier typically points a global
1735 XML Schema element declaration. A property specification typically specifies the schema that
1736 contains this global element declaration. A constraint on the set of values that a property can have
1737 is specified by a QName that identifies a XML Schema type.

1738 <fnp:property name="xs:QName">

1739 [**<fnp:value>**xs:anyType**</fnp:value>** |

1740 **<fnp:constraint>**xs:QName**</fnp:constraint>**]

1741 [**<extensibility-element/>**]*

1742 </fnp:property>

1743

1744 **III. WS-Reliability Feature**

1745 The WS-Reliability feature is identified by the URI

1746 "http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/"

1747 This feature URI identifies the WS-Reliability specification. Understanding this URI implies
1748 understanding the WS-Reliability specification.

1749

1750 **IV. WS-Reliability Properties**

1751 This section identifies properties for the WS-Reliability specification. Typically these properties
1752 would be scoped within the feature identified by the URI

1753 "http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/"

1754

1755 **IV.A. Guaranteed Delivery Property**

1756 This property is identified by the QName "wsrm:GuaranteedDelivery" and corresponds to the
1757 semantics specified by the WS-Reliability guaranteed delivery semantics. The type of this property
1758 is "xs:boolean".

1759

1760 **IV.B. Duplicate Elimination Property**

1761 This property is identified by the QName "wsrm:NoDuplicateDelivery" and corresponds to the
1762 semantics specified by the WS-Reliability duplicate elimination semantics. The type of this
1763 property is "xs:boolean".

1764

1765 **IV.C. Message Ordering Property**

1766 This property is identified by the QName "wsrm:OrderedDelivery" and corresponds to the
1767 semantics specified by the WS-Reliability message ordering semantics. The type of this property
1768 is "xs:boolean".

1769

1770 **IV.D. Reply Pattern Property**

1771 This property is identified by the QName "wsrm:ReplyPattern" and corresponds to the semantics
1772 specified by the WS-Reliability reply pattern options. The type of this property is "xs:String".
1773 (values: Response, Poll, Callback)

1774

1775 **V. Other Reliability Properties**

1776 In addition to the properties defined in section III, there are WS-Reliability properties that are used
1777 on the Sender side (usually the client side and therefore do not occur in the WSDL document).

1778 This section identifies such properties. These properties MUST NOT be specified in the WSDL
1779 document. How the properties are specified and/or represented does not affect interoperability as
1780 these properties are client-side only properties. They are defined here for convenience only.

1781

1782 **V.A. Group Expiry Time**

1783 This property is identified by the QName "wsrm:GroupExpiryTime" and corresponds to the
1784 semantics specified by the WS-Reliability group expiration time. The type of this property is
1785 xs:duration.

1786 Note: The expiry time is calculated at the time a message is sent, but adding this duration to the
1787 time the message is sent.

1788

1789 **V.B. Group Maximum Idle Duration**

1790 This property is identified by the QName "wsrm:GroupMaxIdleDuration" and corresponds to the
1791 semantics specified by the WS-Reliability group maximum idle duration. The type of this property
1792 is xs:duration.

1793

1794 **V.C. Message Expiration Time**

1795 This property is identified by the QName "wsrm:ExpiryTime" and corresponds to the semantics
1796 specified by the WS-Reliability message expiration time. The type of this property is xs:duration.

1797 Note: The expiry time is calculated at the time a message is sent, but adding this duration to the
1798 time the message is sent.

1799

1800 **V.D. Retry Maximum Time**

1801 This property is identified by the QName "wsrm:RetryMaxTimes" and corresponds to the
1802 semantics specified by the WS-Reliability maximum retry times. The type of this property is xs:int.

1803

1804 **V.E. Retry Time Interval**

1805 This property is identified by the QName "wsrm:RetryTimeInterval" and corresponds to the
1806 semantics specified by the WS-Reliability retry time interval. The type of this property is
1807 xs:duration.

1808

1809 **V.F. ReplyTo URI**

1810 This property is identified by the QName "wsrm:ReplyTo" and corresponds to the semantics
1811 specified by the WS-Reliability reply-to. The type of this property is xs:anyURI.

1812

1813 **VI. Schema**

1814 <?xml version="1.0" encoding="UTF-8" ?>

1815 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

```

1816   xmlns:wsm="http://www.oasis-open.org/committees/wsm/schema/1.1/feature/rel/"
1817   targetNamespace="http://www.oasis-open.org/committees/wsm/schema/1.1/feature/rel/"
1818   elementFormDefault="qualified" >
1819
1820   <!-- properties to be used in WSDL -->
1821   <xs:element name="GuaranteedDelivery" type="xs:boolean"/>
1822   <xs:element name="NoDuplicateDelivery" type="xs:boolean"/>
1823   <xs:element name="OrderedDelivery" type="xs:boolean"/>
1824   <xs:element name="ReplyPattern" type="xs:string"/>
1825
1826   <!-- properties to be used on the client side -->
1827   <xs:element name="GroupExpiryTime" type="xs:duration"/>
1828   <xs:element name="GroupMaxIdleDuration" type="xs:duration"/>
1829   <xs:element name="ExpiryTime" type="xs:duration"/>
1830   <xs:element name="RetryMaxTimes" type="xs:int"/>
1831   <xs:element name="RetryTimeInterval" type="xs:duration"/>
1832   <xs:element name="ReplyTo" type="xs:anyURI"/>
1833 </xs:schema>

```

1834

1835 VII. Examples

1836 VII.A Example for the "all" compositor

```

1837 <wsdl11:portType name="Example-1">
1838   <fnp:compositor uri="http://www.oasis-
1839   open.org/committees/wsm/schema/1.1/fnp/compositor/all">
1840     <fnp:feature uri="http://www.oasis-open.org/committees/wsm/schema/1.1/feature/rel/"
1841     <fnp:compositor uri="http://www.oasis-
1842   open.org/committees/wsm/schema/1.1/fnp/compositor/all">
1843     <fnp:property name="wsm:DuplicateElimination">
1844       <fnp:value>true</fnp:value>
1845     </fnp:property>
1846     <fnp:property name="wsm:OrderedDelivery">
1847       <fnp:value>true</fnp:value>
1848     </fnp:property>
1849     <fnp:property name="wsm:GuaranteedDelivery">
1850       <fnp:value>true</fnp:value>
1851     </fnp:property>

```

```
1852     </fnp:compositor>
1853 </fnp:feature>
1854 </fnp:compositor>
1855 ...
1856 </wsdl11:portType>
```

1857 In the example above, the reliability feature identified by URI "http://www.oasis-
1858 open.org/committees/wsrn/schema/1.1/feature/rel/" is required by the portType. This feature
1859 consists of three properties, all of which are required because of the semantics of the 'all'
1860 compositor that composes the three properties.

1861

1862 **VII.B Example for the "choice" compositor:**

```
1863 <wsdl11:binding name="Example-2">
1864   <fnp:compositor uri="http://www.oasis-
1865   open.org/committees/wsrn/schema/1.1/fnp/compositor/all">
1866     <fnp:feature uri="http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/"
1867     <fnp:compositor uri="http://www.oasis-
1868     open.org/committees/wsrn/schema/1.1/fnp/compositors/choice">
1869       <fnp:property name="wsrm:ReplyPattern">
1870         <value>Response</value>
1871       </fnp:property>
1872       <fnp:property name="wsrm:ReplyPattern">
1873         <value>Callback</value>
1874       </fnp:property>
1875       <fnp:property name="wsrm:ReplyPattern">
1876         <value>Poll</value>
1877       </fnp:property>
1878     </fnp:compositor>
1879   </fnp:feature>
1880 </fnp:compositor>
1881 ...
1882 </wsdl11:binding>
```

1883 In the example above, the reliability feature identified by URI "http://www.oasis-
1884 open.org/committees/wsrn/schema/1.1/feature/rel/" is required by the portType. This feature
1885 consists of three properties, of which the client must choose one.

1886

1887 **VII.C Example for the "one-or-more" compositor:**

```
1888 <wsdl11:portType name="Example-3">
```

```

1889   <fnp:compositor uri="http://www.oasis-
1890 open.org/committees/wsrn/schema/1.1/fnp/compositor/all">
1891     <fnp:feature uri="http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/"
1892       <fnp:compositor uri="http://www.oasis-
1893 open.org/committees/wsrn/schema/1.1/fnp/compositor/one-or-more">
1894       <fnp:property name="wsrm:DuplicateElimination">
1895         <fnp:value>true</fnp:value>
1896       </fnp:property>
1897       <fnp:property name="wsrm:OrderedDelivery">
1898         <fnp:value>true</fnp:value>
1899       </fnp:property>
1900       <fnp:property name="wsrm:GuaranteedDelivery">
1901         <fnp:value>true</fnp:value>
1902       </fnp:property>
1903     </fnp:compositor>
1904   </fnp:feature>
1905 </fnp:compositor>
1906 ...
1907 </wsdl11:portType>
1908

```

1909 **VII.D Example for the "zero-or-more" compositor:**

```

1910 <wsdl11:portType name="Example-4">
1911   <fnp:compositor uri="http://www.oasis-
1912 open.org/committees/wsrn/schema/1.1/fnp/compositor/all">
1913     <fnp:feature uri="http://www.oasis-open.org/committees/wsrn/schema/1.1/feature/rel/"
1914       <fnp:compositor uri="http://www.oasis-
1915 open.org/committees/wsrn/schema/1.1/fnp/compositor/zero-or-more">
1916       <fnp:property name="wsrm:DuplicateElimination">
1917         <fnp:value>true</fnp:value>
1918       </fnp:property>
1919       <fnp:property name="wsrm:OrderedDelivery">
1920         <fnp:value>true</fnp:value>
1921       </fnp:property>
1922       <fnp:property name="wsrm:GuaranteedDelivery">
1923         <fnp:value>true</fnp:value>
1924       </fnp:property>
1925     </fnp:compositor>
1926   </fnp:feature>

```

```
1927 </fnp:compositor>
1928 ...
1929 </wsdl11:portType>
```

1930 **Appendix B. Acknowledgments**

- 1931 The following individuals were members of the committee during the development of this
1932 specification:
- 1933 David Ingham, Arjuna Technologies Limited
- 1934 Joseph Chiusano, Booz Allen Hamilton
- 1935 Peter Furniss, Choreology Ltd
- 1936 Jeff Turpin, Cyclone Commerce
- 1937 Pramila Mullan, France Telecom
- 1938 Jacques Durand, Fujitsu
- 1939 Kazunori Iwasa (Secretary), Fujitsu
- 1940 Tom Rutt (Chair), Fujitsu
- 1941 Jishnu Mukerji, Hewlett-Packard
- 1942 Robert Freund, Hitachi
- 1943 Eisaku Nishiyama, Hitachi
- 1944 Nobuyuki Yamamoto, Hitachi
- 1945 Ben Bloch, Individual
- 1946 Mark Hansen, Individual
- 1947 Paolo Romano, Individual
- 1948 Dock Allen, Mitre Corporation
- 1949 Junichi Tatemura, NEC Corporation
- 1950 Alan Weissberger, NEC Corporation
- 1951 Magdolna Gerendai, Nokia
- 1952 Szabolcs Payrits, Nokia
- 1953 Mark Peel, Novell
- 1954 Sunil Kunisetty (Secretary), Oracle
- 1955 Anish Karmarkar, Oracle
- 1956 Jeff Mischkinsky, Oracle
- 1957 Marc Goodner (Secretary), SAP
- 1958 Pete Wenzel, SeeBeyond Technology Corporation
- 1959 Doug Bunting (Secretary), Sun Microsystems

- 1960 Tony Graham, Sun Microsystems
- 1961 Chi-Yuen Ng, University of Hong Kong
- 1962 Patrick Yee, University of Hong Kong
- 1963 Prasad Yendluri, webMethods, Inc.
- 1964 Scott Werden, WRQ, Inc.

1965

1966 And the following people made conditions to produce Ver1.0 of this specification:

- 1967 Colleen Evans, Sonic Software Corporation
- 1968 Dave Chappell, Sonic Software Corporation
- 1969 Doug Bunting, Sun Microsystems, Inc.
- 1970 George Tharakan, Sun Microsystems, Inc.
- 1971 Hisashi Shimamura, NEC Corporation
- 1972 Jacques Durand, Fujitsu Software Corporation
- 1973 Jeff Mischkinsky, Oracle Corporation
- 1974 Katsutoshi Nihei, NEC Corporation
- 1975 Kazunori Iwasa, Fujitsu Limited
- 1976 Martin Chapman, Oracle Corporation
- 1977 Masayoshi Shimamura, Fujitsu Limited
- 1978 Nicholas Kassem, Sun Microsystems, Inc.
- 1979 Nobuyuki Yamamoto, Hitachi Limited
- 1980 Sunil Kunisetty, Oracle Corporation
- 1981 Tetsuya Hashimoto, Hitachi Limited
- 1982 Tom Rutt, Fujitsu Software Corporation
- 1983 Yoshihide Nomura, Fujitsu Limited
- 1984 Akira Ochi, Fujitsu Limited
- 1985 Hirotaka Hara, Fujitsu Limited
- 1986 Hiroyuki Tomisawa, Hitachi Limited
- 1987 Katsuhisa Nakazato, Fujitsu Limited
- 1988 Masahiko Narita, Fujitsu Limited
- 1989 Nobuyuki Saji, NEC Corporation
- 1990 Shuichi Imabayashi, Fujitsu Limited

1991 Appendix C. Revision History

1992 [This appendix is optional, but helpful. It should be removed for specifications that are at OASIS
 1993 Standard level.]

| Rev | Date | By Whom | What |
|------------|-------------|----------------|--|
| 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 |

| Rev | Date | By Whom | What |
|------------|-------------|----------------|--|
| 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 : http://lists.oasis-open.org/archives/wsrn/200310/msg00054.html</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> |

| Rev | Date | By Whom | What |
|------------|-------------|----------------|--|
| WD-0.86 | -2004-01-14 | Kazunori Iwasa | <p>Updated for comments at : http://www.oasis-open.org/archives/wsrn/200401/msg00010.html</p> <p>except for:</p> <ul style="list-style-type: none"> - More faults for Tables1 (Need to list up all faults) - 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.) - Figure1,2,and3 "New processor Entity" (Want to confirm with TC member) -New terminologies for "Group Termination", "Removal", "Complete", and others. (Needs definitions) <p>--</p> <p>And other editorial updates.</p> |

| Rev | Date | By Whom | What |
|------------|-------------|----------------|--|
| WD-0.87 | -2004-01-15 | Kazunori Iwasa | <p>Updated for:</p> <p>Prelim Wed minutes on 1/14/2004 at: http://www.oasis-open.org/archives/wsrn/200401/msg00038.html.</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 -> MessageId)</p> |

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

| Rev | Date | By Whom | What |
|------------|-------------|----------------|---|
| WD-0.90 | -2004-01-26 | Kazunori Iwasa | <p>Updated for remaining action items at: http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5089/Minutes-Jan04f2f.htm</p> <p>This includes :</p> <p>1) 2.4: Message Identifier -> MessageId Sequence Number -> 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 -> RefToMessageIds value -> 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> |

| Rev | Date | By Whom | What |
|------------|-------------|----------------|--|
| | | | <p>Remaining Action items and editorial changes for 2004-01-26(0.90):</p> <ol style="list-style-type: none"> 1. Consistency of word: e.g. sender RMP, sending RMP or sender's RMP 2. Removing MAY, Optional 3. NotSupportedFeaturesFault 4. Explanatin for cardinality and others 5. SOAP1.2 statement in the new section in 3 on rm:fault element 6. Update fault in section3 7. Examples |
| WD-0.91 | -2004-02-02 | Kazunori Iwasa | <p>Updated for remaining action items at:</p> <p>http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5089/Minutes-Jan04f2f.htm</p> <p>or</p> <p>http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5090/Action%20Item%20List%20from%20Jan%20Face%20To%20Face%20Meeting.htm</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 & 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> |

| Rev | Date | By Whom | What |
|------------|-------------|----------------|--|
| 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>http://www.oasis-open.org/archives/wsrn/200402/msg00038.html</p> <p>http://www.oasis-open.org/archives/wsrn/200402/msg00068.html</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>http://www.oasis-open.org/archives/wsrn/200402/msg00080.html</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> |

| Rev | Date | By Whom | What |
|------------|-------------|----------------|---|
| WD-0.98 | -2004-02-26 | Kazunori Iwasa | <p>This was updated with minutes at: http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/5630/MinutesWSRMTTC021704.htm and http://www.oasis-open.org/archives/wsrn/200402/msg00223.html</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: http://www.oasis-open.org/archives/wsrn/200402/msg00161.html</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: http://www.oasis-open.org/apps/org/workgroup/wsrn/email/archives/200403/msg00035.html</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> |

| Rev | Date | By Whom | What |
|------------|-------------|----------------|--|
| WD-0.992 | -2004-03-10 | Kazunori Iwasa | This was updated with a minutes at: http://www.oasis-open.org/archives/wsrn/200403/msg00084.html And some editorial updates. |
| WD-0.993 | -2004-04-19 | Kazunori Iwasa | This was updated with some minutes or proposed resolution at: http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/6167/MinutesWSRMTTC032304-Approved.htm , http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/6252/MinutesWSRMTTC03304.htm , http://www.oasis-open.org/apps/org/workgroup/wsrn/download.php/6328/MinutesWSRMTTC040604.htm , http://www.oasis-open.org/archives/wsrn/200404/msg00034.html , and http://www.oasis-open.org/archives/wsrn/200404/msg00021.html with editorial updates. |
| | | | |
| | | | |

1994

1995

1996 **Appendix D. Futures List**

1997 The features and issues in the table below are listed as forward-looking statements regarding
1998 possible enhancements or the evolution of this specification.

1999

| | Category | Details |
|---|-----------------|---|
| 1 | WSDL | Define WSDL extensions profiling the use of RM SOAP extensions. |
| | | |
| | | |

2000

2001 Appendix E. Notices

2002 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
2003 that might be claimed to pertain to the implementation or use of the technology described in this
2004 document or the extent to which any license under such rights might or might not be available;
2005 neither does it represent that it has made any effort to identify any such rights. Information on
2006 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
2007 website. Copies of claims of rights made available for publication and any assurances of licenses
2008 to be made available, or the result of an attempt made to obtain a general license or permission
2009 for the use of such proprietary rights by implementors or users of this specification, can be
2010 obtained from the OASIS Executive Director.

2011 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
2012 applications, or other proprietary rights which may cover technology that may be required to
2013 implement this specification. Please address the information to the OASIS Executive Director.

2014 **Copyright © OASIS Open 2003-2004. All Rights Reserved.**

2015 This document and translations of it may be copied and furnished to others, and derivative works
2016 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
2017 published and distributed, in whole or in part, without restriction of any kind, provided that the
2018 above copyright notice and this paragraph are included on all such copies and derivative works.
2019 However, this document itself does not be modified in any way, such as by removing the copyright
2020 notice or references to OASIS, except as needed for the purpose of developing OASIS
2021 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
2022 Property Rights document must be followed, or as required to translate it into languages other
2023 than English.

2024 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
2025 successors or assigns.

2026 This document and the information contained herein is provided on an "AS IS" basis and OASIS
2027 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
2028 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
2029 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
2030 PARTICULAR PURPOSE.