



Web Services Reliable Messaging TC WS-Reliability 1.1

Working Draft 1.086, 23 August 2004

Document identifier:

oasis-wsrm-ws_reliability-1.1-spec-wd-1.086

Location:

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

Editors:

Principal editor:

Kazunori Iwasa, Fujitsu Limited <kiwasa@jp.fujitsu.com>

Assisting editors:

Jacques Durand, Fujitsu Software Corporation <jdurand@us.fujitsu.com>

Tom Rutt, Fujitsu Software Corporation <trutt@us.fujitsu.com>

Mark Peel, Novell, Inc. <mpeel@novell.com>

Sunil Kunisetty, Oracle Corporation <Sunil.Kunisetty@oracle.com>

Doug Bunting, Sun Microsystems <Doug.Bunting@sun.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:

Committee members should send comments on this specification to the warm@lists.oasis-open.org list. Others should use the comment form at http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=warm.

For information on whether any patents that may be essential to implementing this specification have been disclosed 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/warm/>).

If necessary, the errata page for this version of of the specification will be located at <http://www.oasis-open.org/committees/warm/documents/errata/1.1/index.html>.

33 Table of Contents

34	1 Introduction.....	6
35	1.1 Purpose of WS-Reliability.....	6
36	1.2 Definition and Scope of Reliable Messaging.....	6
37	1.3 Notational Conventions.....	7
38	1.4 Relation to Other Specifications.....	8
39	1.5 Terminology.....	9
40	2 Messaging Model.....	11
41	2.1 Messaging Context.....	11
42	2.2 RMP Operations and Their Invocation.....	11
43	2.2.1 Binding between WSDL Operation Types and RMP Invocations.....	12
44	2.3 Assumed SOAP Message Exchange Patterns.....	12
45	2.4 Message Reply Patterns.....	13
46	2.4.1 Response RM-Reply Pattern.....	13
47	2.4.2 Callback RM-Reply Pattern.....	14
48	2.4.3 Poll RM-Reply Pattern.....	14
49	2.5 Message Identification and Grouping.....	15
50	3 Reliability Agreement and Features.....	16
51	3.1 RM Agreement.....	16
52	3.1.1 Definition.....	16
53	3.1.2 RM Agreement Items.....	16
54	3.1.3 Scope of an Agreement Item.....	17
55	3.1.4 Rules.....	18
56	3.1.5 Creation, Representation and Deployment of RM Agreements.....	18
57	3.1.6 RM Capability.....	18
58	3.2 Main Reliability Features.....	18
59	3.2.1 Guaranteed Delivery.....	19
60	3.2.2 Duplicate Elimination.....	19
61	3.2.3 Guaranteed Message Ordering.....	20
62	4 Message Format.....	21
63	4.1 Structure.....	21
64	4.2 Request Element.....	23
65	4.2.1 Element: Request/MessageId.....	24
66	4.2.2 Element: Request/ExpiryTime.....	26
67	4.2.3 Element: Request/ReplyPattern.....	27

68	4.2.4 Element: Request/AckRequested.....	28
69	4.2.5 Element: Request/DuplicateElimination.....	29
70	4.2.6 Element: Request/MessageOrder.....	29
71	4.2.7 Example.....	29
72	4.3 PollRequest Element.....	30
73	4.3.1 Element: PollRequest/ReplyTo.....	31
74	4.3.2 Element: PollRequest/RefToMessageIds.....	32
75	4.3.3 Example.....	33
76	4.4 Response Element.....	34
77	4.4.1 Element: Response/NonSequenceReply.....	35
78	4.4.2 Element: Response/SequenceReplies.....	36
79	4.4.3 Example.....	37
80	4.5 Fault Codes For Reliable Messaging Failures.....	37
81	4.5.1 Message Format Faults.....	38
82	4.5.2 Message Processing Faults.....	40
83	4.5.3 RM Fault Examples.....	41
84	4.6 Extensibility Features of Schema.....	41
85	5 Operational Aspects and Semantics.....	43
86	5.1 Message Group Life Cycle.....	43
87	5.1.1 Group Termination.....	43
88	5.1.2 Group Termination Parameters.....	44
89	5.1.3 Termination Rules.....	45
90	5.2 Attachments.....	48
91	6 HTTP Binding.....	49
92	6.1 Reliable Messaging with Response RM-Reply Pattern.....	50
93	6.2 Reliable Messaging with Callback RM-Reply Pattern.....	52
94	6.3 Reliable Messaging with Poll RM-Reply Pattern.....	54
95	6.3.1 Synchronous Poll RM-Reply Pattern.....	54
96	6.3.2 Asynchronous Poll RM-Reply Pattern.....	56
97	7 Conformance.....	59
98	8 References.....	60
99	Appendix A. Schema (Normative).....	62
100	Appendix B. WS-Reliability Features, Properties and Compositors (Normative and Optional).....	63
101	B.1. Introduction.....	63
102	B.2. Conformance.....	63
103	B.3. WSDL Extensibility Elements.....	64
104	B.3.1. Compositor.....	64
105	B.3.2. Feature.....	65

106	B.3.3. Property.....	66
107	B.4. WS-Reliability Feature.....	66
108	B.5. WS-Reliability Properties.....	66
109	B.5.1. Guaranteed Delivery Property.....	66
110	B.5.2. Duplicate Elimination Property.....	66
111	B.5.3. Message Ordering Property.....	66
112	B.5.4. Reply Pattern Property.....	67
113	B.6. Compositor Examples.....	67
114	B.6.1. Example for the "all" compositor.....	67
115	B.6.2. Example for the "choice" compositor.....	68
116	B.6.3. Example for the "one-or-more" compositor.....	69
117	B.6.4. Example for the "zero-or-more" compositor.....	69
118	Appendix C. Acknowledgments.....	70
119	Appendix D. Notices.....	72

120 List of Tables

121	Table 1 Labels.....	8
122	Table 2 Prefixes.....	8
123	Table 3 RM Agreement Items.....	17
124	Table 4 Request Element.....	23
125	Table 5 MessageId Element.....	24
126	Table 6 SequenceNum Element.....	25
127	Table 7 ExpiryTime Element.....	26
128	Table 8 ReplyPattern Element.....	27
129	Table 9 Value Element.....	27
130	Table 10 ReplyTo Element.....	28
131	Table 11 BareURI Element.....	28
132	Table 12 AckRequested Element.....	29
133	Table 13 DuplicateElimination Element.....	29
134	Table 14 MessageOrder Element.....	29
135	Table 15 PollRequest Element.....	31
136	Table 16 ReplyTo Element.....	31
137	Table 17 BareURI Element.....	32
138	Table 18 RefToMessageIds Element.....	32
139	Table 19 SequenceNumRange Element.....	33
140	Table 20 Response Element.....	35
141	Table 21 NonSequenceReply Element.....	35
142	Table 22 SequenceReplies Element.....	36
143	Table 23 ReplyRange Element.....	36
144	Table 24 Invalid Message Format Fault Code Values.....	39
145	Table 25 Messaging Processing Failure Fault Code Values.....	40
146	Table 26 Conditions for terminating a group – Receiving RMP.....	48
147	Table 27 Conditions for terminating a group – Sending RMP.....	48
148	Table 28 WS-Reliability Schema Prefixes.....	62

149 1 Introduction

150 1.1 Purpose of WS-Reliability

151 WS-Reliability is a SOAP-based ([SOAP 1.1] and [SOAP 1.2 Part 1]) specification that fulfills
152 reliable messaging requirements critical to some applications of Web Services. SOAP over HTTP
153 [RFC2616] is not sufficient when an application-level messaging protocol must also guarantee
154 some level of reliability and security. This specification defines reliability in the context of current
155 Web Services standards. This specification has been designed for use in combination with other
156 complementary protocols (see **Section 1.4**) and builds on previous experiences (e.g., ebXML
157 Message Service [ebMS].)

158 1.2 Definition and Scope of Reliable Messaging

159 Reliable Messaging (RM) is the execution of a transport-agnostic, SOAP-based protocol providing
160 quality of service in the reliable delivery of messages. There are two aspects to Reliable
161 Messaging; both must be equally addressed when specifying RM features:

- 162 (1) **The “wire” protocol** aspect. RM is a protocol, including both specific message headers
163 and specific message choreographies, between a sending party and a receiving party.
- 164 (2) **The quality of service (QoS)** aspect. RM defines a quality of messaging service to the
165 communicating parties, viz., the users of the messaging service. This assumes a
166 protocol between these users and the provider of this service (i.e., the reliable
167 messaging middleware). This protocol is defined by a set of abstract operations: Submit,
168 Deliver, Notify, Respond (defined in **Section 1.5**).

169 Reliable messaging requires the definition and enforcement of contracts between:

- 170 • The Sending and Receiving message processors (contracts about the wire protocol)
- 171 • The messaging service provider and the users of the messaging service (contracts about
172 quality of service).

173 Each major RM feature will be defined as a composition of these two types of contract.

174 **Example:** Guaranteed message delivery is defined as both (1) a messaging protocol involving
175 Acknowledgment Indications and specific message headers and (2) as a rule guaranteeing if
176 “Submit” completes successfully for a payload on the sending side, “Deliver” completes
177 successfully for this payload on the receiving side or “Notify” (of failure) will be invoked on the
178 sending side.

179 **Figure 1** shows all of the reliability contracts (both QoS and protocol) binding the Reliable
180 Messaging entities (a producer of reliable messages, a consumer of reliable messages, and the
181 two Reliable Messaging Processors or RMPs). The direction of the arrows for the QoS contract
182 abstract operations, shown in **Figure 1**, represents the direction of information flow associated
183 with the operation.

184 **Note:**

185 This specification does not make any assumption about the implementation of a messaging
186 service user component (Producer or Consumer components in **Figure 1**): such a component
187 could be an application, a queuing or logging system, a database, a SOAP node, or the next
188 handler in the message processing chain. The QoS contracts concern only the conditions of
189 invocation of the “Deliver”, “Submit”, “Respond” and “Notify” operations. The interpretation of
190 these operations is a matter of implementation.

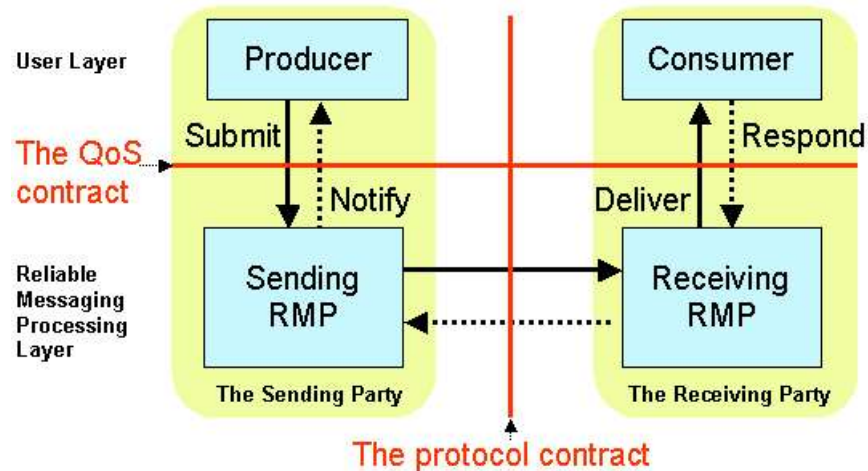


Figure 1 Reliable Messaging Contracts

191 The current specification defines the following reliability features:

- 192 • Guaranteed message delivery, or At-Least-Once delivery semantics.
- 193 • Guaranteed message duplicate elimination, or At-Most-Once delivery semantics.
- 194 • Guaranteed message delivery and duplicate elimination, or Exactly-Once delivery semantics.
- 195
- 196 • Guaranteed message ordering for delivery within a group of messages.

197 Some messaging features are out of scope for this specification. They are:

- 198 • Routing features. This specification addresses end-to-end reliability and is not concerned with intermediaries. The mechanisms described are orthogonal to routing techniques and can be used in combination with them.
- 199
- 200
- 201 • Transactions. Transactional messaging ensures the integrity of exchange patterns that involve possibly several messages. Failure conditions may involve application-level decisions based on message payload interpretation. This specification is concerned with the reliability of individual messages from submission to delivery; it ignores any interpretation of these messages.
- 202
- 203
- 204
- 205

206 Reliability is often associated with quantitative measures in QoS areas other than Web services (e.g., networking). Thresholds such as rate of failures, minimal size of persistent store, average latency, and quantitative measures that may appear in service level agreements (SLAs) are out of scope for this version.

210 1.3 Notational Conventions

211 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].

216 All text in this specification is normative, except the following:

- 217 • examples
- 218 • notes (identified with a preceding "Note" header)

219 • appendices not explicitly identified as normative

220 **Section 4** includes tables to explain each message header element. The meaning of the labels in
221 these tables is as follows:

<i>Label</i>	<i>Meaning</i>
Cardinality	A constraint on the number of instances of the element, as allowed in its enclosing element (e.g., “0 or 1” means means the element may be either absent or present only once in its enclosing element).
Value	A type or format for a value of the element.
Attributes	Attribute names for the element. The type or format for the attribute value is included in parentheses.
Child elements	Elements allowed as direct descendants of the element.

Table 1 Labels

222 This specification uses the following namespace prefixes:

<i>Prefix</i>	<i>Namespace</i>
soap	http://schemas.xmlsoap.org/soap/envelope/
soap12	http://www.w3.org/2003/05/soap-envelope
wstrm	http://docs.oasis-open.org/wstrm/2004/06/ws-reliability-1.1.xsd
xs	http://www.w3.org/2001/XMLSchema/
wSDL11	http://schemas.xmlsoap.org/wSDL/
fnp	http://docs.oasis-open.org/wstrm/2004/06/fnp-1.1.xsd
wstrmfp	http://docs.oasis-open.org/wstrm/2004/06/wstrmfp-1.1.xsd
ref	http://docs.oasis-open.org/wstrm/2004/06/reference-1.1.xsd

Table 2 Prefixes

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

224 XPath [XPath 1.0] is used to refer to header elements, in particular in **Section 4**.

225 **1.4 Relation to Other Specifications**

226 • **W3C SOAP 1.1/1.2:** SOAP 1.1 [SOAP 1.1] and SOAP 1.2 [SOAP 1.2 Part 1] are the
227 base protocols for this specification. This specification defines reliable messaging
228 protocol features expressed as extension header blocks embedded in the SOAP
229 Header.

230 • **OASIS ebXML Message Service Specification 2.0:** The reliable messaging
231 mechanism defined in the ebXML Message Service Specification 2.0 [ebMS] is
232 implemented in a number of products and open source efforts, many of which have
233 undergone interoperability testing. WS-Reliability borrows from this technology.

234 • **OASIS Web Services Security: SOAP Message Security 1.0:** This specification
235 defines reliability independently from security, each of these features mapping to

236 different SOAP header extensions. Although both features can be used in combination,
237 the specification does not attempt to compose them in a more intricate way, nor does it
238 attempt to profile their combination. This specification can be used with OASIS Web
239 Services Security: SOAP Message Security 1.0 [WSS].

- 240 • **WS-I Basic Profile 1.1:** This specification defines how to use reliability in compliance
241 with WS-I Basic Profile 1.1 [WS-I BP 1.1].

242 **1.5 Terminology**

243 Some of these definitions may reference other definitions, either within or outside of the
244 terminology section.

245 **Reliable Messaging (RM):**

246 The act of processing the set of transport-agnostic SOAP Features defined by WS-Reliability,
247 which results in a protocol supporting quality of service features such as guaranteed delivery,
248 duplicate message elimination, and message ordering.

249 **Reliable Messaging Processor (RMP):**

250 A SOAP processor and other infrastructure capable of performing Reliable Messaging as
251 described by this specification. With regard to the transmission of a Reliable Message from one
252 RMP to another, the former is referred to as the Sending RMP and the latter as the Receiving
253 RMP. An RMP may act in both roles.

254 **Reliable Message:**

255 A SOAP message containing a <wsrm:Request> header block.

256 **Payload:**

257 A subset of the message data intended for the Consumer or Producer of the Reliable Message
258 and provided by the Producer or Consumer respectively.

259 **Producer (or Payload Producer)**

260 An abstract component that produces the payload of a message to be sent. An example of a
261 Producer is an application component able to invoke an RMP to send a payload.

262 **Consumer (or Payload Consumer)**

263 An abstract component that consumes the payload of a received message after it has been
264 processed by the Receiving RMP. Examples of Consumers are: an application component called
265 back when a message is received, a queuing device storing received payloads.

266 **Deliver:**

267 An abstract operation that transfers a payload from Receiving RMP to Consumer.

268 **Submit:**

269 An abstract operation that transfers a payload from Producer to Sending RMP – for example, a
270 request to the Sending RMP to handle the payload subject to a reliability agreement.

271 **Respond:**

272 An abstract operation that transfers a payload from Consumer to Receiving RMP as a response to
273 a previously received Reliable Message.

274 **Notify:**

275 An abstract operation that makes available to the Producer a failure status of a previously sent
276 message (e.g., a notification the Sending RMP failed to send a Reliable Message) or transfers a
277 payload received as a response from Sending RMP to Producer.

278 **RMP Operations:**

279 Deliver, Submit, Respond and Notify are also called “RMP operations”. These abstract operations
280 control the transfer of payload data (and, in one case, failure information) between the RMP and a
281 user component (Producer or Consumer). An RMP operation is not necessarily implemented by
282 an RMP, but it must be either supported in some way by an RMP or invoked by the RMP.

283 **Message Identifier:**

284 A message header value or a combination of message header values that uniquely identifies a
285 Reliable Message. This identifier is meaningful only to the reliability features described here.

286 **Duplicate Message:**

287 A message is a duplicate of another message if it has same Message Identifier.

288 **Message Delivery:**

289 Completion of the Deliver operation for a Reliable Message.

290 **Acknowledgment Indication:**

291 An indication that refers to a previous message delivered by the Receiving RMP. An
292 Acknowledgment Indication signals that the acknowledged message has been successfully
293 delivered (that is, the message has satisfied all of the reliability requirements placed on it for
294 delivery).

295 **Reliable Messaging Fault Indication (RM Fault):**

296 An indication referring to a previous message that encountered a Reliable Messaging fault
297 condition at the Receiving RMP: it signals to the Sending RMP of the referred message that there
298 was a failure to invoke the Deliver operation for the message.

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

300 An indication – either an Acknowledgment Indication or a Reliable Messaging Fault Indication –
301 referring to a previous Reliable Message.

302 **Response, Callback and Poll RM-Reply Patterns:**

303 See [Section 2.5](#).

304 **PollRequest Message:**

305 A message from the Sending RMP to the Receiving RMP that requests RM-Replies for its
306 identified set of previously sent Reliable Messages.

307 **Intermediary:**

308 A SOAP node between a Sending RMP and a Receiving RMP.

309 **Publish (an RM-Reply):**

310 The set of mechanisms that make an RM-Reply available to the Sending RMP. The particular
311 mechanism used for a given Publish operation depends on the RM-Reply Pattern ([Section 2.5](#))
312 requested within the Reliable Message that elicited the Publish.

313 2 Messaging Model

314 2.1 Messaging Context

315 The Reliable Messaging Model described in this document makes the following assumptions
316 about SOAP messaging and its relation to the RMP behavior:

- 317 • **Intermediary transparency.** SOAP Intermediaries do not play any active role in the
318 reliability mechanisms. They can be abstracted from the communication between
319 Sending RMP and Receiving RMP: the RMPs are the only parties involved in
320 implementing the RM protocol (e.g., for handling RM-Replies). There is no role for an
321 RMP other than Receiving RMP or Sending RMP. **Figure 2** illustrates this model.
- 322 • **Message integrity.** For the reliability mechanisms described here to fulfill the reliability
323 contract, this specification strongly RECOMMENDS that message header integrity be
324 guaranteed end-to-end by using adequate security options such as those described in
325 Web Services Security: SOAP Message Security 1.0 [WSS].

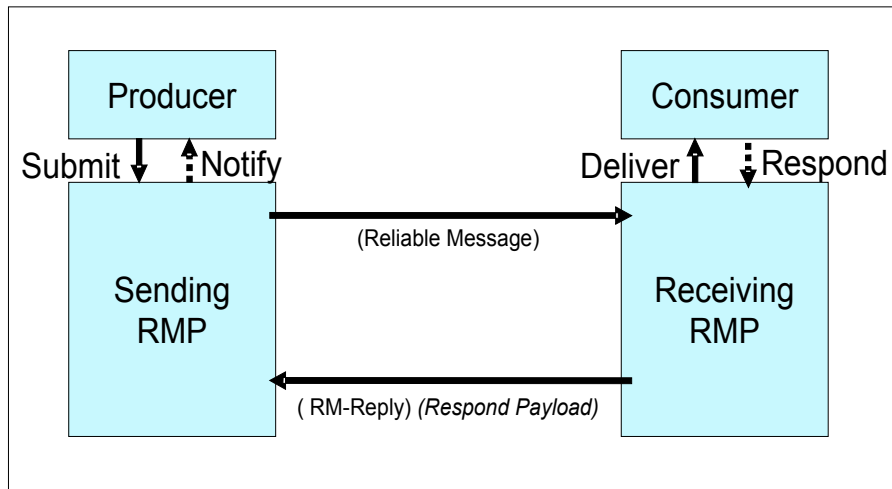


Figure 2 Messaging Model

326 2.2 RMP Operations and Their Invocation

327 Four operations (Submit, Deliver, Respond and Notify) are used to model the reliability contracts
328 between an RMP and its users (Producer and Consumer components).

329 These operations and executable components are defined abstractly to simplify discussion of the
330 WS-Reliability protocol, not to imply a particular API or component separation. No requirement is
331 made herein about how these operations should be implemented, which component should
332 implement them, or whether an implementation should explicitly represent them. The operations
333 themselves describe a transfer of information (payload or failure notice) between an RMP and
334 associated external components (Producer, Consumer).

335 The separations assumed here between the RMPs and their external components indicate the
336 expected value of placing WS-Reliability support within an infrastructure component. However,
337 any implementation choice leading to the externally observable properties describe in this
338 specification is equally valid.

339 For example, a Receiving RMP could put a received payload in a queue; later, an application
340 component gets the payload from that queue. This situation could be modeled in two different
341 ways: (1) the queuing middleware is the Consumer, in which case the delivery is over when the
342 payload is placed in the queue, (2) the application component is the Consumer, in which case the
343 delivery is over when the payload is read by the application. Note that the reliability contracts will
344 differ in each case and that it is an implementation choice to decide the precise point at which the
345 reliability contract is considered fulfilled.

346 The following requirements are associated with the use of RMP operations:

- 347 • For every valid and non-expired message it receives, a Receiving RMP MUST invoke the
348 Deliver operation after the associated reliability requirements (ordering, duplicate
349 elimination) have been satisfied.
- 350 • The Sending RMP is NOT REQUIRED to invoke the Notify operation for communicating
351 the status of every Reliable Message to a Producer. Only the failure status and available
352 Consumer payload cases need be reported.
- 353 • An invocation of Deliver is not always matched by an invocation of Respond; the
354 Consumer is NOT REQUIRED to invoke Respond for every Reliable Message delivered.
355 A Receiving RMP MUST be capable of mapping a pair of Deliver and Respond
356 invocations to an instance of SOAP Request-response MEP (See 2.3)

357 The basic exchange patterns described in the following section derive from the above messaging
358 assumptions. Reliability features defined in this specification will in turn rely on these patterns.

359 **2.2.1 Binding between WSDL Operation Types and RMP Invocations**

360 This specification supports Reliable Messaging capabilities for WSDL 1.1 [WSDL 1.1] One-way
361 and Request-response operation types only. That is, a WSDL instance describing the Consumer
362 interface would use one of these two operations. Assuming a Sending RMP (or S-RMP) and a
363 Receiving RMP (or R-RMP), the operations in such a WSDL instance MUST bind with the RMP
364 operations in the following way:

- 365 • A successful WSDL One-way operation maps to a sequence of RMP invocations of the
366 form: S-RMP.Submit(p) + R-RMP.Deliver(p), where (p) is the payload sent in the request
367 (input message) of the operation described in WSDL.
- 368 • A successful WSDL Request-response operation maps to a sequence of RMP
369 invocations of the form: S-RMP.Submit(p) + R-RMP.Deliver(p) + R-RMP.Respond(p2) +
370 S-RMP.Notify(p2), where (p) is the payload sent in the request and (p2) is the payload
371 returned in the response (output message) of the operation described in WSDL.

372 **2.3 Assumed SOAP Message Exchange Patterns**

373 Although SOAP [SOAP 1.1] was initially defined as a one-way messaging protocol, support for
374 other exchange patterns [SOAP 1.1], message exchange patterns (MEPs) [SOAP 1.2 Part 2], and
375 operations [WSDL 1.1] has been described. For example, SOAP over HTTP was principally
376 described in terms of a request-response exchange pattern in [SOAP 1.1], bound to either One-
377 way or Request-response operations in [WSDL 1.1] and restricted (especially with regard to the
378 meaning of a One-way operation) in [WS-I BP 1.1]. Described below are two MEPs – called here
379 SOAP MEPs – of interest for the RM features specified herein and derived from the terminology in
380 those specifications. We use these terms to describe how the RMPs send and receive SOAP
381 messages over the underlying transfer protocol.

382 An RMP MUST know which SOAP MEP is in use when sending or receiving a Reliable Message.
383 A WSDL instance is just one way among many to specify to an RMP a message's binding to a
384 SOAP MEP.

385 **SOAP One-way MEP:**

386 From an RMP perspective, support for this MEP assumes the following:

- 387 • The Sending RMP (as a SOAP node) is able to initiate the sending of a SOAP envelope
388 over the underlying protocol (i.e., not as a result of a previous protocol action such as an
389 HTTP GET or POST).
- 390 • No response containing a SOAP envelope is sent back – although a non-SOAP
391 response (e.g., an HTTP error code) may be returned.

392 **SOAP Request-response MEP:**

393 From an RMP perspective, support for this MEP assumes the following:

- 394 • The Sending RMP is able to initiate the sending of a SOAP envelope over the underlying
395 protocol.
- 396 • The Receiving RMP can send back a message with a SOAP envelope (called a
397 response) after somehow associating the response with the request.

398 **2.4 Message Reply Patterns**

399 There are three ways to publish an RM-Reply (Acknowledgment Indication or Fault Indication):

400 **2.4.1 Response RM-Reply Pattern**

401 When the Response RM-Reply Pattern is in use, the following sequence of exchanges MUST
402 occur:

403 Step 1: The Sending RMP sends the Reliable Message in a request of a SOAP Request-
404 response MEP instance.

405 Step 2: The Receiving RMP sends the RM-Reply in the response message of the same
406 SOAP MEP instance.

407 **Figure 3** shows this reply pattern.

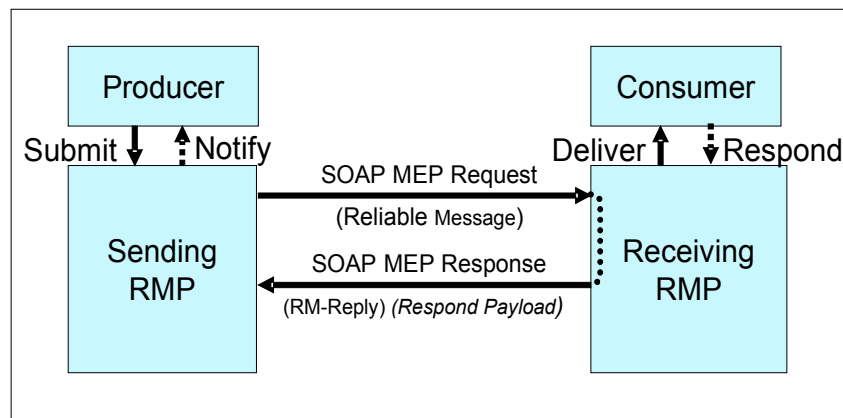


Figure 3 Response RM-Reply Pattern

408 The Response RM-Reply Pattern MUST NOT be used for WSDL One-way operations to the
409 Consumer.

410 2.4.2 Callback RM-Reply Pattern

411 When the Callback RM-Reply Pattern is in use, the following sequence of exchanges MUST
412 occur:

413 Step 1: The Sending RMP sends the Reliable Message in the SOAP MEP instance required
414 by this Producer-Consumer exchange. This MEP instance may be either Request-response
415 or One-way.

416 Step 2: The Receiving RMP sends the RM-Reply. Except when the RM Reply is bundled
417 with a Reliable Message (as described in **Section 4.4**), the RMP MUST send this RM-Reply
418 using a SOAP One-way MEP.

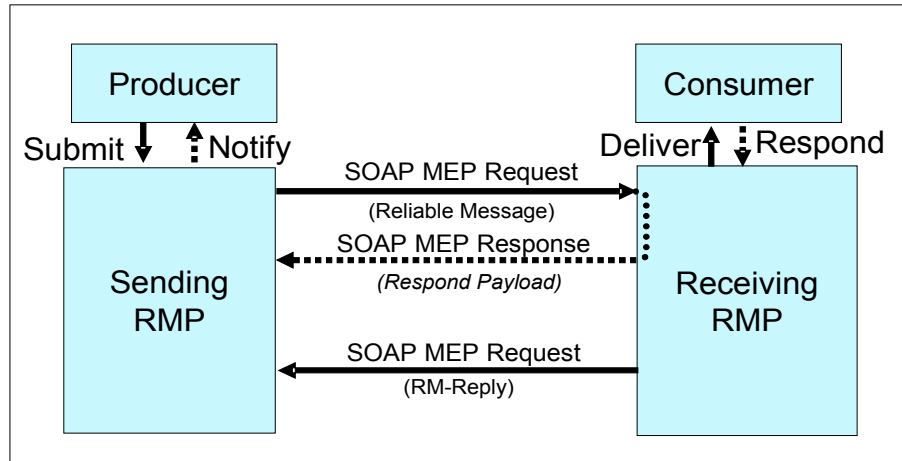


Figure 4 Callback RM-Reply Pattern

419 **Figure 4** shows this reply pattern. The dashed arrows indicate the SOAP message returned when
420 a SOAP Request-response MEP is used to send the Reliable Message.

421 2.4.3 Poll RM-Reply Pattern

422 When the Poll RM-Reply Pattern is in use, the following sequence of exchanges MUST occur:

423 Step 1: The Sending RMP sends the Reliable Message in the SOAP MEP instance required
424 by this Producer-Consumer exchange. This MEP instance may be either Request-response
425 or One-way.

426 Step 2: The Sending RMP issues a message with a PollRequest element in a new SOAP
427 MEP instance; this acts as a request for Acknowledgment. This message MUST NOT
428 contain a payload (as defined in **Section 1.5**). The Sending RMP MUST use the request of
429 a SOAP Request-response MEP instance for a synchronous PollRequest and MUST use a
430 SOAP One-way MEP for an asynchronous PollRequest.

431 Step 3: The Receiving RMP sends the RM-Reply either (if synchronous polling) in the
432 response message of the same SOAP instance that carried the PollRequest or (if
433 asynchronous polling) in a message from a SOAP One-way MEP instance. This message
434 MUST NOT contain a payload.

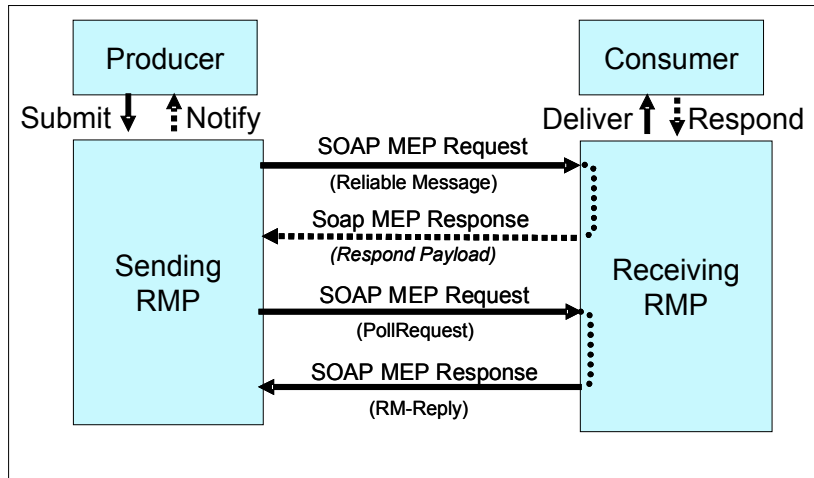


Figure 5 Synchronous Poll RM-Reply Pattern

435 When the Sending RMP of Reliable Messages cannot receive underlying protocol requests (e.g.,
 436 due to security restrictions), it may use the synchronous version of this reply pattern. The Sending
 437 RMP MAY also use this reply pattern (steps 2 and 3 above) to extend other RM-Reply Patterns.
 438 **Figure 5** illustrates the synchronous variant, **Figure 6** the asynchronous.

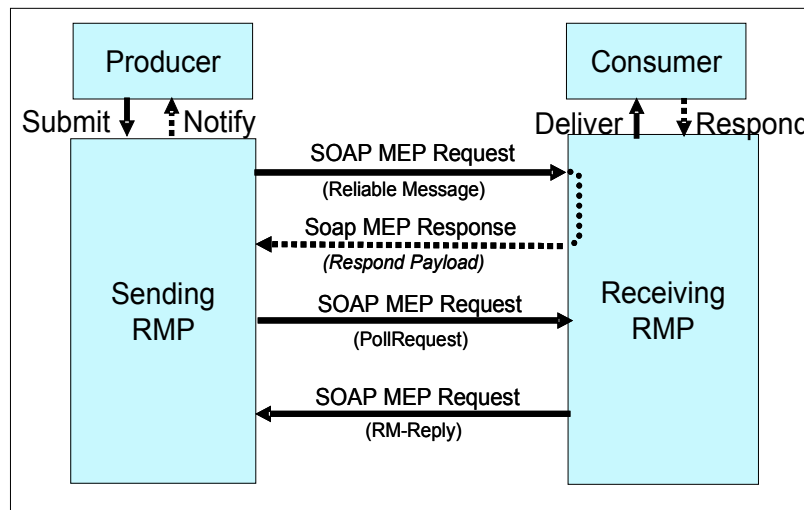


Figure 6 Asynchronous Poll RM-Reply Pattern

439 2.5 Message Identification and Grouping

440 A Reliable Message contains an Identifier that is globally unique and relies on the notion of a
 441 group. A Reliable Message always belongs to a group. The Sending RMP sends a group of
 442 messages to the Receiving RMP as a sequence of individual messages. The Reliable Message
 443 Identifier is a combination of a group ID and an optional sequence number; a sequence number, if
 444 present, is an integer that is unique within a group. More precisely, a message is uniquely
 445 identified as follows:

- 446 1) When there is only one message in the group: the group ID, which is a globally unique
 447 group identifier, may be used alone as Message Identifier. No sequence number is required,
 448 although one is allowed.
- 449 2) When the message belongs to a group of several messages: the message is identified by
 450 the group ID and a unique sequence number.

451 3 Reliability Agreement and Features

452 3.1 RM Agreement

453 3.1.1 Definition

454 An agreement for messaging reliability, or RM Agreement, describes which reliability features a
455 sending party and a receiving party have agreed to use when exchanging a set of messages. The
456 RM Agreement can be seen as a contract at two levels: (1) quality of service (QoS), about the
457 conditions and quality of message delivery to the Consumer and (2) protocol features, including
458 timing parameters and details about choreography between the Sending and Receiving RMPs.

459 3.1.2 RM Agreement Items

460 An RM Agreement is a list of Agreement Items.

461 A Sending RMP MUST be capable of (1) taking knowledge (whether by configuration, an API call,
462 a message, the result of an algorithm or any other means) of a set of values that represent the
463 RM Agreement Items described in this specification and (2) processing them according to the
464 semantics described in this specification.

465 A Receiving RMP MUST be capable of (1) taking knowledge of the Agreement items as they are
466 communicated via the header elements of Reliable Messages and (2) processing them according
467 to the semantics described in this specification.

468 **Table 3** shows the Agreement Items this specification uses. Each item is listed with its possible
469 values:

Name	Value	Definition
GuaranteedDelivery	enabled/disabled	For setting Guaranteed Delivery (see Section 3.2.1 for details).
NoDuplicateDelivery	enabled/disabled	For setting message delivery without duplicates or Duplicate Elimination (see Section 3.2.2 for details).
OrderedDelivery	enabled/disabled	For setting Guaranteed Message Ordering (see Section 3.2.3 for details).
GroupMaxIdleDuration	number of seconds	For setting the elapsed time limit from the last message sent or received in a group, after which the group can be terminated. The value MUST NOT be zero or smaller.
GroupExpiryTime	date/time	For setting the date and time after which the group can be terminated.
ExpiryTime	date/time	For setting the date and time after which a message must not be delivered to the Consumer.
ReplyPattern	"Response", "Callback", "Poll"	For setting the mode of response for Acknowledgments or Faults.

Table 3 RM Agreement Items

470 **3.1.3 Scope of an Agreement Item**

471 There are two scopes to consider:

- 472 • Group scope: All messages sent within a group.
- 473 • Message Scope: A single message.

474 Agreement Items relate to a particular scope: for example, ExpiryTime affects each message
475 separately, while GroupExpiryTime is an Agreement Item about groups.

476 Agreement items applying to the Message Scope MAY be applied to the Group Scope. For
477 example, an RMP implementation may decide to specify the same ExpiryTime value for all
478 messages of a group and not support setting different values for messages in a group. The
479 default scope of applicability for each RM Agreement item is:

480 Message scope:

- 481 • ExpiryTime
- 482 • ReplyPattern

483 Group scope:

- 484 • OrderedDelivery
- 485 • GuaranteedDelivery
- 486 • NoDuplicateDelivery
- 487 • GroupExpiryTime
- 488 • GroupMaxIdleDuration

489 An RMP MUST NOT allow most Agreement items applicable at Group scope to vary between
490 messages of a group. For example, a Sending RMP MUST NOT use different guaranteed delivery
491 modes for different messages of a group. However, it is allowed to dynamically change the value
492 of GroupExpiryTime or GroupMaxIdleDuration pertaining to a group (See **Section 5.1.2**).

493 **3.1.4 Rules**

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

- 496 • If OrderedDelivery is enabled for a group, GuaranteedDelivery and NoDuplicateDelivery
497 MUST also be enabled for that group.
- 498 • If GroupExpiryTime is used for a group, the item GroupMaxIdleDuration MUST NOT be
499 used for this group and vice versa.

500 **3.1.5 Creation, Representation and Deployment of RM Agreements**

501 The concrete representation of an RM Agreement is beyond the scope of this specification, as this
502 may be part of a more general agreement that covers other matters as well as the reliability
503 aspect. However, the RM Agreement determines the use of the reliability protocol and the
504 behavior of RMPs. For these reasons, this specification references the RM Agreement in an
505 abstract way, showing it as a simple list of (name, value) pairs called Agreement Items. This
506 allows a description of the concrete effect of each Agreement Item on the message content and
507 flow. Once there is a broad enough consensus for using a particular representation for
508 agreements, a future version of this specification will define a corresponding binding for RM
509 Agreements.

510 The way RM Agreements are established or communicated to each party is out of scope.
511 However, one of the principles of this specification is that it should not be necessary to deploy an
512 RM Agreement on both RMPs prior to executing business transactions. Only the Sending RMP
513 needs to have knowledge of the RM Agreement initially. No prior communication of the agreement
514 to the receiving party (an RMP and its user) is required. The only input the Receiving RMP will
515 need in order to enforce the reliability requirements will be obtained from the header elements of
516 received messages.

517 **3.1.6 RM Capability**

518 As a way to support the creation of RM Agreements, it may be useful for Web services providers
519 to advertise somehow the reliability features (or RM Agreement Item values) supported by a
520 deployed Web service. In contrast with agreements involving both parties, such reliability features
521 – called RM Capabilities – may conveniently be associated with WSDL definitions. In support of
522 this option, this specification proposes a concrete representation for these capabilities (see
523 **Appendix B**).

524 **3.2 Main Reliability Features**

525 The main reliability features mentioned in **Section 1** are formally described here in terms of
526 requirements. This specification provides the means to enforce these requirements. A detailed
527 description of the protocol features implementing these means is given in **Section 4** and beyond.

528 3.2.1 Guaranteed Delivery

529 Quality of Service requirements:

530 When the GuaranteedDelivery Agreement Item is enabled, one of the two following outcomes
531 SHALL occur for each Submit invocation: either (1) the Receiving RMP successfully delivers
532 (Deliver invocation) the submitted payload to its associated Consumer or (2) the Sending RMP
533 notifies (Notify invocation) the Producer associated with that payload of a delivery failure.

534 **Notes:**

- 535 • This QoS feature guarantees only that the sender will always be notified of a delivery
536 failure when a message is not delivered. It is, however, impossible to guarantee this
537 while at the same time guaranteeing that (1) and (2) will never occur together for the
538 same message. A proper usage by an implementation of the protocol options described
539 in this specification will, however, greatly reduce situations where both (1) and (2) occur.
- 540 • The GuaranteedDelivery agreement is defined for messages resulting from invocations
541 of the Submit operation. An extension of this agreement to messages resulting from
542 invocations of the Respond operation is out of scope for this specification.

543 Protocol requirements:

544 For all messages sent with the GuaranteedDelivery agreement, a Receiving RMP MUST publish
545 the RM-Reply of each such message that has been either delivered or faulted. The Sending RMP
546 MUST poll for all of its sent messages that requested the Poll RM-Reply Pattern.

547 A message resending technique combined with the acknowledgment and fault mechanism
548 described here MUST be used in case of a delivery failure. Parameters that control the resending
549 policy (number of retries, frequency, etc.) are out of the scope of this specification. These
550 parameters may be added to an RM Agreement, although the resending policy may need to be
551 dynamically adjusted depending on network conditions. When resending a message, the
552 message contents must not change.

553 A Receiving RMP MUST NOT publish a Reliable Messaging Fault for a delivered Message. The
554 RMP MUST NOT deliver a message for which a Reliable Messaging Fault has been published.

555 A Sending RMP MUST NOT resend a message for which an RM-Reply with a Fault type other
556 than MessageProcessingFailure has been received and MUST instead notify its Producer of a
557 delivery failure.

558 3.2.2 Duplicate Elimination

559 Quality of Service requirements:

560 When the NoDuplicateDelivery Agreement Item is enabled, a message resulting from a Submit
561 invocation SHALL NOT be delivered twice or more to the Consumer.

562 **Note:**

563 In the current specification, the NoDuplicateDelivery agreement is defined for messages resulting
564 from invocations to the Submit operation. An extension of this agreement to messages resulting
565 from invocations to the Respond operation is out of scope for this specification.

566 Protocol requirements:

567 An implementation of this specification must ensure the following invariants:

- 568 • Message instances resulting from separate invocations of Submit MUST NOT share the
569 same Message Identifier.

570 • When resending a message, the message contents must not change.

571 As a corollary to the above requirements, a Receiving RMP MUST ensure that once a message
572 under this agreement has been delivered to a Consumer, no message with the same identifier
573 received afterward will be delivered to this Consumer.

574 When the Response RM-Reply Pattern is requested with Duplicate Elimination for a Reliable
575 Message, the Receiving RMP cannot deliver that message to the Consumer again (because it is a
576 duplicate of a previously delivered message), and a Consumer response payload is expected, the
577 response of the SOAP MEP instance MUST contain one (but not both) of the following:

578 • a copy of the original response payload returned for that Message (in the SOAP Body) in
579 addition to the Acknowledgment Indication (in the SOAP Header) or

580 • a SOAP server Fault (in the SOAP Body) in addition to the Acknowledgment Indication
581 (in the SOAP Header).

582 The Sending RMP and Producer expect either a complete response or a SOAP Fault when using
583 the Response RM-Reply Pattern; these two allowed behaviors satisfy that expectation.

584 **3.2.3 Guaranteed Message Ordering**

585 Quality of Service requirements:

586 When the OrderedDelivery Agreement Item is enabled, messages resulting from a sequence of
587 Submit invocations SHALL be delivered in the same order to the Consumer. In addition, when the
588 Receiving RMP delivers one of these messages, all previous messages submitted in the
589 sequence MUST already have been delivered (no missing message allowed).

590 **Note:**

591 In the current specification, the OrderedDelivery agreement is defined for messages resulting
592 from invocations of the Submit operation on the Sending RMP. An extension of this agreement to
593 messages resulting from invocations of the Respond operation is out of scope for this
594 specification.

595 Protocol requirements:

596 Ordering is supported only over messages of the same group.

597 An implementation of this specification must ensure the following invariants, regarding the usage
598 of sequence numbers (SequenceNum element):

599 • The Sending RMP MUST reflect the order of the Submit invocations on this RMP in the
600 sequence numbers of the corresponding messages sent.

601 • The Receiving RMP MUST deliver the messages received according to the order
602 expressed by their sequence numbers, which is the same as the submission order.

603 An RMP will terminate the group as specified in **Section 5.1.3.5** (T5) when those conditions arise.

604 **4 Message Format**

605 **4.1 Structure**

606 **Figure 7** shows the structure of reliability SOAP header blocks in the SOAP Envelope, as
 607 specified by the WS-Reliability protocol. On the left side of the figure, a Reliable Message is
 608 characterized by the presence of the wsrn:Request element. On the right side a response to a
 609 Reliable Message contains a wsrn:Response element. Both wsrn:Request and wsrn:Response
 610 elements may be found in the same message.

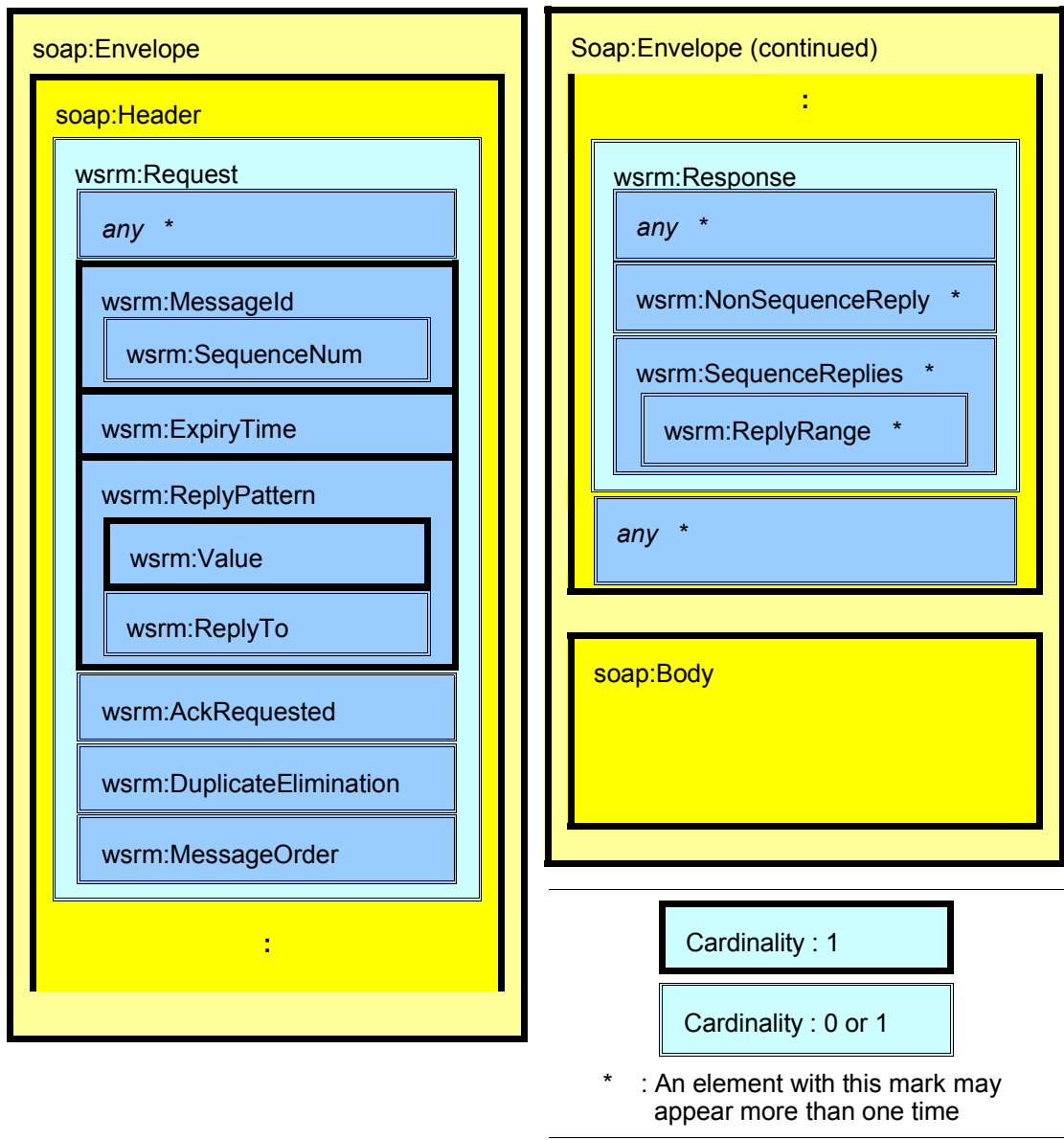


Figure 7 Structure of WS-Reliability elements

611 **Figure 8** shows the structure of PollRequest message embedded in the SOAP Envelope.

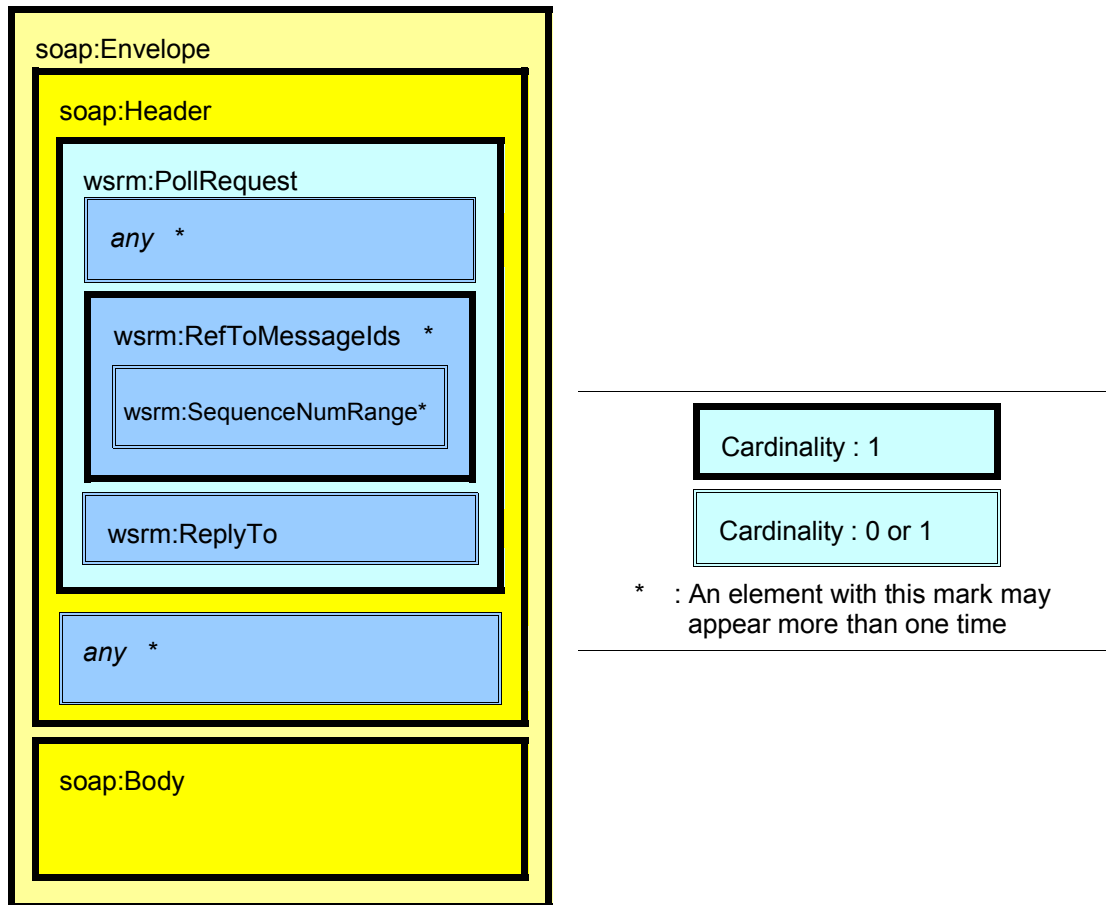


Figure 8 Structure of PollRequest message elements

612 The namespace [XML Namespaces] for reliable messaging defined in this specification is:

613 <http://docs.oasis-open.org/wsm/2004/06/ws-reliability-1.1.xsd>

614 When the text of the specification is shown to be in conflict with schema statements, the schema
615 statements prevail in the absence of an errata addressing the conflict.

616 The schema for some of the elements specified in this section includes the specification of
617 extensibility elements and attributes. The extensibility features expressed formally in the schema
618 are specified in **Section 4.6**.

619 If a message contains additional elements or attributes not described in this specification, the
620 Reliable Messaging Processor MAY ignore them.

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

- 622 • **Request** element
- 623 • **PollRequest** element
- 624 • **Response** element

625 4.2 Request Element

626 The Request element conveys information about the agreement items that apply to the containing
627 Reliable Message. This element includes the following attribute and child elements (see the
628 description of each child element for cardinality requirements):

- 629 • SOAP **mustUnderstand** attribute (see **Appendix A** for details)
- 630 • **MessageId** element
- 631 • **ExpiryTime** element
- 632 • **ReplyPattern** element
- 633 • **AckRequested** element
- 634 • **DuplicateElimination** element
- 635 • **MessageOrder** element

Cardinality	0 or 1
Value	None
Attributes	soap:mustUnderstand (Boolean)
Child elements	MessageId ExpiryTime ReplyPattern AckRequested DuplicateElimination MessageOrder

Table 4 Request Element

636 **Example 1** shows an instance of a Request element.

Example 1 Request Element

```
637 <Request
638   xmlns="http://docs.oasis-open.org/wsrn/2004/06/ws-reliability-1.1.xsd"
639   xmlns:soap12="http://www.w3.org/2003/05/soap-envelope"
640   soap12:mustUnderstand="1">
641   <MessageId groupId="mid://20040202.103832@wsr-sender.org">
642     <SequenceNum number="0"
643       groupExpiryTime="2005-02-02T03:00:33-31:00" />
644   </MessageId>
645   <ExpiryTime>2004-09-07T03:01:03-03:50</ExpiryTime>
646   <ReplyPattern>
647     <Value>Response</Value>
648   </ReplyPattern>
649   <AckRequested/>
650   <DuplicateElimination/>
651   <MessageOrder/>
652 </Request>
```

653 4.2.1 Element: Request/MessageId

654 This element includes the following attribute:

- 655
- a **groupId** attribute

Cardinality	1
Value	None
Attributes	groupId (xs:anyURI)
Child elements	SequenceNum

Table 5 MessageId Element

656 4.2.1.1 Attribute: Request/MessageId@groupId

657 This attribute identifies a message group. The Sending RMP MUST use a distinct globally unique
658 @groupId value for each distinct group of messages. Within any such group, all messages will
659 have the same value for @groupId. This identification (the value) is of type URI as defined in
660 [RFC2396]. It is RECOMMENDED that implementations use the Message-ID schema defined in
661 [RFC2392].

662 4.2.1.2 Element: Request/MessageId/SequenceNum

663 The Sending RMP MUST include the SequenceNum element in all Reliable Messages of a group
664 with more than one message.

665 The SequenceNum element carries the sequence number as well as other attributes that may
666 alter the Receiving RMP's processing of the group. When a message includes a MessageOrder
667 element, the sequence number is used in support of message ordering (**Section 3.2.3**).

668 This element includes the following attributes:

- 669 • a **groupExpiryTime** attribute
- 670 • a **groupMaxIdleDuration** attribute
- 671 • a **number** attribute
- 672 • a **last** attribute

673 In a request message, the sender MAY include either (but not both) @groupExpiryTime or
 674 @groupMaxIdleDuration (see **Section 5.1.2**).

675 **Example 2** illustrates the SequenceNum element with some message fragments:

Example 2 SequenceNum Element

676 1) First message

```
677 <MessageId groupId="mid://20040202.103832@wsr-sender.org">
678   <SequenceNum number="0"
679     groupExpiryTime="2005-02-02T03:00:33-31:00" />
680 </MessageId>
```

681 2) Second message

```
682 <MessageId groupId="mid://20040202.103832@wsr-sender.org">
683   <SequenceNum number="1"
684     groupExpiryTime="2005-02-02T03:00:33-31:00" />
685 </MessageId>
```

686 3) The last message for the group

```
687 <MessageId groupId="mid://20040202.103832@wsr-sender.org">
688   <SequenceNum number="2"
689     groupExpiryTime="2005-02-02T03:00:33-31:00" last="true" />
690 </MessageId>
```

Cardinality	1
Value	None
Attributes	groupExpiryTime (dateTime) groupMaxIdleDuration (duration) number (unsignedLong) last (Boolean)
Child elements	None

Table 6 SequenceNum Element

691 **4.2.1.2.1 Attribute: Request/MessageId/SequenceNum@groupExpiryTime**

692 This attribute represents the GroupExpiryTime agreement item (**Section 3.1.2, Table 3**). It
 693 specifies the the date and time at which the sender wishes the group to terminate. The
 694 @groupExpiryTime value is expressed as UTC and conforms to [XML Schema Part 2] dateTime.

695 The Cardinality of this attribute is 0 or 1. Constraints on the use of this attribute are specified in
696 **Section 5**.

697 **4.2.1.2.2 Attribute: Request/MessageId/SequenceNum@groupMaxIdleDuration** 698

699 This attribute represents the GroupMaxIdleDuration agreement item (**Section 3.1.2, Table 3**). It
700 specifies the maximum idle time for a group. The @groupMaxIdleDuration value conforms to
701 [XML Schema Part 2] duration. The Cardinality of this attribute is 0 or 1. Constraints on the use of
702 this attribute are specified in **Section 5**.

703 **4.2.1.2.3 Attribute: Request/MessageId/SequenceNum@number**

704 This attribute contains the sequence number, which identifies the message within its group
705 (**Section 2.6**) and is used in support of message ordering (**Section 3.2.3**). @number conforms to
706 [XML Schema Part 2] unsignedLong.

707 The Sending RMP MUST set this value to 0 for the first message of a group. The Sending RMP
708 thereafter MUST increment this value by 1 for each message submitted in this group. Once the
709 value reaches the maximum (18446744073709551615, the maximum value for this data type),
710 the group is terminated (see **Section 5**).

711 **4.2.1.2.4 Attribute: Request/MessageId/SequenceNum@last**

712 This attribute indicates whether or not the containing message is the last in a group. The
713 Cardinality of this attribute is 0 or 1. When this attribute is present, its Boolean value has the
714 following meaning:

- 715 • **false**: Indicates the message is not the last message of the group or is not known to be
716 the last message of the group.
- 717 • **true**: Indicates the message is known to be the last message sent within a group of
718 messages.

719 When this attribute is not present, its value defaults to false.

720 **4.2.2 Element: Request/ExpiryTime**

721 The ExpiryTime element represents the ExpiryTime agreement item (**Section 3.1.2, Table 3**). It
722 indicates the ultimate date and time after which the Receiving RMP MUST NOT invoke the Deliver
723 operation for the received message. The message is considered expired if the current time,
724 expressed in UTC, is greater than the value of the ExpiryTime element. When a message expires
725 on the Sending RMP before being successfully sent, a Sending RMP MUST NOT send or resend
726 it and MUST communicate a delivery failure to the Producer. The time is expressed as UTC and
727 conforms to [XML Schema Part 2] dateTime.

Cardinality	1
Value	xs:dateTime
Attributes	None
Child elements	None

Table 7 ExpiryTime Element

728 4.2.3 Element: Request/ReplyPattern

729 A Sending RMP MUST include the ReplyPattern element in a Request element. The ReplyPattern
730 element includes the following child elements:

- 731 • a **Value** element
- 732 • a **ReplyTo** element

Cardinality	1
Value	None
Attributes	None
Child elements	Value ReplyTo

Table 8 ReplyPattern Element

733 4.2.3.1 Element: Request/ReplyPattern/Value

734 The Value element indicates which reply pattern the Sending RMP requests. This element
735 specifies whether the Receiving RMP should send the Acknowledgment Indication or RM Fault
736 Indication back in the response to the reliable message, in a separate callback request, or in the
737 response to a separate poll request. A Sending RMP MUST include the Value element in a
738 ReplyPattern element. This element has one of the following three values:

- 739 • **Response**
- 740 • **Callback**
- 741 • **Poll**

742 These values respectively indicate which of the RM-Reply Patterns – Response, Callback or Poll
743 – is in use, as described in **Section 2.5**.

Cardinality	1
Value	xs:string: Response, Callback or Poll
Attributes	None
Child elements	None

Table 9 Value Element

744 4.2.3.2 Element: Request/ReplyPattern/ReplyTo

745 If the value of the Request/ReplyPattern/Value element is "Callback", the Sending RMP MUST
746 include this element in the Reliable Message. For all other values ("Poll" and "Response") of
747 Request/ReplyPattern/Value element, the Sending RMP MUST NOT include this element. This
748 element specifies the endpoint where the Sending RMP expects to receive a callback containing
749 RM-Reply information.

750 If present, the reference-scheme attribute specifies the format of the single child element of the
 751 ReplyTo element. If the attribute is omitted, the default content of the ReplyTo element is
 752 BareURI.

Cardinality	0 or 1
Value	None
Attributes	reference-scheme
Child elements	{ <i>xs:anyType</i> } (an element representing the reference)

Table 10 ReplyTo Element

753 **4.2.3.2.1 Attribute: Request/ReplyPattern/ReplyTo@reference-scheme**

754 This attribute specifies the format or schema of the child element of
 755 Request/ReplyPattern/ReplyTo. The Sending RMP MUST omit this attribute when the child
 756 element of Request/ReplyPattern/ReplyTo is BareURI. The type of this attribute is xs:anyURI.

757 **4.2.3.2.2 Element: Request/ReplyPattern/ReplyTo/BareURI**

758 This element provides one of the simplest referencing options, the URI of the callback recipient's
 759 endpoint. It is the default content of the Request/ReplyPattern/ReplyTo and PollRequest/ReplyTo
 760 (see **Section 4.3.1**) elements, though the Sending RMP MAY use any other element and scheme
 761 supported by the Receiving RMP. This location (the value) is of type URI as defined in [RFC2396].

762 **Section 6** provides additional information about the specific case for which the content of a
 763 BareURI in a Request or PollRequest element uses the HTTP URI scheme.

Cardinality	0 or 1
Value	xs:anyURI
Attributes	None
Child elements	None

Table 11 BareURI Element

764 **4.2.4 Element: Request/AckRequested**

765 A Sending RMP MUST include the AckRequested element in a message if and only if that
 766 message is subject to the GuaranteedDelivery Agreement Item (refer to **Section 3.2.1** for details);
 767 as described in **Section 3.1.4**, this condition includes all messages subject to the OrderedDelivery
 768 Agreement Item. The Sending RMP uses this element to request the Receiving RMP to publish
 769 an Acknowledgment after the message is delivered to the consumer party or else to publish an
 770 RM Fault Indication. The Receiving RMP MUST publish this information, even for received
 771 messages that are duplicates of previously delivered messages. For example, if the RM-Reply
 772 Pattern is Callback and no fault occurs, an Acknowledgment Indication SHALL be sent back.

773 The Receiving RMP MAY publish an RM Fault Indication for a Reliable Message, even if the
 774 AckRequested element is not present in the Request element for that message.

775 The pattern used to send the Acknowledgment or RM Fault Indication is determined by the value
 776 of the ReplyPattern element.

Cardinality	0 or 1
Value	None
Attributes	None
Child elements	None

Table 12 AckRequested Element

777 **4.2.5 Element: Request/DuplicateElimination**

778 A Sending RMP MUST include the DuplicateElimination element in a message if and only if that
779 message is subject to the NoDuplicateDelivery Agreement Item (refer to **Section 3.2.2** for
780 details); as described in **Section 3.1.4**, this condition includes all messages subject to the
781 OrderedDelivery Agreement Item.

Cardinality	0 or 1
Value	None
Attributes	None
Child elements	None

Table 13 DuplicateElimination Element

782 **4.2.6 Element: Request/MessageOrder**

783 A Sending RMP MUST include the MessageOrder element if and only if that message is subject
784 to the OrderedDelivery Agreement Item (refer to **Section 3.2.3** for details).

785 If the MessageOrder element appears in the message received, the Receiving RMP MUST NOT
786 deliver the message until all messages with the same Request/MessageId@groupid value and a
787 lower Request/MessageId/SequenceNum@number value have been delivered.

Cardinality	0 or 1
Value	None
Attributes	None
Child elements	None

Table 14 MessageOrder Element

788 **4.2.7 Example**

789 The HTTP message below uses the Request element to specify (among other things) that all
790 three reliability features should be used: GuaranteedDelivery ("AckRequested" element),
791 NoDuplicateDelivery ("DuplicateElimination" element), and OrderedDelivery ("MessageOrder"
792 element). The reply pattern is "Poll", meaning that no Acknowledgment or Fault will be sent back
793 unless explicitly requested by another message containing a PollRequest header.

Example 3 Reliable Message with Request header

```
794 POST /abc/servlet/wsrEndpoint HTTP/1.0
795 Content-Type: text/xml; charset=utf-8
796 Host: 192.168.183.100
797 SOAPAction: ""
798 Content-Length: 736
799
800 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
801   <soap:Header>
802     <Request
803       xmlns="http://docs.oasis-open.org/wsr/2004/06/ws-reliability-1.1.xsd"
804       soap:mustUnderstand="1">
805       <MessageId groupId="mid://20040202.103832@wsr-sender.org">
806         <SequenceNum number="0"
807           groupExpiryTime="2005-02-02T03:00:33-31:00" />
808       </MessageId>
809       <ExpiryTime>2004-09-07T03:01:03-03:50</ExpiryTime>
810       <ReplyPattern>
811         <Value>Poll</Value>
812       </ReplyPattern>
813       <AckRequested/>
814       <DuplicateElimination/>
815       <MessageOrder/>
816     </Request>
817   </soap:Header>
818   <soap:Body>
819     <Request xmlns="http://example.org/wsr">Request Message</Request>
820   </soap:Body>
821 </soap:Envelope>
```

822 4.3 PollRequest Element

823 A PollRequest Message requests an RM-Reply for a Reliable Message that had "Poll" as the
824 value of the Request/ReplyPattern/Value element and included the Request/AckRequested
825 element. However, PollRequest Messages can also solicit delivery status for messages that were
826 originally sent with "Response" or "Callback" as the value of the Request/ReplyPattern/Value
827 element and that included the Request/AckRequested element.

828 If a Receiving RMP does not support the use of PollRequest as a general status query
829 mechanism, it MAY return a FeatureNotSupported fault in response to a PollRequest when the
830 relevant ReplyPattern Agreement Item does not have the value "Poll".

831 A Receiving RMP that receives a supported form of PollRequest MUST publish RM-Reply
832 information relevant to non-expired messages identified in that request.

833 This element includes the following attribute and child elements:

- 834 • SOAP **mustUnderstand** attribute (see **Appendix A** for details)
- 835 • a **ReplyTo** element
- 836 • a **RefToMessageIds** element

Cardinality	0 or 1
Value	None
Attributes	soap:mustUnderstand (Boolean)
Child elements	ReplyTo RefToMessageIds

Table 15 PollRequest Element

Example 4 PollRequest Element

```

837 <PollRequest
838   xmlns="http://docs.oasis-open.org/wsrn/2004/06/ws-reliability-1.1.xsd"
839   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
840   soap:mustUnderstand="1">
841   <RefToMessageIds groupId="mid://20040202.103832@wsr-sender.org">
842     <SequenceNumRange from="0" to="5"/>
843     <SequenceNumRange from="15" to="20"/>
844   </RefToMessageIds>
845   <RefToMessageIds groupId="mid://20040202.103811@wsr-sender.org" />
846   <RefToMessageIds groupId="mid://20040202.103807@wsr-sender.org">
847     <SequenceNumRange from="713" to="6150"/>
848   </RefToMessageIds>
849 </PollRequest>

```

850 4.3.1 Element: PollRequest/ReplyTo

851 The Receiving RMP MUST send the RM-Reply information in a new request to the endpoint
852 specified by PollRequest/ReplyTo whenever this element is present. If it is not present, the
853 Receiving RMP MUST send back the RM-Reply on the response to the PollRequest message.

854 **Section 4.2.3.2** provides additional information about the very similar
855 Request/ReplyPattern/ReplyTo element.

Cardinality	0 or 1
Value	None
Attributes	reference-scheme
Child elements	{xs:anyType} (an element representing the reference)

Table 16 ReplyTo Element

856 4.3.1.1 Attribute: PollRequest/ReplyTo@reference-scheme

857 **Section 4.2.3.2.1** provides additional information about the similar
858 Request/ReplyPattern/ReplyTo@reference attribute.

859 **4.3.1.2 Element: PollRequest/ReplyTo/BareURI**

860 **Section 4.2.3.2.2** provides additional information about the similar
861 Request/ReplyPattern/ReplyTo/BareURI element.

Cardinality	0 or 1
Value	xs:anyURI
Attributes	None
Child elements	None

Table 17 BareURI Element

862 **4.3.2 Element: PollRequest/RefToMessagelds**

863 The RefToMessagelds element contains the identifiers of groups and messages whose status the
864 Sending RMP is requesting. This element includes @groupid and zero or more
865 SequenceNumRange elements as follows:

- 866 • a **groupid** attribute
- 867 • zero or more **SequenceNumRange** elements

Cardinality	1 or more
Value	None
Attributes	groupid (URI)
Child elements	SequenceNumRange

Table 18 RefToMessagelds Element

868 When this RefToMessagelds element does not include a SequenceNumRange element, the
869 Receiving RMP MUST return RM-Replies for non-expired messages that were delivered or
870 faulted in that group.

871 When the RefToMessagelds element includes one or more SequenceNumRange element(s), the
872 Receiving RMP MUST return RM-Replies for the non-expired messages that were delivered or
873 faulted in the identified subset of that group. The identified subset includes all Reliable Messages
874 whose MessageId/SequenceNum@number values fall in the range(s) specified in the
875 RefToMessagelds/SequenceNumRange element(s) of the PollRequest.

876 A Sending RMP MAY include multiple RefToMessagelds elements (one for each @groupid value)
877 in a single PollRequest Message to request RM-Replies for multiple groups.

878 **4.3.2.1 Attribute: PollRequest/RefToMessagelds@groupid**

879 The @groupid specifies the group of messages whose status the Sending RMP is requesting.
880 This identification (the value) is of type URI as defined in [RFC2396].

881 **4.3.2.2 Element: PollRequest/RefToMessagelds/SequenceNumRange**

882 The SequenceNumRange element specifies those messages in a group for which the Sending
883 RMP requests status. Attributes @from and @to of this element express an inclusive range for
884 SequenceNum values. This element contains the following two attributes:

885 • a **from** attribute

886 • a **to** attribute

887 When these attributes have the same value, the range is limited to a single message.

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

Table 19 SequenceNumRange Element

888 **4.3.2.2.1 Attribute:**

889 **PollRequest/RefToMessageIds/SequenceNumRange@from**

890 This attribute specifies the lowest SequenceNum@number value of the message range. The
891 value of @from is of type unsignedLong and SHALL be less than or equal to the value of @to.

892 **4.3.2.2.2 Attribute: PollRequest/RefToMessageIds/SequenceNumRange@to**

893 This attribute specifies the highest SequenceNum@number value of the message range. The
894 value of @to is of type unsignedLong and SHALL be greater than or equal to the value of @from.

895 **4.3.3 Example**

896 The HTTP message below uses the PollRequest reliability element, polling the Receiving RMP for
897 the status of messages within the range of sequence numbers 0 to 20 of a particular group. The
898 response to this PollRequest will identify which of those messages have been delivered
899 (Acknowledged).

Example 5 PollRequest Message embedded in HTTP Request

```
900 POST /abc/servlet/wsrEndpoint HTTP/1.0
901 Content-Type: text/xml; charset=utf-8
902 Host: 192.168.183.100
903 SOAPAction: ""
904 Content-Length: 432
905
906 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
907   <soap:Header>
908     <PollRequest
909       xmlns="http://docs.oasis-open.org/wsr/2004/06/ws-reliability-1.1.xsd"
910       soap:mustUnderstand="1">
911       <RefToMessageIds groupId="mid://20040202.103832@wsr-sender.org">
912         <SequenceNumRange from="0" to="20"/>
913       </RefToMessageIds>
914     </PollRequest>
915   </soap:Header>
916   <soap:Body />
917 </soap:Envelope>
```

918 4.4 Response Element

919 The Response element indicates Acknowledgments and Faults for Reliable Messages. This
920 element includes the following attributes:

- 921 • SOAP **mustUnderstand** attribute (see **Appendix A** for details)

922 The Response element SHALL include a list one or more elements in length containing a choice
923 or choices from the following:

- 924 • **NonSequenceReply** element(s)
- 925 • **SequenceReplies** element(s)

926 When the Response occurs under the Response RM-Reply Pattern, the first element in this list
927 describes the status of the received Reliable Message. In this case, when the SequenceReplies
928 element is used, the first contained ReplyRange element will include the received Reliable
929 Message within its range.

930 The Receiving RMP MAY bundle a Response element with a Request element when responding
931 to a message that used the Callback RM-Reply Pattern. In this case, the response and the new
932 Reliable Message MUST share a common destination URI. This enables the combination of an
933 Acknowledgment Indication and the business response to the original message. This also allows
934 a Receiving RMP to bundle an Acknowledgment Indication with another unrelated message to the
935 Sending RMP to reduce network traffic. When combined in a single message, the Request and
936 Response elements are treated separately from the perspective of the abstract model (**Section**
937 **2**); a Receiving RMP component handles the Request element and payload while a Sending RMP
938 handles the Response element.

Cardinality	0 or 1
Value	None
Attributes	soap:mustUnderstand (Boolean)
Child elements	NonSequenceReply SequenceReplies

Table 20 Response Element

939 **Example 6** shows an instance of the Response element.

Example 6 Response Element

```

940 <Response
941   xmlns="http://docs.oasis-open.org/wsrn/2004/06/ws-reliability-1.1.xsd"
942   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
943   soap:mustUnderstand="1">
944   <NonSequenceReply groupId="mid://20040202.103832@wsr-sender.org" />
945   <SequenceReplies groupId="mid://20040202.103807@wsr-sender.org">
946     <ReplyRange from="1" to="4" />
947     <ReplyRange from="5" to="5" fault="wsrm:InvalidRequest" />
948     <ReplyRange from="6" to="42" />
949   </SequenceReplies>
950   <NonSequenceReply groupId="mid://20040202.103811@wsr-sender.org"
951     fault="wsrm:PermanentProcessingFailure" />
952 </Response>

```

953 4.4.1 Element: Response/NonSequenceReply

954 An RM-Reply for a message that does not have a sequence number SHALL include a
955 NonSequenceReply element. This element includes the following attributes:

- 956 • a **groupId** attribute
- 957 • a **fault** attribute

958 The **@fault** indicates a particular fault for the identified message. Without this attribute, the
959 NonSequenceReply element is an Acknowledgment Indication for the message.

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

Table 21 NonSequenceReply Element

960 **4.4.1.1 Attribute: Response/NonSequenceReply@groupId**

961 This attribute specifies the group identifier of a message that did not have a sequence number. A
962 NonSequenceReply element SHALL include the message's @groupId. This identification (the
963 value) is of type URI as defined in [RFC2396].

964 **4.4.1.2 Attribute: Response/NonSequenceReply@fault**

965 This attribute indicates the code of a Reliable Messaging Fault encountered while processing the
966 message. The Cardinality of this attribute is 0 or 1.

967 **4.4.2 Element: Response/SequenceReplies**

968 An RM-Reply for a group (or a subset thereof) whose messages had sequence numbers SHALL
969 include a SequenceReplies element. This element contains a @groupId and 1 or more
970 ReplyRange elements.

Cardinality	0 or more
Value	None
Attributes	groupId (URI)
Child elements	ReplyRange

Table 22 SequenceReplies Element

971 **4.4.2.1 Attribute: Response/SequenceReplies@groupId**

972 The @groupId specifies the message group for which its SequenceReplies element carries the
973 status. A SequenceReplies element SHALL include the group's @groupId. This identification (the
974 value) is of type URI as defined in [RFC2396].

975 **4.4.2.2 Element: Response/SequenceReplies/ReplyRange**

976 The ReplyRange element indicates a range of sequence numbers with a shared delivery status.
977 The @fault indicates a particular, common fault all messages in the range share. Without this
978 attribute, the ReplyRange element is an Acknowledgment Indication for all messages in the range.

Cardinality	1 or more
Value	None
Attributes	from (unsigned Long) to (unsigned Long) fault (QName)
Child elements	None

Table 23 ReplyRange Element

979 **4.4.2.2.1 Attribute: Response/SequenceReplies/ReplyRange@from**

980 This attribute has same type and semantics as in the PollRequest element.

981 **4.4.2.2 Attribute: Response/SequenceReplies/ReplyRange@to**

982 This attribute has same type and semantics as in the PollRequest element.

983 **4.4.2.3 Attribute: Response/SequenceReplies/ReplyRange@fault**

984 This attribute indicates the code of a Reliable Messaging Fault encountered while processing all
985 of the messages in the identified range. The Cardinality of this attribute is 0 or 1.

986 **4.4.3 Example**

987 The message below uses the Response reliability element, which in this case is carrying the
988 response of a previous PollRequest element. The response acknowledges a message specified
989 by the group identifier "mid://20040202.103811@wsr-sender.org" and messages for a group
990 specified by the group identifier "mid://20040202.103832@wsr-sender.org" within the ranges of
991 sequence numbers 0 to 14 and 16 to 20. The response also reports an RM Fault for a message
992 with sequence number 15 for the group.

Example 7 RM-Reply message embedded in HTTP Response

```
993 HTTP/1.0 200 OK
994 Server: WS-ReliabilityServer
995 Date: Mon, 02 Feb 2004 10:38:32 GMT
996 Content-Language: en
997 Content-Type: text/xml; charset=utf-8
998 Content-Length: 593
999
1000 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
1001   <soap:Header>
1002     <Response soap:mustUnderstand="1"
1003       xmlns="http://docs.oasis-open.org/wsr/2004/06/ws-reliability-1.1.xsd">
1004       <NonSequenceReply groupId="mid://20040202.103811@wsr-sender.org"/>
1005       <SequenceReplies groupId="mid://20040202.103832@wsr-sender.org">
1006         <ReplyRange from="0" to="14"/>
1007         <ReplyRange from="15" to="15" fault="InvalidRequest"/>
1008         <ReplyRange from="16" to="20"/>
1009       </SequenceReplies>
1010     </Response>
1011   </soap:Header>
1012   <soap:Body />
1013 </soap:Envelope>
```

1014 **4.5 Fault Codes For Reliable Messaging Failures**

1015 The protocol defines two fault categories:

- 1016 • The Message Format fault set, which includes all faults generated because of a
1017 malformed Reliable Message header.
- 1018 • The Message Processing fault set, which includes all faults generated while processing
1019 the message.

1020 They are explained in detail in the following sections. The Receiving RMP returns these protocol-
1021 specific fault codes within the Response header element. Reliable Message Faults are carried in
1022 the SOAP Header and do not rely exclusively on the SOAP Fault model for the following reasons:

- 1023 • The SOAP Fault model does not allow batching of several faults in the same message.
- 1024 • RM Faults may be carried along with business messages that are unrelated to these
1025 faults; they should not affect the processing of the SOAP body in such messages.

1026 The rules for processing faults are:

- 1027 • The Receiving RMP MUST NOT deliver a message for which an RM Fault is published.
1028 Therefore, the Receiving RMP MUST NOT send an Acknowledgment Indication for such
1029 a message.
- 1030 • If a Reliable Message sent over a SOAP Request-response MEP cannot be delivered to
1031 the Consumer, the response of the SOAP MEP instance SHALL contain a SOAP Fault
1032 (in the SOAP Body) in addition to the appropriate RM Fault (in the SOAP Header). If the
1033 specific RM Fault encountered was due to a problem with the request header element,
1034 the Receiving RMP MUST set the value of the soap:Fault@faultcode attribute to
1035 "soap:Client" (for SOAP 1.1 messages) or the soap12:Fault/Code/Value element to
1036 "soap12:Sender" (for SOAP 1.2 messages). If the specific RM Fault encountered was
1037 due to a problem with processing by the Receiving RMP, the Receiving RMP MUST set
1038 the value of the soap:Fault@faultcode attribute to "soap:Server" (for SOAP 1.1
1039 messages) or the soap12:Fault/Code/Value element to "soap12:Receiver" (for SOAP 1.2
1040 messages). The Sending RMP and Producer expect either a complete response or a
1041 SOAP Fault when using the SOAP Request-response MEP; this requirement satisfies
1042 those expectations. More details are given in **Section 3.2** and in the HTTP Binding
1043 section (**Section 6**).
- 1044 • When a Reliable Message sent over a SOAP One-way MEP cannot be delivered to the
1045 Consumer due to a failure in processing the RM headers, a SOAP Fault SHALL NOT be
1046 returned. The HTTP binding section (**Section 6**) gives more details on the recommended
1047 behavior in such case.

1048 The Fault codes described in **Sections 4.5.1** and **4.5.2** are allowed values for @fault in a
1049 Response element.

1050 **4.5.1 Message Format Faults**

1051 The Receiving RMP publishes these faults when the message format of the Reliable Messaging
1052 Headers is either invalid or wrong.

Local part name	Description and Cause(s)
InvalidRequest	<p>The Request element is wrong or invalid. Examples are:</p> <ol style="list-style-type: none"> 1. Any of the mandatory elements such as MessageId, ExpiryTime or ReplyPattern are missing. 2. AckRequested, DuplicateElimination or MessageOrder elements appear twice. 3. The soap:mustUnderstand attribute is missing.
InvalidPollRequest	<p>The PollRequest element is wrong or invalid. Examples are:</p> <ol style="list-style-type: none"> 1. The soap:mustUnderstand attribute is missing. 2. The RefToMessageIds element is missing.
InvalidMessageId	<p>Used in any of the following cases:</p> <ol style="list-style-type: none"> 1. @groupId (for MessageId or RefToMessageIds) is not present or is present with an invalid value. 2. @number in SequenceNum element is not present or is present with an invalid value. 3. Attributes (from and to) of SequenceNumRange are not present or are present with invalid values.
InvalidMessageParameters	<p>Used in any of the following cases:</p> <ol style="list-style-type: none"> 1. The @groupExpiryTime is wrong or invalid. 2. The @groupMaxIdleDuration is wrong or invalid. 3. Both group parameters are present. 4. SequenceNum@last exists but is not one of the allowed {false true} values.
InvalidReplyPattern	<p>Used in either of the following cases:</p> <ol style="list-style-type: none"> 1. The ReplyPattern format is wrong or invalid. 2. The ReplyTo element is missing for the Callback pattern.
InvalidExpiryTime	<p>The ExpiryTime format is wrong or invalid.</p>

Table 24 Invalid Message Format Fault Code Values

1053 **Note:**

1054 Cases exist in which the Receiving RMP is unable to send RM Fault Indications for messages
1055 with invalid message headers, such as:

- 1056 • The ReplyTo element is missing or invalid in the Callback and asynchronous Poll cases.
- 1057 • The MessageId element is missing for the Request element.
- 1058 • The RefToMessageIds is missing for the PollRequest element.

1059 **4.5.2 Message Processing Faults**

1060 The Receiving RMP publishes these faults when there is an error processing a valid Reliable
 1061 Messaging message.

Local part name	Description and Cause(s)
FeatureNotSupported	The Receiving RMP receives a message with an RM feature that it does not support. An example is an RM message with a MessageOrder element sent to a Receiving RMP that doesn't support Guaranteed Message Ordering.
PermanentProcessingFailure	Permanent and fatal processing failures such as: <ol style="list-style-type: none"> 1. Persistence Storage failures. 2. Message Delivery failures. A PermanentProcessingFailure fault indicates that the failure is fatal and subsequent retries of the same message will also fail.
MessageProcessingFailure	Used in transient failure cases such as: <ol style="list-style-type: none"> 1. The number of buffered requests exceeded the maximum limit. 2. The number of threads reached the maximum limit, etc. 3. The Deliver operation fails. A transient fault, unlike a permanent fault, is temporary; the message may succeed after a subsequent retry.
GroupAborted	All processing for the group associated with the reliable message request has been aborted by the Receiving RMP. The Receiving RMP MUST NOT deliver subsequent messages within that group.

Table 25 Messaging Processing Failure Fault Code Values

1062 4.5.3 RM Fault Examples

Example 8 RM Fault Indication for Reliable Messaging

```
1063 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
1064   <soap:Header>
1065     <Response soap:mustUnderstand="1"
1066       xmlns="http://docs.oasis-open.org/wsr/2004/06/ws-reliability-1.1.xsd">
1067       <SequenceReplies groupId="mid://20040202.103832@wsr-sender.org">
1068         <ReplyRange from="1" to="1" fault="InvalidRequest" />
1069       </SequenceReplies>
1070     </Response>
1071   </soap:Header>
1072   <soap:Body />
1073 </soap:Envelope>
```

1074 If the PollRequest element in **Example 4** was missing the soap:mustUnderstand attribute, the
1075 InvalidPollRequest fault may be sent as follows.

Example 9 RM Fault Indication for PollRequest message

```
1076 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
1077   <soap:Header>
1078     <Response soap:mustUnderstand="1"
1079       xmlns="http://docs.oasis-open.org/wsr/2004/06/ws-reliability-1.1.xsd">
1080     <SequenceReplies groupId="mid://20040202.103832@wsr-sender.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@wsr-sender.org"
1085       fault="InvalidPollRequest"/>
1086     <SequenceReplies groupId="mid://20040202.103807@wsr-sender.org/">
1087       <ReplyRange from="713" to="6150" fault="InvalidPollRequest"/>
1088     </SequenceReplies>
1089   </Response>
1090 </soap:Header>
1091 <soap:Body />
1092 </soap:Envelope>
```

1093 4.6 Extensibility Features of Schema

1094 The core schema for this specification (associated in **Section 1.3, Table 2**, with the “wsrm”
1095 namespace prefix) specifies extension mechanisms for some schema elements.

1096 The following elements (which have a complex sequence type) allow the presence of zero or
1097 more extension elements (of type xs:anyType; that is, any type not defined in this core
1098 namespace is allowed) at the beginning of the sequence, as well as zero or more extension
1099 attributes (with similar namespace restrictions):

- 1100 • Request
- 1101 • Response

1102 • PollRequest

1103 • NonSequenceReply

1104 • SequenceReplies

1105 • ReplyRange

1106 The extensibility of the ReplyTo elements (**Sections 4.2.3.2 and 4.3.1**) is somewhat different; it is
1107 described in the appropriate sections above.

1108 5 Operational Aspects and Semantics

1109 5.1 Message Group Life Cycle

1110 5.1.1 Group Termination

1111 Being able to know when a group may be terminated and its persistent resources reclaimed is
1112 essential for keeping the resource footprint of reliability low. However, this section is not just about
1113 efficient management of resources: it describes normative behavioral rules for RMPs when
1114 handling group termination.

1115 Termination of a group in the Sending RMP and in the Receiving RMP are two distinct events, not
1116 synchronized by any special message but instead occurring as the result of rules applying
1117 separately to the Sending and Receiving RMPs. As a consequence, the termination of a group
1118 may occur at quite different times on the Sending RMP and the Receiving RMP. However, the
1119 lack of synchronization allowed by these termination rules is not consequential.

1120 Groups undergoing termination on the Sending RMP and the Receiving RMP pass through the
1121 following states:

1122 **Group complete:**

- 1123 • The Sending RMP considers a group complete when all of its messages have been sent
1124 and the last sent message has an ending marker (SequenceNum@last="true" or it has a
1125 sequence number with the maximum value). Note that completeness occurs even if not
1126 all of the group's messages have been either acknowledged or faulted (in case
1127 GuaranteedDelivery is enabled).
- 1128 • The Receiving RMP considers a group complete when a message with an ending
1129 marker has been received and all previous messages for this group also have been
1130 received (no number missing in the sequence) although not necessarily delivered yet.

1131 **Group closed:**

- 1132 • When a group is closed in the Sending RMP, the RMP expects to send no new message
1133 in this group. However, the RMP MAY resend messages as needed if
1134 GuaranteedDelivery is enabled. If a new message is submitted for a closed group, the
1135 Sending RMP MUST notify the Producer that the group is closed and MUST NOT send
1136 the message.
- 1137 • When a group is closed in the Receiving RMP, the RMP expects to receive no new
1138 message for this group. After a group is closed and before it is "removed" (see definition
1139 below), a Receiving RMP MUST NOT deliver messages received with this group
1140 identifier, whether or not they are duplicates of previous messages and regardless of
1141 whether they result from a resend of previously failed messages initiated before closing
1142 on the Sending RMP (in case GuaranteedDelivery is enabled).

1143 **Note:**

1144 Due to time-out, a group may be closed without being complete. Once complete, a group will
1145 close (see termination rules).

1146 **Group Removed:**

1147 Group removal occurs at the time the group is closed or afterward. Intuitively, a group is removed
1148 when a Receiving RMP does not need to remember anything about this group, i.e., when there is
1149 no need to check for duplicates of its messages in the future (for example, when all of its
1150 messages have expired).

- 1151 • When a group is removed in the Sending RMP, the RMP is NOT REQUIRED to verify
1152 that future submitted messages are improperly associated with the removed group and
1153 MAY treat them as part of a new group. However, the Sending RMP is responsible for
1154 generating group identifiers, and it SHOULD generate values unique enough to avoid
1155 later reuse of the group identifier of a removed group (for example, generation
1156 mechanisms including a timestamp will make reuse impossible).
- 1157 • When a group is removed in the Receiving RMP, the RMP is no longer supposed to
1158 remember anything about this group. In particular, the group identifier is discarded from
1159 the RMP state. When receiving a message with same group identifier as a removed
1160 group, a Receiving RMP is NOT REQUIRED to confirm whether or not this group
1161 identifier value has already been used; the RMP MAY treat such a message as part of a
1162 new group.

1163 **5.1.2 Group Termination Parameters**

1164 Two RM Agreement Items, GroupExpiryTime and GroupMaxIdleDuration, determine when a
1165 group can be terminated. These two items are considered Group Termination parameters that
1166 control the persistence of the group data. The corresponding message header attributes are
1167 @groupExpiryTime and @groupMaxIdleDuration respectively. The following requirements pertain
1168 to these header attributes:

- 1169 a) The first message in a group (the one with
1170 Request/MessageId/SequenceNum@number=0) indicates which Group Termination (time-
1171 out) parameter is in use for the group. However, the Receiving RMP MUST use the first
1172 message received for this group to indicate which termination parameter is associated with
1173 this group.
 - 1174 • If the first message in the sequence of a group has neither group time-out parameter
1175 present, the group will be terminated according to condition T3, T4 or T5.
 - 1176 • If the first message has one of the two time-out parameters present (either
1177 @groupExpiryTime or @groupMaxIdleDuration), the group will be subject to
1178 termination rules T1 or T2 described below.
 - 1179 • The Receiving RMP MUST return an InvalidMessageParameters fault if both group
1180 persistence parameters are present in any request message.
 - 1181 • If @groupExpiryTime is in use, the Sending RMP MUST NOT send a message in that
1182 group with an ExpiryTime value greater than @groupExpiryTime.
 - 1183 b) The group termination parameter sent on the first message in the group SHALL be used
1184 on all subsequent messages in that group and SHALL be assigned a value.
 - 1185 c) If the Receiving RMP receives a message with a group termination parameter that is not
1186 consistent with the termination parameter used in previous messages for this group, the
1187 Receiving RMP MUST return an InvalidMessageParameters fault.
- 1188 When the group is ordered, the fault SHALL be returned for the message with lowest
1189 sequence number that was found inconsistent in the group. If the group is not required to be
1190 ordered, the fault SHALL be returned for the first message received that was found
1191 inconsistent in the group.

1192 d) The Sending RMP MAY modify either time-out parameter, sending a subsequent
1193 message with the new value. When applying termination rules, the Sending RMP MUST use
1194 the value in the message with the highest sequence number sent for the group. The
1195 Receiving RMP MUST use the value from the message with the highest sequence number
1196 received for the group.

1197 e) @groupMaxIdleDuration can be either increased or decreased without restriction. The
1198 Sending RMP may increase or decrease @groupExpiryTime as long as it is never less than
1199 the max(ExpiryTime) of the messages sent for the group so far.

1200 The Receiving RMP MUST publish an InvalidMessageParameters Fault for a message with
1201 a @groupExpiryTime value less than the max(ExpiryTime) of the messages previously
1202 received for the group.

1203 5.1.3 Termination Rules

1204 Termination is the process by which an RMP discontinues the use of a group, allowing the RMP to
1205 reclaim resources used by the group. Termination typically involves two steps that may occur at
1206 different times: closing and removal. Removal of a group may happen some time after it is closed,
1207 allowing an RMP to filter out potential duplicate messages. The general rule is that a group is
1208 removed once all of its messages have expired. If we define max(ExpiryTime) as the maximum
1209 date and time of all ExpiryTime values of the messages sent for a group (on the Sender side) or
1210 received for a group (on the Receiver side), a group will not be removed before max(ExpiryTime)
1211 occurs.

1212 There are two general indicators an RMP will use to terminate a group:

1213 a) Message Marker: Information within a message (either
1214 Request/MessageId/SequenceNum@last="true" or the maximum sequence number)
1215 indicates the last message for the group. This is used by termination rules T3, T4.

1216 b) Timing: Either the group's lifespan expired or its idle time exceeded a time-out. This is
1217 used by termination rules T1, T2. Or due to message expiration, a group with the ordering
1218 requirement cannot be delivered. This is used by termination rule T5.

1219 These termination rules apply to both ordered and unordered groups. However, these rules do not
1220 apply to groups that contain a single message with no sequence number.

1221 5.1.3.1 Termination by expiration (T1):

1222 Context:

1223 The group specified @groupExpiryTime.

1224 Receiver side:

1225 Triggering event: @groupExpiryTime is in the past.

1226 The RMP MUST close and remove the group.

1227 Sender side:

1228 Triggering event: @groupExpiryTime is in the past (note: in this case, max(ExpiryTime) also is
1229 past).

1230 The RMP MUST close and remove the group.

1231 **5.1.3.2 Termination by idle time-out (T2):**

1232 Context:

1233 The group specified @groupMaxIdleDuration.

1234 Receiver side:

1235 Triggering event: The time since the last received message for the group is over
1236 @groupMaxIdleDuration.

1237 The RMP MUST close the group. But unlike T1, some of its past messages may not have expired
1238 yet. In case Duplicate Elimination is required, the RMP MUST NOT remove the group until max
1239 (ExpiryTime) is reached in order to make sure all potential duplicates for the group will not be
1240 delivered.

1241 Sender side:

1242 Triggering event: The time since the last sent message for the group is over
1243 @groupMaxIdleDuration.

1244 The RMP MUST close the group. If GuaranteedDelivery was required, the RMP MUST remove
1245 the group once it has received either acknowledgment or notification of delivery failure for all sent
1246 messages. If no GuaranteedDelivery was required, the RMP MUST remove the group
1247 immediately.

1248 **5.1.3.3 Termination by completeness (T3):**

1249 Context:

1250 No specific context.

1251 Receiver side:

1252 Triggering event: The RMP receives a message marked last
1253 (Request/MessageId/SequenceNum@last="true"). If all previous messages for the group have
1254 been received, the group is closed immediately. Alternately, the group is closed when the RMP
1255 receives the last missing message in the group.

1256 The RMP MUST close the group. However, its removal is done according to T1 or T2 depending
1257 on which time-out parameter was specified for the group. If no time-out parameter was specified,
1258 the group is removed once all of its messages have expired, i.e., the date and time max
1259 (ExpiryTime) has passed.

1260 **Note:**

1261 In the case in which a message is received with an ending marker before all previous messages
1262 have been received, the group remains active. No termination process is initiated yet.

1263 Sender side:

1264 Triggering event: The RMP sends a message marked last.

1265 All messages of the group have been sent. The RMP MUST close the group. If
1266 GuaranteedDelivery was required, the RMP MUST remove the group once it has received either
1267 acknowledgment or notification of delivery failure for all sent messages. If GuaranteedDelivery
1268 was not required, the RMP MUST remove the group immediately.

1269 **5.1.3.4 Termination by sequence exhaustion (T4):**

1270 Context:

1271 No specific context.

1272 Receiver side:

1273 Triggering event: The RMP receives a message with a sequence number of the maximum value.

1274 If all previous messages for the group have been received, the group is closed immediately.

1275 Alternately, the group is closed when the RMP receives the last missing message in the group.

1276 The group closing and removal follow the rules in T3, the message with the maximum sequence

1277 number acting as a message with the ending mark.

1278 **Note:**

1279 In case a message is received with the maximum sequence number before all previous

1280 messages have been received, the group remains active. No termination process is initiated yet.

1281 Sender side:

1282 Triggering event: The RMP sends a message with a sequence number with the maximum value.

1283 The group closing and removal follow the rules in T3, the message with the maximum sequence

1284 number acting as a message with the ending mark.

1285 **5.1.3.5 Termination by ordering failure (T5):**

1286 Context:

1287 The group requires the Guaranteed Message Ordering reliability feature.

1288 Receiving side:

1289 Triggering event: In an ordered group, a received message expires before delivery or faults with a

1290 fault code other than MessageProcessingFailure. If all previous messages for the group have

1291 been received, the group is closed immediately. Alternately, the group is closed when the RMP

1292 receives the last missing message in the group.

1293 The RMP MUST close the group. The group is removed according to rule T3.

1294 Sender Side:

1295 Triggering event: In an ordered group, an unacknowledged message expires or the RMP receives

1296 an RM Fault for this Reliable Message with a fault code other than MessageProcessingFailure.

1297 The RMP MUST close the group. The group is removed according to rule T3.

1298 **5.1.3.6 Summary of Group Termination Rules**

1299 Conditions for terminating a group in a Receiving RMP:

Group Closing	Group Removal
When @groupExpiryTime has passed.	(after closing) When @groupExpiryTime has passed.
When the @groupMaxIdleDuration time-out has expired.	(after closing) When Max(ExpiryTime) has passed.
When a group is complete.	(after closing) When Max(ExpiryTime) has passed.
When a group is ordered AND an undelivered message expires or faults.	(after closing) When Max(ExpiryTime) has passed.

Table 26 Conditions for terminating a group – Receiving RMP

1300 Conditions for terminating a group in a Sending RMP:

Group Closing	Group Removal
When @groupExpiryTime has passed.	(after closing) When @groupExpiryTime has passed.
When the @groupMaxIdleDuration time-out has expired.	(after closing) In case GuaranteedDelivery is not required, remove the group immediately. Otherwise, remove it if all messages have been either acknowledged or faulted.
When a group is complete.	(after closing) In case GuaranteedDelivery is not required, remove the group immediately. Otherwise, remove it if all messages have been either acknowledged or faulted.
When a group is ordered AND an unacknowledged message expires or faults.	(after closing) Remove the group after all messages have been either acknowledged or faulted.

Table 27 Conditions for terminating a group – Sending RMP

1301 5.2 Attachments

1302 When an RMP implementing this specification uses the W3C Note “SOAP Messages with
1303 Attachments” specification [SOAP with Attachments], it MUST follow the following rules:

- 1304 1) The Sending RMP MUST include the whole SOAP envelope containing the WS-Reliability
1305 header elements in the first MIME part.
- 1306 2) It MUST set the charset parameter of the Content-Type header of the first MIME part to
1307 either UTF-8 or UTF-16.
- 1308 3) It MAY include zero or more additional MIME parts in a Reliable Message.
- 1309 4) The Receiving RMP MUST deliver all MIME parts in a Reliable Message to the
1310 Consumer.

1311 6 HTTP Binding

1312 This section specifies two normative bindings of WS-Reliability header elements to SOAP header
1313 blocks carried in messages using HTTP as a transport protocol:

- 1314 • SOAP 1.1 over HTTP POST binding: An implementation of WS-Reliability MAY support
1315 mapping the WS-Reliability header elements as SOAP header blocks in accordance with
1316 the SOAP 1.1 HTTP Binding specified in Section 6 of [SOAP 1.1]. In that case, the
1317 SOAP Request-response MEP defined in this specification will map to an HTTP request-
1318 response. The SOAP One-way MEP, as defined in **Section 2.3**, maps to the request of
1319 an HTTP request-response.
- 1320 • SOAP 1.2 over HTTP POST binding: An implementation of WS-Reliability MAY support
1321 mapping the WS-Reliability header elements as SOAP header blocks in accordance with
1322 the SOAP 1.2 HTTP binding for the Request-Response MEP specified in Section 7,
1323 "SOAP HTTP Binding", of [SOAP 1.2 Part 2].

1324 If a Reliable Message request is invoked using SOAP 1.1, all subsequent message exchanges
1325 pertaining to that Message Identifier MUST use the SOAP 1.1 protocol. In addition, when an
1326 HTTP binding is used, it is RECOMMENDED the RMP comply with WS-I BP 1.1 [WS-I BP 1.1].
1327 When no WSDL describes the messages being exchanged, the previous WS-I conformance
1328 requirements should be understood as conformance to the subset of the profile requirements
1329 pertaining to the message artifact only.

1330 In case a message encounters a failure in processing the RM headers, the requirements for Fault
1331 handling in **Section 4.5** apply. When using SOAP 1.1, conformance to the WS-I Basic Profile 1.1
1332 requires the following:

- 1333 • For SOAP One-way HTTP binding: the HTTP response entity-body SHALL be empty. If
1334 the RM Fault is a Message Format fault, the HTTP status code SHOULD be "400 Bad
1335 Request" (see R1113 in [WS-I BP 1.1]); otherwise, the RM fault is a Message
1336 Processing fault and the status code SHOULD be "500 Internal Server Error".
- 1337 • For SOAP Request-response HTTP binding: the HTTP response contains a SOAP Fault
1338 element and has the "500 Internal Server Error" HTTP status code (see R1126 in [WS-I
1339 BP 1.1]).

1340 These two requirements for Fault handling apply to all message exchanges described in this
1341 section and its sub-sections.

1342 If a ReplyTo element present in a Request element or Poll Request header element sent using the
1343 SOAP 1.1 protocol uses the wsm:BareURI (the default, described in **Sections 4.2.3.2.2** and
1344 **4.3.1.2**) reference scheme and uses the 'http:' URL scheme, the Receiving RMP MUST send the
1345 WS-Reliability response using the HTTP binding specified in Section 6 of SOAP 1.1.

1346 If a Reliable Message request is invoked using SOAP 1.2, all subsequent message exchanges
1347 pertaining to its Message Identifier MUST use the SOAP 1.2 protocol.

1348 If a ReplyTo element present in a Request element or Poll Request header element sent using the
1349 SOAP 1.2 protocol uses the wsm:BareURI reference scheme and uses the 'http:' URL scheme,
1350 the the Receiving RMP MUST send the WS-Reliability response using the HTTP binding for
1351 Request-Response MEP specified in SOAP 1.2.

1352 The following subsections specify the mapping of WS-Reliability header elements to HTTP
1353 request and response messages for the three RM-Reply Patterns. The Poll RM-Reply Pattern has
1354 two variations: synchronous and asynchronous.

1355 The value of the ReplyPattern/Value element identifies the specific RM-Reply Pattern in use (see
1356 **Section 4.2.3.1** for details).

1357 This specification requires the transport layer to deliver messages to the reliability layer without
1358 corruption. When a request message contains the AckRequested element, the Receiving RMP
1359 MUST send an RM-Reply (an Acknowledgment Indication or an RM Fault Indication) for that
1360 request. For the Callback and Poll RM-Reply Patterns, a Response element can contain multiple
1361 Acknowledgment and/or RM Fault Indications.

1362 For simplicity, the detailed examples show only the use of SOAP 1.1. However, the figures that
1363 show the mapping of WS-Reliability elements to HTTP POST request messages and HTTP
1364 response messages apply to both the SOAP 1.1 over HTTP POST binding and the SOAP 1.2
1365 over HTTP POST binding.

1366 6.1 Reliable Messaging with Response RM-Reply Pattern

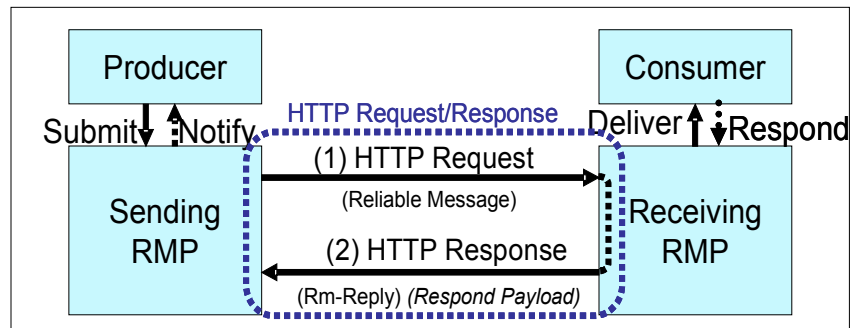


Figure 9 Response RM-Reply Pattern

1367 As described in general for this RM-Reply Pattern (**Section 2.4.1**), the Receiving RMP MUST
1368 return the RM-Reply with the HTTP response on the same HTTP connection used by the Sending
1369 RMP to send the request. This is illustrated in **Figure 9**.

- 1370 • In (1), the Sending RMP initiates an HTTP connection and sends a Message using the
1371 HTTP POST method, as in **Example 10**.
- 1372 • In (2), using the same connection, the Receiving RMP sends back to the Sending RMP
1373 an HTTP response containing an RM-Reply; in **Example 11**, the RM-Reply is an
1374 Acknowledgment Indication.

Example 10 Request Message with Response RM-Reply Pattern

```
1375 POST /abc/servlet/wsrEndpoint HTTP/1.0
1376 Content-Type: text/xml; charset=utf-8
1377 Host: 192.168.183.100
1378 SOAPAction: ""
1379 Content-Length: 755
1380
1381 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1382   <soap:Header>
1383     <Request
1384       xmlns="http://docs.oasis-open.org/wsr/2004/06/ws-reliability-1.1.xsd"
1385       soap:mustUnderstand="1">
1386       <MessageId groupId="mid://20040202.103832@wsr-sender.org">
1387         <SequenceNum number="0"
1388           groupExpiryTime="2005-02-02T03:00:33-31:00" />
1389       </MessageId>
1390       <ExpiryTime>2004-09-07T03:01:03-03:50</ExpiryTime>
1391       <ReplyPattern>
1392         <Value>Response</Value>
1393       </ReplyPattern>
1394       <AckRequested/>
1395       <DuplicateElimination/>
1396       <MessageOrder/>
1397     </Request>
1398   </soap:Header>
1399   <soap:Body>
1400     <Request xmlns="http://example.org/wsr">Request Message</Request>
1401   </soap:Body>
1402 </soap:Envelope>
```

Example 11 Acknowledgment Indication with Response RM-Reply Pattern

```
1403 HTTP/1.0 200 OK
1404 Server: WS-ReliabilityServer
1405 Date: Mon, 02 Feb 2004 10:38:32 GMT
1406 Content-Language: en
1407 Content-Type: text/xml; charset=utf-8
1408 Content-Length: 414
1409
1410 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1411   <soap:Header>
1412     <Response soap:mustUnderstand="1"
1413       xmlns="http://docs.oasis-open.org/wsrn/2004/06/ws-reliability-1.1.xsd">
1414       <SequenceReplies groupId="mid://20040202.103832@wsr-sender.org">
1415         <ReplyRange from="0" to="0"/>
1416       </SequenceReplies>
1417     </Response>
1418   </soap:Header>
1419   <soap:Body />
1420 </soap:Envelope>
```

1421 6.2 Reliable Messaging with Callback RM-Reply Pattern

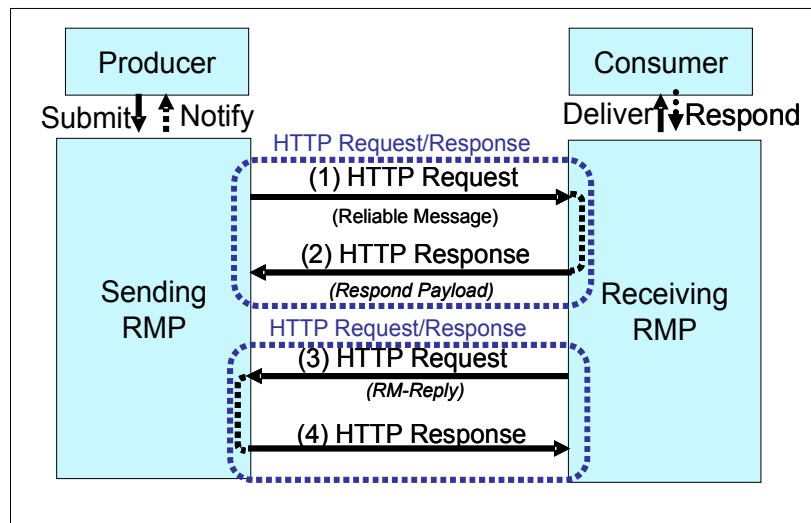


Figure 10 Callback RM-Reply Pattern

1422 As described in general for this RM-Reply Pattern (Section 2.4.2) and as illustrated in Figure 10,
1423 two distinct HTTP request/response exchanges are involved.

- 1424 • In (1), the Sending RMP initiates a new HTTP request and sends a Reliable Message
1425 with the Callback RM Reply Pattern. **Example 12** shows such an HTTP message.
- 1426 • In (2), the HTTP response may have an empty entity-body (in case of a SOAP One-way
1427 MEP instance).
- 1428 • In (3), the Receiving RMP MUST return the RM-Reply on an HTTP connection different
1429 from the one the Sending RMP used to send the message. The direction of the HTTP

1430 connection used by the Receiving RMP is from the Receiving RMP to the Sending RMP.
1431 **Example 14** shows an Acknowledgment Indication as the RM-Reply.

- 1432 • In (4), there is no HTTP entity-body unless the RM-Reply was bundled with a new
1433 Reliable Message on a SOAP Request-response MEP instance.

Example 12 Request Message with Callback RM-Reply Pattern

```
1434 POST /abc/servlet/wsrEndpoint HTTP/1.0
1435 Content-Type: text/xml; charset=utf-8
1436 Host: 192.168.183.100
1437 SOAPAction: ""
1438 Content-Length: 863
1439
1440 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1441   <soap:Header>
1442     <Request
1443       xmlns="http://docs.oasis-open.org/wsr/2004/06/ws-reliability-1.1.xsd"
1444       soap:mustUnderstand="1">
1445       <MessageId groupId="mid://20040202.103832@wsr-sender.org">
1446         <SequenceNum number="0"
1447           groupExpiryTime="2005-02-02T03:00:33-31:00" />
1448       </MessageId>
1449       <ExpiryTime>2004-09-07T03:01:03-03:50</ExpiryTime>
1450       <ReplyPattern>
1451         <Value>Callback</Value>
1452         <ReplyTo>
1453           <BareURI>http://wsr-sender.org/abc/wsrmlistener</BareURI>
1454         </ReplyTo>
1455       </ReplyPattern>
1456       <AckRequested/>
1457       <DuplicateElimination/>
1458       <MessageOrder/>
1459     </Request>
1460   </soap:Header>
1461   <soap:Body>
1462     <Request xmlns="http://example.org/wsr">Request Message</Request>
1463   </soap:Body>
1464 </soap:Envelope>
```

Example 13 HTTP response with no content

```
1465 HTTP/1.0 200 OK
1466 Server: WS-ReliabilityServer
1467 Date: Mon, 02 Feb 2004 10:38:32 GMT
1468 Content-Language: en
1469 Content-Type: text/xml; charset=utf-8
1470 Content-Length: 0
```

Example 14 Acknowledgment Indication with Callback RM-Reply Pattern

```
1471 POST /abc/wsrmlistener HTTP/1.0
1472 Content-Type: text/xml; charset=utf-8
1473 Host: 192.168.183.200
1474 SOAPAction: ""
1475 Content-Length: 414
1476
1477 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
1478   <soap:Header>
1479     <Response soap:mustUnderstand="1"
1480       xmlns="http://docs.oasis-open.org/wsr/2004/06/ws-reliability-1.1.xsd">
1481       <SequenceReplies groupId="mid://20040202.103832@wsr-sender.org">
1482         <ReplyRange from="0" to="0"/>
1483       </SequenceReplies >
1484     </Response>
1485   </soap:Header>
1486   <soap:Body />
1487 </soap:Envelope>
```

1488 6.3 Reliable Messaging with Poll RM-Reply Pattern

1489 The general rules for this RM-Reply Pattern are described in **Section 2.4.3**. When the Sending
1490 RMP issues a PollRequest, the Receiving RMP MAY return the RM-Reply on the HTTP
1491 connection used to send the PollRequest message (synchronous), or it MAY return the RM-Reply
1492 on a different HTTP connection (asynchronous). Whether the RM-Reply corresponding to the
1493 PollRequest is synchronous or asynchronous depends on the presence of a ReplyTo element in
1494 the PollRequest element.

1495 6.3.1 Synchronous Poll RM-Reply Pattern

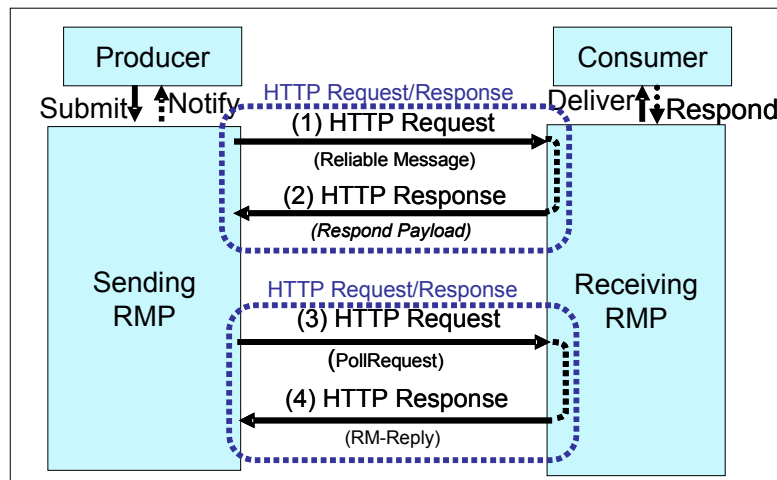


Figure 11 Synchronous Poll RM-Reply Pattern

1496 **Figure 11** illustrates the synchronous variant of the Poll RM Reply Pattern.

- 1497 • In (1), the Sending RMP initiates a new HTTP Request and sends a Reliable Message
1498 with the Poll RM-Reply Pattern.

- 1499 • In (2), the HTTP response may have an empty entity-body (in case of a SOAP One-way
1500 MEP instance).
- 1501 • In (3), at a later time the Sending RMP initiates a different HTTP Request to send a
1502 PollRequest message. The PollRequest does not include the ReplyTo element (see
1503 **Example 15**).
- 1504 • In (4), the Receiving RMP returns the RM-Reply in an HTTP response on the same
1505 HTTP connection used to send the PollRequest, as illustrated in **Figure 11**. The HTTP
1506 response (4) includes an RM-Reply (e.g., an Acknowledgment Indication as in **Example**
1507 **16**).

Example 15 PollRequest message with Synchronous Poll RM-Reply Pattern

```
1508 POST /abc/servlet/wsrmlistener HTTP/1.0
1509 Content-Type: text/xml; charset=utf-8
1510 Host: 192.168.183.100
1511 SOAPAction: ""
1512 Content-Length: 433
1513
1514 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1515   <soap:Header>
1516     <PollRequest
1517       xmlns="http://docs.oasis-open.org/wsrmlistener/2004/06/ws-reliability-1.1.xsd"
1518       soap:mustUnderstand="1">
1519       <RefToMessageIds groupId="mid://20040202.103832@wsr-sender.org">
1520         <SequenceNumRange from="0" to="20"/>
1521       </RefToMessageIds>
1522     </PollRequest>
1523   </soap:Header>
1524   <soap:Body />
1525 </soap:Envelope>
```

Example 16 Synchronous Acknowledgment Indication

```
1526 HTTP/1.0 200 OK
1527 Server: WS-ReliabilityServer
1528 Date: Mon, 02 Feb 2004 10:38:32 GMT
1529 Content-Language: en
1530 Content-Type: text/xml; charset=utf-8
1531 Content-Length: 456
1532
1533 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1534   <soap:Header>
1535     <Response soap:mustUnderstand="1"
1536       xmlns="http://docs.oasis-open.org/wsrn/2004/06/ws-reliability-1.1.xsd">
1537       <SequenceReplies groupId="mid://20040202.103832@wsr-sender.org">
1538         <ReplyRange from="0" to="14"/>
1539         <ReplyRange from="16" to="20"/>
1540       </SequenceReplies>
1541     </Response>
1542   </soap:Header>
1543   <soap:Body />
1544 </soap:Envelope>
```

1545 6.3.2 Asynchronous Poll RM-Reply Pattern

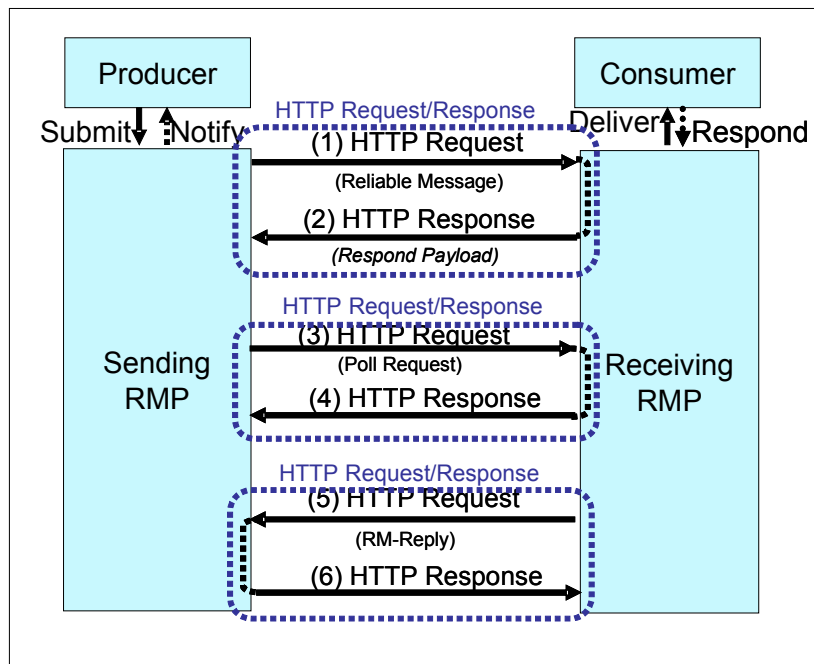


Figure 12 Asynchronous Poll RM-Reply Pattern

1546 **Figure 12** illustrates the asynchronous variant of the Poll RM Reply Pattern.

- 1547 • In (1), the Sending RMP initiates a new HTTP Request and sends a Reliable Message
1548 with the Poll RM-Reply Pattern.

- 1549 • In (2), the HTTP response may have an empty entity-body (in the case of a SOAP One-
1550 way MEP instance).
- 1551 • In (3), the Sending RMP initiates a new HTTP request and sends a PollRequest
1552 message. Note that in **Example 17**, the PollRequest element has a ReplyTo element.
- 1553 • In (4), the HTTP response (4) has no HTTP entity-body (see **Example 13**).
- 1554 • In (5), the Receiving RMP sends the RM-Reply in a different HTTP request to the listener
1555 identified by the ReplyTo element (see **Example 18**).
- 1556 • In (6), the HTTP response has no HTTP entity-body (see **Example 13**).

Example 17 PollRequest message with Asynchronous Poll RM-Reply Pattern

```

1557 POST /abc/servlet/wsrmlistener HTTP/1.0
1558 Content-Type: text/xml; charset=utf-8
1559 Host: 192.168.183.100
1560 SOAPAction: ""
1561 Content-Length: 553
1562
1563 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1564   <soap:Header>
1565     <PollRequest
1566       xmlns="http://docs.oasis-open.org/wsr/2004/06/ws-reliability-1.1.xsd"
1567       soap:mustUnderstand="1">
1568       <RefToMessageIds groupId="mid://20040202.103832@wsr-sender.org">
1569         <SequenceNumRange from="0" to="20"/>
1570       </RefToMessageIds>
1571       <ReplyTo>
1572         <BareURI>http://wsr-sender.org/xyz/servlet/wsrmlistener
1573         </BareURI>
1574       </ReplyTo>
1575     </PollRequest>
1576   </soap:Header>
1577   <soap:Body />
1578 </soap:Envelope>

```

Example 18 Asynchronous Acknowledgment Indication

```
1579 POST /xyz/servlet/wsrmlistener HTTP/1.0
1580 Content-Type: text/xml; charset=utf-8
1581 Host: 192.168.183.200
1582 SOAPAction: ""
1583 Content-Length: 456
1584
1585 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
1586   <soap:Header>
1587     <Response soap:mustUnderstand="1"
1588       xmlns="http://docs.oasis-open.org/wsrml/2004/06/ws-reliability-1.1.xsd">
1589       <SequenceReplies groupId="mid://20040202.103832@wsr-sender.org">
1590         <ReplyRange from="0" to="14"/>
1591         <ReplyRange from="16" to="20"/>
1592       </SequenceReplies>
1593     </Response>
1594   </soap:Header>
1595   <soap:Body />
1596 </soap:Envelope>
```

1597 7 Conformance

1598 In order to conform to this specification, an implementation must satisfy all of the following
1599 conditions:

- 1600 • It has implemented all required syntax, features and behaviors.
- 1601 • It complies with the following interpretation of the keywords OPTIONAL and MAY: as
1602 stated in [RFC2119], when these keywords apply to the behavior of the implementation,
1603 the implementation is free to support these behaviors or not.
- 1604 • It MUST be capable of processing the prescribed failure mechanism for those optional
1605 features it has chosen to implement. If an RMP conforming to this requirement has
1606 implemented an optional feature, syntax or behavior defined in this specification, it can
1607 interoperate with another implementation that has not.
- 1608 • It MUST be capable of generating the prescribed failure mechanism for those optional
1609 features it has not chosen to implement. If an RMP conforming to this requirement has
1610 not implemented an optional feature, syntax or behavior defined in this specification, it
1611 can interoperate with another implementation that has.

1612 8 References

- 1613 [ebMS] "Message Service Specification Version 2.0", OASIS ebXML Messaging Services
1614 Technical Committee, OASIS Standard, 1 April 2002. Available at
1615 <http://www.ebxml.org/specs/ebMS2.pdf>
- 1616 [RFC1738] "Uniform Resource Locators (URL)", T. Berners-Lee et al, RFC 1738, IESG and IETF,
1617 December 1994. Available at
1618 <http://www.ietf.org/rfc/rfc1738.txt>
- 1619 [RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119,
1620 Bradner, S., IESG and IETF, March 1997. Available at
1621 <http://www.ietf.org/rfc/rfc2119.txt>
- 1622 [RFC2392] "Content-ID and Message-ID Uniform Resource Locators", RFC2392, E. Levinson,
1623 IESG and IETF, August 1998. Available at
1624 <http://www.ietf.org/rfc/rfc2392.txt>
- 1625 [RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, Tim Berners-Lee et
1626 al, IESG and IETF, August 1998. Available at
1627 <http://www.ietf.org/rfc/rfc2396.txt>
- 1628 [RFC2616] "Hypertext Transfer Protocol – HTTP/1.1", RFC 2616, R. Fielding et al, IESG and
1629 IETF, June 1999. Available at
1630 <http://www.ietf.org/rfc/rfc2616.txt>
- 1631 [RFC2822] "Internet Message Format", RFC 2822, P. Resnick, Editor, IESG and IETF, April 2001.
1632 Available at
1633 <http://www.ietf.org/rfc/rfc2822.txt>
- 1634 [SOAP 1.1] "Simple Object Access Protocol (SOAP) 1.1", Don Box et al, W3C Note, 8 May, 2000.
1635 Available at
1636 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 1637 [SOAP 1.2 Part 1] "SOAP 1.2 Part 1: Messaging Framework", Martin Gudgin, Marc Hadley, Noah
1638 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, eds., W3C Recommendation, 24
1639 June 2003. Available at
1640 <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>

- 1641 [SOAP 1.2 Part 2] "SOAP 1.2 Part 1: Adjuncts", Martin Gudgin, Marc Hadley, Noah Mendelsohn,
1642 Jean-Jacques Moreau, Henrik Frystyk Nielsen, eds., W3C Recommendation, 24 June 2003.
1643 Available at
1644 <http://www.w3.org/TR/2003/REC-soap12-part2-20030624/>
- 1645 [SOAP with Attachments] "SOAP Messages with Attachments", John J. Barton, Satish Thatte,
1646 Henrik Frystyk Nielsen, W3C Note, 11 December 2000, Available at
1647 <http://www.w3.org/TR/SOAP-attachments>
- 1648 [XML] "Extensible Markup Language (XML) 1.0 (Third Edition)", Tim Bray et al, eds., W3C
1649 Recommendation, first published 10 February 1998, revised 4 February 2004. Available at
1650 <http://www.w3.org/TR/2004/REC-xml-20040204>
- 1651 [XML Namespaces] "Namespaces in XML", Tim Bray et al., eds., W3C Recommendation, 14
1652 January 1999. Available at
1653 <http://www.w3.org/TR/1999/REC-xml-names-19990114/>
- 1654 [XML Schema Part 1] "XML Schema Part 1: Structures", Henry S. Thompson, David Beech,
1655 Murray Maloney, Noah Mendelsohn, eds., W3C Recommendation, 2 May 2001. Available at
1656 <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>
- 1657 [XML Schema Part 2] "XML Schema Part 2: Datatypes", Paul V. Biron and Ashok Malhotra, eds.
1658 W3C Recommendation, 2 May 2001. Available at
1659 <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>
- 1660 [XPath 1.0] "XML Path Language (XPath) Version 1.0", James Clark, Steve DeRose, eds., W3C
1661 Recommendation, 16 November 1999. Available at
1662 <http://www.w3.org/TR/1999/REC-xpath-19991116>
- 1663 [WSDL 1.1] "Web Services Description Language (WSDL) 1.1", Erik Christensen, Francisco
1664 Curbera, Greg Meredith, Sanjiva Weerawarana, eds., W3C Note, 15 March 2001. Available at
1665 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- 1666 [WS-I BP 1.1] "Basic Profile Version 1.1", Keith Ballinger, David Ehnebuske, Christopher Ferris,
1667 Martin Gudgin, Mark Nottingham, Prasad Yendluri, eds., WS-I specification, 8 August 2004.
1668 Available at
1669 <http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-07-21.html>
- 1670 [WSS] "OASIS Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)", Chris
1671 Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds, OASIS Standard 200401, March 2004.
1672 Available at
1673 <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

1674 **Appendix A.Schema (Normative)**

1675 The schemas for this specification have the following URLs and are located using the filenames
1676 shown in the table:

Schema Namespace URL	File name	Prefix
http://docs.oasis-open.org/wsrn/2004/06/ws-reliability-1.1.xsd	ws-reliability-1.1.xsd	wsrn
http://docs.oasis-open.org/wsrn/2004/06/reference-1.1.xsd	reference-1.1.xsd	ref
http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd	fnp-1.1.xsd	fnp
http://docs.oasis-open.org/wsrn/2004/06/wsrmp-1.1.xsd	wsrmp-1.1.xsd	wsrmp

Table 28 WS-Reliability Schema Prefixes

1677 RMPs MUST include the SOAP mustUnderstand attribute (defined in the same namespace used
1678 for the soap:Envelope element) in all Reliable Messaging specified header blocks and MUST
1679 observe the following restrictions:

- 1680 • For SOAP 1.1, the mustUnderstand attribute value is restricted to "1".
- 1681 • For SOAP 1.2, the mustUnderstand attribute value is restricted to "1" or "true".

1682 **Appendix B.WS-Reliability Features, Properties**
1683 **and Compositors (Normative and Optional)**

1684 **B.1. Introduction**

1685 Users of a Web Service need to be aware of the reliability capabilities (RM capabilities) the
1686 service supports or requires. One practical location to advertise these capabilities is in the service
1687 description (WSDL document), which allows publishing both abstract service definitions and
1688 concrete protocol details (bindings). This allows clients (including other Web services) to easily
1689 obtain information about specific capabilities (such as guaranteed delivery, duplicate elimination,
1690 message ordering, and the supported reply patterns) of a specific Web service before calling the
1691 service. While bundling RM capabilities with the service description may not be desirable in all
1692 cases, this convenient approach often should be appropriate. The WSDL annotation mechanism
1693 described here adds such capability assertions in a flexible way.

1694 WS-Reliability uses the WSDL 1.1 extensibility points to define an extensible framework
1695 consisting of features, properties and compositors. This framework addresses the needs of a
1696 reliable Web service to advertise its capabilities and the composability of those capabilities.

1697 The following extensibility elements are relevant to RM capabilities:

- 1698 • **feature** – see [Appendix B.3.2](#).
- 1699 • **property** – see [Appendix B.3.3](#).
- 1700 • **compositor** – see [Appendix B.3.1](#).

1701 An annotation composed with the above extensibility elements will specify the reliability features
1702 and properties associated with specific WSDL constructs. Features and properties represent RM
1703 capabilities; compositors specify how these capabilities are composed.

1704 This would, for example, allow a Web service description to advertise that clients invoking the
1705 service must use duplicate elimination or message ordering.

1706 **B.2. Conformance**

1707 Implementations of WS-Reliability are expected (though not required) to understand the WSDL
1708 extensibility points defined in this section.

1709 Understanding these extensibility points promotes interoperability: a service advertises its
1710 supported and required features when its WSDL document contains these extensibility points.
1711 Therefore it is RECOMMENDED that implementations recognize, understand and support these
1712 extensibility points.

1713 It is also possible for services to advertise features through other channels (such as UDDI) in
1714 addition to these extensibility points.

1715 B.3. WSDL Extensibility Elements

1716 B.3.1. Compositor

1717 The compositor semantics describe how features and properties are composed for the enclosing
1718 component (or WSDL 1.1 element). The compositor's semantics determine whether the usage of
1719 composed elements by a client to the service is required or optional. All of the RM capabilities
1720 represented by these elements must be supported by the service. A compositor element can
1721 occur as a child element of wsdl11:portType, wsdl11:operation (which itself may be a child of
1722 wsdl11:portType or wsdl11:binding), wsdl11:binding, wsdl11:service and wsdl11:port. The
1723 compositor element uses the extensibility defined by WSDL 1.1. A compositor element specifies
1724 the semantics for combining its children elements. These children elements can be additional
1725 compositors, features, properties or extensibility elements.

1726 A compositor element is expressed by the following pseudo-syntax:

```
1727 <fnp:compositor uri="..." name="NCName"?>  
1728 [fnp:feature/> | <fnp:property/> | <fnp:compositor/> |  
1729 <extensibility-element/>]+  
1730 </fnp:compositor>
```

1731 The uri attribute of the compositor specifies its semantics. Four different compositors (URIs) and
1732 their capability-related semantics are described below. It is possible to provide additional
1733 compositors by using other URIs. The possibility of additional compositors and the existence of
1734 extensibility points (represented by "<extensibility-element>") make the framework extensible. The
1735 optional @name identifies the compositor. An element built with such compositors represents an
1736 RM capability.

1737 • **all:** this compositor specifies that a service invocation **MUST** comply with all of the
1738 children elements representing RM capability assertions. This compositor is identified by
1739 the URI:

1740 <http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositors/all>

1741 • **choice:** this compositor specifies that a service invocation **MUST** comply with exactly
1742 one of the possibly many children elements representing RM capability assertions. This
1743 compositor is identified by the URI:

1744 <http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositors/choice>

1745 • **one-or-more:** this compositor specifies that a service invocation **MUST** comply with at
1746 least one of the possibly many children elements representing RM capability assertions.
1747 This compositor is identified by the URI:

1748 <http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositors/one-or-more>

1749 • **zero-or-more:** this compositor specifies that a service invocation **MAY** comply with one
1750 or more of the children elements representing RM capability assertions. This compositor
1751 is identified by the URI:

1752 <http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositors/zero-or-more>

1753 Examples for each compositor are provided in **Appendix B.7** below.

1754 Compositors specified at different WSDL components are implicitly aggregated using the 'all'
1755 compositor at the dependent WSDL component. Consider the example below:

```

1756 <wsdl11:definitions>
1757   ...
1758   <wsdl11:portType name="myPortType">
1759     <fnp:compositor uri="..." name="A">
1760       ...
1761     </fnp:compositor>
1762     ...
1763   </wsdl11:portType>
1764   <wsdl11:binding name="myBinding" type="myPortType">
1765     <fnp:compositor uri="..." name="B">
1766       ...
1767     </fnp:compositor>
1768     ...
1769   <wsdl11:binding>
1770     <wsdl11:service name="myService">
1771       <wsdl11:port name="myPort" binding="myBinding">
1772         ...
1773       </wsdl11:port>
1774     </wsdl11:service>
1775   </wsdl11:definitions>

```

1776 The compositor specified at the wsdl11:portType "myPortType" and the compositor specified at
1777 wsdl11:binding "myBinding" are aggregated at the dependent wsdl11:port "myPort" using the 'all'
1778 compositor. The equivalent compositor at "myPort" is

```

1779 <fnp:compositor
1780   uri="http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositor/all">
1781   <fnp:compositor uri="..." name="A">
1782   </fnp:compositor>
1783   <fnp:compositor uri="..." name="B">
1784     ...
1785   </fnp:compositor>
1786 </fnp:compositor>

```

1787 **B.3.2. Feature**

1788 A feature describes an abstract RM capability or assertion associated with a WSDL element. A
1789 feature can occur only as a child of a compositor.

1790 The enclosing compositor(s) define(s) whether or not the usage of a feature is required. A feature
1791 is identified by a URI. Recognizing the URI of a feature implies understanding the feature
1792 identified by that URI.

1793 A feature element is expressed by the following pseudo-syntax:

```

1794 <fnp:feature uri="...">
1795   [<fnp:compositor/> | <extensibility-element/>]*
1796 </fnp:feature>

```

1797 **B.3.3. Property**

1798 A property is identified by a QName. A property is an assertion or constraint on a specific RM
1799 capability and its value(s). A property can occur only as a child of a compositor.

1800 Typically, properties are (but are not required to be) associated with a feature and are described in
1801 a feature specification. The QName identifier of a property uniquely identifies the property.
1802 Recognizing the property QName identifier implies understanding the semantics associated with
1803 that property. The property QName identifier typically points to a global XML Schema element
1804 declaration. A property specification typically specifies the schema containing this global element
1805 declaration. There may be a constraint on the set of values a property can have; such a constraint
1806 is specified by a QName identifying an XML Schema type.

```
1807 <fnp:property name="xs:QName">  
1808   [<fnp:value>xs:anyType</fnp:value> |  
1809     <fnp:constraint>xs:QName</fnp:constraint>]  
1810   [<extensibility-element/>]*  
1811 </fnp:property>
```

1812 **B.4. WS-Reliability Feature**

1813 The WS-Reliability feature is identified by the URI

1814 <http://docs.oasis-open.org/wsm/2004/06/wsmfp-1.1.xsd>

1815 This feature URI identifies the WS-Reliability specification. Understanding this URI implies
1816 understanding the WS-Reliability specification.

1817 **B.5. WS-Reliability Properties**

1818 This section identifies properties for the WS-Reliability specification. Typically these properties are
1819 scoped within the feature identified by the URI

1820 <http://docs.oasis-open.org/wsm/2004/06/wsmfp-1.1.xsd>

1821 **B.5.1. Guaranteed Delivery Property**

1822 This property is identified by the QName "wsmfp:GuaranteedDelivery" and corresponds to the
1823 semantics specified by the WS-Reliability guaranteed delivery semantics. The type of this property
1824 is "xs:boolean".

1825 **B.5.2. Duplicate Elimination Property**

1826 This property is identified by the QName "wsmfp:NoDuplicateDelivery" and corresponds to the
1827 semantics specified by the WS-Reliability duplicate elimination semantics. The type of this
1828 property is "xs:boolean".

1829 **B.5.3. Message Ordering Property**

1830 This property is identified by the QName "wsmfp:OrderedDelivery" and corresponds to the
1831 semantics specified by the WS-Reliability message ordering semantics. The type of this property
1832 is "xs:boolean".

1833 B.5.4. Reply Pattern Property

1834 This property is identified by the QName "wsrmfp:ReplyPattern" and corresponds to the semantics
1835 specified by the WS-Reliability reply pattern options. The type of this property is "xs:string".
1836 (values: Response, Poll, Callback)

1837 B.6. Compositor Examples

1838 B.6.1. Example for the "all" compositor

```
1839 <wsdl111:portType name="Example-1">  
1840   <fnp:compositor  
1841     uri="http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositor/all">  
1842     <fnp:feature  
1843       uri="http://docs.oasis-open.org/wsrn/2004/06/wsrnfp-1.1.xsd"  
1844       <fnp:compositor uri=  
1845         "http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositor/all">  
1846         <fnp:property name="wsrmfp:NoDuplicateDelivery">  
1847           <fnp:value>true</fnp:value>  
1848         </fnp:property>  
1849         <fnp:property name="wsrmfp:OrderedDelivery">  
1850           <fnp:value>true</fnp:value>  
1851         </fnp:property>  
1852         <fnp:property name="wsrmfp:GuaranteedDelivery">  
1853           <fnp:value>true</fnp:value>  
1854         </fnp:property>  
1855       </fnp:compositor>  
1856     </fnp:feature>  
1857   </fnp:compositor>  
1858   ...  
1859 </wsdl111:portType>
```

1860 In the example above, the reliability feature identified by URI "[http://docs.oasis-](http://docs.oasis-open.org/wsrn/2004/06/wsrnfp-1.1.xsd)
1861 [open.org/wsrn/2004/06/wsrnfp-1.1.xsd](http://docs.oasis-open.org/wsrn/2004/06/wsrnfp-1.1.xsd)" is required by the portType. This feature consists of
1862 three properties, all of which are required because of the semantics of the 'all' compositor that
1863 composes the three properties.

1864 **B.6.2. Example for the "choice" compositor:**

```
1865 <wsdl11:binding name="Example-2">
1866   <fnp:compositor
1867     uri="http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositor/all">
1868     <fnp:feature
1869       uri="http://docs.oasis-open.org/wsrn/2004/06/wsrnfp-1.1.xsd"
1870       <fnp:compositor uri=
1871         "http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositors/choice">
1872         <fnp:property name="wsrnf:ReplyPattern">
1873           <value>Response</value>
1874         </fnp:property>
1875         <fnp:property name="wsrnf:ReplyPattern">
1876           <value>Callback</value>
1877         </fnp:property>
1878         <fnp:property name="wsrnf:ReplyPattern">
1879           <value>Poll</value>
1880         </fnp:property>
1881       </fnp:compositor>
1882     </fnp:feature>
1883   </fnp:compositor>
1884   ...
1885 </wsdl11:binding>
```

1886 In the example above, the reliability feature identified by URI "[http://docs.oasis-](http://docs.oasis-open.org/wsrn/2004/06/wsrnfp-1.1.xsd)
1887 [open.org/wsrn/2004/06/wsrnfp-1.1.xsd](http://docs.oasis-open.org/wsrn/2004/06/wsrnfp-1.1.xsd)" is required by the portType. This feature consists of
1888 three properties composed by the 'choice' compositor; the client must choose one.

1889 **B.6.3. Example for the "one-or-more" compositor:**

```
1890 <wsdl11:portType name="Example-3">
1891   <fnp:compositor
1892     uri="http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositor/all">
1893     <fnp:feature
1894       uri="http://docs.oasis-open.org/wsrn/2004/06/wsrnfp-1.1.xsd" >
1895       <fnp:compositor uri=
1896         "http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositor/one-or-more">
1897         <fnp:property name="wsrnfp:NoDuplicateDelivery">
1898           <fnp:value>true</fnp:value>
1899         </fnp:property>
1900         <fnp:property name="wsrnfp:OrderedDelivery">
1901           <fnp:value>true</fnp:value>
1902         </fnp:property>
1903         <fnp:property name="wsrnfp:GuaranteedDelivery">
1904           <fnp:value>true</fnp:value>
1905         </fnp:property>
1906       </fnp:compositor>
1907     </fnp:feature>
1908   </fnp:compositor>
1909   ...
1910 </wsdl11:portType>
```

1911 **B.6.4. Example for the "zero-or-more" compositor:**

```
1912 <wsdl11:portType name="Example-4">
1913   <fnp:compositor
1914     uri="http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositor/all">
1915     <fnp:feature
1916       uri="http://docs.oasis-open.org/wsrn/2004/06/wsrnfp-1.1.xsd"
1917       <fnp:compositor uri=
1918         "http://docs.oasis-open.org/wsrn/2004/06/fnp-1.1.xsd/compositor/zero-or-more">
1919         <fnp:property name="wsrnfp:NoDuplicateDelivery">
1920           <fnp:value>true</fnp:value>
1921         </fnp:property>
1922         <fnp:property name="wsrnfp:OrderedDelivery">
1923           <fnp:value>true</fnp:value>
1924         </fnp:property>
1925         <fnp:property name="wsrnfp:GuaranteedDelivery">
1926           <fnp:value>true</fnp:value>
1927         </fnp:property>
1928       </fnp:compositor>
1929     </fnp:feature>
1930   </fnp:compositor>
1931   ...
1932 </wsdl11:portType>
```

1933 **Appendix C.Acknowledgments**

1934 The following individuals were members of the committee during the development of this
1935 specification:

- 1936 • David Ingham, Arjuna Technologies Limited
- 1937 • Joseph Chiusano, Booz Allen Hamilton
- 1938 • Peter Furniss, Choreology Ltd
- 1939 • Jeff Turpin, Cyclone Commerce
- 1940 • Pramila Mullan, France Telecom
- 1941 • Jacques Durand, Fujitsu
- 1942 • Kazunori Iwasa (Secretary), Fujitsu
- 1943 • Tom Rutt (Chair), Fujitsu
- 1944 • Jishnu Mukerji, Hewlett-Packard
- 1945 • Robert Freund, Hitachi
- 1946 • Eisaku Nishiyama, Hitachi
- 1947 • Nobuyuki Yamamoto, Hitachi
- 1948 • Ben Bloch, Individual
- 1949 • Mark Hansen, Individual
- 1950 • Paolo Romano, Individual
- 1951 • Dock Allen, Mitre Corporation
- 1952 • Junichi Tatemura, NEC Corporation
- 1953 • Alan Weissberger, NEC Corporation
- 1954 • Magdolna Gerendai, Nokia
- 1955 • Szabolcs Payrits, Nokia
- 1956 • Mark Peel, Novell
- 1957 • Sunil Kunisetty (Secretary), Oracle
- 1958 • Anish Karmarkar, Oracle
- 1959 • Jeff Mischkinisky, Oracle
- 1960 • Marc Goodner (Secretary), SAP
- 1961 • Pete Wenzel, SeeBeyond Technology Corporation

- 1962 • Doug Bunting (Secretary), Sun Microsystems
- 1963 • Tony Graham, Sun Microsystems
- 1964 • Chi-Yuen Ng, University of Hong Kong
- 1965 • Patrick Yee, University of Hong Kong
- 1966 • Prasad Yendluri, webMethods, Inc.
- 1967 • Scott Werden, WRQ, Inc.

1968 And the following people made contributions to produce Ver 1.0 of this specification:

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

1979

Appendix D. Notices

1980 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
1981 that might be claimed to pertain to the implementation or use of the technology described in this
1982 document or the extent to which any license under such rights might or might not be available;
1983 neither does it represent that it has made any effort to identify any such rights. Information on
1984 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
1985 website. Copies of claims of rights made available for publication and any assurances of licenses
1986 to be made available, or the result of an attempt made to obtain a general license or permission
1987 for the use of such proprietary rights by implementors or users of this specification, can be
1988 obtained from the OASIS Executive Director.

1989 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
1990 applications, or other proprietary rights which may cover technology that may be required to
1991 implement this specification. Please address the information to the OASIS Executive Director.

1992 **Copyright © OASIS Open 2003-2004. All Rights Reserved.**

1993 This document and translations of it may be copied and furnished to others, and derivative works
1994 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
1995 published and distributed, in whole or in part, without restriction of any kind, provided that the
1996 above copyright notice and this paragraph are included on all such copies and derivative works.
1997 However, this document itself does not be modified in any way, such as by removing the copyright
1998 notice or references to OASIS, except as needed for the purpose of developing OASIS
1999 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
2000 Property Rights document must be followed, or as required to translate it into languages other
2001 than English.

2002 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
2003 successors or assigns.

2004 This document and the information contained herein is provided on an "AS IS" basis and OASIS
2005 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
2006 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
2007 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
2008 PARTICULAR PURPOSE.