



Web Services Base Notification 1.3 (WS-BaseNotification)

Public Review Draft 01, 07 July 2005

Document identifier

wsn-ws_base_notification-1.3-spec-pr-01

Location:

http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-01.pdf

Editors:

Steve Graham, IBM <sggraham@us.ibm.com>

David Hull, Tibco <dmh@tibco.com>

Bryan Murray, Hewlett-Packard Company <bryan.murray@hp.com>

Abstract:

The Event-driven, or Notification-based, interaction pattern is a commonly used pattern for inter-object communications. Examples exist in many domains, for example in publish/subscribe systems provided by Message Oriented Middleware vendors, or in system and device management domains. This notification pattern is increasingly being used in a Web services context.

WS-Notification is a family of related specifications that define a standard Web services approach to notification using a topic-based publish/subscribe pattern. It includes: standard message exchanges to be implemented by service providers that wish to participate in Notifications, standard message exchanges for a notification broker service provider (allowing publication of messages from entities that are not themselves service providers), operational requirements expected of service providers and requestors that participate in notifications, and an XML model that describes topics. The WS-Notification family of documents includes three normative specifications: [WS-BaseNotification], [WS-BrokeredNotification], and WS-Topics.

Status:

On July 7th, 2005, the OASIS WS-Notification Technical Committee approved this document for publication as a Public Review Draft. Committee members should send comments on this specification to the wsn@lists.oasis-open.org list. Others may submit comments to the TC via the web form found on the TC's web page at <http://www.oasis->

35 open.org/committees/wsn. Click the button for "Send A Comment" at the top of the page.
36 Submitted comments (for this work as well as other works of that TC) are publicly
37 archived and can be viewed at:
38 <http://lists.oasis-open.org/archives/wsn-comment/>.
39 For information on whether any patents have been disclosed that may be essential to
40 implementing this specification, and any offers of patent licensing terms, please refer to
41 the Intellectual Property Rights section of the WSN TC web page ([http://www.oasis-
open.org/committees/wsn/](http://www.oasis-
42 open.org/committees/wsn/)).

43 **Table of Contents**

44	1	Introduction.....	5
45	1.1	Goals and Requirements.....	5
46	1.1.1	Requirements.....	5
47	1.1.2	Non-Goals.....	6
48	1.2	Terminology	6
49	1.3	Namespaces	7
50	1.4	Fault Definitions.....	7
51	2	Terminology and Concepts	8
52	2.1	Definitions	8
53	2.2	Production vs. Delivery.....	9
54	3	NotificationConsumer Interface.....	10
55	3.1	Notification Metadata.....	10
56	3.2	Notify.....	11
57	3.2.1	Example SOAP Encoding of the Notify Message	11
58	4	NotificationProducer Interface.....	13
59	4.1	NotificationProducer Resource Properties.....	13
60	4.2	Subscribe	14
61	4.2.1	Example SOAP Encoding of the Subscribe Message Exchange	20
62	4.3	GetCurrentMessage	21
63	4.3.1	Example SOAP Encoding of the GetCurrentMessage Message Exchange	22
64	5	Pull-Style Notification.....	24
65	5.1	PullPoint Interface	24
66	5.1.1	Accumulating Notification Messages.....	24
67	5.1.2	GetMessages.....	24
68	5.1.3	Example SOAP Encoding of the GetMessages Message Exchange	26
69	5.1.4	Destroy	27
70	5.1.5	Example SOAP Encoding of the Destroy Message Exchange.....	27
71	5.2	Create PullPoint Interface.....	28
72	5.2.1	Example SOAP Encoding of the CreatePullPoint Message Exchange.....	29
73	6	SubscriptionManager Interface.....	30
74	6.1	Base Subscription Manager.....	30
75	6.1.1	Renew	30
76	6.1.2	Unsubscribe	32
77	6.2	Pausable Subscription Manager	34
78	6.2.1	PauseSubscription	34
79	6.2.2	Example SOAP Encoding of the PauseSubscription Message Exchange.....	35

80	6.3	ResumeSubscription	35
81	6.3.1	Example SOAP Encoding of the ResumeSubscription Message Exchange.....	36
82	6.4	Subscriptions as WS-Resources.....	37
83	6.4.1	Subscription Resource Properties.....	37
84	7	Security Considerations	39
85	7.1	Securing the Message Exchanges.....	39
86	7.2	Securing Subscriptions and Notifications	39
87	8	References.....	41
88	8.1	Normative.....	41
89	8.2	Non-Normative	41
90		Appendix A. Acknowledgments	41
91		Appendix B. XML Schema.....	42
92		Appendix C. WSDL 1.1	54
93		Appendix D. Revision History	63
94		Appendix E. Notices.....	65
95			

96 1 Introduction

97 The Event-driven, or Notification-based, interaction pattern is commonly used in inter-object
98 communications. Examples exist in many domains, for example in publish/subscribe systems
99 provided by Message Oriented Middleware vendors, or in system and device management
100 domains.

101 The WS-Notification family of specifications defines a standard Web services approach to
102 notification. This document is the base specification on which all the other specifications in the
103 family depend. It defines the normative Web services interfaces for two of the important roles in
104 the notification pattern, namely the NotificationProducer and NotificationConsumer roles. This
105 specification includes standard message exchanges to be implemented by service providers that
106 wish to act in these roles, along with operational requirements expected of them.

107 In the remainder of this section and section 2 we will give a brief introduction to the Notification
108 pattern, and the terms we will use in this specification.

109 In the Notification pattern a Web service, or other entity, disseminates information to a set of
110 other Web services, without having to have prior knowledge of these other Web services.

111 This specification defines a role called the NotificationProducer. A NotificationProducer is capable
112 of producing a set of Notification messages. A NotificationProducer accepts incoming Subscribe
113 requests. Each Subscribe request contains a reference to a NotificationConsumer and identifies
114 the subset of the Notifications the NotificationProducer should produce. This subset can be
115 described by identifying one or more boolean filters, including filtering by Topic, as discussed in
116 [WS-Topics]. The NotificationProducer agrees to produce Notification Messages as requested in
117 the Subscribe request, or returns a fault if the subscription cannot be handled.

118 The production of Notifications may be realized in a number of ways. One particular configuration,
119 in which the NotificationProducer reproduces Notifications produced by other entities, is
120 described in the [WS-Brokered Notification] specification. Alternatively, a NotificationProducer
121 may produce Notifications itself. An implementer interested only in such direct, point-to-point,
122 notification need only refer to this WS-BaseNotification specification.

123 1.1 Goals and Requirements

124 The goal of WS-BaseNotification is to standardize the terminology, concepts, operations, WSDL
125 and XML needed to express the basic roles involved in Web services publish and subscribe for
126 notification message exchange.

127 1.1.1 Requirements

128 In meeting these goals, the WS-BaseNotification specification must explicitly address the
129 following requirements:

- 130 • **Must support resource-constrained devices.** The specifications must be factored in a way
131 that allows resource-constrained devices to participate in the Notification pattern. Such
132 devices will be able to send information to, and receive information from Web services,
133 without having to implement all the features of the specifications.
- 134 • **Must provide runtime metadata:** There must be a mechanism that lets a potential
135 Subscriber discover what elements are provided for subscription by a NotificationProducer,
136 and in what formats the subscription for notification can be made.

137 In addition, the WS-BaseNotification must allow for the following requirements to be met:

- 138 • **WS-BaseNotification must be independent of binding-level details:** Transport protocol
139 details must be orthogonal to the subscription and the delivery of the notifications, so that the
140 specification can be used over a variety of different transports.
- 141 • **Must allow for Message Oriented Middleware implementations.** The design of the WS-
142 BaseNotification must allow a service that is acting as a NotificationProducer to delegate its
143 implementation of WS-BaseNotification semantics to a Message Oriented Middleware
144 provider.
- 145 • **Relationship to other WS-* specifications:** WS-BaseNotification must be composable with
146 other Web services specifications.

147 1.1.2 Non-Goals

148 The following topics are outside the scope of the WS-BaseNotification specification:

- 149 • **Defining the format of notification payloads:** The data carried in Notification payloads is
150 application-domain specific, and WS-BaseNotification does not prescribe any particular
151 format for this data.
- 152 • **Defining any Notifications.** The WS-BaseNotification specification does not define any
153 “standard” or “built-in” notification situations or messages.
- 154 • **Defining the mapping between Situations and Notifications.** The WS-BaseNotification
155 specification does not define the circumstances under which a potential producer of
156 information should decide if and when it should actually produce notifications. However they
157 do define the form and semantics of the notification once it has decided to do so.
- 158 • **Defining the means by which NotificationProducers are discovered by subscribers.** It is
159 beyond the scope of this specification to define the mechanisms for runtime discovery of
160 NotificationProducers.
- 161 • **Defining message ordering or interleaving policies for delivery is beyond the scope of**
162 **WS-BaseNotification.**

163

164 1.2 Terminology

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

168 When describing abstract data models, this specification uses the notational convention used by
169 the [XML Infoset]. Specifically, abstract property names always appear in square brackets (e.g.,
170 [some property]).

171 This specification uses a notational convention, referred to as “Pseudo-schemas” in a fashion
172 similar to the WSDL 2.0 Part 1 specification [WSDL 2.0]. A Pseudo-schema uses a BNF-style
173 convention to describe attributes and elements:

- 174 • ‘?’ denotes optionality (i.e. zero or one occurrences),
175 • ‘*’ denotes zero or more occurrences,
176 • ‘+’ one or more occurrences,
177 • ‘[’ and ‘]’ are used to form groups,
178 • ‘|’ represents choice.

- Attributes are conventionally assigned a value which corresponds to their type, as defined in the normative schema.

```

181 <!-- sample pseudo-schema -->
182 <element
183   required_attribute_of_type_QName="xs:QName"
184   optional_attribute_of_type_string="xs:string"? >
185   <required_element />
186   <optional_element /> ?
187   <one_or_more_of_these_elements /> +
188   [ <choice_1 /> | <choice_2 /> ] *
189 </element>

```

Where there is disagreement between the separate XML schema and WSDL files describing the messages defined by this specification and the normative descriptive text (excluding any pseudo-schema) in this document, the normative descriptive text will take precedence over the separate files. The separate files take precedence over any pseudo-schema and over any schema and WSDL included in the appendices.

1.3 Namespaces

The following namespaces are used in this document:

Prefix	Namespace
s	http://schemas.xmlsoap.org/soap/envelope/ OR http://www.w3.org/2003/05/soap-envelope
xsd	http://www.w3.org/2001/XMLSchema
wsa	http://www.w3.org/2005/03/addressing
wsrf-rp	http://docs.oasis-open.org/wsrp/rp-1
wsrf-bf	http://docs.oasis-open.org/wsrp/bf-1
wsnt	http://docs.oasis-open.org/wsn/b-1
wsntw	http://docs.oasis-open.org/wsn/bw-1

1.4 Fault Definitions

- All faults generated by a NotificationProducer or SubscriptionManager SHOULD be compliant with the WS-BaseFaults [WS-BaseFaults] specification.
- All faults defined by this specification MUST use the following URI for the WS-Addressing [action] Message Addressing Property:
<http://docs.oasis-open.org/wsn/fault>.

203 2 Terminology and Concepts

204 2.1 Definitions

205 The following definitions outline the terminology and usage in this specification.

206 **Situation:**

- 207 • A Situation is some occurrence known to a NotificationProducer and of potential interest
208 to third parties.
- 209 • A Situation could be a change of the internal state of a resource or could be
210 environmental, such as a timer event. It could also be an external event, such as a piece
211 of news that has been supplied by a news-feed service.
- 212 • WS-Notification does not specify what a Situation is or is not, nor does it define the
213 relationship between a Situation and the Notification(s) that are used to describe it.

214 **Notification:**

- 215 • A Notification is an artifact of a Situation containing information about that Situation that
216 some entity wishes to communicate to other entities.
- 217 • A Notification is represented as an XML element with a Namespace qualified QName and
218 a type defined using XML Schema.
- 219 • A typical usage pattern is to define a single Notification type (to be precise, its defining
220 XML element) for each kind of Situation, containing information pertinent to that kind of
221 Situation; in this case one can think of a Notification instance as in some sense *being* (or
222 at least representing) the Situation.
- 223 • A designer could choose to associate several different Notification types with a Situation,
224 for example, describing different aspects of the Situation, destined for different target
225 recipients, etc. Conversely it is possible that several essentially different Situations give
226 rise to Notifications of the same type.

227 **NotificationProducer:**

- 228 • A NotificationProducer is a Web service that implements the message exchanges
229 associated with the NotificationProducer interface.
- 230 • A NotificationProducer is capable of producing Notifications for those
231 NotificationConsumers for which Subscriptions have been registered, based on
232 Situations that occur and on the parameters supplied with the requests from which the
233 Subscriptions were created.
- 234 • A Web Service that implements the message exchanges associated with
235 NotificationProducer may directly produce Notifications itself, or it may be a
236 NotificationBroker, reproducing Notifications that were produced by separate Publisher
237 and/or NotificationProducer entities.
- 238 • It is the factory for Subscription resources.

239 **NotificationConsumer:**

- 240 • A NotificationConsumer is an endpoint, represented by a WS-Addressing endpoint
241 reference, designated to receive Notifications produced by a NotificationProducer as a
242 result of a subscription.
- 243 • A NotificationConsumer may accept the generic Notify message, or it may be able to
244 process one or more domain-specific Notification types.

- 245 **Subscription:**
- 246
- 247
- 248
- 249
- 250
- 251
- 252
- A Subscription represents the relationship between a NotificationConsumer and a NotificationProducer, including any filtering parameters such as Topic and various other optional filter expressions, along with any relevant policies and context information.
 - A Subscription resource is created when a Subscriber sends the SubscribeRequest message to a NotificationProducer.
 - Subscription resources are manipulated by messages sent to the SubscriptionManager Web service associated with the Subscription resource.

253 **SubscriptionManager**

- 254
- 255
- 256
- 257
- 258
- 259
- 260
- 261
- 262
- A SubscriptionManager is an endpoint, represented by an endpoint reference [WS-Addressing] that implements message exchanges associated with the SubscriptionManager interface.
 - A SubscriptionManager provides operations that allow a service requestor to query and manipulate Subscription resources that it manages
 - A SubscriptionManager is subordinate to the NotificationProducer, and MAY be implemented by the NotificationProducer service provider. However WS-Notification permits it to be implemented by a separate service provider, should an implementer so desire.

263 **Subscriber:**

- 264
- 265
- 266
- 267
- A Subscriber is any entity that sends the SubscribeRequest message to a NotificationProducer.
 - Note that a Subscriber may be a different entity from the NotificationConsumer for which Notifications are actually produced.

268 **2.2 Production vs. Delivery**

269 Various statements in this document refer to a NotificationProducer *producing* a Notification.

270 Producing a Notification means: *supplying a Notification to a delivery mechanism for transmission*

271 *to a NotificationConsumer*. Depending on the actual delivery mechanism, this transmission might

272 be reliable or might be done on a best-effort basis. A Notification which is never produced will

273 definitely never be delivered, but the converse is not necessarily true: a Notification which *is*

274 produced may or may not actually be delivered, depending on the delivery mechanism, the

275 validity of the NotificationConsumer address, the state of the network, and so forth.

276 **3 NotificationConsumer Interface**

277 WS-BaseNotification allows a NotificationConsumer to receive a Notification in one of two forms:

- 278 1. The NotificationConsumer MAY simply receive the “raw” Notification (i.e. the application-
279 specific content).
- 280 2. The NotificationConsumer MAY receive the Notification data as a Notify message as
281 described below.

282 The second option (the Notify message) provides a well specified mechanism by which the
283 NotificationProducer can supply additional information defined by WS-BaseNotification (such as
284 the Topic) in addition to the application-specific Notification content. It also allows the
285 NotificationConsumer to receive a wide range of Notifications without having to explicitly provide
286 support for each one in its WSDL. This form of Notification also allows a batch of several
287 Notifications to be combined into a single physical message.

288 When a Subscriber sends a Subscribe request message, it indicates which form of Notification is
289 required (the raw Notification, or the Notify Message). The NotificationProducer MUST observe
290 this component of the Subscription and use the form that has been requested.

291 This means that a NotificationConsumer MAY choose to implement the Notify Message, or to
292 implement raw Notification(s) explicitly (or both). When requesting creation of a Subscription on
293 behalf of a NotificationConsumer, a Subscriber SHOULD ensure that the NotificationConsumer is
294 able to handle the form of Notification it has requested for the given Subscription.

295 **3.1 Notification Metadata**

296 In order to inform the NotificationConsumer about the Situation that produced a Notification and
297 to allow the NotificationConsumer to manipulate the underlying subscription, the
298 NotificationProducer MAY include any combination of the following metadata elements in the
299 Notifications it produces:

300 wsnt:SubscriptionReference

301 An EndpointReference to the Subscription WS-Resource [WS-Resource] that is
302 associated with the Notify message.

303 wsnt:Topic

304 A TopicExpression describing exactly one Topic, which MUST be the Topic that is
305 associated with the Notification. This element describes the Topic that matched to a
306 subscription, causing the NotificationProducer to send the Notify message to the
307 NotificationConsumer.

308 wsnt:Topic/@Dialect

309 The dialect used in the TopicExpression. This MUST be the same dialect used by the
310 Subscriber when it created the Subscription that yielded this Notify message.

311 wsnt:ProducerReference

312 An EndpointReference to the NotificationProducer that produced the Notification artifact.

313

314 When using the Notify message format, these elements MUST appear within the
315 wsnt:NotificationMessage element as described below. When using the raw message format,
316 these elements MAY appear as components of the message (such as SOAP header elements) in
317 accordance with any contract with the NotificationConsumer.

318 3.2 Notify

319 A NotificationProducer MAY produce a Notify message containing one or more Notification(s).
320 The format of the Notify message is:

```
321 ...  
322 <wsnt:Notify>  
323   <wsnt:NotificationMessage>  
324     <wsnt:SubscriptionReference>  
325       wsa:EndpointReference  
326     </wsnt:SubscriptionReference> ?  
327     <wsnt:Topic Dialect="xsd:anyURI">  
328       {any} *  
329     </wsnt:Topic>?  
330     <wsnt:ProducerReference>  
331       wsa:EndpointReference  
332     </wsnt:ProducerReference> ?  
333     <wsnt:Message>  
334       {any}  
335     </wsnt:Message>  
336   <wsnt:NotificationMessage> +  
337   {any} *  
338 </wsnt:Notify>  
339 ...
```

340 The WS-Addressing [action] Message Addressing Property MUST contain the URI
341 <http://docs.oasis-open.org/wsn/bw-1/NotificationConsumer/Notify>.

342 The components of the Notify message are further described as follows:

343 /wsnt:Notify

344 Contains a collection of one or more Notifications.

345 /wsnt:Notify/wsnt:NotificationMessage

346 Contains a Notification payload along with any metadata components as defined in
347 section 3.1.

348 /wsnt:Notify/wsnt:NotificationMessage/wsnt:Message

349 A copy of the actual Notification payload.

350 /wsnt:Notify/{any}

351 The Notify message also allows for open content, in order to accommodate elements that
352 may be needed by extensions built on BaseNotification, including those providing
353 additional filtering mechanisms.

354

355 No response is expected from the NotificationConsumer upon receipt of this message.

356 3.2.1 Example SOAP Encoding of the Notify Message

357 The following is a non-normative example of a Notify message using SOAP:

```
358 <s:Envelope ... >  
359   <s:Header>
```

```

360     <wsa:Action>
361         http://docs.oasis-open.org/wsn/bw-1/NotificationConsumer/Notify
362     </wsa:Action>
363     ...
364 </s:Header>
365 <s:Body>
366     <wsnt:Notify>
367         <wsnt:NotificationMessage>
368             <wsnt:SubscriptionReference>
369                 <wsa:Address>
370                     http://www.producer.org/SubsMgrEndpoint
371                 </wsa:Address>
372             </wsnt:SubscriptionReference>
373             <wsnt:Topic Dialect=
374 "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
375                 npex:SomeTopic
376             </wsnt:Topic>
377             <wsnt:ProducerReference>
378                 <wsa:Address>
379                     http://www.producer.org/ProducerEndpoint
380                 </wsa:Address>
381             </wsnt:ProducerReference>
382             <wsnt:Message>
383                 <npex:NotifyContent>exampleNotifyContent</npex:NotifyContent>
384             </wsnt:Message>
385             <wsnt:NotificationMessage>
386         </wsnt:Notify>
387     </s:Body>
388 </s:Envelope>

```

389 4 NotificationProducer Interface

390 This section describes the message exchanges that a NotificationProducer MUST support. These
391 message exchanges allow the NotificationProducer to advertise its support for one or more
392 Topics, and allow a Subscriber to create Subscriptions.

393 4.1 NotificationProducer Resource Properties

394 In addition to the message exchanges described in this specification, a NotificationProducer MAY
395 also support the required message exchanges defined in the WS-ResourceProperties
396 specification and MAY support the optional message exchanges defined in the WS-
397 ResourceProperties specification. In such a case, this specification defines several resource
398 properties, which MUST conform to the following schema fragment for content:

```
399 ...  
400     targetNamespace="http://docs.oasis-open.org/wsn/b-1"  
401     ...  
402  
403     <xsd:element name="TopicExpression" type="wsnt:TopicExpressionType" />  
404  
405     <xsd:element name="FixedTopicSet" type="xsd:boolean" default="true" />  
406  
407     <xsd:element name="TopicExpressionDialect" type="xsd:anyURI" />  
408     ...
```

409 These properties must also conform to the following schema fragment for cardinality:

```
410  
411     <xsd:element ref="wsnt:TopicExpression"  
412         minOccurs="0" maxOccurs="unbounded" />  
413  
414     <xsd:element ref="wsnt:FixedTopicSet "  
415         minOccurs="0" maxOccurs="1" />  
416  
417     <xsd:element ref="wsnt:TopicExpressionDialect "  
418         minOccurs="0" maxOccurs="unbounded" />  
419
```

420

421 These Resource Property elements are further constrained as follows:

422 /wsnt:TopicExpression

423 This resource property contains a collection of topics supported by the
424 NotificationProducer. The set of topics is expressed using one or more
425 wsnt:TopicExpression resource property element(s). Each wsnt:TopicExpression
426 resource property element is a TopicExpression. The dialect of TopicExpression used
427 can be any dialect. It is RECOMMENDED to use one of the TopicExpression dialects
428 described in [WS-Topics]. A NotificationProducer MAY use an expression that refers to
429 multiple topics, if the dialect used has this capability. The same topic may be referred to

430 by multiple wsnt:TopicExpression resource property element(s), for example, in different
431 resource property elements each using a different dialect.

432 If a topic is identified by one of the wsnt:TopicExpression resource property elements, a
433 Subscriber can reasonably expect that the NotificationProducer will not return an
434 InvalidTopicExpressionFault for subscription requests for the topic. It is not a guarantee
435 that it will receive any Notifications; the NotificationProducer may not actually produce
436 any Notifications on the particular topic during the time that the Subscriber is registered.

437 /wsnt:FixedTopicSet

438 Indicates if the collection of topics contained within the wsnt:TopicExpression resource
439 property may change. This value is “true” if the collection of topics supported by the
440 NotificationProducer does not change and “false” if the NotificationProducer allows the
441 collection to change (for example if it allows additional topics to be supported should
442 publishers or subscribers request them). This property is optional and defaults to “true” if
443 missing.

444 /wsnt:TopicExpressionDialect

445 Indicates one or more TopicExpression dialect(s) that are supported by the
446 NotificationProducer. If a URI corresponding to a dialect appears in this resource
447 property, a subscriber is assured that a subscription request containing a valid topic
448 expression using that dialect will be accepted by the NotificationProducer. The
449 TopicExpressionDialect property is a “fixed property”, i.e. its value, for any given
450 NotificationProducer WS-Resource, does not change over time.

451 **4.2 Subscribe**

452 A NotificationProducer is capable of producing a sequence of zero or more Notifications. A
453 Subscriber can register the interest of a NotificationConsumer to receive a subset of this
454 sequence. A Subscriber sends a Subscribe message to a NotificationProducer in order to register
455 this interest.

456 If the processing of a Subscribe message is successful, the NotificationProducer MUST produce
457 a response message, as described below, containing an endpoint reference representing a
458 Subscription created as a result of processing the Subscribe request.

459 Sending two identical Subscribe messages to a NotificationProducer MUST result in the creation
460 of two Subscriptions. The NotificationConsumer will be associated with both Subscriptions with
461 the result that two copies of any matching Notification will be produced for that consumer.

462 A given NotificationConsumer may be the object of more than one Subscription, and separate
463 NotificationConsumers may subscribe to the same subset of Notifications. In such situations, WS-
464 BaseNotification places no restrictions on the order in which Notifications are produced.

465 Notifications for different NotificationConsumers may be produced in different orders, even when
466 the associated subscription requests are otherwise identical, and Notifications from separate
467 Subscriptions with the same NotificationConsumer may be interleaved in any manner.

468 NotificationProducers MAY advertise more constrained behavior through policy assertions or
469 other means.

470 The format of the Subscribe message is:

```
471 ...  
472 <wsnt:Subscribe>
```

```

473 <wsnt:ConsumerReference>
474   wsa:EndpointReference
475 </wsnt:ConsumerReference>
476 <wsnt:Filter>
477   [ <wsnt:TopicExpression Dialect="xsd:anyURI">
478     {any} ?
479   </wsnt:TopicExpression> |
480   <wsnt:ProducerProperties Dialect="xsd:anyURI">
481     {any} ?
482   </wsnt:ProducerProperties> |
483   <wsnt:MessageContent Dialect="xsd:anyURI">
484     {any}
485   </wsnt:MessageContent> |
486   {any} *
487   ] *
488 </wsnt:Filter> ?
489 <wsnt:InitialTerminationTime>
490   [xsd:dateTime | xsd:duration]
491 </wsnt:InitialTerminationTime> ?
492 <wsnt:SubscriptionPolicy>
493   [ <wsnt:UseRaw/> |
494     {any}
495   ] *
496 </wsnt:SubscriptionPolicy> ?
497 {any}*
498 </wsnt:Subscribe>
499 ...

```

500 The WS-Addressing [action] Message Addressing Property MUST contain the URI
501 <http://docs.oasis-open.org/wsn/bw-1/NotificationProducer/SubscribeRequest>.

502 The components of the Subscribe message are further described as follows:

503 /wsnt:ConsumerReference

504 An endpoint reference element, as defined by WS-Addressing [WS-Addressing], used to
505 identify the NotificationConsumer.

506 This component SHOULD provide all the necessary information to specify how the
507 NotificationProducer should send notifications to the NotificationConsumer. However,
508 there may be cases when the ConsumerReference EPR may not include all the details
509 that the NotificationProducer expects.

510 The NotificationProducer should specify via WSDL, policy assertions, meta-data or by
511 some other means, the information it expects to be present in a ConsumerReference. If a
512 ConsumerReference does not contain sufficient information, the NotificationProducer
513 MAY choose to fault or it MAY choose to use out of band mechanisms to obtain the
514 required information.

515 In cases where the optional wsnt:UseRaw policy component is not specified, the Web
516 service identified by the endpoint reference MUST implement the message exchanges
517 defined by NotificationConsumer (i.e., the Notify message).

518 /wsnt:Filter

519 The Filter component is the means by which a Subscriber expresses the subset of
520 Notifications that the NotificationConsumer should receive. This subset is expressed by
521 the child elements of the Filter. The child elements are a sequence of zero or more
522 expressions evaluating to a Boolean that constrain the set of possible Notifications. Each
523 expression is evaluated in a manner specific to that kind of expression (see below); the
524 order and timing of the evaluation is determined by the NotificationProducer. All filter
525 expressions MUST evaluate to true in order for the notification message to be sent.

526 If no Filter component appears in a Subscribe message, then the Subscriber's intent is
527 for the NotificationConsumer to receive every message produced by the
528 NotificationProducer. The NotificationProducer MAY reject Subscribe requests that do not
529 contain a Filter component.

530 The NotificationProducer MUST respond with an InvalidFilterFault message if any child
531 expression element is not supported by the NotificationProducer. For example, if the
532 NotificationProducer does not support the concept of Topics, it MUST respond with an
533 InvalidFilterFault message if a Subscribe message contains a Filter component that
534 includes a TopicExpression child. The fault MUST include the filter QNames for the
535 filters it did not understand.

536 This specification defines the filters TopicExpression, ProducerProperties, and
537 MessageContent. A NotificationProducer MAY accept these filters and/or any other filters
538 that may be defined.

539 /wsnt:Filter/wsnt:TopicExpression

540 A filter limiting notification messages to those that are associated with at least one topic
541 matching the TopicExpression. The TopicExpression identifies one or more topics
542 supported by the NotificationProducer.

543 /wsnt:Filter /wsnt:TopicExpression/@Dialect

544 A REQUIRED attribute of type URI that identifies the language of the TopicExpression.
545 WS-Topics defines an initial set of standard URIs for TopicExpressions. Designers MAY
546 define and use other domain-specific URIs to identify the dialect of the TopicExpression.

547 The NotificationProducer MAY refuse to process the Subscribe request if the dialect used
548 by the Subscriber in the TopicExpression is not one of the dialects supported by the
549 NotificationProducer.

550 The NotificationProducer MUST respond with a TopicExpressionDialectUnknownFault if it
551 understands the TopicExpression element, but does not understand the specified
552 TopicExpression dialect. Note that a NotificationProducer may understand the meaning
553 of a given dialect URI without actually supporting that dialect.

554 If the TopicExpression is incompatible with or does not comply with the rules of the
555 dialect, the NotificationProducer MUST respond with an InvalidTopicExpressionFault.

556 If the TopicExpression dialect is understood and the expression references a topic which
557 is not supported by the NotificationProducer, the NotificationProducer MAY return a
558 TopicNotSupportedFault.

559 /wsnt:Filter /wsnt:ProducerProperties

560 This component contains a filter expression evaluated on the ResourceProperties [WS-
561 ResourceProperties] of the NotificationProducer. The expression MUST be a Boolean
562 expression.

563 /wsnt:Filter /wsnt:ProducerProperties/@Dialect

564 This attribute contains a URI specifying the type of ProducerProperties filter expression
565 contained by the element. Some standard URIs are defined by the WS-
566 ResourceProperties specification. Designers MAY define and use other domain-specific
567 URIs to identify the dialect of the ProducerProperties filter expression.

568 The NotificationProducer MAY refuse to process the Subscribe request if the Dialect
569 used by the Subscriber is not one of the dialects supported by the NotificationProducer.

570 /wsnt:Filter /wsnt:MessageContent

571 A filter limiting notification messages to the set for which the specified expression
572 evaluated over the Notification Message to be produced evaluates to true A
573 wsnt:MessageContent expression MUST evaluate to a Boolean. The evaluation context
574 is Notification payload.

575 /wsnt:Filter /wsnt:MessageContent/@Dialect

576 This attribute contains a URI specifying the type of MessageContent filter expression
577 contained by the element. There are two well known dialects identified by this
578 specification, corresponding to two versions of the XPath language.

579 <http://www.w3.org/TR/1999/REC-xpath-19991116>

580 This URI identifies the XPath 1.0 language. The contents of the
581 MessageContent expression MUST be a string containing a valid XPath
582 1.0 expression.

583 <http://www.w3.org/TR/2005/WD-xpath20-20050404>

584 This URI identifies the Xpath 2.0 (working draft) language. The contents
585 of the MessageContent expression MUST be a string containing a valid
586 XPath 2.0 expression. Note: an additional URI will be added to represent
587 the W3C Recommendation form of the XPath 2.0 language.

588 Designers MAY define and use other domain-specific URIs to identify the dialect of the
589 MessageContent filter expression.

590 The NotificationProducer MAY refuse to process the Subscribe request if the dialect used
591 by the Subscriber is not one of the dialects supported by the NotificationProducer.

592 /wsnt:InitialTerminationTime

593 This component contains the service requestor's suggestion for the initial termination
594 time of the Subscription being created. There are two forms of this component, absolute
595 time and duration. If the type of this component is xsd:dateTime, the value of the
596 component is to be interpreted as an absolute time. If the type of this component is
597 xsd:duration, the value of the component is to be interpreted as a duration to be added to
598 the current time. All time measurements are determined by the NotificationProducer.

599 The resulting absolute time, whether computed from a duration or given explicitly in the
600 request message, is used to initialize the value of the TerminationTime resource property
601 of the Subscription resource.

602 If the NotificationProducer is unable or unwilling to set the TerminationTime resource
603 property of the Subscription resource to the requested time or a value greater, or if this
604 requested time is not in the future, then the NotificationProducer MUST return an
605 UnacceptableInitialTerminationTimeFault message.

606 The use of the xsi:nil attribute with value "true" indicates there is no scheduled
607 termination time requested for the Subscription, implying that the requested Subscription
608 has infinite duration.

609 If the element does not include the time zone designation, the value of the element
610 MUST be interpreted as universal time (UTC) time.

611 If this component is not included, the initial value of the TerminationTime resource
612 property is dependent on the implementation of the NotificationProducer.

613 /wsnt:SubscriptionPolicy

614 This optional component is an open component intended to be used in an application
615 specific way to specify policy related requirements/assertions associated with the
616 subscribe requests. This mechanism could be used to govern the message rate (e.g.
617 maximum 3 messages per second), reliability of the Notification delivery, etc. The
618 semantics of how the NotificationProducer MUST or MAY react to the policy
619 requirements and assertions appearing in this component are specific to the actual policy
620 grammar used.

621 If this component is not specified in the Subscribe request message, then the
622 NotificationProducer SHOULD use other means (such as directly contacting the
623 NotificationConsumer) to resolve any policy-related inquiries.

624 /wsnt:SubscriptionPolicy/wsnt:UseRaw

625 An element whose presence indicates that the NotificationProducer is to produce
626 Notifications without using the Notify wrapper. This component is optional, if missing the
627 default behavior is to use the Notify wrapper for all notification messages. This element
628 MUST NOT occur more than once in a Subscribe request message.

629 If this element is not present then the NotificationProducer MUST use the wsnt:Notify
630 message in Notifications produced for this subscription. In this case the
631 NotificationConsumer referred to in the wsnt:ConsumerReference element SHOULD
632 implement the Notify message and include a Notify operation in its portType definition.

633 If this element is present then the NotificationProducer SHOULD produce the raw
634 Notification itself.

635 The response to the Subscribe request message is a message of the following form:

```
636 ...  
637 <wsnt:SubscribeResponse>  
638   <wsnt:SubscriptionReference>  
639     wsa:EndpointReference  
640   </wsnt:SubscriptionReference>  
641   <wsnt:CurrentTime>xsd:dateTime</wsnt:CurrentTime> ?  
642   <wsnt:TerminationTime>xsd:dateTime</wsnt:TerminationTime> ?  
643   {any} *  
644 </wsnt:SubscribeResponse>
```

645

...

646 The WS-Addressing [action] Message Addressing Property MUST contain the URI
647 <http://docs.oasis-open.org/wsn/bw-1/NotificationProducer/SubscribeResponse>.

648 The contents of the SubscribeResponse message are further described as follows:

649 /wsnt:SubscriptionReference

650 A WS-Resource reference to the Subscription WS-Resource created as a result of the
651 Subscribe message.

652 /wsnt:CurrentTime

653 This OPTIONAL component SHOULD be returned if the SubscriptionManager uses
654 scheduled termination from WS-ResourceLifetime. The value of this component is the
655 value of the CurrentTime resource property of the Subscription at the time the response
656 message is created.

657 /wsnt:TerminationTime

658 This OPTIONAL component SHOULD be returned if the SubscriptionManager uses
659 scheduled termination from WS-ResourceLifetime. The value of this component is the
660 value of the TerminationTime resource property of the Subscription at the time the
661 response message is created.

662 If the NotificationProducer does not respond to the Subscribe request message with the
663 SubscribeResponse message, then it MUST send a fault. This specification defines the following
664 faults associated with failure to process the Subscribe request message:

665

666 ResourceUnknownFault

667 • The NotificationProducer is a WS-Resource, and the resource identified in the message
668 is not known to the Web service. This fault is specified by the WS-Resource [WS-
669 Resource] specification.

670 InvalidFilterFault

671 • The Subscribe message contained a filter that was not understood or supported by the
672 NotificationProducer.

673 TopicExpressionDialectUnknownFault

674 • The Subscribe message contained a TopicExpression filter having a dialect that was not
675 understood or supported by the NotificationProducer.

676 InvalidTopicExpressionFault

677 • The Subscribe message contained a TopicExpression filter where the contents of the
678 filter did not match the dialect specified.

679 TopicNotSupportedFault

680 • The Subscribe message contained a TopicExpression filter that referenced a topic that
681 was not supported by the NotificationProducer.

682 InvalidProducerPropertiesExpressionFault

683 • The Subscribe message contained a ProducerProperties filter that did not represent a
684 valid boolean expression.

685 InvalidMessageContentExpressionFault

- 686 • The Subscribe message contained a MessageContent filter that did not represent a valid
687 boolean expression.
- 688 InvalidUseRawValueFault
- 689 • The RawValue element appeared more than once in the SubscriptionPolicy.
- 690 UnacceptableInitialTerminationTimeFault
- 691 • The value of InitialTerminationTime specified in the Subscribe message was not
692 acceptable to the NotificationProducer. The NotificationProducer MAY include a hint in
693 the fault message indicating acceptable values for InitialTerminationTime.
- 694 SubscribeCreationFailedFault
- 695 • The NotificationProducer failed to process the Subscribe message. The
696 NotificationProducer SHOULD use a more specific fault message if possible. The
697 NotificationProducer MAY include a hint in the fault message indicating why it failed to
698 process the Subscribe message.

699 4.2.1 Example SOAP Encoding of the Subscribe Message Exchange

700 The following is a non-normative example of a Subscribe request message using SOAP:

```

701 <s:Envelope ... >
702   <s:Header>
703     <wsa:Action>
704       http://docs.oasis-open.org/wsn/bw-
705 1/NotificationProducer/SubscribeRequest
706     </wsa:Action>
707     ...
708   </s:Header>
709   <s:Body>
710     <wsnt:Subscribe>
711       <wsnt:ConsumerReference>
712         <wsa:Address>
713           http://www.consumer.org/ConsumerEndpoint
714         </wsa:Address>
715       </wsnt:ConsumerReference>
716       <wsnt:Filter>
717         <wsnt:TopicExpression Dialect=
718 "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
719           npex:SomeTopic
720         </wsnt:TopicExpression>
721         <wsnt:MessageContent
722           Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
723           boolean(ncex:Producer="15")
724         </wsnt:MessageContent>
725       </wsnt:Filter>
726       <wsnt:InitialTerminationTime>
727         2005-12-25T00:00:00.000000Z
728       </wsnt:InitialTerminationTime>
729     </wsnt:Subscribe>
730   </s:Body>

```

731

```
</s:Envelope>
```

732

733 The following is a non-normative example of a Subscribe response message using SOAP:

734

```
<s:Envelope ... >
```

735

```
<s:Header>
```

736

```
<wsa:Action>
```

737

```
http://docs.oasis-open.org/wsn/bw-
```

738

```
1/NotificationProducer/SubscribeResponse
```

739

```
</wsa:Action>
```

740

```
...
```

741

```
</s:Header>
```

742

```
<s:Body>
```

743

```
<wsnt:SubscribeResponse>
```

744

```
<wsnt:SubscriptionReference>
```

745

```
<wsa:Address>
```

746

```
http://www.producer.org/SubsMgrEndpoint
```

747

```
</wsa:Address>
```

748

```
</wsnt:SubscriptionReference>
```

749

```
</wsnt:SubscribeResponse>
```

750

```
</s:Body>
```

751

```
</s:Envelope>
```

752 4.3 GetCurrentMessage

753

In response to a GetCurrentMessage message, the NotificationProducer MAY return the last Notification published to a given Topic. This is a non-destructive read, allowing a newly-subscribed NotificationConsumer to get the last Notification that other NotificationConsumers have received.

757

In certain circumstances, a NotificationProducer MAY choose to not cache the last Notification to one or more Topics it supports. In such cases, the NotificationProducer MUST respond with a fault message indicating that no current message is available on that Topic. The NotificationProducer MAY choose to communicate its caching policy by some means not specified in this document, such as using a policy assertion.

762

The format of the GetCurrentMessage request message is:

763

```
...
```

764

```
<wsnt:GetCurrentMessage>
```

765

```
<wsnt:Topic Dialect="xsd:anyURI">
```

766

```
{any}
```

767

```
</wsnt:Topic>
```

768

```
{any} *
```

769

```
</wsnt:GetCurrentMessage>
```

770

```
...
```

771

The WS-Addressing [action] Message Addressing Property MUST contain the URI

772

```
http://docs.oasis-open.org/wsn/bw-1/NotificationProducer/GetCurrentMessageRequest.
```

773

The components of the GetCurrentMessage request message are further described as follows:

774 /wsnt:Topic
775 A TopicExpression that identifies exactly one Topic.

776 If the NotificationProducer successfully processes the GetCurrentMessage request, it MUST
777 respond with a GetCurrentMessageResponse message. This response has the following form:

```
778 ...  
779 <wsnt:GetCurrentMessageResponse>  
780   {any} *  
781 </wsnt:GetCurrentMessageResponse>  
782 ...
```

783 The WS-Addressing [action] Message Addressing Property MUST contain the URI
784 <http://docs.oasis-open.org/wsn/bw-1/NotificationProducer/GetCurrentMessageResponse>.

785 The contents of the GetCurrentMessage response message are further described as follows:

786 /wsnt:GetCurrentMessageResponse/{any}

787 Contains the last Notification associated with the Topic identified by the request
788 message.

789 If the NotificationProducer does not respond to the GetCurrentMessage request message with
790 the GetCurrentMessageResponse message, then it MUST send a fault. This specification defines
791 the following faults associated with failure to process the GetCurrentMessage request message:

792 ResourceUnknownFault

793 • The NotificationProducer is acting as a WS-Resource, and the resource identified in the
794 message (which follows the WS-Resource Access Pattern) is not known to the Web
795 service. This fault is specified by the WS-Resource [WS-Resource] specification.

796 TopicExpressionDialectUnknownFault

797 • The Topic element of the GetCurrentMessage message had a dialect that was not
798 understood or supported by the NotificationProducer.

799 InvalidTopicExpressionFault

800 • The Topic element of the GetCurrentMessage message had contents that did not match
801 the dialect specified.

802 TopicNotSupportedFault

803 • The Topic element of the GetCurrentMessage message referenced a topic that was not
804 supported by the NotificationProducer.

805 MultipleTopicsSpecifiedFault

806 • The GetCurrentMessage message referenced more than one topic.

807 NoCurrentMessageOnTopicFault

808 • The topic referenced in the GetCurrentMessage message does not have any pending
809 messages.

810 **4.3.1 Example SOAP Encoding of the GetCurrentMessage Message** 811 **Exchange**

812 The following is a non-normative example of a GetCurrentMessage request message using
813 SOAP:

```

814 <s:Envelope ... >
815   <s:Header>
816     <wsa:Action>
817       http://docs.oasis-open.org/wsn/bw-
818 1/NotificationProducer/GetCurrentMessageRequest
819     </wsa:Action>
820     ...
821   </s:Header>
822   <s:Body>
823     <wsnt:GetCurrentMessage>
824       <wsnt:Topic Dialect=
825         "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
826         npex:SomeTopic
827       </wsnt:Topic>
828     </wsnt:GetCurrentMessage>
829   </s:Body>
830 </s:Envelope>

```

831 The following is a non-normative example of a GetCurrentMessage response message using
832 SOAP:

```

833 <s:Envelope ... >
834   <s:Header>
835     <wsa:Action>
836       http://docs.oasis-open.org/wsn/bw-
837 1/NotificationProducer/GetCurrentMessageResponse
838     </wsa:Action>
839     ...
840   </s:Header>
841   <s:Body>
842     <wsnt:GetCurrentMessageResponse>
843       <npex:NotifyContent>exampleNotifyContent</npex:NotifyContent>
844     </wsnt:GetCurrentMessageResponse>
845   </s:Body>
846 </s:Envelope>

```

847 **5 Pull-Style Notification**

848 There are certain circumstances in which the basic “push-style” of NotificationMessage delivery is
849 not appropriate. For example, certain NotificationConsumers are behind a firewall such that the
850 NotificationProducer cannot initiate a message exchange to send the Notification. A similar
851 circumstance exists for NotificationConsumers that are unable or unwilling to provide an endpoint
852 to which the NotificationProducer can send Notification Messages. In other situations, the
853 NotificationConsumer prefers to control the timing of receipt of Notification Messages, instead of
854 receiving Notification Messages at unpredictable intervals, it may prefer to “pull” “retrieve” the
855 Notification Messages at a time of its own choosing.

856 For these and other reasons, WS-BaseNotification defines a pair of portTypes: a PullPoint
857 interface, defining an endpoint that accumulates Notification Messages and allows a requestor to
858 retrieve accumulated Notification Messages and a CreatePullPoint interface that acts as a factory
859 for PullPoint resources. The intended pattern of use is that a Subscriber or other party creates a
860 PullPoint through the factory interface, and then uses it as the ConsumerReference in one or
861 more Subscribe requests. The actual consumer then pulls Notifications from the PullPoint.

862 **5.1 PullPoint Interface**

863 In support of the “pull-style” of Notification Message delivery, this specification describes a
864 PullPoint resource by defining a PullPoint interface. The PullPoint interface provides the means
865 by which NotificationProducers can insert Notification Messages into the PullPoint, by which
866 requestors can retrieve messages from the PullPoint and the means by which requestors can
867 destroy the PullPoint. Note: the PullPoint MAY be a WS-Resource, and if it is, request messages
868 defined in this specification MUST follow the WS-Resource Access Pattern defined by [WS-
869 Resource] and the PullPoint WS-Resource MUST support the immediate termination interface
870 defined by WS-RF Resource Lifetime and it MAY support the scheduled termination interface
871 defined by WS-RF Resource Lifetime.

872 **5.1.1 Accumulating Notification Messages**

873 The PullPoint interface supports the NotificationConsumer interface (as defined in section 3). This
874 interface allows NotificationProducers to send Notification Messages to the PullPoint using either
875 a “raw” or a “wrapped” approach. Notification Messages received by the PullPoint through its
876 NotificationConsumer interface are accumulated by the PullPoint on a best effort basis. If the
877 PullPoint is no longer capable of accumulating additional Notification Messages, it MAY ignore
878 Notification Messages sent to its NotificationConsumer interface, or it MAY dispose of any
879 previously accumulated Notification Messages and continue to accumulate incoming Notification
880 Messages. The PullPoint interface does not define additional constraints on its use of the
881 NotificationConsumer interface.

882 **5.1.2 GetMessages**

883 The PullPoint interface provides a message exchange to allow requestors to retrieve (or pull)
884 Notification Messages from the PullPoint.

885 If a requestor wishes to retrieve Notification Messages accumulated by the PullPoint, it sends a
886 GetMessages request to the PullPoint endpoint. The format of the GetMessages request
887 message is:

888
889
890
891
892
893

```
<wsnt:GetMessages>  
  <wsnt:MaximumNumber>xsd:nonNegativeInteger</wsnt:MaximumNumber> ?  
  {any} *  
</wsnt:GetMessages>
```

894 The WS-Addressing [action] Message Addressing Property MUST contain the URI

895 <http://docs.oasis-open.org/wsn/bw-1/PullPoint/GetMessagesRequest>.

896 The components of the GetMessages request message are further described as follows:

897 /wsnt:GetMessages

898 This component requests that messages held by the PullPoint be returned. A response is
899 generated without waiting for messages to be accumulated by the PullPoint.

900 /wsnt:GetMessages/wsnt:MaximumNumber

901 This optional non-negative integer indicates the maximum number of accumulated
902 Notification Messages to be returned in the response message. If the total number of
903 Notification Messages accumulated by the PullPoint is less than or equal to this number,
904 all of the Notification Messages are returned in the response message. If the total
905 number of accumulated Notification Messages exceeds this value, the PullPoint MUST
906 return exactly this number of accumulated Notification Messages; the implementation of
907 the PullPoint is free to determine which subset of messages are returned in the response
908 message in this case. If the MaximumNumber component is not present in the request
909 message, the requestor is asking to receive all messages held by the PullPoint.

910 The response of the GetMessages request message is a message of the following form:

911
912
913
914
915
916

```
<wsnt:GetMessagesResponse>  
  <wsnt:NotificationMessage /> *  
  {any} *  
</wsnt:GetMessagesResponse>
```

917 The WS-Addressing [action] Message Addressing Property MUST contain the URI

918 <http://docs.oasis-open.org/wsn/bw-1/PullPoint/GetMessagesResponse>.

919 The contents of the GetMessages response message are further described as follows:

920 /wsnt:GetMessagesResponse

921 This component contains zero or more accumulated Notification Messages. The number
922 of messages appearing is limited by the wsnt:MaximumNumber component of the
923 GetMessages request message. Each accumulated Notification Message appearing in a
924 GetMessagesResponse is "removed" from the PullPoint and MUST not appear in the
925 response message of subsequent GetMessages requests.

926 /wsnt:GetMessagesResponse/wsnt:NotificationMessage

927 The content of this component is an accumulated NotificationMessage. The
928 NotificationMessage component is described as part of the Notify message (section 3.2)
929 the GetMessagesResponse message does not define additional constraints on the
930 NotificationMessage component. If the accumulated NotificationMessage was added to

931 the PullPoint through a Notify message, the contents of the NotificationMessage is
932 exactly the contents of the NotificationMessage component of the Notify message. If the
933 accumulated NotificationMessage was added to the PullPoint through a “raw” message,
934 then the contents of the NotificationMessage is formed by inserting the body of the “raw”
935 message into the wsnt:Message component of the
936 /wsnt:GetMessagesResponse/wsnt:NotificationMessage.

937 If the PullPoint does not respond to the GetMessages request message with the
938 GetMessagesResponse message, then it MUST send a fault. This specification defines the
939 following faults associated with failure to process the GetMessages request message:

940 ResourceUnknownFault

- 941 • The resource identified in the message (which follows the WS-Resource Access Pattern)
942 is not known to the Web service. This fault is specified by the WS-Resource [WS-
943 Resource] specification.

944 **5.1.3 Example SOAP Encoding of the GetMessages Message** 945 **Exchange**

946 The following is a non-normative example of a GetMessages request message using SOAP:

```
947 <s:Envelope ... >  
948   <s:Header>  
949     <wsa:Action>  
950       http://docs.oasis-open.org/wsn/bw-1/PullPoint/GetMessagesRequest  
951     </wsa:Action>  
952     ...  
953   </s:Header>  
954   <s:Body>  
955     <wsnt:GetMessages>  
956       <wsnt:MaximumNumber>10</wsnt:MaximumNumber>  
957     </wsnt:GetMessages>  
958   </s:Body>  
959 </s:Envelope>
```

960 The following is a non-normative example of a GetMessages response message using SOAP:

```
961 <s:Envelope ... >  
962   <s:Header>  
963     <wsa:Action>  
964       http://docs.oasis-open.org/wsn/bw-1/PullPoint/GetMessagesResponse  
965     </wsa:Action>  
966     ...  
967   </s:Header>  
968   <s:Body>  
969     <wsnt:GetMessagesResponse>  
970       <wsnt:NotificationMessage>  
971         ...  
972       </wsnt:NotificationMessage>  
973       <wsnt:NotificationMessage>  
974         ...
```

```
975     </wsnt:NotificationMessage>
976     </wsnt:GetMessagesResponse>
977     </s:Body>
978 </s:Envelope>
```

979 **5.1.4 Destroy**

980 The PullPoint interface provides a destroy operation, providing a means by which a requestor can
981 terminate the PullPoint resource. To terminate a PullPoint resource, a requestor **MUST** send a
982 Destroy request message to the PullPoint. The Destroy request message has the following form:

```
983
984 <wsnt:Destroy>
985   {any} *
986 </wsnt:Destroy>
987
```

988 The WS-Addressing [action] Message Addressing Property **MUST** contain the URI

989 <http://docs.oasis-open.org/wsn/bw-1/PullPoint/DestroyRequest>.

990 Upon receipt of the Destroy request, the PullPoint **MUST** attempt to destroy itself. If the Destroy
991 request message is successfully processed, the PullPoint **MUST** respond with the following
992 message:

```
993
994 <wsnt:DestroyResponse>
995   {any} *
996 </wsnt:DestroyResponse>
997
```

998 The WS-Addressing [action] Message Addressing Property **MUST** contain the URI

999 <http://docs.oasis-open.org/wsn/bw-1/PullPoint/DestroyResponse>.

1000 If the PullPoint does not respond to the Destroy request message with the DestroyResponse
1001 message, then it **MUST** send a fault. This specification defines the following faults associated
1002 with failure to process the Destroy request message:

1003 UnableToDestroyPullPointFault.

- 1004 • The PullPoint was unable to destroy the PullPoint resource for some reason..

1005 **5.1.5 Example SOAP Encoding of the Destroy Message Exchange**

1006 The following is a non-normative example of a Destroy request message using SOAP:

```
1007 <s:Envelope ... >
1008   <s:Header>
1009     <wsa:Action>
1010       http://docs.oasis-open.org/wsn/bw-1/PullPoint/DestroyRequest
1011     </wsa:Action>
1012     ...
1013   </s:Header>
1014   <s:Body>
```

```
1015     <wsnt:Destroy/>
1016   </s:Body>
1017 </s:Envelope>
```

1018 The following is a non-normative example of a Destroy response message using SOAP:

```
1019 <s:Envelope ... >
1020   <s:Header>
1021     <wsa:Action>
1022       http://docs.oasis-open.org/wsn/bw-1/PullPoint/DestroyResponse
1023     </wsa:Action>
1024     ...
1025   </s:Header>
1026   <s:Body>
1027     <wsnt:DestroyResponse/>
1028   </s:Body>
1029 </s:Envelope>
```

1030 5.2 Create PullPoint Interface

1031 This specification describes a CreatePullPoint interface standardizing the means by which a
1032 PullPoint resource is created.

1033 If a requestor wishes to create a new PullPoint resource, it MUST send a CreatePullPoint request
1034 to an endpoint supporting the PullPoint interface. The CreatePullPoint request message has the
1035 following form:

```
1036 <wsnt:CreatePullPoint>
1037   {any} *
1038 </wsnt:CreatePullPoint>
```

1041 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1042 <http://docs.oasis-open.org/wsn/bw-1/CreatePullPoint/CreatePullPointRequest>

1043 Upon receipt of the CreatePullPoint request, the receiver MUST attempt to create a new PullPoint
1044 resource. If the CreatePullPoint request message is successfully processed, the receiver MUST
1045 respond with the following message:

```
1046 <wsnt:CreatePullPointResponse>
1047   <wsnt:PullPoint>wsa:EndpointReferenceType</wsnt:PullPoint>
1048   {any} *
1049 </wsnt:CreatePullPointResponse>
```

1052 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1053 <http://docs.oasis-open.org/wsn/bw-1/CreatePullPoint/CreatePullPointResponse>.

1054 The contents of the CreatePullPoint response message are further described as follows:

1055 /wsnt:CreatePullPointResponse/wsnt:PullPoint

1056 This component is an EndpointReference, as defined by WS-Addressing, that is a
1057 reference to the PullPoint resource created during the processing of the CreatePullPoint
1058 request message.

1059 If the receiver does not respond to the CreatePullPoint request message with the
1060 CreatePullPointResponse message, then it MUST send a fault. This specification defines the
1061 following faults associated with failure to process the Destroy request message:

- 1062 • UnableToCreatePullPointFault.

1063

1064 **5.2.1 Example SOAP Encoding of the CreatePullPoint Message** 1065 **Exchange**

1066 The following is a non-normative example of a CreatePullPoint request message using SOAP:

```
1067 <s:Envelope ... >  
1068   <s:Header>  
1069     <wsa:Action>  
1070     http://docs.oasis-open.org/wsn/bw-1/PullPoint/CreatePullPointRequest  
1071     </wsa:Action>  
1072     ...  
1073   </s:Header>  
1074   <s:Body>  
1075     <wsnt:CreatePullPoint/>  
1076   </s:Body>  
1077 </s:Envelope>
```

1078 The following is a non-normative example of a CreatePullPoint response message using SOAP:

```
1079 <s:Envelope ... >  
1080   <s:Header>  
1081     <wsa:Action>  
1082     http://docs.oasis-open.org/wsn/bw-1/PullPoint/CreatePullPointResponse  
1083     </wsa:Action>  
1084     ...  
1085   </s:Header>  
1086   <s:Body>  
1087     <wsnt:CreatePullPointResponse>  
1088       <wsnt:PullPoint>  
1089         <wsa:Address>...</wsa:Address>  
1090         ...  
1091       </wsnt:PullPoint>  
1092     </wsnt:CreatePullPointResponse>  
1093   </s:Body>  
1094 </s:Envelope>
```

1095 **6 SubscriptionManager Interface**

1096 The SubscriptionManager interface defines message exchanges to manipulate Subscription
1097 resources. There are two styles of SubscriptionManager interface: base and pausable. All
1098 SubscriptionManagers MUST implement the message exchanges described in the Base
1099 Subscription Manager section.

1100 For those environments that choose to implement a Subscription as a WS-Resource, the
1101 SubscriptionManager MUST implement the resource properties and message exchanges defined
1102 in Subscriptions as WS-Resources section. Both the Base Subscription Manager and the
1103 Pausable Subscription Manager MAY expose Subscriptions as WS-Resources.

1104 **6.1 Base Subscription Manager**

1105 The basic behavior of a SubscriptionManager is to renew the duration of a Subscription resource
1106 and terminate a Subscription.

1107 **6.1.1 Renew**

1108 To modify the current lifetime of a Subscription, a requestor sends a Renew request message to
1109 the SubscriptionManager. The Renew request message MUST have the following form:

```
1110 ...  
1111 <wsnt:Renew>  
1112   <wsnt:TerminationTime>  
1113     [xsd:dateTime | xsd:duration]  
1114   </wsnt:TerminationTime>  
1115   {any}  
1116 </wsnt:Renew>  
1117 ...
```

1118 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1119 <http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/RenewRequest>.

1120 The components of the Renew request message are further described as follows:

1121 /wsnt:TerminationTime

1122 This component contains the service requestor's suggestion for change in expiration, or
1123 termination time of the Subscription resource. There are two forms of this component,
1124 absolute time and duration. If the type of this component is xsd:dateTime, the value of the
1125 component is to be interpreted as an "absolute time". If the type of this component is
1126 xsd:duration, the value of the component is to be interpreted as a "duration".

1127 The duration form is used to "compute" the "absolute time" form in the following fashion.
1128 The value of this component in "absolute time" form is computed by adding the
1129 xsd:duration value to the current time value of the SubscriptionManager.

1130 The "absolute time" form (whether computed from a duration, or contained within the
1131 request message) is used to reset the value of the expiration or termination time
1132 component of the Subscription resource.

1133 If the SubscriptionManager is unable or unwilling to reset the termination time component
1134 of the Subscription resource to the given value of the "absolute time" form or a value

1135 greater, then the Renew request MUST fault. If the value is not “in the future” relative to
1136 the current time as known by the SubscriptionManager, the Renew request MUST fault.

1137 The use of the xsi:nil attribute with value “true” indicates there is no scheduled
1138 termination time requested for the Subscription, implying that the Subscription should be
1139 renewed to have an infinite duration. If the element does not include the time zone
1140 designation, the value of the element MUST be interpreted as universal time (UTC) time.

1141 If the SubscriptionManager successfully processes the Renew request message, it MUST
1142 respond with a RenewResponse message. This response has the following form:

1143
1144
1145
1146
1147
1148
1149

```
...  
<wsnt:RenewResponse>  
  <wsnt:TerminationTime>xsd:dateTime</wsnt:TerminationTime>  
  <wsnt:CurrentTime>xsd:dateTime</wsnt:CurrentTime>?  
  {any}  
</wsnt:RenewResponse>  
...
```

1150 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1151 <http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/RenewResponse>.

1152 The contents of the RenewResponse message are further described as follows:

1153 /wsnt:TerminationTime

1154 The value of this component is the value of the termination time component of the
1155 Subscription at the time the response message is created.

1156 /wsnt:CurrentTime

1157 This OPTIONAL component contains the value of the CurrentTime according to the
1158 SubscriptionManager at the time the response message is created.

1159 If the SubscriptionManager does not respond to the Renew request message with the
1160 RenewResponse message, then it MUST send a fault. This specification defines the following
1161 faults associated with failure to process the Renew request message:

1162 ResourceUnknownFault

- 1163 • The SubscriptionManager is acting as a WS-Resource, and the resource identified in the
1164 message (which follows the WS-Resource Access Pattern) is not known to the Web
1165 service. This fault is specified by the WS-Resource [WS-Resource] specification.

1166 UnacceptableTerminationTimeFault

- 1167 • The value of InitialTerminationTime specified in the Subscribe message was not
1168 acceptable to the NotificationProducer. The NotificationProducer MAY include a hint in
1169 the fault message indicating acceptable values for InitialTerminationTime.

1170 **6.1.1.1 Example SOAP Encoding of the Renew Message** 1171 **Exchange**

1172 The following is a non-normative example of a Renew request message using SOAP:

1173
1174
1175

```
<s:Envelope ... >  
  <s:Header>  
    <wsa:Action>
```

```
1176 http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/RenewRequest
1177 </wsa:Action>
1178 ...
1179 </s:Header>
1180 <s:Body>
1181 <wsnt:Renew>
1182 <wsnt:TerminationTime>P1D</wsnt:TerminationTime>
1183 </wsnt:Renew>
1184 </s:Body>
1185 </s:Envelope>
```

1186 The following is a non-normative example of a Renew response message using SOAP:

```
1187 <s:Envelope ... >
1188 <s:Header>
1189 <wsa:Action>
1190 http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/RenewResponse
1191 </wsa:Action>
1192 ...
1193 </s:Header>
1194 <s:Body>
1195 <wsnt:RenewResponse>
1196 <wsnt:TerminationTime>
1197 2005-12-26T00:00:00.000000Z
1198 </wsnt:TerminationTime>
1199 </wsnt:RenewResponse>
1200 </s:Body>
1201 </s:Envelope>
```

1202 6.1.2 Unsubscribe

1203 To terminate a Subscription, a requestor sends an Unsubscribe request message to the
1204 SubscriptionManager. The Unsubscribe request message MUST have the following form:

```
1205 ...
1206 <wsnt:Unsubscribe>
1207 {any}
1208 </wsnt:Unsubscribe>
1209 ...
```

1210 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1211 <http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/UnsubscribeRequest>.

1212 Upon receipt of the Unsubscribe request, the SubscriptionManager MUST attempt to destroy the
1213 Subscription resource. If the SubscriptionManager successfully processes the Unsubscribe
1214 request message, it MUST respond with the following message:

```
1215 ...
1216 <wsnt:UnsubscribeResponse>
1217 {any}
1218 </wsnt:UnsubscribeResponse>
```


1219

...

1220 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1221 <http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/UnsubscribeResponse>.

1222 If the SubscriptionManager does not respond to the Unsubscribe request message with the
1223 UnsubscribeResponse message, then it MUST send a fault. This specification defines the
1224 following faults associated with failure to process the Unsubscribe request message:

1225 ResourceUnknownFault

1226 • The SubscriptionManager is acting as a WS-Resource, and the resource identified in the
1227 message (which follows the WS-Resource Access Pattern) is not known to the Web
1228 service. This fault is specified by the WS-Resource [WS-Resource] specification.

1229 UnableToDestroySubscriptionFault

1230 • The SubscriptionManager was unable to destroy the Subscription resource for some
1231 reason.

1232 **6.1.2.1 Example SOAP Encoding of the Unsubscribe Message** 1233 **Exchange**

1234 The following is a non-normative example of an Unsubscribe request message using SOAP:

```
1235 <s:Envelope ... >  
1236   <s:Header>  
1237     <wsa:Action>  
1238     http://docs.oasis-open.org/wsn/bw-  
1239     1/SubscriptionManager/UnsubscribeRequest  
1240     </wsa:Action>  
1241     ...  
1242   </s:Header>  
1243   <s:Body>  
1244     <wsnt:Unsubscribe />  
1245   </s:Body>  
1246 </s:Envelope>
```

1247 The following is a non-normative example of an Unsubscribe response message using SOAP:

```
1248 <s:Envelope ... >  
1249   <s:Header>  
1250     <wsa:Action>  
1251     http://docs.oasis-open.org/wsn/bw-  
1252     1/SubscriptionManager/UnsubscribeResponse  
1253     </wsa:Action>  
1254     ...  
1255   </s:Header>  
1256   <s:Body>  
1257     <wsnt:UnsubscribeResponse />  
1258   </s:Body>  
1259 </s:Envelope>
```

1260 6.2 Pausable Subscription Manager

1261 A Pausable Subscription Manager implements all the message exchanges defined in the Base
1262 Subscription Manager and augments this with a set of advanced capabilities to allow third parties
1263 to pause and resume subscriptions.

1264 6.2.1 PauseSubscription

1265 To temporarily suspend the production of Notifications on the given Subscription, a requestor
1266 MAY send a PauseSubscription request message to the SubscriptionManager. The
1267 PauseSubscription request message has the following form:

```
1268 ...  
1269 <wsnt:PauseSubscription>{any}</wsnt:PauseSubscription>  
1270 ...
```

1271 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1272 <http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/PauseSubscriptionRequest>.

1273 Upon successful processing of this message the Subscription is in the *paused* state. Production
1274 of Notifications can be resumed by sending the ResumeSubscription request message (see 6.3).

1275 In an asynchronous implementation, a NotificationProducer may continue producing notifications
1276 while the PauseSubscription request is in transit. In general, there is no way of knowing exactly
1277 when the pause will take effect.

1278 Note: transitioning between the paused state and the resumed state has no effect on the
1279 TerminationTime of the Subscription resource. A Subscription's TerminationTime may elapse
1280 while it is in a paused state, initiating the scheduled destruction of the Subscription resource.

1281 When a Subscription is resumed after having been paused, the NotificationProducer MAY choose
1282 from one of the courses of action listed below, or any other action. NotificationProducers MAY
1283 advertise their behavior in this situation via policy assertions. In the absence of a specific policy
1284 assertion, Subscribers SHOULD NOT assume any particular behavior on the part of the
1285 NotificationProducer. Possible actions include

- 1286 1. Produce Notifications for all the Situations related to the Subscription that occurred while
1287 the Subscription was paused (as well as for any new Situations that occur after the
1288 Subscription has been resumed).
- 1289 2. Produce a Notification for the last Situation that occurred while the Subscription was
1290 paused (as well as for any new situations that occur after the topic has been resumed).
- 1291 3. Produce no Notifications until a Situation occurs after the Subscription has been
1292 resumed.

1293 If the SubscriptionManager accepts the PauseSubscription request message, it MUST respond
1294 with a message of the following form:

```
1295 ...  
1296 <wsnt:PauseSubscriptionResponse>{any}</wsnt:PauseSubscriptionResponse>  
1297 ...
```

1298 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1299 <http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/PauseSubscriptionResponse>.

1300 If the SubscriptionManager does not respond to the PauseSubscription request message with the
1301 PauseSubscription response message, then it MUST send a fault. This specification defines the
1302 following faults associated with failure to process the PauseSubscription request message:

1303 ResourceUnknownFault

- 1304 • The resource identified in the message is not known to the Web service. This fault is
1305 specified by the WS-Resource [WS-Resource] specification.

1306 PauseFailedFault

- 1307 • The Pause operation could not be performed on the Subscription.

1308 6.2.2 Example SOAP Encoding of the PauseSubscription Message 1309 Exchange

1310 The following is a non-normative example of a PauseSubscription request message using SOAP:

```
1311 <s:Envelope ... >  
1312   <s:Header>  
1313     <wsa:Action>  
1314       http://docs.oasis-open.org/wsn/bw-  
1315 1/SubscriptionManager/PauseSubscriptionRequest  
1316     </wsa:Action>  
1317     ...  
1318   </s:Header>  
1319   <s:Body>  
1320     <wsnt:PauseSubscription/>  
1321   </s:Body>  
1322 </s:Envelope>
```

1323 The following is a non-normative example of a PauseSubscription response message using
1324 SOAP:

```
1325 <s:Envelope ... >  
1326   <s:Header>  
1327     <wsa:Action>  
1328       http://docs.oasis-open.org/wsn/bw-  
1329 1/SubscriptionManager/PauseSubscriptionResponse  
1330     </wsa:Action>  
1331     ...  
1332   </s:Header>  
1333   <s:Body>  
1334     <wsnt:PauseSubscriptionResponse/>  
1335   </s:Body>  
1336 </s:Envelope>
```

1337 6.3 ResumeSubscription

1338 If a requestor wishes to resume the production of Notifications on the given Subscription, it must
1339 send a ResumeSubscription request message. The ResumeSubscription request message has
1340 the following form:

1341 ...
1342 <wsnt:ResumeSubscription>
1343 {any}
1344 </wsnt:ResumeSubscription>
1345 ...

1346 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1347 <http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/ResumeSubscriptionRequest>.

1348 Upon successful processing of this message the Subscription is no longer in the *paused* state.
1349 If a Subscription is not in the *paused* state, and it is the subject of a ResumeSubscription request
1350 message, no change occurs.

1351 If the SubscriptionManager accepts the ResumeSubscription request message, it MUST respond
1352 with a message of the following form:

1353 ...
1354 <wsnt:ResumeSubscriptionResponse>
1355 {any}
1356 </wsnt:ResumeSubscriptionResponse>
1357 ...

1358 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1359 [http://docs.oasis-open.org/wsn/bw-](http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/ResumeSubscriptionResponse)
1360 [1/SubscriptionManager/ResumeSubscriptionResponse](http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/ResumeSubscriptionResponse).

1361 If the SubscriptionManager does not respond to the ResumeSubscriptionMessage request
1362 message with the ResumeSubscriptionResponse message, then it MUST send a fault. This
1363 specification defines the following faults associated with failure to process the
1364 ResumeSubscriptionMessage request message:

1365 ResourceUnknownFault

- 1366 • The resource identified in the message is not known to the Web service. This fault is
1367 specified by the WS-Resource [WS-Resource] specification.

1368 ResumeFailedFault

- 1369 • The Resume operation could not be performed on the Subscription.

1370 **6.3.1 Example SOAP Encoding of the ResumeSubscription Message** 1371 **Exchange**

1372 The following is a non-normative example of a ResumeSubscription request message using
1373 SOAP:

1374 <s:Envelope ... >
1375 <s:Header>
1376 <wsa:Action>
1377 [http://docs.oasis-open.org/wsn/bw-](http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/ResumeSubscriptionRequest)
1378 [1/SubscriptionManager/ResumeSubscriptionRequest](http://docs.oasis-open.org/wsn/bw-1/SubscriptionManager/ResumeSubscriptionRequest)
1379 </wsa:Action>
1380 ...
1381 </s:Header>
1382 <s:Body>

```
1383 <wsnt:ResumeSubscription/>
1384 </s:Body>
1385 </s:Envelope>
```

1386 The following is a non-normative example of a ResumeSubscription response message using
1387 SOAP:

```
1388 <s:Envelope ... >
1389 <s:Header>
1390 <wsa:Action>
1391 http://docs.oasis-open.org/wsn/bw-
1392 1/SubscriptionManager/ResumeSubscriptionResponse
1393 </wsa:Action>
1394 ...
1395 </s:Header>
1396 <s:Body>
1397 <wsnt:ResumeSubscriptionResponse/>
1398 </s:Body>
1399 </s:Envelope>
```

1400 **6.4 Subscriptions as WS-Resources**

1401 Either Base Subscription Managers or Pausable Subscription Managers MAY expose
1402 subscriptions as WS-Resources. A Subscription Manager that exposes Subscriptions as WS-
1403 Resources MUST support the required message exchanges associated with the WS-
1404 ResourceProperties specification [WS-ResourceProperties] and MAY support the optional
1405 message exchanges defined by WS-ResourceProperties.

1406 It is RECOMMENDED that Subscription Managers that expose Subscriptions as WS-Resources
1407 support the message exchanges and resource properties defined for both forms of resource
1408 lifetime (immediate and scheduled destruction) by WS-ResourceLifetime [WS-ResourceLifetime].
1409 These message exchanges MAY be used to define the means by which Subscription resources
1410 can be explicitly destroyed, or destroyed using a scheduled (time-based) mechanism. Such
1411 Subscription Managers MAY support other means of destroying subscriptions.

1412 Subscription Managers MUST (for example in its WSDL definition) declare to requestors which
1413 means of Subscription destruction is supported.

1414 **6.4.1 Subscription Resource Properties**

1415 In cases where subscriptions are exposed as WS-Resources, this specification defines several
1416 resource properties, which MUST conform to the following schema fragment for content:

```
1417 ...
1418 targetNamespace="http://docs.oasis-open.org/wsn/b-1"
1419 ...
1420
1421 <xsd:element name="ConsumerReference"
1422 type="wsa:EndpointReferenceType" />
1423 <xsd:element name="Filter"
1424 type="wsnt:FilterType" />
```

1425
1426

```
<xsd:element name="CreationTime"
  type="xsd:dateTime" />
```

1427 These properties must also conform to the following schema fragment for cardinality:

1428
1429
1430
1431
1432
1433
1434
1435

```
<xsd:element ref="wsnt:ConsumerReference"
  minOccurs="1" maxOccurs="1" />
<xsd:element ref="wsnt:Filter"
  minOccurs="0" maxOccurs="1" />
<xsd:element ref="wsnt:SubscriptionPolicy"
  minOccurs="0" maxOccurs="1" />
<xsd:element ref="wsnt:CreationTime"
  minOccurs="0" maxOccurs="1" />
```

1436 These Resource Property elements are further described as follows:

1437 /wsnt:ConsumerReference and /wsnt:Filter and /wsnt:SubscriptionPolicy

1438 These elements are as defined in the description of the Subscribe request message (see
1439 4.2). SubscriptionManagers MAY prohibit any combination of these properties from being
1440 set. If setting a given property is disallowed, the SubscriptionManager MUST fault on
1441 receiving a request to change it. SubscriptionManagers SHOULD describe using
1442 metadata techniques defined by WS-ResourceFramework which properties may be set.

1443 If any of these properties is settable, it is implementation dependent whether production
1444 of notifications will be attempted without interruption when a change is made. The
1445 SubscriptionManager may choose to treat the change in properties as equivalent to a
1446 cancellation followed by the creation of a new subscription. In this case, messages may
1447 potentially be lost in the interim.

1448 A SubscriptionManager MAY choose to provide a stronger guarantee of continuity of
1449 notification production, and if so it SHOULD advertise this capability, for example in a
1450 policy assertion. In the absence of any such policy assertion, a subscriber SHOULD
1451 NOT make any assumption of continuity.

1452 /wsnt:CreationTime

1453 Indicates the date and time at which the Subscription was created. This is an optional
1454 component, supporting resource constrained devices which cannot associate a creation
1455 time with subscription resources they create. This property is read-only.
1456 SubscriptionManagers MUST fault on receiving a request to change it.

1457 If the SubscriptionManager also accepts the SetResourceProperties request message as defined
1458 in WS-ResourceProperties, the following properties MAY be modified by the requestor:

1459 /wsnt:ConsumerReference and /wsnt:Filter and /wsnt:SubscriptionPolicy

1460 **7 Security Considerations**

1461 This section deals with the security aspects of WS-BaseNotification. It deals with (a) securing the
1462 message exchanges defined in this specification, and (b) authorization and denial of service
1463 considerations specific to the Notification pattern.

1464 **7.1 Securing the Message Exchanges**

1465 In the Notification pattern, Notifications are sent to a NotificationConsumer, Subscribers
1466 exchange SubscriptionRequest and SubscriptionResponse messages with NotificationProducers,
1467 and any party with access to a SubscriptionManager endpoint may perform operations on the
1468 underlying Subscription resources. In cases where this communication must be secured it is
1469 RECOMMENDED that this be done using the mechanisms described in WS-Security.

1470 Communication between a NotificationProducer and NotificationConsumer will typically comprise
1471 a number of Notification messages. In cases where this communication must be secure, it is
1472 RECOMMENDED that a security context be established, allowing for potentially more efficient
1473 means of authentication. Note that the keys used to secure this channel may differ from any keys
1474 used in the Subscribe request/response operation that created the associated Subscription.

1475 **7.2 Securing Subscriptions and Notifications**

1476 As WS-BaseNotification provides mechanisms for subscribing to topics, security policies should
1477 be established such that

- 1478 1. only authorized principals can subscribe to receive Notifications
- 1479 2. only authorized principals can modify or delete Subscriptions

1480 It is recommended that the authorization policies be specified at the granularity of the Topic, if
1481 Topics are supported. It should be noted that even though Subscriptions may be done by
1482 authorized principals, the Notifications may be produced for NotificationConsumers whose
1483 identity may be different from the Subscriber. Message protection policies as outlined in the
1484 previous section can be used to ensure that sensitive Notifications are not delivered to malicious
1485 endpoints. For example, a key may need to be specified or generated during the process of
1486 Subscription, so that the Notifications can be encrypted using the key to ensure confidentiality of
1487 the messages. The mechanism by which the key is specified is governed by the Subscription
1488 policy.

1489 Given that WS-BaseNotification may use WS-ResourceProperties and WS-ResourceLifetime, the
1490 security considerations outlined in WS-those specifications need to be taken into account where
1491 appropriate. Authorization policies for those Resource Properties should be put in place so that
1492 the implications of providing the state information (through GetResourceProperty request
1493 messages) or through notification of state change and modification of the resource properties
1494 (through SetResourceProperty request messages), are taken into account.

1495 This specification provides a mechanism by which Subscribers can specify a subscription policy.
1496 Such a policy may contain security policy about protecting the message exchanges resulting from
1497 the Subscription. Security policy for Subscription message exchanges needs to take this into
1498 consideration so that the Subscription policies are protected. Also, given this policy may be
1499 contained in the resource properties of the subscription maintained by the SubscriptionManager,
1500 the resource properties must be appropriately secured.

1501 In addition to the usual concerns of authorization and message integrity which apply to all web
1502 services, notification presents issues all its own due to the third-party nature of subscription.
1503 Since the NotificationProducer is agreeing to produce Notifications for a consumer based on the
1504 requests of a Subscriber, it must assure itself that there is no harm in producing these
1505 Notifications. A malicious Subscriber may request Notifications be sent to a party that is not
1506 authorized to receive them. It may also mount DOS attacks by requesting large volumes of
1507 Notifications be sent to parties that cannot handle them.

1508
1509 The NotificationProducer may address these risks in many different ways, including but not
1510 limited to:

- 1511 • Simply trusting all Subscribers, perhaps because all parties are known to be on a closed,
1512 trusted, network, or because the consequences of unauthorized Subscriptions are
1513 otherwise known to be negligible.
- 1514 • Requiring all Subscribers to provide secure credentials proving that they are trusted to
1515 make subscriptions.
- 1516 • Refusing to produce notifications for NotificationConsumers that are not known to be
1517 authorized.
- 1518 • Explicitly confirming with NotificationConsumers whether they wish to receive the
1519 Notifications that the Subscriber has requested.
- 1520 • Some combination of the above, depending on the identity of the Subscriber and
1521 NotificationProducer

1522 NotificationProducers SHOULD advertise, whether through policy assertions or other means,
1523 what security measures they take.

1524 8 References

1525 8.1 Normative

- 1526 [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 1527
- 1528
- 1529 [URI] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.
- 1530
- 1531
- 1532 [WS-Addressing] <http://www.w3.org/TR/ws-addr-core>
- 1533 [WS-BrokeredNotification] <http://docs.oasis-open.org/wsn/wsn-WS-BrokeredNotification-1.3-cd-01.pdf>
- 1534
- 1535 [WS-Resource] http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-cd-01.pdf
- 1536
- 1537 [WS-ResourceLifetime] http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-cd-01.pdf
- 1538
- 1539 [WS-ResourceProperties] http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-cd-01.pdf
- 1540
- 1541 [WS-Topics] <http://docs.oasis-open.org/wsn/wsn-WS-Topics-1.3-cd-01.pdf>
- 1542 [XML-Infoset] <http://www.w3.org/TR/xml-infoset/>
- 1543 [XPath] <http://www.w3.org/TR/xpath>

1544 8.2 Non-Normative

- 1545 [SOAP 1.1] <http://www.w3.org/TR/SOAP>
- 1546 [SOAP 1.2] <http://www.w3.org/TR/soap12-part1/>
- 1547 [WS-Security] <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>
- 1548

1549 Appendix A. Acknowledgments

1550 Special thanks to the Global Grid Forum's Open Grid Services Infrastructure working group,
1551 which defined the OGSi v1.0 [OGSI] specification which was a large inspiration for the ideas
1552 expressed in this specification.

1553 The following individuals were members of the committee during the development of this
1554 specification:

1555

1556 Sid Askary, Fred Carter (AmberPoint), Martin Chapman (Oracle), Dave Chappell (Sonic
1557 Software), Rick Cobb (KnowNow), Ugo Corda (SeeBeyond Technology Corporation), John Fuller,
1558 Stephen Graham (IBM), David Hull (Tibco), Hideharu Kato (Hitachi), Lily Liu (webMethods, Inc.),
1559 Tom Maguire (IBM), Susan Malaika (IBM), Samuel Meder (Argonne National Laboratory), Bryan
1560 Murray (Hewlett-Packard), Peter Niblett (IBM), Sanjay Patil (SAP), Mark Peel (Novell), Matt
1561 Roberts (IBM), Igor Sedukhin (Computer Associates), David Snelling (Fujitsu), Latha Srinivasan

1562 (Hewlett-Packard), William Vambenepe (Hewlett-Packard) and Kirk Wilson (Computer
1563 Associates).

1564

1565 In addition, the following people made contributions to this specification:

1566 Tim Banks (IBM), Nick Butler (IBM), Doug Davis (IBM), John Dinger (IBM), Don Ferguson (IBM),
1567 Jeff Frey (IBM), Andreas Koeppel (SAP), Heather Kreger (IBM), Amy Lewis (TIBCO Software),
1568 Kevin Liu (SAP), Nataraj Nagaratnam (IBM), Martin Nally (IBM), Jeff Nick (IBM), Jay Parikh
1569 (Akamai Technologies), Claus von Riegen (SAP), Rick Rineholt (IBM), John Rofrano (IBM),
1570 Shivajee Samdarshi (TIBCO Software), Eugène Sindambiwe (SAP), Jay Unger (IBM), Bill Weihl
1571 (Akamai Technologies), Mark Weitzel (IBM), and Dan Wolfson (IBM).

1572 **Appendix B. XML Schema**

1573 The XML types and elements used in this specification are defined in the following XML Schema:

```
1574 <?xml version="1.0" encoding="UTF-8"?>
```

```
1575 <!--
```

```
1576
```

```
1577 OASIS takes no position regarding the validity or scope of any  
1578 intellectual property or other rights that might be claimed to pertain  
1579 to the implementation or use of the technology described in this  
1580 document or the extent to which any license under such rights might or  
1581 might not be available; neither does it represent that it has made any  
1582 effort to identify any such rights. Information on OASIS's procedures  
1583 with respect to rights in OASIS specifications can be found at the  
1584 OASIS website. Copies of claims of rights made available for  
1585 publication and any assurances of licenses to be made available, or the  
1586 result of an attempt made to obtain a general license or permission for  
1587 the use of such proprietary rights by implementors or users of this  
1588 specification, can be obtained from the OASIS Executive Director.
```

```
1589
```

```
1590 OASIS invites any interested party to bring to its attention any  
1591 copyrights, patents or patent applications, or other proprietary rights  
1592 which may cover technology that may be required to implement this  
1593 specification. Please address the information to the OASIS Executive  
1594 Director.
```

```
1595
```

```
1596 Copyright (C) OASIS Open (2004-2005). All Rights Reserved.
```

```
1597
```

1598 This document and translations of it may be copied and furnished to
1599 others, and derivative works that comment on or otherwise explain it or
1600 assist in its implementation may be prepared, copied, published and
1601 distributed, in whole or in part, without restriction of any kind,
1602 provided that the above copyright notice and this paragraph are
1603 included on all such copies and derivative works. However, this
1604 document itself may not be modified in any way, such as by removing the
1605 copyright notice or references to OASIS, except as needed for the
1606 purpose of developing OASIS specifications, in which case the
1607 procedures for copyrights defined in the OASIS Intellectual Property
1608 Rights document must be followed, or as required to translate it into
1609 languages other than English.

1610

1611 The limited permissions granted above are perpetual and will not be
1612 revoked by OASIS or its successors or assigns.

1613

1614 This document and the information contained herein is provided on an
1615 "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
1616 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
1617 INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
1618 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1619

1620 -->

1621

1622 <xsd:schema
1623 targetNamespace="http://docs.oasis-open.org/wsn/b-1"
1624 xmlns:wsnt="http://docs.oasis-open.org/wsn/b-1"
1625 xmlns:wsa="http://www.w3.org/2005/03/addressing"
1626 xmlns:wsrf-bf="http://docs.oasis-open.org/wsrf/bf-1"
1627 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1628 elementFormDefault="qualified" attributeFormDefault="unqualified">

1629

1630 <!-- ===== Imports ===== -->

1631

1632 <xsd:import namespace="http://www.w3.org/2005/03/addressing"
1633 schemaLocation="http://www.w3.org/2005/03/addressing"
1634 />

1635

1636 <xsd:import namespace="http://docs.oasis-open.org/wsrf/bf-1"
1637 schemaLocation="http://docs.oasis-open.org/wsrf/bf-1"
1638 />

1639

1640 <!-- ===== Misc. Helper Types ===== -->

1641

1642 <xsd:complexType name="QueryExpressionType" mixed="true">
1643 <xsd:sequence>
1644 <xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
1645 </xsd:sequence>

```

1646     <xsd:attribute name="Dialect" type="xsd:anyURI" use="required"/>
1647 </xsd:complexType>
1648
1649 <xsd:complexType name="TopicExpressionType" mixed="true">
1650   <xsd:sequence>
1651     <xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
1652   </xsd:sequence>
1653   <xsd:attribute name="Dialect" type="xsd:anyURI" use="required" />
1654   <xsd:anyAttribute/>
1655 </xsd:complexType>
1656
1657 <xsd:complexType name="FilterType">
1658   <xsd:sequence>
1659     <xsd:any minOccurs="0" maxOccurs="unbounded"/>
1660   </xsd:sequence>
1661 </xsd:complexType>
1662
1663 <!-- ===== Resource Property Related ===== -->
1664 <!-- ===== Resource Properties for NotificationProducer ===== -->
1665 <xsd:element name="TopicExpression" type="wsnt:TopicExpressionType"/>
1666 <xsd:element name="FixedTopicSet" type="xsd:boolean" default="true"/>
1667 <xsd:element name="TopicExpressionDialect" type="xsd:anyURI"/>
1668
1669 <xsd:element name="NotificationProducerRP">
1670   <xsd:complexType>
1671     <xsd:sequence>
1672       <xsd:element ref="wsnt:TopicExpression"
1673         minOccurs="0" maxOccurs="unbounded" />
1674       <xsd:element ref="wsnt:FixedTopicSet"
1675         minOccurs="0" maxOccurs="1" />
1676       <xsd:element ref="wsnt:TopicExpressionDialect"
1677         minOccurs="0" maxOccurs="unbounded" />
1678     </xsd:sequence>
1679   </xsd:complexType>
1680 </xsd:element>
1681
1682 <!-- ===== Resource Properties for SubscriptionManager ===== -->
1683 <xsd:element name="ConsumerReference"
1684   type="wsa:EndpointReferenceType" />
1685 <xsd:element name="Filter" type="wsnt:FilterType" />
1686 <xsd:element name="SubscriptionPolicy">
1687   <xsd:complexType>
1688     <xsd:sequence>
1689       <xsd:any minOccurs="0" maxOccurs="unbounded"
1690 processContents="lax"/>
1691     </xsd:sequence>
1692   </xsd:complexType>
1693 </xsd:element>

```

```

1694
1695 <xsd:element name="CreationTime" type="xsd:dateTime" />
1696
1697 <xsd:element name="SubscriptionManagerRP" >
1698   <xsd:complexType>
1699     <xsd:sequence>
1700       <xsd:element ref="wsnt:ConsumerReference"
1701         minOccurs="1" maxOccurs="1" />
1702       <xsd:element ref="wsnt:Filter"
1703         minOccurs="0" maxOccurs="1" />
1704       <xsd:element ref="wsnt:SubscriptionPolicy"
1705         minOccurs="0" maxOccurs="1" />
1706       <xsd:element ref="wsnt:CreationTime"
1707         minOccurs="0" maxOccurs="1" />
1708     </xsd:sequence>
1709   </xsd:complexType>
1710 </xsd:element>
1711
1712 <!-- ===== Notification Metadata ===== -->
1713 <xsd:element name="SubscriptionReference"
1714   type="wsa:EndpointReferenceType" />
1715 <xsd:element name="Topic"
1716   type="wsnt:TopicExpressionType" />
1717 <xsd:element name="ProducerReference"
1718   type="wsa:EndpointReferenceType" />
1719
1720 <!-- ===== Message Helper Types ===== -->
1721 <xsd:complexType name="NotificationMessageHolderType" >
1722   <xsd:sequence>
1723     <xsd:element ref="wsnt:SubscriptionReference"
1724       minOccurs="0" maxOccurs="1" />
1725     <xsd:element ref="wsnt:Topic"
1726       minOccurs="0" maxOccurs="1" />
1727     <xsd:element ref="wsnt:ProducerReference"
1728       minOccurs="0" maxOccurs="1" />
1729     <xsd:element name="Message">
1730       <xsd:complexType>
1731         <xsd:sequence>
1732           <xsd:any namespace="##any" processContents="lax"
1733             minOccurs="1" maxOccurs="1"/>
1734         </xsd:sequence>
1735       </xsd:complexType>
1736     </xsd:element>
1737   </xsd:sequence>
1738 </xsd:complexType>
1739 <xsd:element name="NotificationMessage"
1740   type="wsnt:NotificationMessageHolderType"/>
1741

```

```

1742 <!-- ===== Message Types for NotificationConsumer ===== -->
1743 <xsd:element name="Notify" >
1744 <xsd:complexType>
1745 <xsd:sequence>
1746 <xsd:element ref="wsnt:NotificationMessage"
1747 minOccurs="1" maxOccurs="unbounded" />
1748 <xsd:any namespace="##any" processContents="lax"
1749 minOccurs="0" maxOccurs="unbounded" />
1750 </xsd:sequence>
1751 </xsd:complexType>
1752 </xsd:element>
1753
1754 <!-- ===== Message Types for NotificationProducer ===== -->
1755
1756 <xsd:simpleType name="AbsoluteOrRelativeTimeType">
1757 <xsd:union memberTypes="xsd:dateTime xsd:duration" />
1758 </xsd:simpleType>
1759
1760 <xsd:element name="CurrentTime" type="xsd:dateTime" />
1761
1762 <xsd:element name="TerminationTime"
1763 nillable="true" type="xsd:dateTime" />
1764
1765 <xsd:element name="ProducerProperties"
1766 type="wsnt:QueryExpressionType" />
1767
1768 <xsd:element name="MessageContent"
1769 type="wsnt:QueryExpressionType" />
1770
1771 <xsd:element name="UseRaw"><xsd:complexType/></xsd:element>
1772
1773 <xsd:element name="Subscribe" >
1774 <xsd:complexType>
1775 <xsd:sequence>
1776 <xsd:element name="ConsumerReference"
1777 type="wsa:EndpointReferenceType"
1778 minOccurs="1" maxOccurs="1" />
1779 <xsd:element name="Filter"
1780 type="wsnt:FilterType"
1781 minOccurs="0" maxOccurs="1" />
1782 <xsd:element name="InitialTerminationTime"
1783 type="wsnt:AbsoluteOrRelativeTimeType"
1784 nillable="true"
1785 minOccurs="0" maxOccurs="1" />
1786 <xsd:element name="SubscriptionPolicy"
1787 minOccurs="0" maxOccurs="1">
1788 <xsd:complexType>
1789 <xsd:sequence>

```

```

1790         <xsd:any namespace="##any" processContents="lax"
1791             minOccurs="0" maxOccurs="unbounded" />
1792     </xsd:sequence>
1793 </xsd:complexType>
1794 </xsd:element>
1795     <xsd:any namespace="##other" processContents="lax"
1796         minOccurs="0" maxOccurs="unbounded" />
1797 </xsd:sequence>
1798 </xsd:complexType>
1799 </xsd:element>
1800
1801 <xsd:element name="SubscribeResponse">
1802     <xsd:complexType>
1803         <xsd:sequence>
1804             <xsd:element name="SubscriptionReference"
1805                 type="wsa:EndpointReferenceType"
1806                 minOccurs="0" maxOccurs="1" />
1807             <xsd:element ref="wsnt:CurrentTime"
1808                 minOccurs="0" maxOccurs="1" />
1809             <xsd:element ref="wsnt:TerminationTime"
1810                 minOccurs="0" maxOccurs="1" />
1811             <xsd:any namespace="##other" processContents="lax"
1812                 minOccurs="0" maxOccurs="unbounded" />
1813         </xsd:sequence>
1814     </xsd:complexType>
1815 </xsd:element>
1816
1817 <xsd:element name="GetCurrentMessage">
1818     <xsd:complexType>
1819         <xsd:sequence>
1820             <xsd:element name="Topic"
1821                 type="wsnt:TopicExpressionType" />
1822             <xsd:any namespace="##other" processContents="lax"
1823                 minOccurs="0" maxOccurs="unbounded" />
1824         </xsd:sequence>
1825     </xsd:complexType>
1826 </xsd:element>
1827
1828 <xsd:element name="GetCurrentMessageResponse">
1829     <xsd:complexType>
1830         <xsd:sequence>
1831             <xsd:any namespace="##other" processContents="lax"
1832                 minOccurs="0" maxOccurs="unbounded" />
1833         </xsd:sequence>
1834     </xsd:complexType>
1835 </xsd:element>
1836
1837 <xsd:complexType name="SubscribeCreationFailedFaultType">

```

```

1838     <xsd:complexContent>
1839         <xsd:extension base="wsrf-bf:BaseFaultType" />
1840     </xsd:complexContent>
1841 </xsd:complexType>
1842 <xsd:element name="SubscribeCreationFailedFault"
1843             type="wsnt:SubscribeCreationFailedFaultType" />
1844
1845 <xsd:complexType name="InvalidFilterFaultType">
1846     <xsd:complexContent>
1847         <xsd:extension base="wsrf-bf:BaseFaultType">
1848             <xsd:sequence>
1849                 <xsd:element name="UnknownFilter" type="xsd:QName"
1850                             minOccurs="1" maxOccurs="unbounded" />
1851             </xsd:sequence>
1852         </xsd:extension>
1853     </xsd:complexContent>
1854 </xsd:complexType>
1855 <xsd:element name="InvalidFilterFault"
1856             type="wsnt:InvalidFilterFaultType" />
1857
1858 <xsd:complexType name="TopicExpressionDialectUnknownFaultType">
1859     <xsd:complexContent>
1860         <xsd:extension base="wsrf-bf:BaseFaultType" />
1861     </xsd:complexContent>
1862 </xsd:complexType>
1863 <xsd:element name="TopicExpressionDialectUnknownFault"
1864             type="wsnt:TopicExpressionDialectUnknownFaultType" />
1865
1866 <xsd:complexType name="InvalidTopicExpressionFaultType">
1867     <xsd:complexContent>
1868         <xsd:extension base="wsrf-bf:BaseFaultType" />
1869     </xsd:complexContent>
1870 </xsd:complexType>
1871 <xsd:element name="InvalidTopicExpressionFault"
1872             type="wsnt:InvalidTopicExpressionFaultType" />
1873
1874 <xsd:complexType name="TopicNotSupportedFaultType">
1875     <xsd:complexContent>
1876         <xsd:extension base="wsrf-bf:BaseFaultType" />
1877     </xsd:complexContent>
1878 </xsd:complexType>
1879 <xsd:element name="TopicNotSupportedFault"
1880             type="wsnt:TopicNotSupportedFaultType" />
1881
1882 <xsd:complexType name="MultipleTopicsSpecifiedFaultType">
1883     <xsd:complexContent>
1884         <xsd:extension base="wsrf-bf:BaseFaultType" />
1885     </xsd:complexContent>

```



```

1886 </xsd:complexType>
1887 <xsd:element name="MultipleTopicsSpecifiedFault"
1888     type="wsnt:MultipleTopicsSpecifiedFaultType" />
1889
1890 <xsd:complexType name="InvalidProducerPropertiesExpressionFaultType">
1891     <xsd:complexContent>
1892         <xsd:extension base="wsrf-bf:BaseFaultType" />
1893     </xsd:complexContent>
1894 </xsd:complexType>
1895 <xsd:element name="InvalidProducerPropertiesExpressionFault"
1896     type="wsnt:InvalidProducerPropertiesExpressionFaultType" />
1897
1898 <xsd:complexType name="InvalidMessageContentExpressionFaultType">
1899     <xsd:complexContent>
1900         <xsd:extension base="wsrf-bf:BaseFaultType" />
1901     </xsd:complexContent>
1902 </xsd:complexType>
1903 <xsd:element name="InvalidMessageContentExpressionFault"
1904     type="wsnt:InvalidMessageContentExpressionFaultType" />
1905
1906 <xsd:complexType name="InvalidUseRawValueFaultType">
1907     <xsd:complexContent>
1908         <xsd:extension base="wsrf-bf:BaseFaultType" />
1909     </xsd:complexContent>
1910 </xsd:complexType>
1911 <xsd:element name="InvalidUseRawValueFault"
1912     type="wsnt:InvalidUseRawValueFaultType" />
1913
1914 <xsd:complexType name="UnacceptableInitialTerminationTimeFaultType">
1915     <xsd:complexContent>
1916         <xsd:extension base="wsrf-bf:BaseFaultType">
1917             <xsd:sequence>
1918                 <xsd:element name="MinimumTime" type="xsd:dateTime" />
1919                 <xsd:element name="MaximumTime" type="xsd:dateTime"
1920                     minOccurs="0" />
1921             </xsd:sequence>
1922         </xsd:extension>
1923     </xsd:complexContent>
1924 </xsd:complexType>
1925 <xsd:element name="UnacceptableInitialTerminationTimeFault"
1926     type="wsnt:UnacceptableInitialTerminationTimeFaultType" />
1927
1928 <xsd:complexType name="NoCurrentMessageOnTopicFaultType">
1929     <xsd:complexContent>
1930         <xsd:extension base="wsrf-bf:BaseFaultType" />
1931     </xsd:complexContent>
1932 </xsd:complexType>
1933 <xsd:element name="NoCurrentMessageOnTopicFault"

```

```

1934         type="wsnt:NoCurrentMessageOnTopicFaultType" />
1935
1936 <!-- ===== Message Types for PullPoint ===== -->
1937 <xsd:element name="GetMessages">
1938   <xsd:complexType>
1939     <xsd:sequence>
1940       <xsd:element name="MaximumNumber"
1941         type="xsd:nonNegativeInteger" />
1942       <xsd:any namespace="##other" processContents="lax"
1943         minOccurs="0" maxOccurs="unbounded" />
1944     </xsd:sequence>
1945     <xsd:anyAttribute/>
1946   </xsd:complexType>
1947 </xsd:element>
1948
1949 <xsd:element name="GetMessagesResponse">
1950   <xsd:complexType>
1951     <xsd:sequence>
1952       <xsd:element ref="wsnt:NotificationMessage"
1953         minOccurs="0" maxOccurs="unbounded" />
1954       <xsd:any namespace="##other" processContents="lax"
1955         minOccurs="0" maxOccurs="unbounded" />
1956     </xsd:sequence>
1957     <xsd:anyAttribute/>
1958   </xsd:complexType>
1959 </xsd:element>
1960
1961 <xsd:element name="Destroy">
1962   <xsd:complexType>
1963     <xsd:sequence>
1964       <xsd:any namespace="##other" processContents="lax"
1965         minOccurs="0" maxOccurs="unbounded" />
1966     </xsd:sequence>
1967     <xsd:anyAttribute/>
1968   </xsd:complexType>
1969 </xsd:element>
1970
1971 <xsd:element name="DestroyResponse">
1972   <xsd:complexType>
1973     <xsd:sequence>
1974       <xsd:any namespace="##other" processContents="lax"
1975         minOccurs="0" maxOccurs="unbounded" />
1976     </xsd:sequence>
1977     <xsd:anyAttribute/>
1978   </xsd:complexType>
1979 </xsd:element>
1980
1981 <xsd:complexType name="UnableToDestroyPullPointType">

```

```

1982     <xsd:complexContent>
1983         <xsd:extension base="wsrf-bf:BaseFaultType" />
1984     </xsd:complexContent>
1985 </xsd:complexType>
1986 <xsd:element name="UnableToDestroyPullPoint"
1987             type="wsnt:UnableToDestroyPullPointType" />
1988
1989 <!-- ===== Message Types for Create PullPoint ===== -->
1990 <xsd:element name="CreatePullPoint">
1991     <xsd:complexType>
1992         <xsd:sequence>
1993             <xsd:any namespace="##other" processContents="lax"
1994                 minOccurs="0" maxOccurs="unbounded" />
1995         </xsd:sequence>
1996         <xsd:anyAttribute/>
1997     </xsd:complexType>
1998 </xsd:element>
1999
2000 <xsd:element name="CreatePullPointResponse">
2001     <xsd:complexType>
2002         <xsd:sequence>
2003             <xsd:element name="PullPoint"
2004                 type="wsa:EndpointReferenceType" />
2005             <xsd:any namespace="##other" processContents="lax"
2006                 minOccurs="0" maxOccurs="unbounded" />
2007         </xsd:sequence>
2008         <xsd:anyAttribute/>
2009     </xsd:complexType>
2010 </xsd:element>
2011
2012 <xsd:complexType name="UnableToCreatePullPointType">
2013     <xsd:complexContent>
2014         <xsd:extension base="wsrf-bf:BaseFaultType" />
2015     </xsd:complexContent>
2016 </xsd:complexType>
2017 <xsd:element name="UnableToCreatePullPoint"
2018             type="wsnt:UnableToCreatePullPointType" />
2019
2020 <!-- ===== Message Types for Base SubscriptionManager ===== -->
2021 <xsd:element name="Renew">
2022     <xsd:complexType>
2023         <xsd:sequence>
2024             <xsd:element name="TerminationTime"
2025                 type="wsnt:AbsoluteOrRelativeTimeType"
2026                 nillable="true"
2027                 minOccurs="1" maxOccurs="1" />
2028             <xsd:any namespace="##other" processContents="lax"
2029                 minOccurs="0" maxOccurs="unbounded" />

```

```

2030     </xsd:sequence>
2031   </xsd:complexType>
2032 </xsd:element>
2033
2034 <xsd:element name="RenewResponse">
2035   <xsd:complexType>
2036     <xsd:sequence>
2037       <xsd:element ref="wsnt:TerminationTime"
2038         minOccurs="1" maxOccurs="1" />
2039       <xsd:element ref="wsnt:CurrentTime"
2040         minOccurs="0" maxOccurs="1" />
2041       <xsd:any namespace="##other" processContents="lax"
2042         minOccurs="0" maxOccurs="unbounded" />
2043     </xsd:sequence>
2044   </xsd:complexType>
2045 </xsd:element>
2046
2047 <xsd:complexType name="UnacceptableTerminationTimeFaultType">
2048   <xsd:complexContent>
2049     <xsd:extension base="wsrf-bf:BaseFaultType">
2050       <xsd:sequence>
2051         <xsd:element name="MinimumTime" type="xsd:dateTime" />
2052         <xsd:element name="MaximumTime" type="xsd:dateTime"
2053           minOccurs="0" />
2054       </xsd:sequence>
2055     </xsd:extension>
2056   </xsd:complexContent>
2057 </xsd:complexType>
2058 <xsd:element name="UnacceptableTerminationTimeFault"
2059   type="wsnt:UnacceptableTerminationTimeFaultType" />
2060
2061 <xsd:element name="Unsubscribe">
2062   <xsd:complexType>
2063     <xsd:sequence>
2064       <xsd:any namespace="##other" processContents="lax"
2065         minOccurs="0" maxOccurs="unbounded" />
2066     </xsd:sequence>
2067   </xsd:complexType>
2068 </xsd:element>
2069
2070 <xsd:element name="UnsubscribeResponse">
2071   <xsd:complexType>
2072     <xsd:sequence>
2073       <xsd:any namespace="##other" processContents="lax"
2074         minOccurs="0" maxOccurs="unbounded" />
2075     </xsd:sequence>
2076   </xsd:complexType>
2077 </xsd:element>

```

```

2078
2079 <xsd:complexType name="UnableToDestroySubscriptionFaultType">
2080 <xsd:complexContent>
2081 <xsd:extension base="wsrf-bf:BaseFaultType" />
2082 </xsd:complexContent>
2083 </xsd:complexType>
2084 <xsd:element name="UnableToDestroySubscriptionFault"
2085 <xsd:complexType name="UnableToDestroySubscriptionFaultType" />
2086
2087 <!-- ===== Message Types for Pausable SubscriptionManager ===== -->
2088
2089 <xsd:element name="PauseSubscription">
2090 <xsd:complexType>
2091 <xsd:sequence>
2092 <xsd:any namespace="##other" processContents="lax"
2093 minOccurs="0" maxOccurs="unbounded" />
2094 </xsd:sequence>
2095 </xsd:complexType>
2096 </xsd:element>
2097
2098 <xsd:element name="PauseSubscriptionResponse" >
2099 <xsd:complexType>
2100 <xsd:sequence>
2101 <xsd:any namespace="##other" processContents="lax"
2102 minOccurs="0" maxOccurs="unbounded" />
2103 </xsd:sequence>
2104 </xsd:complexType>
2105 </xsd:element>
2106
2107 <xsd:element name="ResumeSubscription">
2108 <xsd:complexType>
2109 <xsd:sequence>
2110 <xsd:any namespace="##other" processContents="lax"
2111 minOccurs="0" maxOccurs="unbounded" />
2112 </xsd:sequence>
2113 </xsd:complexType>
2114 </xsd:element>
2115
2116 <xsd:element name="ResumeSubscriptionResponse">
2117 <xsd:complexType>
2118 <xsd:sequence>
2119 <xsd:any namespace="##other" processContents="lax"
2120 minOccurs="0" maxOccurs="unbounded" />
2121 </xsd:sequence>
2122 </xsd:complexType>
2123 </xsd:element>
2124
2125 <xsd:complexType name="PauseFailedFaultType">

```

```

2126     <xsd:complexContent>
2127         <xsd:extension base="wsrf-bf:BaseFaultType" />
2128     </xsd:complexContent>
2129 </xsd:complexType>
2130 <xsd:element name="PauseFailedFault"
2131             type="wsnt:PauseFailedFaultType" />
2132
2133 <xsd:complexType name="ResumeFailedFaultType">
2134     <xsd:complexContent>
2135         <xsd:extension base="wsrf-bf:BaseFaultType" />
2136     </xsd:complexContent>
2137 </xsd:complexType>
2138 <xsd:element name="ResumeFailedFault"
2139             type="wsnt:ResumeFailedFaultType" />
2140
2141 </xsd:schema>

```

2142 **Appendix C. WSDL 1.1**

2143 The following illustrates the WSDL 1.1 for the Web service methods described in this
2144 specification:

```

2145 <?xml version="1.0" encoding="utf-8"?>
2146 <!--
2147
2148 OASIS takes no position regarding the validity or scope of any
2149 intellectual property or other rights that might be claimed to pertain
2150 to the implementation or use of the technology described in this
2151 document or the extent to which any license under such rights might or
2152 might not be available; neither does it represent that it has made any
2153 effort to identify any such rights. Information on OASIS's procedures
2154 with respect to rights in OASIS specifications can be found at the
2155 OASIS website. Copies of claims of rights made available for
2156 publication and any assurances of licenses to be made available, or the
2157 result of an attempt made to obtain a general license or permission for
2158 the use of such proprietary rights by implementors or users of this
2159 specification, can be obtained from the OASIS Executive Director.
2160
2161 OASIS invites any interested party to bring to its attention any
2162 copyrights, patents or patent applications, or other proprietary rights
2163 which may cover technology that may be required to implement this
2164 specification. Please address the information to the OASIS Executive
2165 Director.
2166
2167 Copyright (C) OASIS Open (2004-2005). All Rights Reserved.
2168

```

2169 This document and translations of it may be copied and furnished to
2170 others, and derivative works that comment on or otherwise explain it or
2171 assist in its implementation may be prepared, copied, published and
2172 distributed, in whole or in part, without restriction of any kind,
2173 provided that the above copyright notice and this paragraph are
2174 included on all such copies and derivative works. However, this
2175 document itself may not be modified in any way, such as by removing the
2176 copyright notice or references to OASIS, except as needed for the
2177 purpose of developing OASIS specifications, in which case the
2178 procedures for copyrights defined in the OASIS Intellectual Property
2179 Rights document must be followed, or as required to translate it into
2180 languages other than English.

2181

2182 The limited permissions granted above are perpetual and will not be
2183 revoked by OASIS or its successors or assigns.

2184

2185 This document and the information contained herein is provided on an
2186 "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
2187 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
2188 INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
2189 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

2190

2191 -->

2192 <wsdl:definitions name="WS-BaseNotification"
2193 targetNamespace="http://docs.oasis-open.org/wsn/bw-1"
2194 xmlns:wsntw="http://docs.oasis-open.org/wsn/bw-1"
2195 xmlns:wsnt="http://docs.oasis-open.org/wsn/b-1"
2196 xmlns:wsa="http://www.w3.org/2005/03/addressing"
2197 xmlns:wsrw="http://docs.oasis-open.org/wsrw/rw-1"
2198 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2199 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2200 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

2201

2202 <!-- ===== Imports ===== -->

2203 <wsdl:import
2204 namespace="http://docs.oasis-open.org/wsrw/rw-1"
2205 location="http://docs.oasis-open.org/wsrw/rw-1"/>

2206

2207 <!-- ===== Types Definitions ===== -->

2208 <wsdl:types>

2209 <xsd:schema>

2210 <xsd:import
2211 namespace="http://docs.oasis-open.org/wsn/b-1"
2212 schemaLocation="http://docs.oasis-open.org/wsn/b-1"/>

2213 </xsd:schema>

2214 </wsdl:types>

2215

2216 <!-- ===== NotificationConsumer::Notify =====

```

2217     Notify(
2218         NotificationMessage
2219             (SubscriptionReference, TopicExpression, ProducerReference,
2220             Message)*
2221     returns: n/a (one way)
2222 -->
2223     <wsdl:message name="Notify">
2224         <wsdl:part name="Notify" element="wsnt:Notify"/>
2225     </wsdl:message>
2226
2227 <!-- ===== NotificationProducer::Subscribe =====
2228     Subscribe(
2229         (ConsumerEndpointReference, [Filter], [SubscriptionPolicy],
2230         [InitialTerminationTime])
2231     returns: WS-Resource qualified EPR to a Subscription
2232 -->
2233     <wsdl:message name="SubscribeRequest" >
2234         <wsdl:part name="SubscribeRequest"
2235             element="wsnt:Subscribe"/>
2236     </wsdl:message>
2237
2238     <wsdl:message name="SubscribeResponse">
2239         <wsdl:part name="SubscribeResponse"
2240             element="wsnt:SubscribeResponse"/>
2241     </wsdl:message>
2242
2243     <wsdl:message name="SubscribeCreationFailedFault">
2244         <wsdl:part name="SubscribeCreationFailedFault"
2245             element="wsnt:SubscribeCreationFailedFault" />
2246     </wsdl:message>
2247
2248     <wsdl:message name="TopicExpressionDialectUnknownFault">
2249         <wsdl:part name="TopicExpressionDialectUnknownFault"
2250             element="wsnt:TopicExpressionDialectUnknownFault" />
2251     </wsdl:message>
2252
2253     <wsdl:message name="InvalidFilterFault">
2254         <wsdl:part name="InvalidFilterFault"
2255             element="wsnt:InvalidFilterFault" />
2256     </wsdl:message>
2257
2258     <wsdl:message name="InvalidProducerPropertiesExpressionFault">
2259         <wsdl:part name="InvalidProducerPropertiesExpressionFault"
2260             element="wsnt:InvalidProducerPropertiesExpressionFault" />
2261     </wsdl:message>
2262
2263     <wsdl:message name="InvalidMessageContentExpressionFault">
2264         <wsdl:part name="InvalidMessageContentExpressionFault"

```



```

2265         element="wsnt:InvalidMessageContentExpressionFault" />
2266     </wsdl:message>
2267
2268     <wsdl:message name="InvalidUseRawValueFault">
2269         <wsdl:part name="InvalidUseRawValueFault"
2270             element="wsnt:InvalidUseRawValueFault" />
2271     </wsdl:message>
2272
2273     <wsdl:message name="UnacceptableInitialTerminationTimeFault">
2274         <wsdl:part name="UnacceptableInitialTerminationTimeFault"
2275             element="wsnt:UnacceptableInitialTerminationTimeFault"/>
2276     </wsdl:message>
2277
2278 <!-- ===== NotificationProducer::GetCurrentMessage =====
2279     GetCurrentMessage(topicExpression)
2280     returns: a NotificationMessage (xsd:any)
2281 -->
2282     <wsdl:message name="GetCurrentMessageRequest">
2283         <wsdl:part name="GetCurrentMessageRequest"
2284             element="wsnt:GetCurrentMessage"/>
2285     </wsdl:message>
2286
2287     <wsdl:message name="GetCurrentMessageResponse">
2288         <wsdl:part name="GetCurrentMessageResponse"
2289             element="wsnt:GetCurrentMessageResponse"/>
2290     </wsdl:message>
2291
2292     <wsdl:message name="InvalidTopicExpressionFault">
2293         <wsdl:part name="InvalidTopicExpressionFault"
2294             element="wsnt:InvalidTopicExpressionFault" />
2295     </wsdl:message>
2296
2297     <wsdl:message name="TopicNotSupportedFault">
2298         <wsdl:part name="TopicNotSupportedFault"
2299             element="wsnt:TopicNotSupportedFault" />
2300     </wsdl:message>
2301
2302     <wsdl:message name="MultipleTopicsSpecifiedFault">
2303         <wsdl:part name="MultipleTopicsSpecifiedFault"
2304             element="wsnt:MultipleTopicsSpecifiedFault" />
2305     </wsdl:message>
2306
2307     <wsdl:message name="NoCurrentMessageOnTopicFault">
2308         <wsdl:part name="NoCurrentMessageOnTopicFault"
2309             element="wsnt:NoCurrentMessageOnTopicFault" />
2310     </wsdl:message>
2311
2312 <!-- ===== PullPoint::GetMessages =====

```

```

2313     GetMessages (MaximumNumber )
2314     returns: NotificationMessage list
2315     -->
2316     <wsdl:message name="GetMessagesRequest">
2317         <wsdl:part name="GetMessagesRequest"
2318             element="wsnt:GetMessages" />
2319     </wsdl:message>
2320
2321     <wsdl:message name="GetMessagesResponse">
2322         <wsdl:part name="GetMessagesResponse"
2323             element="wsnt:GetMessagesResponse" />
2324     </wsdl:message>
2325
2326 <!-- ===== PullPoint::Destroy =====
2327     Destroy()
2328     returns: void
2329     -->
2330     <wsdl:message name="DestroyRequest">
2331         <wsdl:part name="DestroyRequest"
2332             element="wsnt:Destroy" />
2333     </wsdl:message>
2334
2335     <wsdl:message name="DestroyResponse">
2336         <wsdl:part name="DestroyResponse"
2337             element="wsnt:DestroyResponse" />
2338     </wsdl:message>
2339
2340     <wsdl:message name="UnableToDestroyPullPoint">
2341         <wsdl:part name="UnableToDestroyPullPoint"
2342             element="wsnt:UnableToDestroyPullPoint" />
2343     </wsdl:message>
2344
2345 <!-- ===== PullPoint::CreatePullPoint =====
2346     CreatePullPoint()
2347     returns: PullPoint (wsa:EndpointReference)
2348     -->
2349     <wsdl:message name="CreatePullPointRequest">
2350         <wsdl:part name="CreatePullPointRequest"
2351             element="wsnt:CreatePullPoint" />
2352     </wsdl:message>
2353
2354     <wsdl:message name="CreatePullPointResponse">
2355         <wsdl:part name="CreatePullPointResponse"
2356             element="wsnt:CreatePullPointResponse" />
2357     </wsdl:message>
2358
2359     <wsdl:message name="UnableToCreatePullPoint">
2360         <wsdl:part name="UnableToCreatePullPoint"

```

```

2361         element="wsnt:UnableToCreatePullPoint"/>
2362     </wsdl:message>
2363
2364     <!-- ===== SubscriptionManager::Renew =====
2365         Renew( Duration | AbsoluteTime)
2366         returns: (New Termination Time [CurrentTime])
2367     -->
2368     <wsdl:message name="RenewRequest">
2369         <wsdl:part name="RenewRequest"
2370             element="wsnt:Renew"/>
2371     </wsdl:message>
2372
2373     <wsdl:message name="RenewResponse">
2374         <wsdl:part name="RenewResponse"
2375             element="wsnt:RenewResponse"/>
2376     </wsdl:message>
2377
2378     <wsdl:message name="UnacceptableTerminationTimeFault">
2379         <wsdl:part name="UnacceptableTerminationTimeFault"
2380             element="wsnt:UnacceptableTerminationTimeFault" />
2381     </wsdl:message>
2382
2383     <!-- ===== SubscriptionManager::Unsubscribe =====
2384         Unsubscribe()
2385         returns: empty
2386     -->
2387     <wsdl:message name="UnsubscribeRequest">
2388         <wsdl:part name="UnsubscribeRequest"
2389             element="wsnt:Unsubscribe"/>
2390     </wsdl:message>
2391
2392     <wsdl:message name="UnsubscribeResponse">
2393         <wsdl:part name="UnsubscribeResponse"
2394             element="wsnt:UnsubscribeResponse"/>
2395     </wsdl:message>
2396
2397     <wsdl:message name="UnableToDestroySubscriptionFault">
2398         <wsdl:part name="UnableToDestroySubscriptionFault"
2399             element="wsnt:UnableToDestroySubscriptionFault" />
2400     </wsdl:message>
2401
2402     <!-- ===== SubscriptionManager::PauseSubscription =====
2403         PauseSubscription()
2404         returns: empty
2405     -->
2406     <wsdl:message name="PauseSubscriptionRequest">
2407         <wsdl:part name="PauseSubscriptionRequest"
2408             element="wsnt:PauseSubscription"/>

```

```

2409     </wsdl:message>
2410
2411     <wsdl:message name="PauseSubscriptionResponse">
2412         <wsdl:part name="PauseSubscriptionResponse"
2413             element="wsnt:PauseSubscriptionResponse"/>
2414     </wsdl:message>
2415
2416     <wsdl:message name="PauseFailedFault">
2417         <wsdl:part name="PauseFailedFault"
2418             element="wsnt:PauseFailedFault" />
2419     </wsdl:message>
2420
2421 <!-- ===== SubscriptionManager::ResumeSubscription =====
2422     ResumeSubscription()
2423     returns: empty
2424 -->
2425     <wsdl:message name="ResumeSubscriptionRequest">
2426         <wsdl:part name="ResumeSubscriptionRequest"
2427             element="wsnt:ResumeSubscription"/>
2428     </wsdl:message>
2429
2430     <wsdl:message name="ResumeSubscriptionResponse">
2431         <wsdl:part name="ResumeSubscriptionResponse"
2432             element="wsnt:ResumeSubscriptionResponse"/>
2433     </wsdl:message>
2434
2435     <wsdl:message name="ResumeFailedFault">
2436         <wsdl:part name="ResumeFailedFault"
2437             element="wsnt:ResumeFailedFault" />
2438     </wsdl:message>
2439
2440 <!-- ===== PortType Definitions ===== -->
2441 <!-- ===== NotificationConsumer PortType Definition ===== -->
2442     <wsdl:portType name="NotificationConsumer">
2443         <wsdl:operation name="Notify">
2444             <wsdl:input message="wsntw:Notify" />
2445         </wsdl:operation>
2446     </wsdl:portType>
2447
2448 <!-- ===== NotificationProducer PortType Definition ===== -->
2449     <wsdl:portType name="NotificationProducer">
2450         <wsdl:operation name="Subscribe">
2451             <wsdl:input message="wsntw:SubscribeRequest" />
2452             <wsdl:output message="wsntw:SubscribeResponse" />
2453             <wsdl:fault name="ResourceUnknownFault"
2454                 message="wsrf-rw:ResourceUnknownFault" />
2455             <wsdl:fault name="InvalidFilterFault"
2456                 message="wsntw:InvalidFilterFault" />

```

```

2457     <wsdl:fault name="TopicExpressionDialectUnknownFault"
2458             message="wsntw:TopicExpressionDialectUnknownFault" />
2459     <wsdl:fault name="InvalidTopicExpressionFault"
2460             message="wsntw:InvalidTopicExpressionFault" />
2461     <wsdl:fault name="TopicNotSupportedFault"
2462             message="wsntw:TopicNotSupportedFault" />
2463     <wsdl:fault name="InvalidProducerPropertiesExpressionFault"
2464             message="wsntw:InvalidProducerPropertiesExpressionFault" />
2465     <wsdl:fault name="InvalidMessageContentExpressionFault"
2466             message="wsntw:InvalidMessageContentExpressionFault" />
2467     <wsdl:fault name="InvalidUseRawValueFault"
2468             message="wsntw:InvalidUseRawValueFault" />
2469     <wsdl:fault name="UnacceptableInitialTerminationTimeFault"
2470             message="wsntw:UnacceptableInitialTerminationTimeFault" />
2471     <wsdl:fault name="SubscribeCreationFailedFault"
2472             message="wsntw:SubscribeCreationFailedFault" />
2473 </wsdl:operation>
2474
2475 <wsdl:operation name="GetCurrentMessage">
2476     <wsdl:input message="wsntw:GetCurrentMessageRequest" />
2477     <wsdl:output message="wsntw:GetCurrentMessageResponse" />
2478     <wsdl:fault name="ResourceUnknownFault"
2479             message="wsrf-rw:ResourceUnknownFault" />
2480     <wsdl:fault name="TopicExpressionDialectUnknownFault"
2481             message="wsntw:TopicExpressionDialectUnknownFault" />
2482     <wsdl:fault name="InvalidTopicExpressionFault"
2483             message="wsntw:InvalidTopicExpressionFault" />
2484     <wsdl:fault name="TopicNotSupportedFault"
2485             message="wsntw:TopicNotSupportedFault" />
2486     <wsdl:fault name="NoCurrentMessageOnTopicFault"
2487             message="wsntw:NoCurrentMessageOnTopicFault" />
2488 </wsdl:operation>
2489 </wsdl:portType>
2490
2491 <!-- ===== PullPoint PortType Definition ===== -->
2492 <wsdl:portType name="PullPoint">
2493     <wsdl:operation name="GetMessages">
2494         <wsdl:input name="GetMessagesRequest"
2495                 message="wsntw:GetMessagesRequest" />
2496         <wsdl:output name="GetMessagesResponse"
2497                 message="wsntw:GetMessagesResponse" />
2498         <wsdl:fault name="ResourceUnknownFault"
2499                 message="wsrf-rw:ResourceUnknownFault" />
2500     </wsdl:operation>
2501
2502     <wsdl:operation name="Destroy">
2503         <wsdl:input name="DestroyRequest"
2504                 message="wsntw:DestroyRequest" />

```

```

2505         <wsdl:output name="DestroyResponse"
2506             message="wsntw:DestroyResponse" />
2507         <wsdl:fault name="UnableToDestroyPullPoint"
2508             message="wsntw:UnableToDestroyPullPoint" />
2509     </wsdl:operation>
2510 </wsdl:portType>
2511
2512 <!-- ===== CreatePullPoint PortType Definition ===== -->
2513 <wsdl:portType name="CreatePullPoint">
2514     <wsdl:operation name="CreatePullPoint">
2515         <wsdl:input name="CreatePullPointRequest"
2516             message="wsntw:CreatePullPointRequest" />
2517         <wsdl:output name="CreatePullPointResponse"
2518             message="wsntw:CreatePullPointResponse" />
2519         <wsdl:fault name="UnableToCreatePullPoint"
2520             message="wsntw:UnableToCreatePullPoint" />
2521     </wsdl:operation>
2522 </wsdl:portType>
2523
2524 <!-- ===== SubscriptionManager PortType Definition ===== -->
2525 <wsdl:portType name="SubscriptionManager">
2526     <wsdl:operation name="Renew">
2527         <wsdl:input name="RenewRequest"
2528             message="wsntw:RenewRequest" />
2529         <wsdl:output name="RenewResponse"
2530             message="wsntw:RenewResponse" />
2531         <wsdl:fault name="ResourceUnknownFault"
2532             message="wsrf-rw:ResourceUnknownFault" />
2533         <wsdl:fault name="UnacceptableTerminationTimeFault"
2534             message=
2535                 "wsntw:UnacceptableTerminationTimeFault" />
2536     </wsdl:operation>
2537     <wsdl:operation name="Unsubscribe">
2538         <wsdl:input name="UnsubscribeRequest"
2539             message="wsntw:UnsubscribeRequest" />
2540         <wsdl:output name="UnsubscribeResponse"
2541             message="wsntw:UnsubscribeResponse" />
2542         <wsdl:fault name="ResourceUnknownFault"
2543             message="wsrf-rw:ResourceUnknownFault" />
2544         <wsdl:fault name="UnableToDestroySubscriptionFault"
2545             message=
2546                 "wsntw:UnableToDestroySubscriptionFault" />
2547     </wsdl:operation>
2548 </wsdl:portType>
2549
2550 <!-- ===== PausableSubscriptionManager PortType Definition ===== -->
2551 <wsdl:portType name="PausableSubscriptionManager">
2552     <!-- ===== Extends: SubscriptionManager ===== -->

```

```

2553     <wsdl:operation name="Renew">
2554         <wsdl:input name="RenewRequest"
2555             message="wsntw:RenewRequest" />
2556         <wsdl:output name="RenewResponse"
2557             message="wsntw:RenewResponse" />
2558         <wsdl:fault name="ResourceUnknownFault"
2559             message="wsrf-rw:ResourceUnknownFault" />
2560         <wsdl:fault name="UnacceptableTerminationTimeFault"
2561             message=
2562             "wsntw:UnacceptableTerminationTimeFault" />
2563     </wsdl:operation>
2564     <wsdl:operation name="Unsubscribe">
2565         <wsdl:input name="UnsubscribeRequest"
2566             message="wsntw:UnsubscribeRequest" />
2567         <wsdl:output name="UnsubscribeResponse"
2568             message="wsntw:UnsubscribeResponse" />
2569         <wsdl:fault name="ResourceUnknownFault"
2570             message="wsrf-rw:ResourceUnknownFault" />
2571         <wsdl:fault name="UnableToDestroySubscriptionFault"
2572             message=
2573             "wsntw:UnableToDestroySubscriptionFault" />
2574     </wsdl:operation>
2575
2576     <!-- === PausableSubscriptionManager specific operations === -->
2577     <wsdl:operation name="PauseSubscription">
2578         <wsdl:input message="wsntw:PauseSubscriptionRequest" />
2579         <wsdl:output message="wsntw:PauseSubscriptionResponse" />
2580         <wsdl:fault name="ResourceUnknownFault"
2581             message="wsrf-rw:ResourceUnknownFault" />
2582         <wsdl:fault name="PauseFailedFault"
2583             message="wsntw:PauseFailedFault" />
2584     </wsdl:operation>
2585     <wsdl:operation name="ResumeSubscription">
2586         <wsdl:input message="wsntw:ResumeSubscriptionRequest" />
2587         <wsdl:output message="wsntw:ResumeSubscriptionResponse" />
2588         <wsdl:fault name="ResourceUnknownFault"
2589             message="wsrf-rw:ResourceUnknownFault" />
2590         <wsdl:fault name="ResumeFailedFault"
2591             message="wsntw:ResumeFailedFault" />
2592     </wsdl:operation>
2593 </wsdl:portType>
2594 </wsdl:definitions>

```

2595 **Appendix D. Revision History**

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

wsn-ws_base_notification-1.3-spec-pr-01

7/8/2005

Copyright © OASIS Open 2004-2005. All Rights Reserved.

Page 63 of 66

Rev	Date	By Whom	What
wd-01	2004-05-18	Steve Graham	Initial version created from submission by contributing companies. Minor modifications made to reflect OASIS formatting and namespace URI choices.
wd-02	2004-05-24	Steve Graham	Added requirements section draft and updated acknowledgments appendix.
wd-03	2004-06-07	Steve Graham	Minor modifications on wd-02 as reported by Peter N. and William V.
wd-04	2004-06-21	Steve Graham	Finalize list of acknowledgments and annotate title page with result of TC vote.
wd-03	2004-07-12	Steve Graham	"rename" 04 back to 03 to minimize reference rework in WSRF.
wd-04	2004-09-20	Steve Graham	WSN 2.3, 2.11, 2.17, 2.18, 2.19, 2.21, 2.22, 2.29
1.3, 01a	2004-09-27	Steve Graham	Rename to follow agreed upon format, add resolution to WSN 5.2
1.3, 01b	2004-11-17	Steve Graham	Post Oct f2f edits
1.3, 01c	2004-12-7	Steve Graham, David Hull	WSN 1,7, 2.7, 2.13, 2.14, 2.15, 2.24 and WSN2.17 revisited
1.3, 01d	2004-12-8	Steve Graham	Fix Action URIs and other edits per Tom Maguire's comments
1.3, 01e	2005-02-19	Bryan Murray	Editorial text and formatting, WSN2.1, WSN2.37, WSN2.38
1.3,01e	2005-02-25	Steve Graham	Mod's to Bryan's edits, WSN 2.34, WSN2.35
1.3, 01e	2005-03-02	Steve Graham	Minor typos reported by Matt Haynos' team.
1.3, 01f	2005-05-09	Bryan Murray, David Hull	WSN1.5, WSN2.1 (InvalidFilterFault), WSN2.32, WSN2.39, WSN2.43, WSN2.45, WSN2.46
1.3, 01g	2005-05-18	Bryan Murray	Per face-to-face agreement on hygiene issues. WSN 2.48 (but see comment to list from dmh) (Partially?) WSN 2.49 WSN 2.50 WSN 2.53
1.3.01g	2005-05-19	David Hull	Integrated definitions from White Paper,

Rev	Date	By Whom	What
			including updates for WSN2.39 compliance. NotificationMessage is not Notification throughout. Began integrating security section from whitepaper, along with proposed text for WSN 2.24 Fixed a few more WS 2.39 stragglers
1.3.01 h	2005-05-27	David Hull	WSN 2.51, WSN 2.52, (also it seems, 2.33, 2.48, 2.1 (pending David raising omnibus issue), 2.23 (partial), 2.25, 2.34, 2.46)
1.3.01.i	2005-06-02	Steve Graham	Modifications on 01g and 01h text, completion of schema and WSDL related items and 2.47, 2.55, 2.41, 2.23
1.3.01.j	2005-06-09	Bryan Murray	AI 106, AI 109, WSN1.4, section 7.1 cleanup, remove ref to Primer
1.3.01.k	2005-06-29	David Hull	Integrated review comments for 2.1, 2.25, 2.33, 2.39, 2.48, 2.50, 2.52. Pseudo-schema descriptions of open content.
1.3.01.l	2005-07-06	David Hull	Integrated review from chairs: 2.6, 2.34, 2.52, 2.55, typos, abstract, acknowledgements.
1.3.01.m	2005-07-07	David Hull	Amended requirement for composability, per committee vote. Formatting in section 4.2

2596 **Appendix E. Notices**

2597 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
2598 that might be claimed to pertain to the implementation or use of the technology described in this
2599 document or the extent to which any license under such rights might or might not be available;
2600 neither does it represent that it has made any effort to identify any such rights. Information on
2601 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
2602 website. Copies of claims of rights made available for publication and any assurances of licenses
2603 to be made available, or the result of an attempt made to obtain a general license or permission
2604 for the use of such proprietary rights by implementors or users of this specification, can be
2605 obtained from the OASIS Executive Director.
2606

2607 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
2608 applications, or other proprietary rights which may cover technology that may be required to
2609 implement this specification. Please address the information to the OASIS Executive Director.

2610

2611 Copyright (C) OASIS Open (2004-2005). All Rights Reserved.

2612

2613 This document and translations of it may be copied and furnished to others, and derivative works
2614 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
2615 published and distributed, in whole or in part, without restriction of any kind, provided that the
2616 above copyright notice and this paragraph are included on all such copies and derivative works.
2617 However, this document itself may not be modified in any way, such as by removing the copyright
2618 notice or references to OASIS, except as needed for the purpose of developing OASIS
2619 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
2620 Property Rights document must be followed, or as required to translate it into languages other
2621 than English.

2622

2623 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
2624 successors or assigns.

2625

2626 This document and the information contained herein is provided on an "AS IS" basis and OASIS
2627 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
2628 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
2629 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
2630 PARTICULAR PURPOSE.

2631