



Web Services Base Notification 1.3 (WS-BaseNotification)

Committee Specification, 31 July 2006

Document identifier

wsn-ws_base_notification-1.3-spec-cs-01

Location:

http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-cs-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:

This document is published by the OASIS WS-Notification Technical Committee as a "Committee Specification",

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

35 web page at <http://www.oasis-open.org/committees/wsn>. Click the button for "Send A
36 Comment" at the top of the page. Submitted comments (for this work as well as other
37 works of the TC) are publicly 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-
42 open.org/committees/wsn/](http://www.oasis-open.org/committees/wsn/)).

43 The errata document for this specification is maintained at:
44 http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-errata.pdf

Table of Contents

46	1	Introduction	5
47	1.1	Goals and Requirements	5
48	1.1.1	Requirements.....	5
49	1.1.2	Non-Goals	6
50	1.2	Terminology.....	6
51	1.3	Namespaces	7
52	1.4	Fault Definitions.....	8
53	2	Terminology and Concepts	9
54	2.1	Definitions.....	9
55	2.2	Production vs. Delivery	10
56	3	NotificationConsumer Interface	11
57	3.1	Notification Metadata	11
58	3.2	Notify	12
59	3.2.1	Example SOAP Encoding of the Notify Message.....	13
60	4	NotificationProducer Interface	14
61	4.1	NotificationProducer Resource Properties.....	14
62	4.2	Subscribe	15
63	4.2.1	Example SOAP Encoding of the Subscribe Message Exchange.....	21
64	4.3	GetCurrentMessage.....	22
65	4.3.1	Example SOAP Encoding of the GetCurrentMessage Message Exchange	24
66	5	Pull-Style Notification.....	25
67	5.1	PullPoint Interface	25
68	5.1.1	Accumulating Notification Messages.....	25
69	5.1.2	GetMessages.....	26
70	5.1.3	Example SOAP Encoding of the GetMessages Message Exchange.....	27
71	5.1.4	DestroyPullPoint	28
72	5.1.5	Example SOAP Encoding of the DestroyPullPoint Message Exchange	28
73	5.2	Create PullPoint Interface	29
74	5.2.1	Example SOAP Encoding of the CreatePullPoint Message Exchange.....	30
75	6	SubscriptionManager Interface.....	31
76	6.1	Base Subscription Manager	31
77	6.1.1	Renew.....	31
78	6.1.2	Unsubscribe	33
79	6.2	Pausable Subscription Manager	34
80	6.2.1	PauseSubscription	35
81	6.2.2	Example SOAP Encoding of the PauseSubscription Message Exchange.....	36
82	6.3	ResumeSubscription	36

83	6.3.1 Example SOAP Encoding of the ResumeSubscription Message Exchange.....	37
84	6.4 Subscriptions as WS-Resources.....	38
85	6.4.1 Subscription Resource Properties	38
86	7 Security Considerations	40
87	7.1 Securing the Message Exchanges.....	40
88	7.2 Securing Subscriptions and Notifications.....	40
89	8 References.....	42
90	8.1 Normative	42
91	8.2 Non-Normative	42
92	Appendix A. Acknowledgments	44
93	Appendix B. XML Schema.....	45
94	Appendix C. WSDL 1.1.....	56
95	Appendix D. Revision History	65
96	Appendix E. Notices	68
97		

98 1 Introduction

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

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

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

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

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

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

125 1.1 Goals and Requirements

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

129 1.1.1 Requirements

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

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

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

149 1.1.2 Non-Goals

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

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

165 1.2 Terminology

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

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

172 This specification uses a notational convention, referred to as “Pseudo-schemas”. A Pseudo-
173 schema uses a BNF-style 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.

- 179 • Attributes are conventionally assigned a value which corresponds to their type, as
- 180 defined in the normative schema.
- 181 • Elements with simple content are conventionally assigned a value which corresponds to
- 182 the type of their content, as defined in the normative schema.
- 183 • The use of {any} indicates the presence of an element wildcard (<xs:any/>).
- 184 • The use of @{any} indicates the presence of an attribute wildcard (<xs:anyAttribute/>).
- 185 • In the interest of brevity, some extensibility points have been omitted from the Pseudo-
- 186 schemas.

```

187 <!-- sample pseudo-schema -->
188 <element
189   required_attribute_of_type_QName="xs:QName"
190   optional_attribute_of_type_string="xs:string"? >
191   <required_element />
192   <optional_element /> ?
193   <one_or_more_of_these_elements /> +
194   [ <choice_1 /> | <choice_2 /> ] *
195 </element>

```

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

201 1.3 Namespaces

202 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/2005/08/addressing
wsrf-rp	http://docs.oasis-open.org/wsrp/rp-2
wsrf-bf	http://docs.oasis-open.org/wsrp/bf-2
wsnt	http://docs.oasis-open.org/wsn/b-2
wsntw	http://docs.oasis-open.org/wsn/bw-2
wstop	http://docs.oasis-open.org/wsn/t-1

203 **1.4 Fault Definitions**

204 All faults generated by a NotificationProducer or SubscriptionManager SHOULD be compliant
205 with the WS-BaseFaults [WS-BaseFaults] specification.

206 All faults defined by this specification MUST use the following URI for the WS-Addressing [action]
207 Message Addressing Property:

208 <http://docs.oasis-open.org/wsn/fault>.

209

2 Terminology and Concepts

210

2.1 Definitions

211

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

212

Situation:

213

- A Situation is some occurrence known to a NotificationProducer and of potential interest to third parties.

214

215

- A Situation could be a change of the internal state of a resource or could be environmental, such as a timer event. It could also be an external event, such as a piece of news that has been supplied by a news-feed service.

216

217

218

- WS-Notification does not specify what a Situation is or is not, nor does it define the relationship between a Situation and the Notification(s) that are used to describe it.

219

220

Notification:

221

- A Notification is an artifact of a Situation containing information about that Situation that some entity wishes to communicate to other entities.

222

223

- A Notification is represented as an XML element with a Namespace qualified QName and a type defined using XML Schema.

224

225

- A typical usage pattern is to define a single Notification type (to be precise, its defining XML element) for each kind of Situation, containing information pertinent to that kind of Situation; in this case one can think of a Notification instance as in some sense *being* (or at least representing) the Situation.

226

227

228

229

- A designer could choose to associate several different Notification types with a Situation, for example, describing different aspects of the Situation, destined for different target recipients, etc. Conversely it is possible that several essentially different Situations give rise to Notifications of the same type.

230

231

232

233

NotificationProducer:

234

- A NotificationProducer is a Web service that implements the message exchanges associated with the NotificationProducer interface.

235

236

- A NotificationProducer is capable of producing Notifications for those NotificationConsumers for which Subscriptions have been registered, based on Situations that occur and on the parameters supplied with the requests from which the Subscriptions were created.

237

238

239

240

- A Web Service that implements the message exchanges associated with NotificationProducer may directly produce Notifications itself, or it may be a NotificationBroker, reproducing Notifications that were produced by separate Publisher and/or NotificationProducer entities.

241

242

243

244

- It is the factory for Subscription resources.

245

NotificationConsumer:

- 246
- 247
- 248
- A NotificationConsumer is an endpoint, represented by a WS-Addressing endpoint reference, designated to receive Notifications produced by a NotificationProducer as a result of a subscription.
- 249
- A NotificationConsumer may accept the generic Notify message, or it may be able to process one or more domain-specific Notification types.
- 250

251 **Subscription:**

- 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.
- 253
- A Subscription resource is created when a Subscriber sends the SubscribeRequest message to a NotificationProducer.
- 254
- Subscription resources are manipulated by messages sent to the SubscriptionManager Web service associated with the Subscription resource.
- 255
- 256
- 257
- 258

259 **SubscriptionManager**

- 260
- A SubscriptionManager is an endpoint, represented by an endpoint reference [WS-Addressing] that implements message exchanges associated with the SubscriptionManager interface.
- 261
- A SubscriptionManager provides operations that allow a service requestor to query and manipulate Subscription resources that it manages.
- 262
- 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
- 264
- 265
- 266
- 267
- 268

269 **Subscriber:**

- 270
- A Subscriber is any entity that sends the SubscribeRequest message to a NotificationProducer.
- 271
- Note that a Subscriber may be a different entity from the NotificationConsumer for which Notifications are actually produced.
- 272
- 273

274 **2.2 Production vs. Delivery**

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

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

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

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

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

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

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

3 NotificationConsumer Interface

282

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

284 1. The NotificationConsumer MAY simply receive the “raw” Notification (i.e. the application-
285 specific content).

286 2. The NotificationConsumer MAY receive the Notification data as a Notify message as
287 described below.

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

294 When a Subscriber sends a Subscribe request message, it indicates which form of Notification is
295 required (the raw Notification, or the Notify Message). The NotificationProducer MUST observe
296 this component of the Subscription and use the form that has been requested, if it is able. If it
297 does not support the form requested, it MUST fault.

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

3.1 Notification Metadata

302

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

307 wsnt:SubscriptionReference

308 An EndpointReference to the Subscription that is associated with the Notify message.

309 wsnt:Topic

310 A TopicExpression describing exactly one Topic, which MUST be the Topic that is
311 associated with the Notification. This element describes the Topic that matched to a
312 subscription, causing the NotificationProducer to send the Notify message to the
313 NotificationConsumer.

314 wsnt:Topic/@Dialect

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

317 wsnt:ProducerReference

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

319

320 When using the Notify message format, these elements, if present, MUST take the form
321 described below. When using the raw message format, these elements MAY appear as

322 components of the message (such as SOAP 1.1 [SOAP 1.1] or SOAP 1.2[SOAP 1.2] header
323 elements) in accordance with any contract with the NotificationConsumer.

324 When producing a raw Notification, the NotificationProducer SHOULD utilise the Action URI
325 defined below for the Notify operation except in situations where the NotificationProducer has
326 knowledge which would cause it to use some alternative URI.

327

328 3.2 Notify

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

```
331 ...  
332 <wsnt:Notify>  
333   <wsnt:NotificationMessage>  
334     <wsnt:SubscriptionReference>  
335       wsa:EndpointReferenceType  
336     </wsnt:SubscriptionReference> ?  
337     <wsnt:Topic Dialect="xsd:anyURI">  
338       {any} ?  
339     </wsnt:Topic>?  
340     <wsnt:ProducerReference>  
341       wsa:EndpointReferenceType  
342     </wsnt:ProducerReference> ?  
343     <wsnt:Message>  
344       {any}  
345     </wsnt:Message>  
346   </wsnt:NotificationMessage> +  
347   {any} *  
348 </wsnt:Notify>  
349 ...
```

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

351 <http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify>.

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

353 /wsnt:Notify

354 Contains a collection of one or more Notifications.

355 /wsnt:Notify/wsnt:NotificationMessage

356 Contains a Notification payload along with any metadata components as defined in
357 section 3.1.

358 /wsnt:Notify/wsnt:NotificationMessage/wsnt:Message

359 A copy of the actual Notification payload.

360 /wsnt:Notify/{any}

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

364

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

366 3.2.1 Example SOAP Encoding of the Notify Message

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

```
368 <s:Envelope ... >
369   <s:Header>
370     <wsa:Action>
371       http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify
372     </wsa:Action>
373     ...
374   </s:Header>
375   <s:Body>
376     <wsnt:Notify>
377       <wsnt:NotificationMessage>
378         <wsnt:SubscriptionReference>
379           <wsa:Address>
380             http://www.example.org/SubscriptionManager
381           </wsa:Address>
382         </wsnt:SubscriptionReference>
383         <wsnt:Topic Dialect=
384           "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
385           npex:SomeTopic
386         </wsnt:Topic>
387         <wsnt:ProducerReference>
388           <wsa:Address>
389             http://www.example.org/NotificationProducer
390           </wsa:Address>
391         </wsnt:ProducerReference>
392         <wsnt:Message>
393           <npex:NotifyContent>exampleNotifyContent</npex:NotifyContent>
394         </wsnt:Message>
395       </wsnt:NotificationMessage>
396     </wsnt:Notify>
397   </s:Body>
398 </s:Envelope>
```

399

4 NotificationProducer Interface

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

403

4.1 NotificationProducer Resource Properties

404 In addition to the message exchanges described in this specification, a NotificationProducer MAY
405 also support the required message exchanges defined in the [WS-ResourceProperties]
406 specification and MAY support the optional message exchanges defined in the WS-
407 ResourceProperties specification. In such a case, this specification defines several resource
408 properties, which MUST conform to the following schema fragment for content:.

409
410
411
412
413
414
415
416
417
418
419
420

```
...  
targetNamespace="http://docs.oasis-open.org/wsn/b-2"  
...  
<xsd:element name="TopicExpression" type="wsnt:TopicExpressionType" />  
<xsd:element name="FixedTopicSet" type="xsd:boolean" default="true" />  
  
<xsd:element name="TopicExpressionDialect" type="xsd:anyURI" />  
  
<xsd:element name="wstop:TopicSet" type="wstop:TopicSetType" />  
...
```

421

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

422
423
424
425
426
427
428
429
430
431
432
433
434
435

```
<xsd:element ref="wsnt:TopicExpression"  
minOccurs="0" maxOccurs="unbounded" />  
  
<xsd:element ref="wsnt:FixedTopicSet"  
minOccurs="0" maxOccurs="1" />  
  
<xsd:element ref="wsnt:TopicExpressionDialect"  
minOccurs="0" maxOccurs="unbounded" />  
  
<xsd:element ref="wstop:TopicSet"  
minOccurs="0" maxOccurs="1" />
```

436

437 These Resource Property elements are further constrained as follows:

438

/wsnt:TopicExpression

439

This resource property contains a collection of topics supported by the NotificationProducer.

440

The set of topics is expressed using one or more wsnt:TopicExpression resource property

441

element(s). Each wsnt:TopicExpression resource property element is a TopicExpression. The

442

dialect of TopicExpression used can be any dialect. It is RECOMMENDED to use one of the

443 TopicExpression dialects described in [WS-Topics]. A NotificationProducer MAY use an
444 expression that refers to multiple topics, if the dialect used has this capability. The same topic
445 may be referred to by multiple wsnt:TopicExpression resource property element(s), for
446 example, in different resource property elements each using a different dialect.

447 If a topic is identified by one of the wsnt:TopicExpression resource property elements, a
448 Subscriber can reasonably expect that the NotificationProducer will not return an
449 InvalidTopicExpressionFault for subscription requests for the topic. It is not a guarantee that
450 the NotificationConsumer will receive any Notifications. For example, the NotificationProducer
451 may not actually produce any Notifications on the particular topic during the time that the
452 Subscriber is registered.

453 /wsnt:FixedTopicSet

454 Indicates if the collection of topics contained within the wsnt:TopicExpression resource
455 property may change. This value is “true” if the collection of topics supported by the
456 NotificationProducer does not change and “false” if the NotificationProducer allows the
457 collection to change (for example if it allows additional topics to be supported should
458 publishers or subscribers request them). This property defaults to “true” if missing.

459 /wsnt:TopicExpressionDialect

460 Indicates one or more TopicExpression dialect(s) that are supported by the
461 NotificationProducer. If a URI corresponding to a dialect appears in this resource property, a
462 subscriber is assured that a subscription request containing a valid topic expression using
463 that dialect will be accepted by the NotificationProducer. The TopicExpressionDialect
464 property is a “fixed property”, i.e. its value, for any given NotificationProducer, does not
465 change over time.

466 /wstop:TopicSet

467 This resource property contains the collection of topics supported by the NotificationProducer
468 expressed, as a single XML element as described in [WS-Topics].

469

470 **4.2 Subscribe**

471 A NotificationProducer is capable of producing a sequence of zero or more Notifications. A
472 Subscriber can register the interest of a NotificationConsumer to receive a subset of this
473 sequence. A Subscriber sends a Subscribe message to a NotificationProducer in order to register
474 this interest.

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

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

481 A given NotificationConsumer may be the object of more than one Subscription, and separate
482 NotificationConsumers may subscribe to the same subset of Notifications. In such situations, WS-
483 BaseNotification places no restrictions on the order in which Notifications are produced.
484 Notifications for different NotificationConsumers may be produced in different orders, even when
485 the associated subscription requests are otherwise identical, and Notifications from separate
486 Subscriptions with the same NotificationConsumer may be interleaved in any manner.

487 NotificationProducers MAY advertise more constrained behavior through policy assertions or
488 other means.

489 The format of the Subscribe message is:

```
490 ...  
491 <wsnt:Subscribe>  
492   <wsnt:ConsumerReference>  
493     wsa:EndpointReferenceType  
494   </wsnt:ConsumerReference>  
495   <wsnt:Filter>  
496     [ <wsnt:TopicExpression Dialect="xsd:anyURI">  
497       {any} ?  
498     </wsnt:TopicExpression> |  
499     <wsnt:ProducerProperties Dialect="xsd:anyURI">  
500       {any} ?  
501     </wsnt:ProducerProperties> |  
502     <wsnt:MessageContent Dialect="xsd:anyURI">  
503       {any} ?  
504     </wsnt:MessageContent> |  
505     {any} *  
506   ] *  
507 </wsnt:Filter> ?  
508 <wsnt:InitialTerminationTime>  
509   [xsd:dateTime | xsd:duration]  
510 </wsnt:InitialTerminationTime> ?  
511 <wsnt:SubscriptionPolicy>  
512   [ <wsnt:UseRaw/> |  
513     {any}  
514   ] *  
515 </wsnt:SubscriptionPolicy> ?  
516 {any}*  
517 </wsnt:Subscribe>  
518 ...
```

519 The WS-Addressing [action] Message Addressing Property MUST contain the URI
520 <http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/SubscribeRequest>.

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

522 /wsnt:Subscribe/wsnt:ConsumerReference

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

525 This component SHOULD provide all the necessary information to specify how the
526 NotificationProducer should send notifications to the NotificationConsumer. However, there
527 may be cases when the ConsumerReference EPR may not include all the details that the
528 NotificationProducer expects.

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

534 In cases where the wsnt:UseRaw policy component is not specified, the Web service
535 identified by the endpoint reference SHOULD implement the Notify message, as the
536 NotificationProducer will by default produce Notifications in this form.

537 /wsnt:Subscribe/wsnt:Filter

538 The Filter component is the means by which a Subscriber expresses the subset of
539 Notifications that the NotificationConsumer should receive. This subset is expressed by the
540 child elements of the Filter. The child elements are a sequence of zero or more expressions
541 evaluating to a Boolean that constrain the set of possible Notifications. Each expression is
542 evaluated in a manner specific to that kind of expression (see below); the order and timing of
543 the evaluation is determined by the NotificationProducer. The NotificationProducer MUST
544 NOT produce the Notification if any of the filter expressions evaluates to false.

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

548 The NotificationProducer MUST respond with an InvalidFilterFault message if any child
549 expression element is not supported by the NotificationProducer. For example, if the
550 NotificationProducer does not support the concept of Topics, it MUST respond with an
551 InvalidFilterFault message if a Subscribe message contains a Filter component that includes
552 a TopicExpression child. The fault MUST include the filter QNames for the filters it did not
553 understand.

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

557 /wsnt:Subscribe/wsnt:Filter/wsnt:TopicExpression

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

561 /wsnt:Subscribe/wsnt:Filter/wsnt:TopicExpression/@Dialect

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

565 The NotificationProducer MAY refuse to process the Subscribe request if the dialect used by
566 the Subscriber in the TopicExpression is not one of the dialects supported by the
567 NotificationProducer.

568 The NotificationProducer MUST respond with a TopicExpressionDialectUnknownFault if it
569 understands the TopicExpression element, but does not understand the specified
570 TopicExpression dialect. Note that a NotificationProducer may understand the meaning of a
571 given dialect URI without actually supporting that dialect.

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

574 If the TopicExpression dialect is understood and the expression references a topic which is
575 not supported by the NotificationProducer, the NotificationProducer MAY return a
576 TopicNotSupportedFault.

577 /wsnt:Subscribe/wsnt:Filter/wsnt:ProducerProperties

578 This component contains a filter expression evaluated on the ResourceProperties [WS-
579 ResourceProperties] of the NotificationProducer (if the NotificationProducer is acting as a
580 WS-Resource [WS-Resource]). The expression MUST be a Boolean expression.

581 /wsnt:Subscribe/wsnt:Filter/wsnt:ProducerProperties/@Dialect

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

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

588 /wsnt:Subscribe/wsnt:Filter/wsnt:MessageContent

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

592 /wsnt:Subscribe/wsnt:Filter/wsnt:MessageContent/@Dialect

593 This attribute contains a URI specifying the type of MessageContent filter expression
594 contained by the element. This specification identifies the XPath 1.0 dialect corresponding to
595 version 1.0 of the XPath language [XPath]:

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

597 This URI identifies the XPath 1.0 language. The contents of the MessageContent expression
598 MUST be a string containing a valid XPath 1.0 expression.

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

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

603 /wsnt:Subscribe/wsnt:InitialTerminationTime

604 This component contains the service requestor's suggestion for the initial termination time of
605 the Subscription being created. There are two forms of this component, absolute time and
606 duration. If the type of this component is xsd:dateTime, the value of the component is to be
607 interpreted as an absolute time. If the type of this component is xsd:duration, the value of the
608 component is to be interpreted as a duration to be added to the current time. All time
609 measurements are determined by the NotificationProducer.

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

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

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

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

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

624 /wsnt:Subscribe/wsnt:SubscriptionPolicy

625 This component, which can be omitted, is an open component intended to be used in an
626 application specific way to specify policy related requirements/assertions associated with the
627 subscribe requests. This mechanism could be used to govern the message rate (e.g.
628 maximum 3 messages per second), reliability of the Notification delivery, etc. The semantics
629 of how the NotificationProducer MUST or MAY react to the policy requirements and
630 assertions appearing in this component are specific to the actual policy grammar used.

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

634 /wsnt:Subscribe/wsnt:SubscriptionPolicy/wsnt:UseRaw

635 An element whose presence indicates that the NotificationProducer is to produce
636 Notifications without using the wsnt:Notify wrapper. This element SHOULD NOT occur more
637 than once in a Subscribe request message, as only its presence or absence is significant.

638 The presence of this element is interpreted as a request for raw Notification. If raw
639 notification is requested, then:

640 • If the NotificationProducer supports this form of notification, then the NotificationProducer
641 MUST produce the Notification message in raw form.

642 • If the NotificationProducer does not support this form of notification, it MUST return a
643 UnsupportedPolicyRequestFault message containing a wsnt:UseRaw QName. The
644 NotificationProducer MUST NOT return an UnrecognizedPolicyRequestFault in response
645 to this element.

646 The absence of this element is interpreted as a request for notification using the wsnt:Notify
647 wrapper. If notification using the Notify wrapper is requested, then the NotificationConsumer
648 referred to in the wsnt:ConsumerReference element SHOULD implement the Notify message
649 and include a Notify operation in its portType definition, and:

650 • If the NotificationProducer supports this form of notification, then the NotificationProducer
651 MUST produce Notification messages using the Notify wrapper

652 • If the NotificationProducer does not support this form of notification, it MUST return a
653 NotifyMessageNotSupportedFault message.

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

```
655 ...  
656 <wsnt:SubscribeResponse>  
657   <wsnt:SubscriptionReference>  
658     wsa:EndpointReferenceType  
659   </wsnt:SubscriptionReference>  
660   <wsnt:CurrentTime>xsd:dateTime</wsnt:CurrentTime> ?  
661   <wsnt:TerminationTime>xsd:dateTime</wsnt:TerminationTime> ?  
662   {any}*  
663 </wsnt:SubscribeResponse>
```

664

...

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

666 <http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/SubscribeResponse>.

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

668 /wsnt:SubscribeResponse/wsnt:SubscriptionReference

669 A reference to the Subscription created as a result of the Subscribe message.

670 /wsnt:SubscribeResponse/wsnt:CurrentTime

671 This OPTIONAL component SHOULD be returned if the SubscriptionManager uses
672 scheduled termination from WS-ResourceLifetime. The value of this component is the value
673 of the CurrentTime resource property of the Subscription at the time the response message is
674 created.

675 /wsnt:SubscribeResponse/wsnt:TerminationTime

676 This OPTIONAL component SHOULD be returned if the SubscriptionManager uses
677 scheduled termination from [WS-ResourceLifetime]. The value of this component is the value
678 of the TerminationTime resource property of the Subscription at the time the response
679 message is created.

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

683

684 ResourceUnknownFault

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

688 InvalidFilterFault

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

691 TopicExpressionDialectUnknownFault

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

694 InvalidTopicExpressionFault

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

697 TopicNotSupportedFault

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

700 InvalidProducerPropertiesExpressionFault

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

703 InvalidMessageContentExpressionFault

- 704 • The Subscribe message contained a MessageContent filter that did not represent a valid
705 boolean expression.
- 706 UnrecognizedPolicyRequestFault
- 707 • The NotificationProducer does not recognize one or more policy requests carried in the
708 SubscriptionPolicy element. This fault SHOULD contain elements representing the
709 offending policy requests.
- 710 UnsupportedPolicyRequestFault
- 711 • The NotificationProducer recognizes, but does not support one or more policy requests
712 carried in the SubscriptionPolicy element. This fault SHOULD contain elements
713 representing the offending policy requests.
- 714 NotifyMessageNotSupportedFault
- 715 • The NotificationProducer does not support the wsnt:Notify wrapper and was not able to
716 determine an alternative.
- 717 UnacceptableInitialTerminationTimeFault
- 718 • The value of InitialTerminationTime specified in the Subscribe message was not
719 acceptable to the NotificationProducer. The NotificationProducer MAY include a hint in
720 the fault message indicating acceptable values for InitialTerminationTime.
- 721 SubscribeCreationFailedFault
- 722 • The NotificationProducer failed to process the Subscribe message. The
723 NotificationProducer SHOULD use a more specific fault message if possible. The
724 NotificationProducer MAY include a hint in the fault message indicating why it failed to
725 process the Subscribe message.

726 4.2.1 Example SOAP Encoding of the Subscribe Message Exchange

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

```

728   <s:Envelope ... >
729    <s:Header>
730      <wsa:Action>
731        http://docs.oasis-open.org/wsn/bw-
732        2/NotificationProducer/SubscribeRequest
733      </wsa:Action>
734      ...
735    </s:Header>
736    <s:Body>
737      <wsnt:Subscribe>
738        <wsnt:ConsumerReference>
739          <wsa:Address>
740            http://www.example.org/NotificationConsumer
741          </wsa:Address>
742        </wsnt:ConsumerReference>
743        <wsnt:Filter>
744          <wsnt:TopicExpression Dialect=
745            "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
746            ncex:SomeTopic
747          </wsnt:TopicExpression>
748        <wsnt:MessageContent
749          Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
750          boolean(ncex:Producer="15")

```

```
751     </wsnt:MessageContent>
752   </wsnt:Filter>
753   <wsnt:InitialTerminationTime>
754     2005-12-25T00:00:00.000000Z
755   </wsnt:InitialTerminationTime>
756 </wsnt:Subscribe>
757 </s:Body>
758 </s:Envelope>
```

759

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

```
761 <s:Envelope ... >
762   <s:Header>
763     <wsa:Action>
764       http://docs.oasis-open.org/wsn/bw-
765 2/NotificationProducer/SubscribeResponse
766     </wsa:Action>
767     ...
768   </s:Header>
769   <s:Body>
770     <wsnt:SubscribeResponse>
771       <wsnt:SubscriptionReference>
772         <wsa:Address>
773           http://www.example.org/SubscriptionManager
774         </wsa:Address>
775       </wsnt:SubscriptionReference>
776     </wsnt:SubscribeResponse>
777   </s:Body>
778 </s:Envelope>
```

779 **4.3 GetCurrentMessage**

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

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

789 The format of the GetCurrentMessage request message is:

```
790 ...
791 <wsnt:GetCurrentMessage>
792   <wsnt:Topic Dialect="xsd:anyURI">
793     {any} ?
794   </wsnt:Topic>
795   {any} *
796 </wsnt:GetCurrentMessage>
797 ...
```

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

799 <http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/GetCurrentMessageRequest>.

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

801 `/wsnt:GetCurrentMessage/wsnt:Topic`

802 A `TopicExpression` that identifies exactly one `Topic`.

803 If the `NotificationProducer` successfully processes the `GetCurrentMessage` request, it MUST

804 respond with a `GetCurrentMessageResponse` message. This response has the following form:

```
805 ...
806 <wsnt:GetCurrentMessageResponse>
807   {any}
808 </wsnt:GetCurrentMessageResponse>
809 ...
```

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

811 <http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/GetCurrentMessageResponse>.

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

813 `/wsnt:GetCurrentMessageResponse/{any}`

814 Contains the last `Notification` associated with the `Topic` identified by the request message.

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

818 `ResourceUnknownFault`

819 • The `NotificationProducer` is acting as a `WS-Resource`, and the resource identified in the
820 message is not known to the Web service. This fault is specified by the `WS-Resource`
821 `[WS-Resource]` specification.

822 `TopicExpressionDialectUnknownFault`

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

825 `InvalidTopicExpressionFault`

826 • The `Topic` element of the `GetCurrentMessage` message had contents that did not match
827 the dialect specified.

828 `TopicNotSupportedFault`

829 • The `Topic` element of the `GetCurrentMessage` message referenced a topic that was not
830 supported by the `NotificationProducer`.

831 `MultipleTopicsSpecifiedFault`

832 • The `GetCurrentMessage` message referenced more than one topic.

833 `NoCurrentMessageOnTopicFault`

834 • The topic referenced in the `GetCurrentMessage` message does not have any pending
835 messages.

836
837

4.3.1 Example SOAP Encoding of the GetCurrentMessage Message Exchange

838 The following is a non-normative example of a GetCurrentMessage request message using
839 SOAP:

```
840 <s:Envelope ... >  
841   <s:Header>  
842     <wsa:Action>  
843       http://docs.oasis-open.org/wsn/bw-  
844 2/NotificationProducer/GetCurrentMessageRequest  
845     </wsa:Action>  
846     ...  
847   </s:Header>  
848   <s:Body>  
849     <wsnt:GetCurrentMessage>  
850       <wsnt:Topic Dialect=  
851         "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">  
852         npex:SomeTopic  
853       </wsnt:Topic>  
854     </wsnt:GetCurrentMessage>  
855   </s:Body>  
856 </s:Envelope>
```

857 The following is a non-normative example of a GetCurrentMessage response message using
858 SOAP:

```
859 <s:Envelope ... >  
860   <s:Header>  
861     <wsa:Action>  
862       http://docs.oasis-open.org/wsn/bw-  
863 2/NotificationProducer/GetCurrentMessageResponse  
864     </wsa:Action>  
865     ...  
866   </s:Header>  
867   <s:Body>  
868     <wsnt:GetCurrentMessageResponse>  
869       <npex:NotifyContent>exampleNotifyContent</npex:NotifyContent>  
870     </wsnt:GetCurrentMessageResponse>  
871   </s:Body>  
872 </s:Envelope>
```


873

5 Pull-Style Notification

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

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

5.1 PullPoint Interface

888
889 In support of the “pull-style” of Notification Message delivery, this specification describes a
890 PullPoint resource by defining a PullPoint interface. The PullPoint interface provides the means
891 by which NotificationProducers can insert Notification Messages into the PullPoint, by which
892 requestors can retrieve messages from the PullPoint and the means by which requestors can
893 destroy the PullPoint.

894 A PullPoint MAY be a WS-Resource, and if it is, the PullPoint WS-Resource MUST support the
895 immediate termination interface defined by WS-ResourceLifetime and it MAY support the
896 scheduled termination interface defined by WS-ResourceLifetime. If the PullPoint supports the
897 scheduled termination interface, the initial termination time is set by the factory that creates it.
898 This termination time may be queried and set via the standard WS-ResourceLifetime message
899 exchanges. PullPoints SHOULD NOT be destroyed if subscriptions involving them still exist.

900

5.1.1 Accumulating Notification Messages

901
902 The PullPoint interface supports the NotificationConsumer interface (as defined in section 3). This
903 interface allows NotificationProducers to send Notification Messages to the PullPoint using
904 the Notify message format. Notification Messages received by the PullPoint through its
905 NotificationConsumer interface are accumulated by the PullPoint on a best effort basis. If the
906 PullPoint is no longer capable of accumulating additional Notification Messages, it MAY ignore
907 Notification Messages sent to its NotificationConsumer interface, or it MAY dispose of any
908 previously accumulated Notification Messages and continue to accumulate incoming Notification
909 Messages. The PullPoint interface does not define additional constraints on its use of the
910 NotificationConsumer interface.

911 5.1.2 GetMessages

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

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

```
917 <wsnt:GetMessages>  
918   <wsnt:MaximumNumber>xsd:nonNegativeInteger</wsnt:MaximumNumber> ?  
919   {any} *  
920 </wsnt:GetMessages>
```

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

924 <http://docs.oasis-open.org/wsn/bw-2/PullPoint/GetMessagesRequest>.

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

926 /wsnt:GetMessages

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

929 /wsnt:GetMessages/wsnt:MaximumNumber

930 This non-negative integer, which can be omitted, indicates the maximum number of
931 accumulated Notification Messages to be returned in the response message. If the total
932 number of Notification Messages accumulated by the PullPoint is less than or equal to
933 this number, all of the Notification Messages are returned in the response message. If the
934 total number of accumulated Notification Messages exceeds this value, the PullPoint
935 MUST return exactly this number of accumulated Notification Messages; the
936 implementation of the PullPoint is free to determine which subset of messages are
937 returned in the response message in this case. If the MaximumNumber component is not
938 present in the request message, the requestor is asking to receive all messages held by
939 the PullPoint.

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

```
941 <wsnt:GetMessagesResponse>  
942   <wsnt:NotificationMessage /> *  
943   {any} *  
944 </wsnt:GetMessagesResponse>
```

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

948 <http://docs.oasis-open.org/wsn/bw-2/PullPoint/GetMessagesResponse>.

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

950 /wsnt:GetMessagesResponse

951 This component contains zero or more accumulated Notification Messages. The number
952 of messages appearing is limited by the wsnt:MaximumNumber component of the
953 GetMessages request message. Each accumulated Notification Message appearing in a

954 GetMessagesResponse is “removed” from the PullPoint and MUST NOT appear in the
955 response message of subsequent GetMessages requests.

956 /wsnt:GetMessagesResponse/wsnt:NotificationMessage

957 The content of this component is an accumulated NotificationMessage. The
958 NotificationMessage component is described as part of the Notify message (section 3.2)
959 the GetMessagesResponse message does not define additional constraints on the
960 NotificationMessage component. The content of the NotificationMessage is exactly the
961 content of the NotificationMessage component of the accumulated Notify message.

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

965 ResourceUnknownFault

966 • The PullPoint is acting as a WS-Resource, and the resource identified in the request
967 message is not known to the Web service.. This fault is specified by the WS-Resource
968 [WS-Resource] specification.

969 UnableToGetMessagesFault

970 • The PullPoint was unable to return messages for some reason

971 **5.1.3 Example SOAP Encoding of the GetMessages Message** 972 **Exchange**

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

```
974 <s:Envelope ... >  
975   <s:Header>  
976     <wsa:Action>  
977       http://docs.oasis-open.org/wsn/bw-2/PullPoint/GetMessagesRequest  
978     </wsa:Action>  
979     ...  
980   </s:Header>  
981   <s:Body>  
982     <wsnt:GetMessages>  
983       <wsnt:MaximumNumber>10</wsnt:MaximumNumber>  
984     </wsnt:GetMessages>  
985   </s:Body>  
986 </s:Envelope>
```

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

```
988 <s:Envelope ... >  
989   <s:Header>  
990     <wsa:Action>  
991       http://docs.oasis-open.org/wsn/bw-2/PullPoint/GetMessagesResponse  
992     </wsa:Action>  
993     ...  
994   </s:Header>  
995   <s:Body>  
996     <wsnt:GetMessagesResponse>  
997       <wsnt:NotificationMessage>  
998       ...  
999     </wsnt:NotificationMessage>
```

```
1000     <wsnt:NotificationMessage>
1001         ...
1002     </wsnt:NotificationMessage>
1003 </wsnt:GetMessagesResponse>
1004 </s:Body>
1005 </s:Envelope>
```

1006 **5.1.4 DestroyPullPoint**

1007 The PullPoint interface provides a destroy operation, providing a means by which a requestor can
1008 terminate the PullPoint resource. The DestroyPullPoint request message has the following form:

```
1009 <wsnt:DestroyPullPoint>
1010     {any} *
1011 </wsnt:DestroyPullPoint>
```

1014 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1015 <http://docs.oasis-open.org/wsn/bw-2/PullPoint/DestroyPullPointRequest>.

1016 Upon receipt of the DestroyPullPoint request, the PullPoint MUST attempt to destroy itself. If the
1017 DestroyPullPoint request message is successfully processed, the PullPoint MUST respond with
1018 the following message:

```
1019 <wsnt:DestroyPullPointResponse>
1020     {any} *
1021 </wsnt:DestroyPullPointResponse>
```

1024 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1025 <http://docs.oasis-open.org/wsn/bw-2/PullPoint/DestroyPullPointResponse>.

1026 If the PullPoint does not respond to the DestroyPullPoint request message with the
1027 DestroyPullPointResponse message, then it MUST send a fault. This specification defines the
1028 following faults associated with failure to process the DestroyPullPoint request message:

1029 ResourceUnknownFault

- 1030 • The PullPoint is acting as a WS-Resource, and the resource identified in the request
1031 message is not known to the Web service. This fault is specified by the WS-Resource
1032 [WS-Resource] specification.

1033

1034 UnableToDestroyPullPointFault.

- 1035 • The PullPoint was unable to destroy the PullPoint resource for some reason.

1036 **5.1.5 Example SOAP Encoding of the DestroyPullPoint Message** 1037 **Exchange**

1038 The following is a non-normative example of a DestroyPullPoint request message using SOAP:

```
1039 <s:Envelope ... >
1040 <s:Header>
```

```
1041     <wsa:Action>
1042         http://docs.oasis-open.org/wsn/bw-
1043     2/PullPoint/DestroyPullPointRequest
1044     </wsa:Action>
1045     ...
1046 </s:Header>
1047 <s:Body>
1048     <wsnt:DestroyPullPoint/>
1049 </s:Body>
1050 </s:Envelope>
```

1051 The following is a non-normative example of a DestroyPullPoint response message using SOAP:

```
1052 <s:Envelope ... >
1053 <s:Header>
1054     <wsa:Action>
1055         http://docs.oasis-open.org/wsn/bw-
1056     2/PullPoint/DestroyPullPointResponse
1057     </wsa:Action>
1058     ...
1059 </s:Header>
1060 <s:Body>
1061     <wsnt:DestroyPullPointResponse/>
1062 </s:Body>
1063 </s:Envelope>
```

1064 **5.2 Create PullPoint Interface**

1065 This specification describes a CreatePullPoint interface standardizing the means by which a
1066 PullPoint resource is created.

1067 If a requestor wishes to create a new PullPoint resource, it MUST send a CreatePullPoint request
1068 to an endpoint supporting the PullPoint interface. The CreatePullPoint request message has the
1069 following form:

```
1070 <wsnt:CreatePullPoint>
1071     {any} *
1072 </wsnt:CreatePullPoint>
```

1075 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1076 <http://docs.oasis-open.org/wsn/bw-2/CreatePullPoint/CreatePullPointRequest>

1077 Upon receipt of the CreatePullPoint request, the receiver MUST attempt to create a new PullPoint
1078 resource. If the CreatePullPoint request message is successfully processed, the receiver MUST
1079 respond with the following message:

```
1080 <wsnt:CreatePullPointResponse>
1081     <wsnt:PullPoint>wsa:EndpointReferenceType</wsnt:PullPoint>
1082     {any} *
1083 </wsnt:CreatePullPointResponse>
```

1086 The WS-Addressing [action] Message Addressing Property MUST contain the URI

11087 <http://docs.oasis-open.org/wsn/bw-2/CreatePullPoint/CreatePullPointResponse>.
11088 The contents of the CreatePullPoint response message are further described as follows:
11089 /wsnt:CreatePullPointResponse/wsnt:PullPoint
11090 This component is an EndpointReference, as defined by WS-Addressing, that is a
11091 reference to the PullPoint resource created during the processing of the CreatePullPoint
11092 request message.

11093 A PullPoint created by the CreatePullPoint operation MAY choose to ignore Notifications received
11094 through the NotificationConsumer interface. A Pull Point MAY accumulate Notifications through
11095 other, implementation-defined mechanisms. Pull point factories SHOULD indicate through policy
11096 assertions whether pull points they create will ignore Notifications received through the
11097 NotificationConsumer interface.

11098 If the receiver does not respond to the CreatePullPoint request message with the
11099 CreatePullPointResponse message, then it MUST send a fault. This specification defines the
11100 following fault associated with failure to process the DestroyPullPoint request message:

- 11101 • UnableToCreatePullPointFault.

11102

11103 **5.2.1 Example SOAP Encoding of the CreatePullPoint Message** 11104 **Exchange**

11105 The following is a non-normative example of a CreatePullPoint request message using SOAP:

```
11106 <s:Envelope ... >  
11107   <s:Header>  
11108     <wsa:Action>  
11109       http://docs.oasis-open.org/wsn/bw-2/PullPoint/CreatePullPointRequest  
11110     </wsa:Action>  
11111     ...  
11112   </s:Header>  
11113   <s:Body>  
11114     <wsnt:CreatePullPoint/>  
11115   </s:Body>  
11116 </s:Envelope>
```

11117 The following is a non-normative example of a CreatePullPoint response message using SOAP:

```
11118 <s:Envelope ... >  
11119   <s:Header>  
11120     <wsa:Action>  
11121       http://docs.oasis-open.org/wsn/bw-2/PullPoint/CreatePullPointResponse  
11122     </wsa:Action>  
11123     ...  
11124   </s:Header>  
11125   <s:Body>  
11126     <wsnt:CreatePullPointResponse>  
11127       <wsnt:PullPoint>  
11128         <wsa:Address>...</wsa:Address>  
11129         ...  
11130       </wsnt:PullPoint>  
11131     </wsnt:CreatePullPointResponse>  
11132   </s:Body>  
11133 </s:Envelope>
```

1134

6 SubscriptionManager Interface

1135 The SubscriptionManager interface defines message exchanges to manipulate Subscription
1136 resources. There are two styles of SubscriptionManager interface: base and pausable. All
1137 SubscriptionManagers MUST implement the message exchanges described in the Base
1138 Subscription Manager section.

1139 For those environments that choose to implement a Subscription as a WS-Resource, the
1140 SubscriptionManager MUST implement the resource properties and message exchanges defined
1141 in Section 6.4 (Subscriptions as WS-Resources). Both the Base Subscription Manager and the
1142 Pausable Subscription Manager MAY expose Subscriptions as WS-Resources.

6.1 Base Subscription Manager

1144 The basic behavior of a SubscriptionManager is to renew the duration of a Subscription resource
1145 and terminate a Subscription.

6.1.1 Renew

1147 To modify the current lifetime of a Subscription, a requestor sends a Renew request message to
1148 the SubscriptionManager. The Renew request message MUST have the following form:

1149
1150
1151
1152
1153
1154
1155
1156

```
...  
<wsnt:Renew>  
  <wsnt:TerminationTime>  
    [xsd:dateTime | xsd:duration]  
  </wsnt:TerminationTime>  
  {any} *  
</wsnt:Renew>  
...
```

1157 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1158 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewRequest>.

1159 The components of the Renew request message are further described as follows:

1160 /wsnt:Renew/wsnt:TerminationTime

1161 This component contains the service requestor's suggestion for change in expiration, or
1162 termination time of the Subscription resource. There are two forms of this component,
1163 absolute time and duration. If the type of this component is xsd:dateTime, the value of the
1164 component is to be interpreted as an "absolute time". If the type of this component is
1165 xsd:duration, the value of the component is to be interpreted as a "duration".

1166 The duration form is used to "compute" the "absolute time" form in the following fashion. The
1167 value of this component in "absolute time" form is computed by adding the xsd:duration value
1168 to the current time value of the SubscriptionManager.

1169 The "absolute time" form (whether computed from a duration, or contained within the request
1170 message) is used to reset the value of the expiration or termination time component of the
1171 Subscription resource.

1172 If the SubscriptionManager is unable or unwilling to reset the termination time component of
1173 the Subscription resource to the given value of the “absolute time” form or a value greater,
1174 then the Renew request MUST fault. If the value is not “in the future” relative to the current
1175 time as known by the SubscriptionManager, the Renew request MUST fault.

1176 The use of the xsi:nil attribute with value “true” indicates there is no scheduled termination
1177 time requested for the Subscription, implying that the Subscription should be renewed to
1178 have an infinite duration. If the element does not include the time zone designation, the value
1179 of the element MUST be interpreted as universal time (UTC) time.

1180 If the SubscriptionManager successfully processes the Renew request message, it MUST
1181 respond with a RenewResponse message. This response has the following form:

```
1182 ...  
1183 <wsnt:RenewResponse>  
1184   <wsnt:TerminationTime>xsd:dateTime</wsnt:TerminationTime>  
1185   <wsnt:CurrentTime>xsd:dateTime</wsnt:CurrentTime>?  
1186   {any} *  
1187 </wsnt:RenewResponse>  
1188 ...
```

1189 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1190 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewResponse>.

1191 The contents of the RenewResponse message are further described as follows:

1192 /wsnt:RenewResponse/wsnt:TerminationTime

1193 The value of this component is the value of the termination time component of the
1194 Subscription at the time the response message is created.

1195 /wsnt:RenewResponse/wsnt:CurrentTime

1196 This OPTIONAL component contains the value of the CurrentTime according to the
1197 SubscriptionManager at the time the response message is created.

1198 If the SubscriptionManager does not respond to the Renew request message with the
1199 RenewResponse message, then it MUST send a fault. This specification defines the following
1200 faults associated with failure to process the Renew request message:

1201 ResourceUnknownFault

1202 • The SubscriptionManager is acting as a WS-Resource, and the resource identified in the
1203 message is not known to the Web service. This fault is specified by the WS-Resource
1204 [WS-Resource] specification.

1205 UnacceptableTerminationTimeFault

1206 • The value of TerminationTime specified in the Renew message was not acceptable to the
1207 NotificationProducer. The NotificationProducer MAY include a hint in the fault message
1208 indicating acceptable values for InitialTerminationTime.

1209 **6.1.1.1 Example SOAP Encoding of the Renew Message Exchange**

1210 The following is a non-normative example of a Renew request message using SOAP:

```
1211 <s:Envelope ... >  
1212   <s:Header>  
1213     <wsa:Action>
```



```
1214 http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewRequest
1215 </wsa:Action>
1216 ...
1217 </s:Header>
1218 <s:Body>
1219 <wsnt:Renew>
1220 <wsnt:TerminationTime>PlD</wsnt:TerminationTime>
1221 </wsnt:Renew>
1222 </s:Body>
1223 </s:Envelope>
```

1224 The following is a non-normative example of a Renew response message using SOAP:

```
1225 <s:Envelope ... >
1226 <s:Header>
1227 <wsa:Action>
1228 http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewResponse
1229 </wsa:Action>
1230 ...
1231 </s:Header>
1232 <s:Body>
1233 <wsnt:RenewResponse>
1234 <wsnt:TerminationTime>
1235 2005-12-26T00:00:00.00000Z
1236 </wsnt:TerminationTime>
1237 </wsnt:RenewResponse>
1238 </s:Body>
1239 </s:Envelope>
```

1240 6.1.2 Unsubscribe

1241 To terminate a Subscription, a requestor sends an Unsubscribe request message to the
1242 SubscriptionManager. The Unsubscribe request message MUST have the following form:

```
1243 ...
1244 <wsnt:Unsubscribe>
1245 {any} *
1246 </wsnt:Unsubscribe>
1247 ...
```

1248 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1249 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeRequest>.

1250 Upon receipt of the Unsubscribe request, the SubscriptionManager MUST attempt to destroy the
1251 Subscription resource. If the SubscriptionManager successfully processes the Unsubscribe
1252 request message, it MUST respond with the following message:

```
1253 ...
1254 <wsnt:UnsubscribeResponse>
1255 {any} *
1256 </wsnt:UnsubscribeResponse>
1257 ...
```

1258 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1259 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeResponse>.

- 1260 If the SubscriptionManager does not respond to the Unsubscribe request message with the
1261 UnsubscribeResponse message, then it MUST send a fault. This specification defines the
1262 following faults associated with failure to process the Unsubscribe request message:
- 1263 ResourceUnknownFault
- 1264 • The SubscriptionManager is acting as a WS-Resource, and the resource identified in the
1265 message is not known to the Web service. This fault is specified by the WS-Resource
1266 [WS-Resource] specification.
- 1267 UnableToDestroySubscriptionFault
- 1268 • The SubscriptionManager was unable to destroy the Subscription resource for some
1269 reason.

1270 **6.1.2.1 Example SOAP Encoding of the Unsubscribe Message**

1271 **Exchange**

1272 The following is a non-normative example of an Unsubscribe request message using SOAP:

```
1273 <s:Envelope ... >  
1274   <s:Header>  
1275     <wsa:Action>  
1276     http://docs.oasis-open.org/wsn/bw-  
1277     2/SubscriptionManager/UnsubscribeRequest  
1278     </wsa:Action>  
1279     ...  
1280   </s:Header>  
1281   <s:Body>  
1282     <wsnt:Unsubscribe />  
1283   </s:Body>  
1284 </s:Envelope>
```

1285 The following is a non-normative example of an Unsubscribe response message using SOAP:

```
1286 <s:Envelope ... >  
1287   <s:Header>  
1288     <wsa:Action>  
1289     http://docs.oasis-open.org/wsn/bw-  
1290     2/SubscriptionManager/UnsubscribeResponse  
1291     </wsa:Action>  
1292     ...  
1293   </s:Header>  
1294   <s:Body>  
1295     <wsnt:UnsubscribeResponse />  
1296   </s:Body>  
1297 </s:Envelope>
```

1298 **6.2 Pausable Subscription Manager**

1299 A Pausable Subscription Manager implements all the message exchanges defined in the Base
1300 Subscription Manager and augments this with a set of advanced capabilities to allow third parties
1301 to pause and resume subscriptions.

1302 6.2.1 PauseSubscription

1303 To temporarily suspend the production of Notifications on the given Subscription, a requestor
1304 MAY send a PauseSubscription request message to the SubscriptionManager. The
1305 PauseSubscription request message has the following form:

```
1306 ...  
1307 <wsnt:PauseSubscription>{any}*</wsnt:PauseSubscription>  
1308 ...
```

1309 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1310 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/PauseSubscriptionRequest>.

1311 Upon successful processing of this message the Subscription is in the *paused* state. Production
1312 of Notifications can be resumed by sending the ResumeSubscription request message (see 6.3).

1313 In an asynchronous implementation, a NotificationProducer may continue producing notifications
1314 while the PauseSubscription request is in transit. In general, there is no way of knowing exactly
1315 when the pause will take effect.

1316 Note: transitioning between the paused state and the resumed state has no effect on the
1317 TerminationTime of the Subscription resource. A Subscription's TerminationTime may elapse
1318 while it is in a paused state, initiating the scheduled destruction of the Subscription resource.

1319 When a Subscription is resumed after having been paused, the NotificationProducer MAY choose
1320 from one of the courses of action listed below, or any other action. NotificationProducers MAY
1321 advertise their behavior in this situation via policy assertions. In the absence of a specific policy
1322 assertion, Subscribers SHOULD NOT assume any particular behavior on the part of the
1323 NotificationProducer. Possible actions include

- 1324 1. Produce Notifications for all the Situations related to the Subscription that occurred while
1325 the Subscription was paused (as well as for any new Situations that occur after the
1326 Subscription has been resumed).
- 1327 2. Produce a Notification for the last Situation that occurred while the Subscription was
1328 paused (as well as for any new situations that occur after the topic has been resumed).
- 1329 3. Produce no Notifications until a Situation occurs after the Subscription has been
1330 resumed.

1331 If the SubscriptionManager accepts the PauseSubscription request message, it MUST respond
1332 with a message of the following form:

```
1333 ...  
1334 <wsnt:PauseSubscriptionResponse>{any}*</wsnt:PauseSubscriptionResponse>  
1335 ...
```

1336 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1337 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/PauseSubscriptionResponse>.

1338 If the SubscriptionManager does not respond to the PauseSubscription request message with the
1339 PauseSubscription response message, then it MUST send a fault. This specification defines the
1340 following faults associated with failure to process the PauseSubscription request message:

1341 ResourceUnknownFault

- 1342 • The resource identified in the message is not known to the Web service. This fault is
1343 specified by the WS-Resource [WS-Resource] specification.

1344 PauseFailedFault

- 1345
- The Pause operation could not be performed on the Subscription.

1346 6.2.2 Example SOAP Encoding of the PauseSubscription Message Exchange

1347

1348 The following is a non-normative example of a PauseSubscription request message using SOAP:

```
1349 <s:Envelope ... >
1350   <s:Header>
1351     <wsa:Action>
1352       http://docs.oasis-open.org/wsn/bw-
1353 2/SubscriptionManager/PauseSubscriptionRequest
1354     </wsa:Action>
1355     ...
1356   </s:Header>
1357   <s:Body>
1358     <wsnt:PauseSubscription/>
1359   </s:Body>
1360 </s:Envelope>
```

1361 The following is a non-normative example of a PauseSubscription response message using SOAP:

1362

```
1363 <s:Envelope ... >
1364   <s:Header>
1365     <wsa:Action>
1366       http://docs.oasis-open.org/wsn/bw-
1367 2/SubscriptionManager/PauseSubscriptionResponse
1368     </wsa:Action>
1369     ...
1370   </s:Header>
1371   <s:Body>
1372     <wsnt:PauseSubscriptionResponse/>
1373   </s:Body>
1374 </s:Envelope>
```

1375 6.3 ResumeSubscription

1376 If a requestor wishes to resume the production of Notifications on the given Subscription, it must
1377 send a ResumeSubscription request message. The ResumeSubscription request message has
1378 the following form:

```
1379 ...
1380 <wsnt:ResumeSubscription>
1381   {any} *
1382 </wsnt:ResumeSubscription>
1383 ...
```

1384 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1385 `http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/ResumeSubscriptionRequest.`

1386 Upon successful processing of this message the Subscription is no longer in the *paused* state.

1387 If a Subscription is not in the *paused* state, and it is the subject of a ResumeSubscription request
1388 message, no change occurs.

1389 If the SubscriptionManager accepts the ResumeSubscription request message, it MUST respond
1390 with a message of the following form:

```
1391 ...  
1392 <wsnt:ResumeSubscriptionResponse>  
1393   {any} *  
1394 </wsnt:ResumeSubscriptionResponse>  
1395 ...
```

1396 The WS-Addressing [action] Message Addressing Property MUST contain the URI

```
1397   http://docs.oasis-open.org/wsn/bw-  
1398   2/SubscriptionManager/ResumeSubscriptionResponse.
```

1399 If the SubscriptionManager does not respond to the ResumeSubscriptionMessage request
1400 message with the ResumeSubscriptionResponse message, then it MUST send a fault. This
1401 specification defines the following faults associated with failure to process the
1402 ResumeSubscriptionMessage request message:

1403 ResourceUnknownFault

- 1404 • The resource identified in the message is not known to the Web service. This fault is
1405 specified by the WS-Resource [WS-Resource] specification.

1406 ResumeFailedFault

- 1407 • The Resume operation could not be performed on the Subscription.

1408 **6.3.1 Example SOAP Encoding of the ResumeSubscription Message** 1409 **Exchange**

1410 The following is a non-normative example of a ResumeSubscription request message using
1411 SOAP:

```
1412 <s:Envelope ... >  
1413   <s:Header>  
1414     <wsa:Action>  
1415       http://docs.oasis-open.org/wsn/bw-  
1416       2/SubscriptionManager/ResumeSubscriptionRequest  
1417     </wsa:Action>  
1418     ...  
1419   </s:Header>  
1420   <s:Body>  
1421     <wsnt:ResumeSubscription/>  
1422   </s:Body>  
1423 </s:Envelope>
```

1424 The following is a non-normative example of a ResumeSubscription response message using
1425 SOAP:

```
1426 <s:Envelope ... >  
1427   <s:Header>  
1428     <wsa:Action>  
1429       http://docs.oasis-open.org/wsn/bw-  
1430       2/SubscriptionManager/ResumeSubscriptionResponse  
1431     </wsa:Action>  
1432     ...  
1433   </s:Header>  
1434   <s:Body>
```

1435
1436
1437

```
<wsnt:ResumeSubscriptionResponse/>
</s:Body>
</s:Envelope>
```

1438

6.4 Subscriptions as WS-Resources

1439 Either Base Subscription Managers or Pausable Subscription Managers MAY expose
1440 subscriptions as WS-Resources. A Subscription Manager that exposes Subscriptions as WS-
1441 Resources MUST support the required message exchanges associated with the WS-
1442 ResourceProperties specification [WS-ResourceProperties] and MAY support the OPTIONAL
1443 message exchanges defined by WS-ResourceProperties.

1444 If a SubscriptionManager exposes Subscriptions as WS-Resources then these WS-Resources
1445 MUST support the immediate termination interface defined by Web Services Resource Lifetime
1446 [WS-ResourceLifetime] and MAY support the scheduled termination interface defined by Web
1447 Services Resource Lifetime. Such Subscription Managers MAY support other means of
1448 destroying Subscriptions

1449 A Subscription Manager MUST declare to requestors (for example in a WSDL definition) which
1450 means of Subscription destruction are supported.

1451

6.4.1 Subscription Resource Properties

1452 In cases where subscriptions are exposed as WS-Resources, this specification defines several
1453 resource properties, which MUST conform to the following schema fragment for content:.

1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465

```
...
  targetNamespace="http://docs.oasis-open.org/wsn/b-2"
...
<xsd:element name="ConsumerReference"
  type="wsa:EndpointReferenceType" />
<xsd:element name="Filter"
  type="wsnt:FilterType" />
<xsd:element name="SubscriptionPolicy"
  type="wsnt:SubscriptionPolicyType" />
<xsd:element name="CreationTime"
  type="xsd:dateTime" />
```

1466

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

1467
1468
1469
1470
1471
1472
1473
1474

```
<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" />
```

1475

These Resource Property elements are further described as follows:

1476

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

1477

These elements are as defined in the description of the Subscribe request message (see

1478

4.2). SubscriptionManagers MAY prohibit any combination of these properties from being set.

1479 If setting a given property is disallowed, the SubscriptionManager MUST fault on receiving a
1480 request to change it. SubscriptionManagers SHOULD describe using metadata techniques
1481 defined by WS-ResourceFramework which properties may be set.

1482 If any of these properties is settable, it is implementation dependent whether production of
1483 notifications will be attempted without interruption when a change is made. The
1484 SubscriptionManager may choose to treat the change in properties as equivalent to a
1485 cancellation followed by the creation of a new subscription. In this case, messages may
1486 potentially be lost in the interim.

1487 A SubscriptionManager MAY choose to provide a stronger guarantee of continuity of
1488 notification production, and if so it SHOULD advertise this capability, for example in a policy
1489 assertion. In the absence of any such policy assertion, a subscriber SHOULD NOT make
1490 any assumption of continuity.

1491 /wsnt:CreationTime

1492 Indicates the date and time at which the Subscription was created. This component MAY be
1493 omitted, for example by resource constrained devices which cannot associate a creation time
1494 with subscription resources they create. This property is read-only. SubscriptionManagers
1495 MUST fault on receiving a request to change it.

1496 If the SubscriptionManager also accepts the SetResourceProperties request message as defined
1497 in WS-ResourceProperties, the following properties MAY be modified by the requestor:
1498 /wsnt:ConsumerReference and /wsnt:Filter and /wsnt:SubscriptionPolicy

1499 **7 Security Considerations**

1500 This section deals with the security aspects of WS-BaseNotification. It deals with (a) securing the
1501 message exchanges defined in this specification, and (b) authorization and denial of service
1502 considerations specific to the Notification pattern.

1503 **7.1 Securing the Message Exchanges**

1504 In the Notification pattern, Notifications are sent to a NotificationConsumer, Subscribers
1505 exchange SubscriptionRequest and SubscriptionResponse messages with NotificationProducers,
1506 and any party with access to a SubscriptionManager endpoint may perform operations on the
1507 underlying Subscription resources. In cases where this communication must be secured it is
1508 RECOMMENDED that this be done using the mechanisms described in WS-Security.

1509 Communication between a NotificationProducer and NotificationConsumer will typically comprise
1510 a number of Notification messages. In cases where this communication must be secure, it is
1511 RECOMMENDED that a security context be established, allowing for potentially more efficient
1512 means of authentication. Note that the keys used to secure this channel may differ from any keys
1513 used in the Subscribe request/response operation that created the associated Subscription.

1514 **7.2 Securing Subscriptions and Notifications**

1515 As WS-BaseNotification provides mechanisms for subscribing to topics, security policies should
1516 be established such that

- 1517 1. only authorized principals can subscribe to receive Notifications
- 1518 2. only authorized principals can modify or delete Subscriptions

1519 It is recommended that the authorization policies be specified at the granularity of the Topic, if
1520 Topics are supported. It should be noted that even though Subscriptions may be done by
1521 authorized principals, the Notifications may be produced for NotificationConsumers whose
1522 identity may be different from the Subscriber. Message protection policies as outlined in the
1523 previous section can be used to ensure that sensitive Notifications are not delivered to malicious
1524 endpoints. For example, a key may need to be specified or generated during the process of
1525 Subscription, so that the Notifications can be encrypted using the key to ensure confidentiality of
1526 the messages. The mechanism by which the key is specified is governed by the Subscription
1527 policy.

1528 Given that WS-BaseNotification may use WS-ResourceProperties and WS-ResourceLifetime, the
1529 security considerations outlined in WS- those specifications need to be taken into account where
1530 appropriate. Authorization policies for those Resource Properties should be put in place so that
1531 the implications of providing the state information (through GetResourceProperty request
1532 messages) or through notification of state change and modification of the resource properties
1533 (through SetResourceProperty request messages), are taken into account.

1534 This specification provides a mechanism by which Subscribers can specify a subscription policy.
1535 Such a policy may contain security policy about protecting the message exchanges resulting from
1536 the Subscription. Security policy for Subscription message exchanges needs to take this into
1537 consideration so that the Subscription policies are protected. Also, given this policy may be
1538 contained in the resource properties of the subscription maintained by the SubscriptionManager,
1539 the resource properties must be appropriately secured.

1540 In addition to the usual concerns of authorization and message integrity which apply to all web
1541 services, notification presents issues all its own due to the third-party nature of subscription.
1542 Since the NotificationProducer is agreeing to produce Notifications for a consumer based on the
1543 requests of a Subscriber, it must assure itself that there is no harm in producing these
1544 Notifications. A malicious Subscriber may request Notifications be sent to a party that is not
1545 authorized to receive them. It may also mount DOS attacks by requesting large volumes of
1546 Notifications be sent to parties that cannot handle them.

1547
1548 The NotificationProducer may address these risks in many different ways, including but not
1549 limited to:

- 1550 • Simply trusting all Subscribers, perhaps because all parties are known to be on a closed,
1551 trusted, network, or because the consequences of unauthorized Subscriptions are
1552 otherwise known to be negligible.
- 1553 • Requiring all Subscribers to provide secure credentials proving that they are trusted to
1554 make subscriptions.
- 1555 • Refusing to produce notifications for NotificationConsumers that are not known to be
1556 authorized.
- 1557 • Explicitly confirming with NotificationConsumers whether they wish to receive the
1558 Notifications that the Subscriber has requested.
- 1559 • Some combination of the above, depending on the identity of the Subscriber and
1560 NotificationProducer

1561 NotificationProducers SHOULD advertise, whether through policy assertions or other means,
1562 what security measures they take.

1563

8 References

1564

8.1 Normative

1565

[RFC2119]

1566

S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", IETF RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>

1567

1568

[URI]

1569

T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998. <http://www.ietf.org/rfc/rfc2396.txt>

1570

1571

1572

[WS-Addressing]

1573

"Web Services Addressing 1.0 - Core", W3C Recommendation.

1574

<http://www.w3.org/TR/ws-addr-core>

1575

[WS-BrokeredNotification]

1576

"Web Services Brokered Notification 1.3".

1577

http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-cs-01.pdf

1578

[WS-Resource]

1579

"Web Services Resource 1.2", OASIS Standard.

1580

http://docs.oasis-open.org/wsr/wsr-ws_resource-1.2-spec-os.pdf

1581

[WS-ResourceLifetime]

1582

"Web Services Resource Lifetime 1.2", OASIS Standard.

1583

http://docs.oasis-open.org/wsr/wsr-ws_resource_lifetime-1.2-spec-os.pdf

1584

[WS-ResourceProperties]

1585

"Web Services Resource Properties 1.2", OASIS Standard.

1586

http://docs.oasis-open.org/wsr/wsr-ws_resource_properties-1.2-spec-os.pdf

1587

[WS-BaseFaults]

1588

"Web Services Base Faults 1.2", OASIS Standard.

1589

http://docs.oasis-open.org/wsr/wsr-ws_base_faults-1.2-spec-os.pdf

1590

[WS-Topics]

1591

"Web Services Topics 1.3".

1592

http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-cs-01.pdf

1593

[XML-Infoset]

1594

"XML Information Set", W3C Recommendation.

1595

<http://www.w3.org/TR/xml-infoset/>

1596

[XPath]

1597

"XML Path Language (XPath) Version 1.0", W3C Recommendation.

1598

<http://www.w3.org/TR/xpath>

1599

8.2 Non-Normative

1600

[SOAP 1.1]

1601

"Simple Object Access Protocol (SOAP) 1.1"

1602

<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

1603
1604
1605
1606
1607
1608
1609

[SOAP 1.2]

“SOAP Version 1.2 Part 1: Messaging Framework”, W3C Recommendation.
<http://www.w3.org/TR/soap12-part1/>

[WS-Security]

“Web Services Security: SOAP Message Security 1.0”, OASIS Standard.
<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

1610

Appendix A. Acknowledgments

1611 Special thanks to the Global Grid Forum's Open Grid Services Infrastructure working group,
1612 which defined the OGSI v1.0 [OGSI] specification which was a large inspiration for the ideas
1613 expressed in this specification.

1614 The following individuals were members of the committee during the development of this
1615 specification:

1616

1617 Sid Askary, Fred Carter (AmberPoint), Martin Chapman (Oracle), Dave Chappell (Sonic
1618 Software), Rick Cobb (KnowNow), Ugo Corda (SeeBeyond Technology Corporation), John Fuller,
1619 Stephen Graham (IBM), David Hull (Tibco), Hideharu Kato (Hitachi), Lily Liu (webMethods, Inc.),
1620 Tom Maguire (IBM), Susan Malaika (IBM), Samuel Meder (Argonne National Laboratory), Bryan
1621 Murray (Hewlett-Packard), Peter Niblett (IBM), Sanjay Patil (SAP), Mark Peel (Novell), Matt
1622 Roberts (IBM), Igor Sedukhin (Computer Associates), David Snelling (Fujitsu), Latha Srinivasan
1623 (Hewlett-Packard), William Vambenepe (Hewlett-Packard) and Kirk Wilson (Computer
1624 Associates).

1625

1626 In addition, the following people made contributions to this specification:

1627 Tim Banks (IBM), Nick Butler (IBM), Doug Davis (IBM), John Dinger (IBM), Don Ferguson (IBM),
1628 Jeff Frey (IBM), Andreas Koeppel (SAP), Heather Kreger (IBM), Amy Lewis (TIBCO Software),
1629 Kevin Liu (SAP), Nataraj Nagaratnam (IBM), Martin Nally (IBM), Jeff Nick (IBM), Jay Parikh
1630 (Akamai Technologies), Claus von Riegen (SAP), Rick Rineholt (IBM), John Rofrano (IBM),
1631 Shivajee Samdarshi (TIBCO Software), Eugène Sindambiwe (SAP), Jay Unger (IBM), Bill Weihl
1632 (Akamai Technologies), Mark Weitzel (IBM), and Dan Wolfson (IBM).

1633

Appendix B. XML Schema

1634

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

1635

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

1636

```
<!--
```

1637

1638

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

1640

1641

1642

1643

1644

1645

1646

1647

1648

1649

1650

1651

1652

1653

1654

1655

1656

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

1657

Copyright (C) OASIS Open (2004-2006). All Rights Reserved.

1658

1659

1660

1661

1662

1663

1664

1665

1666

1667

1668

1669

1670

1671

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

1672

1673

1674

1675

1676

1677

1678

1679

1680

1681

1682

1683

1684

```
-->
```

```
<xsd:schema
```

```
targetNamespace="http://docs.oasis-open.org/wsn/b-2"
```

```

1685   xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
1686   xmlns:wsa="http://www.w3.org/2005/08/addressing"
1687   xmlns:wsrfrf-bf="http://docs.oasis-open.org/wsrfrf/bf-2"
1688   xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"
1689   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1690   elementFormDefault="qualified" attributeFormDefault="unqualified">
1691
1692   <!-- ===== Imports ===== -->
1693
1694   <xsd:import namespace="http://www.w3.org/2005/08/addressing"
1695             schemaLocation="http://www.w3.org/2005/08/addressing/ws-
1696   addr.xsd"
1697             />
1698
1699   <xsd:import namespace="http://docs.oasis-open.org/wsrfrf/bf-2"
1700             schemaLocation="http://docs.oasis-open.org/wsrfrf/bf-2.xsd"
1701             />
1702   <xsd:import namespace="http://docs.oasis-open.org/wsn/t-1"
1703             schemaLocation="http://docs.oasis-open.org/wsn/t-1.xsd"
1704             />
1705
1706   <!-- ===== Misc. Helper Types ===== -->
1707
1708   <xsd:complexType name="QueryExpressionType" mixed="true">
1709     <xsd:sequence>
1710       <xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
1711     </xsd:sequence>
1712     <xsd:attribute name="Dialect" type="xsd:anyURI" use="required"/>
1713   </xsd:complexType>
1714
1715   <xsd:complexType name="TopicExpressionType" mixed="true">
1716     <xsd:sequence>
1717       <xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
1718     </xsd:sequence>
1719     <xsd:attribute name="Dialect" type="xsd:anyURI" use="required" />
1720     <xsd:anyAttribute/>
1721   </xsd:complexType>
1722
1723   <xsd:complexType name="FilterType">
1724     <xsd:sequence>
1725       <xsd:any minOccurs="0" maxOccurs="unbounded" />
1726     </xsd:sequence>
1727   </xsd:complexType>
1728
1729   <xsd:complexType name="SubscriptionPolicyType">
1730     <xsd:sequence>
1731       <xsd:any minOccurs="0" maxOccurs="unbounded"
1732   processContents="lax"/>
1733     </xsd:sequence>
1734   </xsd:complexType>
1735
1736   <!-- ===== Resource Property Related ===== -->
1737   <!-- ===== Resource Properties for NotificationProducer ===== -->
1738   <xsd:element name="TopicExpression" type="wsnt:TopicExpressionType"/>
1739   <xsd:element name="FixedTopicSet" type="xsd:boolean" default="true"/>
1740   <xsd:element name="TopicExpressionDialect" type="xsd:anyURI"/>
1741

```

```

1742 <xsd:element name="NotificationProducerRP">
1743   <xsd:complexType>
1744     <xsd:sequence>
1745       <xsd:element ref="wsnt:TopicExpression"
1746         minOccurs="0" maxOccurs="unbounded" />
1747       <xsd:element ref="wsnt:FixedTopicSet"
1748         minOccurs="0" maxOccurs="1" />
1749       <xsd:element ref="wsnt:TopicExpressionDialect"
1750         minOccurs="0" maxOccurs="unbounded" />
1751       <xsd:element ref="wstop:TopicSet"
1752         minOccurs="0" maxOccurs="1" />
1753     </xsd:sequence>
1754   </xsd:complexType>
1755 </xsd:element>
1756
1757 <!-- ===== Resource Properties for SubscriptionManager ===== -->
1758 <xsd:element name="ConsumerReference"
1759   type="wsa:EndpointReferenceType" />
1760 <xsd:element name="Filter" type="wsnt:FilterType" />
1761 <xsd:element name="SubscriptionPolicy"
1762   type="wsnt:SubscriptionPolicyType" />
1763
1764
1765 <xsd:element name="CreationTime" type="xsd:dateTime" />
1766
1767 <xsd:element name="SubscriptionManagerRP" >
1768   <xsd:complexType>
1769     <xsd:sequence>
1770       <xsd:element ref="wsnt:ConsumerReference"
1771         minOccurs="1" maxOccurs="1" />
1772       <xsd:element ref="wsnt:Filter"
1773         minOccurs="0" maxOccurs="1" />
1774       <xsd:element ref="wsnt:SubscriptionPolicy"
1775         minOccurs="0" maxOccurs="1" />
1776       <xsd:element ref="wsnt:CreationTime"
1777         minOccurs="0" maxOccurs="1" />
1778     </xsd:sequence>
1779   </xsd:complexType>
1780 </xsd:element>
1781
1782 <!-- ===== Notification Metadata ===== -->
1783 <xsd:element name="SubscriptionReference"
1784   type="wsa:EndpointReferenceType" />
1785 <xsd:element name="Topic"
1786   type="wsnt:TopicExpressionType" />
1787 <xsd:element name="ProducerReference"
1788   type="wsa:EndpointReferenceType" />
1789
1790 <!-- ===== Message Helper Types ===== -->
1791 <xsd:complexType name="NotificationMessageHolderType" >
1792   <xsd:sequence>
1793     <xsd:element ref="wsnt:SubscriptionReference"
1794       minOccurs="0" maxOccurs="1" />
1795     <xsd:element ref="wsnt:Topic"
1796       minOccurs="0" maxOccurs="1" />
1797     <xsd:element ref="wsnt:ProducerReference"
1798       minOccurs="0" maxOccurs="1" />

```

```

1799     <xsd:element name="Message">
1800         <xsd:complexType>
1801             <xsd:sequence>
1802                 <xsd:any namespace="##any" processContents="lax"
1803                     minOccurs="1" maxOccurs="1"/>
1804             </xsd:sequence>
1805         </xsd:complexType>
1806     </xsd:element>
1807 </xsd:sequence>
1808 </xsd:complexType>
1809 <xsd:element name="NotificationMessage"
1810     type="wsnt:NotificationMessageHolderType"/>
1811
1812 <!-- ===== Message Types for NotificationConsumer ===== -->
1813 <xsd:element name="Notify" >
1814     <xsd:complexType>
1815         <xsd:sequence>
1816             <xsd:element ref="wsnt:NotificationMessage"
1817                 minOccurs="1" maxOccurs="unbounded" />
1818             <xsd:any namespace="##other" processContents="lax"
1819                 minOccurs="0" maxOccurs="unbounded"/>
1820         </xsd:sequence>
1821     </xsd:complexType>
1822 </xsd:element>
1823
1824 <!-- ===== Message Types for NotificationProducer ===== -->
1825
1826 <xsd:simpleType name="AbsoluteOrRelativeTimeType">
1827     <xsd:union memberTypes="xsd:dateTime xsd:duration" />
1828 </xsd:simpleType>
1829
1830 <xsd:element name="CurrentTime" type="xsd:dateTime" />
1831
1832 <xsd:element name="TerminationTime"
1833     nillable="true" type="xsd:dateTime" />
1834
1835 <xsd:element name="ProducerProperties"
1836     type="wsnt:QueryExpressionType" />
1837
1838 <xsd:element name="MessageContent"
1839     type="wsnt:QueryExpressionType" />
1840
1841 <xsd:element name="UseRaw"><xsd:complexType/></xsd:element>
1842
1843 <xsd:element name="Subscribe" >
1844     <xsd:complexType>
1845         <xsd:sequence>
1846             <xsd:element name="ConsumerReference"
1847                 type="wsa:EndpointReferenceType"
1848                 minOccurs="1" maxOccurs="1" />
1849             <xsd:element name="Filter"
1850                 type="wsnt:FilterType"
1851                 minOccurs="0" maxOccurs="1" />
1852             <xsd:element name="InitialTerminationTime"
1853                 type="wsnt:AbsoluteOrRelativeTimeType"
1854                 nillable="true"
1855                 minOccurs="0" maxOccurs="1" />

```



```

1856     <xsd:element name="SubscriptionPolicy"
1857         minOccurs="0" maxOccurs="1">
1858         <xsd:complexType>
1859             <xsd:sequence>
1860                 <xsd:any namespace="##any" processContents="lax"
1861                     minOccurs="0" maxOccurs="unbounded" />
1862             </xsd:sequence>
1863         </xsd:complexType>
1864     </xsd:element>
1865     <xsd:any namespace="##other" processContents="lax"
1866         minOccurs="0" maxOccurs="unbounded" />
1867 </xsd:sequence>
1868 </xsd:complexType>
1869 </xsd:element>
1870
1871 <xsd:element name="SubscribeResponse">
1872     <xsd:complexType>
1873         <xsd:sequence>
1874             <xsd:element name="SubscriptionReference"
1875                 type="wsa:EndpointReferenceType"
1876                 minOccurs="1" maxOccurs="1" />
1877             <xsd:element ref="wsnt:CurrentTime"
1878                 minOccurs="0" maxOccurs="1" />
1879             <xsd:element ref="wsnt:TerminationTime"
1880                 minOccurs="0" maxOccurs="1" />
1881             <xsd:any namespace="##other" processContents="lax"
1882                 minOccurs="0" maxOccurs="unbounded" />
1883         </xsd:sequence>
1884     </xsd:complexType>
1885 </xsd:element>
1886
1887 <xsd:element name="GetCurrentMessage">
1888     <xsd:complexType>
1889         <xsd:sequence>
1890             <xsd:element name="Topic"
1891                 type="wsnt:TopicExpressionType" />
1892             <xsd:any namespace="##other" processContents="lax"
1893                 minOccurs="0" maxOccurs="unbounded" />
1894         </xsd:sequence>
1895     </xsd:complexType>
1896 </xsd:element>
1897
1898 <xsd:element name="GetCurrentMessageResponse">
1899     <xsd:complexType>
1900         <xsd:sequence>
1901             <xsd:any namespace="##other" processContents="lax"
1902                 minOccurs="0" maxOccurs="unbounded" />
1903         </xsd:sequence>
1904     </xsd:complexType>
1905 </xsd:element>
1906
1907 <xsd:complexType name="SubscribeCreationFailedFaultType">
1908     <xsd:complexContent>
1909         <xsd:extension base="wsrf-bf:BaseFaultType" />
1910     </xsd:complexContent>
1911 </xsd:complexType>
1912 <xsd:element name="SubscribeCreationFailedFault"

```

```

1913         type="wsnt:SubscribeCreationFailedFaultType" />
1914
1915 <xsd:complexType name="InvalidFilterFaultType">
1916   <xsd:complexContent>
1917     <xsd:extension base="wsrf-bf:BaseFaultType">
1918       <xsd:sequence>
1919         <xsd:element name="UnknownFilter" type="xsd:QName"
1920           minOccurs="1" maxOccurs="unbounded" />
1921       </xsd:sequence>
1922     </xsd:extension>
1923   </xsd:complexContent>
1924 </xsd:complexType>
1925 <xsd:element name="InvalidFilterFault"
1926   type="wsnt:InvalidFilterFaultType" />
1927
1928 <xsd:complexType name="TopicExpressionDialectUnknownFaultType">
1929   <xsd:complexContent>
1930     <xsd:extension base="wsrf-bf:BaseFaultType" />
1931   </xsd:complexContent>
1932 </xsd:complexType>
1933 <xsd:element name="TopicExpressionDialectUnknownFault"
1934   type="wsnt:TopicExpressionDialectUnknownFaultType" />
1935
1936 <xsd:complexType name="InvalidTopicExpressionFaultType">
1937   <xsd:complexContent>
1938     <xsd:extension base="wsrf-bf:BaseFaultType" />
1939   </xsd:complexContent>
1940 </xsd:complexType>
1941 <xsd:element name="InvalidTopicExpressionFault"
1942   type="wsnt:InvalidTopicExpressionFaultType" />
1943
1944 <xsd:complexType name="TopicNotSupportedFaultType">
1945   <xsd:complexContent>
1946     <xsd:extension base="wsrf-bf:BaseFaultType" />
1947   </xsd:complexContent>
1948 </xsd:complexType>
1949 <xsd:element name="TopicNotSupportedFault"
1950   type="wsnt:TopicNotSupportedFaultType" />
1951
1952 <xsd:complexType name="MultipleTopicsSpecifiedFaultType">
1953   <xsd:complexContent>
1954     <xsd:extension base="wsrf-bf:BaseFaultType" />
1955   </xsd:complexContent>
1956 </xsd:complexType>
1957 <xsd:element name="MultipleTopicsSpecifiedFault"
1958   type="wsnt:MultipleTopicsSpecifiedFaultType" />
1959
1960 <xsd:complexType name="InvalidProducerPropertiesExpressionFaultType">
1961   <xsd:complexContent>
1962     <xsd:extension base="wsrf-bf:BaseFaultType" />
1963   </xsd:complexContent>
1964 </xsd:complexType>
1965 <xsd:element name="InvalidProducerPropertiesExpressionFault"
1966   type="wsnt:InvalidProducerPropertiesExpressionFaultType" />
1967
1968 <xsd:complexType name="InvalidMessageContentExpressionFaultType">
1969   <xsd:complexContent>

```

1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026

```
<xsd:extension base="wsrf-bf:BaseFaultType" />
</xsd:complexContent>
</xsd:complexType>
<xsd:element name="InvalidMessageContentExpressionFault"
  type="wsnt:InvalidMessageContentExpressionFaultType" />
<xsd:complexType name="UnrecognizedPolicyRequestFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType">
      <xsd:sequence>
        <xsd:element name="UnrecognizedPolicy" type="xsd:QName"
          minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="UnrecognizedPolicyRequestFault"
  type="wsnt:UnrecognizedPolicyRequestFaultType" />
<xsd:complexType name="UnsupportedPolicyRequestFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType">
      <xsd:sequence>
        <xsd:element name="UnsupportedPolicy" type="xsd:QName"
          minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="UnsupportedPolicyRequestFault"
  type="wsnt:UnsupportedPolicyRequestFaultType" />
<xsd:complexType name="NotifyMessageNotSupportedFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType" />
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="NotifyMessageNotSupportedFault"
  type="wsnt:NotifyMessageNotSupportedFaultType" />
<xsd:complexType name="UnacceptableInitialTerminationTimeFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType">
      <xsd:sequence>
        <xsd:element name="MinimumTime" type="xsd:dateTime" />
        <xsd:element name="MaximumTime" type="xsd:dateTime"
          minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="UnacceptableInitialTerminationTimeFault"
  type="wsnt:UnacceptableInitialTerminationTimeFaultType" />
<xsd:complexType name="NoCurrentMessageOnTopicFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType" />
```

```

2027     </xsd:complexContent>
2028 </xsd:complexType>
2029 <xsd:element name="NoCurrentMessageOnTopicFault"
2030     type="wsnt:NoCurrentMessageOnTopicFaultType" />
2031
2032 <!-- ===== Message Types for PullPoint ===== -->
2033 <xsd:element name="GetMessages">
2034     <xsd:complexType>
2035         <xsd:sequence>
2036             <xsd:element name="MaximumNumber"
2037                 type="xsd:nonNegativeInteger"
2038                 minOccurs="0" />
2039             <xsd:any namespace="##other" processContents="lax"
2040                 minOccurs="0" maxOccurs="unbounded" />
2041         </xsd:sequence>
2042         <xsd:anyAttribute/>
2043     </xsd:complexType>
2044 </xsd:element>
2045
2046 <xsd:element name="GetMessagesResponse">
2047     <xsd:complexType>
2048         <xsd:sequence>
2049             <xsd:element ref="wsnt:NotificationMessage"
2050                 minOccurs="0" maxOccurs="unbounded" />
2051             <xsd:any namespace="##other" processContents="lax"
2052                 minOccurs="0" maxOccurs="unbounded" />
2053         </xsd:sequence>
2054         <xsd:anyAttribute/>
2055     </xsd:complexType>
2056 </xsd:element>
2057
2058 <xsd:element name="DestroyPullPoint">
2059     <xsd:complexType>
2060         <xsd:sequence>
2061             <xsd:any namespace="##other" processContents="lax"
2062                 minOccurs="0" maxOccurs="unbounded" />
2063         </xsd:sequence>
2064         <xsd:anyAttribute/>
2065     </xsd:complexType>
2066 </xsd:element>
2067
2068 <xsd:element name="DestroyPullPointResponse">
2069     <xsd:complexType>
2070         <xsd:sequence>
2071             <xsd:any namespace="##other" processContents="lax"
2072                 minOccurs="0" maxOccurs="unbounded" />
2073         </xsd:sequence>
2074         <xsd:anyAttribute/>
2075     </xsd:complexType>
2076 </xsd:element>
2077
2078 <xsd:complexType name="UnableToGetMessagesFaultType">
2079     <xsd:complexContent>
2080         <xsd:extension base="wsrf-bf:BaseFaultType" />
2081     </xsd:complexContent>
2082 </xsd:complexType>
2083

```

```

2084 <xsd:element name="UnableToGetMessagesFault"
2085         type="wsnt:UnableToGetMessagesFaultType" />
2086
2087 <xsd:complexType name="UnableToDestroyPullPointFaultType">
2088     <xsd:complexContent>
2089         <xsd:extension base="wsrf-bf:BaseFaultType" />
2090     </xsd:complexContent>
2091 </xsd:complexType>
2092
2093 <xsd:element name="UnableToDestroyPullPointFault"
2094         type="wsnt:UnableToDestroyPullPointFaultType" />
2095
2096 <!-- ===== Message Types for Create PullPoint ===== -->
2097 <xsd:element name="CreatePullPoint">
2098     <xsd:complexType>
2099         <xsd:sequence>
2100             <xsd:any namespace="##other" processContents="lax"
2101                 minOccurs="0" maxOccurs="unbounded" />
2102         </xsd:sequence>
2103         <xsd:anyAttribute />
2104     </xsd:complexType>
2105 </xsd:element>
2106
2107 <xsd:element name="CreatePullPointResponse">
2108     <xsd:complexType>
2109         <xsd:sequence>
2110             <xsd:element name="PullPoint"
2111                 type="wsa:EndpointReferenceType" />
2112             <xsd:any namespace="##other" processContents="lax"
2113                 minOccurs="0" maxOccurs="unbounded" />
2114         </xsd:sequence>
2115         <xsd:anyAttribute />
2116     </xsd:complexType>
2117 </xsd:element>
2118
2119 <xsd:complexType name="UnableToCreatePullPointFaultType">
2120     <xsd:complexContent>
2121         <xsd:extension base="wsrf-bf:BaseFaultType" />
2122     </xsd:complexContent>
2123 </xsd:complexType>
2124 <xsd:element name="UnableToCreatePullPointFault"
2125         type="wsnt:UnableToCreatePullPointFaultType" />
2126
2127 <!-- ===== Message Types for Base SubscriptionManager ===== -->
2128 <xsd:element name="Renew">
2129     <xsd:complexType>
2130         <xsd:sequence>
2131             <xsd:element name="TerminationTime"
2132                 type="wsnt:AbsoluteOrRelativeTimeType"
2133                 nillable="true"
2134                 minOccurs="1" maxOccurs="1" />
2135             <xsd:any namespace="##other" processContents="lax"
2136                 minOccurs="0" maxOccurs="unbounded" />
2137         </xsd:sequence>
2138     </xsd:complexType>
2139 </xsd:element>
2140

```

```

2141 <xsd:element name="RenewResponse">
2142   <xsd:complexType>
2143     <xsd:sequence>
2144       <xsd:element ref="wsnt:TerminationTime"
2145         minOccurs="1" maxOccurs="1" />
2146       <xsd:element ref="wsnt:CurrentTime"
2147         minOccurs="0" maxOccurs="1" />
2148       <xsd:any namespace="##other" processContents="lax"
2149         minOccurs="0" maxOccurs="unbounded" />
2150     </xsd:sequence>
2151   </xsd:complexType>
2152 </xsd:element>
2153
2154 <xsd:complexType name="UnacceptableTerminationTimeFaultType">
2155   <xsd:complexContent>
2156     <xsd:extension base="wsrf-bf:BaseFaultType">
2157       <xsd:sequence>
2158         <xsd:element name="MinimumTime" type="xsd:dateTime"/>
2159         <xsd:element name="MaximumTime" type="xsd:dateTime"
2160           minOccurs="0"/>
2161       </xsd:sequence>
2162     </xsd:extension>
2163   </xsd:complexContent>
2164 </xsd:complexType>
2165 <xsd:element name="UnacceptableTerminationTimeFault"
2166   type="wsnt:UnacceptableTerminationTimeFaultType"/>
2167
2168 <xsd:element name="Unsubscribe">
2169   <xsd:complexType>
2170     <xsd:sequence>
2171       <xsd:any namespace="##other" processContents="lax"
2172         minOccurs="0" maxOccurs="unbounded" />
2173     </xsd:sequence>
2174   </xsd:complexType>
2175 </xsd:element>
2176
2177 <xsd:element name="UnsubscribeResponse">
2178   <xsd:complexType>
2179     <xsd:sequence>
2180       <xsd:any namespace="##other" processContents="lax"
2181         minOccurs="0" maxOccurs="unbounded" />
2182     </xsd:sequence>
2183   </xsd:complexType>
2184 </xsd:element>
2185
2186 <xsd:complexType name="UnableToDestroySubscriptionFaultType">
2187   <xsd:complexContent>
2188     <xsd:extension base="wsrf-bf:BaseFaultType" />
2189   </xsd:complexContent>
2190 </xsd:complexType>
2191 <xsd:element name="UnableToDestroySubscriptionFault"
2192   type="wsnt:UnableToDestroySubscriptionFaultType"/>
2193
2194 <!-- ===== Message Types for Pausable SubscriptionManager ===== -->
2195
2196 <xsd:element name="PauseSubscription">
2197   <xsd:complexType>

```

```

2198     <xsd:sequence>
2199         <xsd:any namespace="##other" processContents="lax"
2200             minOccurs="0" maxOccurs="unbounded" />
2201     </xsd:sequence>
2202 </xsd:complexType>
2203 </xsd:element>
2204
2205 <xsd:element name="PauseSubscriptionResponse" >
2206     <xsd:complexType>
2207         <xsd:sequence>
2208             <xsd:any namespace="##other" processContents="lax"
2209                 minOccurs="0" maxOccurs="unbounded" />
2210         </xsd:sequence>
2211     </xsd:complexType>
2212 </xsd:element>
2213
2214 <xsd:element name="ResumeSubscription">
2215     <xsd:complexType>
2216         <xsd:sequence>
2217             <xsd:any namespace="##other" processContents="lax"
2218                 minOccurs="0" maxOccurs="unbounded" />
2219         </xsd:sequence>
2220     </xsd:complexType>
2221 </xsd:element>
2222
2223 <xsd:element name="ResumeSubscriptionResponse">
2224     <xsd:complexType>
2225         <xsd:sequence>
2226             <xsd:any namespace="##other" processContents="lax"
2227                 minOccurs="0" maxOccurs="unbounded" />
2228         </xsd:sequence>
2229     </xsd:complexType>
2230 </xsd:element>
2231
2232 <xsd:complexType name="PauseFailedFaultType">
2233     <xsd:complexContent>
2234         <xsd:extension base="wsrf-bf:BaseFaultType" />
2235     </xsd:complexContent>
2236 </xsd:complexType>
2237 <xsd:element name="PauseFailedFault"
2238     type="wsnt:PauseFailedFaultType" />
2239
2240 <xsd:complexType name="ResumeFailedFaultType">
2241     <xsd:complexContent>
2242         <xsd:extension base="wsrf-bf:BaseFaultType" />
2243     </xsd:complexContent>
2244 </xsd:complexType>
2245 <xsd:element name="ResumeFailedFault"
2246     type="wsnt:ResumeFailedFaultType" />
2247
2248 </xsd:schema>

```

2249

Appendix C. WSDL 1.1

2250 The following illustrates the WSDL 1.1 for the Web service methods described in this
2251 specification:

```
2252 <?xml version="1.0" encoding="utf-8"?>
2253 <!--
2254
2255 OASIS takes no position regarding the validity or scope of any
2256 intellectual property or other rights that might be claimed to pertain
2257 to the implementation or use of the technology described in this
2258 document or the extent to which any license under such rights might or
2259 might not be available; neither does it represent that it has made any
2260 effort to identify any such rights. Information on OASIS's procedures
2261 with respect to rights in OASIS specifications can be found at the
2262 OASIS website. Copies of claims of rights made available for
2263 publication and any assurances of licenses to be made available, or the
2264 result of an attempt made to obtain a general license or permission for
2265 the use of such proprietary rights by implementors or users of this
2266 specification, can be obtained from the OASIS Executive Director.
2267
2268 OASIS invites any interested party to bring to its attention any
2269 copyrights, patents or patent applications, or other proprietary rights
2270 which may cover technology that may be required to implement this
2271 specification. Please address the information to the OASIS Executive
2272 Director.
2273
2274 Copyright (C) OASIS Open (2004-2006). All Rights Reserved.
2275
2276 This document and translations of it may be copied and furnished to
2277 others, and derivative works that comment on or otherwise explain it or
2278 assist in its implementation may be prepared, copied, published and
2279 distributed, in whole or in part, without restriction of any kind,
2280 provided that the above copyright notice and this paragraph are
2281 included on all such copies and derivative works. However, this
2282 document itself may not be modified in any way, such as by removing the
2283 copyright notice or references to OASIS, except as needed for the
2284 purpose of developing OASIS specifications, in which case the
2285 procedures for copyrights defined in the OASIS Intellectual Property
2286 Rights document must be followed, or as required to translate it into
2287 languages other than English.
2288
2289 The limited permissions granted above are perpetual and will not be
2290 revoked by OASIS or its successors or assigns.
2291
2292 This document and the information contained herein is provided on an
2293 "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
2294 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
2295 INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
2296 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
2297
2298 -->
2299 <wsdl:definitions name="WS-BaseNotification"
2300 targetNamespace="http://docs.oasis-open.org/wsn/bw-2"
```

wsn-ws_base_notification-1.3-spec-cs-01

7/31/2006

Copyright © OASIS Open 2004-2006. All Rights Reserved.

Page 56 of 68


```

2301   xmlns:wsntw="http://docs.oasis-open.org/wsn/bw-2"
2302   xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
2303   xmlns:wsa="http://www.w3.org/2005/08/addressing"
2304   xmlns:wsrfrw="http://docs.oasis-open.org/wsrfrw-2"
2305   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2306   xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/">
2307
2308   <!-- ===== Imports ===== -->
2309   <wSDL:import
2310       namespace="http://docs.oasis-open.org/wsrfrw-2"
2311       location="http://docs.oasis-open.org/wsrfrw-2.wSDL"/>
2312
2313   <!-- ===== Types Definitions ===== -->
2314   <wSDL:types>
2315     <xsd:schema>
2316       <xsd:import
2317         namespace="http://docs.oasis-open.org/wsn/b-2"
2318         schemaLocation="http://docs.oasis-open.org/wsn/b-2.xsd"/>
2319     </xsd:schema>
2320   </wSDL:types>
2321
2322   <!-- ===== NotificationConsumer::Notify ===== -->
2323   Notify(
2324     NotificationMessage
2325     (SubscriptionReference, TopicExpression, ProducerReference,
2326     Message)*
2327   returns: n/a (one way)
2328   -->
2329   <wSDL:message name="Notify">
2330     <wSDL:part name="Notify" element="wsnt:Notify"/>
2331   </wSDL:message>
2332
2333   <!-- ===== NotificationProducer::Subscribe ===== -->
2334   Subscribe(
2335     (ConsumerEndpointReference, [Filter], [SubscriptionPolicy],
2336     [InitialTerminationTime])
2337   returns: WS-Resource qualified EPR to a Subscription
2338   -->
2339   <wSDL:message name="SubscribeRequest" >
2340     <wSDL:part name="SubscribeRequest"
2341       element="wsnt:Subscribe"/>
2342   </wSDL:message>
2343
2344   <wSDL:message name="SubscribeResponse">
2345     <wSDL:part name="SubscribeResponse"
2346       element="wsnt:SubscribeResponse"/>
2347   </wSDL:message>
2348
2349   <wSDL:message name="SubscribeCreationFailedFault">
2350     <wSDL:part name="SubscribeCreationFailedFault"
2351       element="wsnt:SubscribeCreationFailedFault" />
2352   </wSDL:message>
2353
2354   <wSDL:message name="TopicExpressionDialectUnknownFault">
2355     <wSDL:part name="TopicExpressionDialectUnknownFault"
2356       element="wsnt:TopicExpressionDialectUnknownFault" />
2357   </wSDL:message>

```

```

2358
2359 <wsdl:message name="InvalidFilterFault">
2360   <wsdl:part name="InvalidFilterFault"
2361     element="wsnt:InvalidFilterFault" />
2362 </wsdl:message>
2363
2364 <wsdl:message name="InvalidProducerPropertiesExpressionFault">
2365   <wsdl:part name="InvalidProducerPropertiesExpressionFault"
2366     element="wsnt:InvalidProducerPropertiesExpressionFault" />
2367 </wsdl:message>
2368
2369 <wsdl:message name="InvalidMessageContentExpressionFault">
2370   <wsdl:part name="InvalidMessageContentExpressionFault"
2371     element="wsnt:InvalidMessageContentExpressionFault" />
2372 </wsdl:message>
2373
2374 <wsdl:message name="UnrecognizedPolicyRequestFault">
2375   <wsdl:part name="UnrecognizedPolicyRequestFault"
2376     element="wsnt:UnrecognizedPolicyRequestFault" />
2377 </wsdl:message>
2378
2379 <wsdl:message name="UnsupportedPolicyRequestFault">
2380   <wsdl:part name="UnsupportedPolicyRequestFault"
2381     element="wsnt:UnsupportedPolicyRequestFault" />
2382 </wsdl:message>
2383
2384 <wsdl:message name="NotifyMessageNotSupportedFault">
2385   <wsdl:part name="NotifyMessageNotSupportedFault"
2386     element="wsnt:NotifyMessageNotSupportedFault" />
2387 </wsdl:message>
2388
2389 <wsdl:message name="UnacceptableInitialTerminationTimeFault">
2390   <wsdl:part name="UnacceptableInitialTerminationTimeFault"
2391     element="wsnt:UnacceptableInitialTerminationTimeFault" />
2392 </wsdl:message>
2393
2394 <!-- ===== NotificationProducer::GetCurrentMessage =====
2395 GetCurrentMessage(topicExpression)
2396 returns: a NotificationMessage (xsd:any)
2397 -->
2398 <wsdl:message name="GetCurrentMessageRequest">
2399   <wsdl:part name="GetCurrentMessageRequest"
2400     element="wsnt:GetCurrentMessage" />
2401 </wsdl:message>
2402
2403 <wsdl:message name="GetCurrentMessageResponse">
2404   <wsdl:part name="GetCurrentMessageResponse"
2405     element="wsnt:GetCurrentMessageResponse" />
2406 </wsdl:message>
2407
2408 <wsdl:message name="InvalidTopicExpressionFault">
2409   <wsdl:part name="InvalidTopicExpressionFault"
2410     element="wsnt:InvalidTopicExpressionFault" />
2411 </wsdl:message>
2412
2413 <wsdl:message name="TopicNotSupportedFault">
2414   <wsdl:part name="TopicNotSupportedFault"

```

```

2415         element="wsnt:TopicNotSupportedFault" />
2416     </wsdl:message>
2417
2418     <wsdl:message name="MultipleTopicsSpecifiedFault">
2419         <wsdl:part name="MultipleTopicsSpecifiedFault"
2420             element="wsnt:MultipleTopicsSpecifiedFault" />
2421     </wsdl:message>
2422
2423     <wsdl:message name="NoCurrentMessageOnTopicFault">
2424         <wsdl:part name="NoCurrentMessageOnTopicFault"
2425             element="wsnt:NoCurrentMessageOnTopicFault" />
2426     </wsdl:message>
2427
2428 <!-- ===== PullPoint::GetMessages =====>
2429     GetMessages(MaximumNumber)
2430     returns: NotificationMessage list
2431 -->
2432     <wsdl:message name="GetMessagesRequest">
2433         <wsdl:part name="GetMessagesRequest"
2434             element="wsnt:GetMessages" />
2435     </wsdl:message>
2436
2437     <wsdl:message name="GetMessagesResponse">
2438         <wsdl:part name="GetMessagesResponse"
2439             element="wsnt:GetMessagesResponse" />
2440     </wsdl:message>
2441
2442     <wsdl:message name="UnableToGetMessagesFault">
2443         <wsdl:part name="UnableToGetMessagesFault"
2444             element="wsnt:UnableToGetMessagesFault" />
2445     </wsdl:message>
2446
2447
2448 <!-- ===== PullPoint::DestroyPullPoint =====>
2449     DestroyPullPoint()
2450     returns: void
2451 -->
2452     <wsdl:message name="DestroyPullPointRequest">
2453         <wsdl:part name="DestroyPullPointRequest"
2454             element="wsnt:DestroyPullPoint" />
2455     </wsdl:message>
2456
2457     <wsdl:message name="DestroyPullPointResponse">
2458         <wsdl:part name="DestroyPullPointResponse"
2459             element="wsnt:DestroyPullPointResponse" />
2460     </wsdl:message>
2461
2462     <wsdl:message name="UnableToDestroyPullPointFault">
2463         <wsdl:part name="UnableToDestroyPullPointFault"
2464             element="wsnt:UnableToDestroyPullPointFault" />
2465     </wsdl:message>
2466
2467 <!-- ===== PullPoint::CreatePullPoint =====>
2468     CreatePullPoint()
2469     returns: PullPoint (wsa:EndpointReference)
2470 -->
2471     <wsdl:message name="CreatePullPointRequest">

```

```

2472     <wsdl:part name="CreatePullPointRequest"
2473         element="wsnt:CreatePullPoint" />
2474 </wsdl:message>
2475
2476 <wsdl:message name="CreatePullPointResponse">
2477     <wsdl:part name="CreatePullPointResponse"
2478         element="wsnt:CreatePullPointResponse" />
2479 </wsdl:message>
2480
2481 <wsdl:message name="UnableToCreatePullPointFault">
2482     <wsdl:part name="UnableToCreatePullPointFault"
2483         element="wsnt:UnableToCreatePullPointFault" />
2484 </wsdl:message>
2485
2486 <!-- ===== SubscriptionManager::Renew =====
2487 Renew( Duration | AbsoluteTime)
2488 returns: (New Termination Time [CurrentTime])
2489 -->
2490 <wsdl:message name="RenewRequest">
2491     <wsdl:part name="RenewRequest"
2492         element="wsnt:Renew" />
2493 </wsdl:message>
2494
2495 <wsdl:message name="RenewResponse">
2496     <wsdl:part name="RenewResponse"
2497         element="wsnt:RenewResponse" />
2498 </wsdl:message>
2499
2500 <wsdl:message name="UnacceptableTerminationTimeFault">
2501     <wsdl:part name="UnacceptableTerminationTimeFault"
2502         element="wsnt:UnacceptableTerminationTimeFault" />
2503 </wsdl:message>
2504
2505 <!-- ===== SubscriptionManager::Unsubscribe =====
2506 Unsubscribe()
2507 returns: empty
2508 -->
2509 <wsdl:message name="UnsubscribeRequest">
2510     <wsdl:part name="UnsubscribeRequest"
2511         element="wsnt:Unsubscribe" />
2512 </wsdl:message>
2513
2514 <wsdl:message name="UnsubscribeResponse">
2515     <wsdl:part name="UnsubscribeResponse"
2516         element="wsnt:UnsubscribeResponse" />
2517 </wsdl:message>
2518
2519 <wsdl:message name="UnableToDestroySubscriptionFault">
2520     <wsdl:part name="UnableToDestroySubscriptionFault"
2521         element="wsnt:UnableToDestroySubscriptionFault" />
2522 </wsdl:message>
2523
2524 <!-- ===== SubscriptionManager::PauseSubscription =====
2525 PauseSubscription()
2526 returns: empty
2527 -->
2528 <wsdl:message name="PauseSubscriptionRequest">

```

```

2529     <wsdl:part name="PauseSubscriptionRequest"
2530           element="wsnt:PauseSubscription" />
2531   </wsdl:message>
2532
2533   <wsdl:message name="PauseSubscriptionResponse">
2534     <wsdl:part name="PauseSubscriptionResponse"
2535           element="wsnt:PauseSubscriptionResponse" />
2536   </wsdl:message>
2537
2538   <wsdl:message name="PauseFailedFault">
2539     <wsdl:part name="PauseFailedFault"
2540           element="wsnt:PauseFailedFault" />
2541   </wsdl:message>
2542
2543   <!-- ===== SubscriptionManager::ResumeSubscription =====
2544   ResumeSubscription()
2545   returns: empty
2546   -->
2547   <wsdl:message name="ResumeSubscriptionRequest">
2548     <wsdl:part name="ResumeSubscriptionRequest"
2549           element="wsnt:ResumeSubscription" />
2550   </wsdl:message>
2551
2552   <wsdl:message name="ResumeSubscriptionResponse">
2553     <wsdl:part name="ResumeSubscriptionResponse"
2554           element="wsnt:ResumeSubscriptionResponse" />
2555   </wsdl:message>
2556
2557   <wsdl:message name="ResumeFailedFault">
2558     <wsdl:part name="ResumeFailedFault"
2559           element="wsnt:ResumeFailedFault" />
2560   </wsdl:message>
2561
2562   <!-- ===== PortType Definitions ===== -->
2563   <!-- ===== NotificationConsumer PortType Definition ===== -->
2564   <wsdl:portType name="NotificationConsumer">
2565     <wsdl:operation name="Notify">
2566       <wsdl:input message="wsntw:Notify" />
2567     </wsdl:operation>
2568   </wsdl:portType>
2569
2570   <!-- ===== NotificationProducer PortType Definition ===== -->
2571   <wsdl:portType name="NotificationProducer">
2572     <wsdl:operation name="Subscribe">
2573       <wsdl:input message="wsntw:SubscribeRequest" />
2574       <wsdl:output message="wsntw:SubscribeResponse" />
2575       <wsdl:fault name="ResourceUnknownFault"
2576             message="wsrf-rw:ResourceUnknownFault" />
2577       <wsdl:fault name="InvalidFilterFault"
2578             message="wsntw:InvalidFilterFault" />
2579       <wsdl:fault name="TopicExpressionDialectUnknownFault"
2580             message="wsntw:TopicExpressionDialectUnknownFault" />
2581       <wsdl:fault name="InvalidTopicExpressionFault"
2582             message="wsntw:InvalidTopicExpressionFault" />
2583       <wsdl:fault name="TopicNotSupportedFault"
2584             message="wsntw:TopicNotSupportedFault" />
2585       <wsdl:fault name="InvalidProducerPropertiesExpressionFault"

```

```

2586         message="wsntw:InvalidProducerPropertiesExpressionFault" />
2587     <wsdl:fault name="InvalidMessageContentExpressionFault"
2588         message="wsntw:InvalidMessageContentExpressionFault" />
2589     <wsdl:fault name="UnacceptableInitialTerminationTimeFault"
2590         message="wsntw:UnacceptableInitialTerminationTimeFault" />
2591     <wsdl:fault name="UnrecognizedPolicyRequestFault"
2592         message="wsntw:UnrecognizedPolicyRequestFault" />
2593     <wsdl:fault name="UnsupportedPolicyRequestFault"
2594         message="wsntw:UnsupportedPolicyRequestFault" />
2595     <wsdl:fault name="NotifyMessageNotSupportedFault"
2596         message="wsntw:NotifyMessageNotSupportedFault" />
2597     <wsdl:fault name="SubscribeCreationFailedFault"
2598         message="wsntw:SubscribeCreationFailedFault" />
2599 </wsdl:operation>
2600
2601 <wsdl:operation name="GetCurrentMessage">
2602     <wsdl:input message="wsntw:GetCurrentMessageRequest" />
2603     <wsdl:output message="wsntw:GetCurrentMessageResponse" />
2604     <wsdl:fault name="ResourceUnknownFault"
2605         message="wsrf-rw:ResourceUnknownFault" />
2606     <wsdl:fault name="TopicExpressionDialectUnknownFault"
2607         message="wsntw:TopicExpressionDialectUnknownFault" />
2608     <wsdl:fault name="InvalidTopicExpressionFault"
2609         message="wsntw:InvalidTopicExpressionFault" />
2610     <wsdl:fault name="TopicNotSupportedFault"
2611         message="wsntw:TopicNotSupportedFault" />
2612     <wsdl:fault name="NoCurrentMessageOnTopicFault"
2613         message="wsntw:NoCurrentMessageOnTopicFault" />
2614     <wsdl:fault name="MultipleTopicsSpecifiedFault"
2615         message="wsntw:MultipleTopicsSpecifiedFault" />
2616 </wsdl:operation>
2617 </wsdl:portType>
2618
2619 <!-- ===== PullPoint PortType Definition ===== -->
2620 <wsdl:portType name="PullPoint">
2621     <wsdl:operation name="GetMessages">
2622         <wsdl:input name="GetMessagesRequest"
2623             message="wsntw:GetMessagesRequest" />
2624         <wsdl:output name="GetMessagesResponse"
2625             message="wsntw:GetMessagesResponse" />
2626         <wsdl:fault name="ResourceUnknownFault"
2627             message="wsrf-rw:ResourceUnknownFault" />
2628         <wsdl:fault name="UnableToGetMessagesFault"
2629             message="wsntw:UnableToGetMessagesFault" />
2630     </wsdl:operation>
2631
2632     <wsdl:operation name="DestroyPullPoint">
2633         <wsdl:input name="DestroyPullPointRequest"
2634             message="wsntw:DestroyPullPointRequest" />
2635         <wsdl:output name="DestroyPullPointResponse"
2636             message="wsntw:DestroyPullPointResponse" />
2637         <wsdl:fault name="ResourceUnknownFault"
2638             message="wsrf-rw:ResourceUnknownFault" />
2639         <wsdl:fault name="UnableToDestroyPullPointFault"
2640             message="wsntw:UnableToDestroyPullPointFault" />
2641     </wsdl:operation>
2642

```

```

2643     <wsdl:operation name="Notify">
2644         <wsdl:input message="wsntw:Notify" />
2645     </wsdl:operation>
2646 </wsdl:portType>
2647
2648 <!-- ===== CreatePullPoint PortType Definition ===== -->
2649 <wsdl:portType name="CreatePullPoint">
2650     <wsdl:operation name="CreatePullPoint">
2651         <wsdl:input name="CreatePullPointRequest"
2652             message="wsntw:CreatePullPointRequest" />
2653         <wsdl:output name="CreatePullPointResponse"
2654             message="wsntw:CreatePullPointResponse" />
2655         <wsdl:fault name="UnableToCreatePullPointFault"
2656             message="wsntw:UnableToCreatePullPointFault" />
2657     </wsdl:operation>
2658 </wsdl:portType>
2659
2660 <!-- ===== SubscriptionManager PortType Definition ===== -->
2661 <wsdl:portType name="SubscriptionManager">
2662     <wsdl:operation name="Renew">
2663         <wsdl:input name="RenewRequest"
2664             message="wsntw:RenewRequest" />
2665         <wsdl:output name="RenewResponse"
2666             message="wsntw:RenewResponse" />
2667         <wsdl:fault name="ResourceUnknownFault"
2668             message="wsrf-rw:ResourceUnknownFault" />
2669         <wsdl:fault name="UnacceptableTerminationTimeFault"
2670             message=
2671             "wsntw:UnacceptableTerminationTimeFault" />
2672     </wsdl:operation>
2673     <wsdl:operation name="Unsubscribe">
2674         <wsdl:input name="UnsubscribeRequest"
2675             message="wsntw:UnsubscribeRequest" />
2676         <wsdl:output name="UnsubscribeResponse"
2677             message="wsntw:UnsubscribeResponse" />
2678         <wsdl:fault name="ResourceUnknownFault"
2679             message="wsrf-rw:ResourceUnknownFault" />
2680         <wsdl:fault name="UnableToDestroySubscriptionFault"
2681             message=
2682             "wsntw:UnableToDestroySubscriptionFault" />
2683     </wsdl:operation>
2684 </wsdl:portType>
2685
2686 <!-- ===== PausableSubscriptionManager PortType Definition ===== -->
2687 <wsdl:portType name="PausableSubscriptionManager">
2688     <!-- ===== Extends: SubscriptionManager ===== -->
2689     <wsdl:operation name="Renew">
2690         <wsdl:input name="RenewRequest"
2691             message="wsntw:RenewRequest" />
2692         <wsdl:output name="RenewResponse"
2693             message="wsntw:RenewResponse" />
2694         <wsdl:fault name="ResourceUnknownFault"
2695             message="wsrf-rw:ResourceUnknownFault" />
2696         <wsdl:fault name="UnacceptableTerminationTimeFault"
2697             message=
2698             "wsntw:UnacceptableTerminationTimeFault" />
2699     </wsdl:operation>

```

```

2700 <wsdl:operation name="Unsubscribe">
2701   <wsdl:input name="UnsubscribeRequest"
2702             message="wsntw:UnsubscribeRequest" />
2703   <wsdl:output name="UnsubscribeResponse"
2704             message="wsntw:UnsubscribeResponse" />
2705   <wsdl:fault name="ResourceUnknownFault"
2706             message="wsrf-rw:ResourceUnknownFault" />
2707   <wsdl:fault name="UnableToDestroySubscriptionFault"
2708             message=
2709             "wsntw:UnableToDestroySubscriptionFault" />
2710 </wsdl:operation>
2711
2712 <!-- === PausableSubscriptionManager specific operations === -->
2713 <wsdl:operation name="PauseSubscription">
2714   <wsdl:input message="wsntw:PauseSubscriptionRequest"/>
2715   <wsdl:output message="wsntw:PauseSubscriptionResponse"/>
2716   <wsdl:fault name="ResourceUnknownFault"
2717             message="wsrf-rw:ResourceUnknownFault" />
2718   <wsdl:fault name="PauseFailedFault"
2719             message="wsntw:PauseFailedFault" />
2720 </wsdl:operation>
2721 <wsdl:operation name="ResumeSubscription">
2722   <wsdl:input message="wsntw:ResumeSubscriptionRequest"/>
2723   <wsdl:output message="wsntw:ResumeSubscriptionResponse"/>
2724   <wsdl:fault name="ResourceUnknownFault"
2725             message="wsrf-rw:ResourceUnknownFault" />
2726   <wsdl:fault name="ResumeFailedFault"
2727             message="wsntw:ResumeFailedFault" />
2728 </wsdl:operation>
2729 </wsdl:portType>
2730 </wsdl:definitions>

```


Appendix D. Revision History

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.

Rev	Date	By Whom	What
			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, 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
1.3.02.a	2005-09-13	David Hull	Issues 2.57, 2.59. 2.60, 2.61
1.3.02.a	2005-09-14	Bryan Murray	Changed namespace and [action] URIs to b-2 and bw-2 Added resolution to 2.58
1.3.02b	2005-10-10	David Hull	Issues 2.63, 4.2, AI 132, 134, 136, cleanup of references.

Rev	Date	By Whom	What
1.3.02c	2005-10-11	David Hull	Issues 2.62, 4.2, AI 132, 134, 138, 139, 140
1.3.02d	2005-10-19	David Hull	Various errata
1.3.02f	2005-11-21	David Hull	Issue 2.62 and various errata
1.3.03a	2006-03-27	David Hull	Issues 2.64, 2.65, 2.66, 2.67 and various errata
1.3.04	2006-05-08	Peter Niblett	Miscellaneous pseudo-schema corrections. Changed schema for GetMessages/MaximumNumber to be minOccurs="0"

2732

Appendix E. Notices

2733 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
2734 that might be claimed to pertain to the implementation or use of the technology described in this
2735 document or the extent to which any license under such rights might or might not be available;
2736 neither does it represent that it has made any effort to identify any such rights. Information on
2737 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
2738 website. Copies of claims of rights made available for publication and any assurances of licenses
2739 to be made available, or the result of an attempt made to obtain a general license or permission
2740 for the use of such proprietary rights by implementors or users of this specification, can be
2741 obtained from the OASIS Executive Director.

2742

2743 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
2744 applications, or other proprietary rights which may cover technology that may be required to
2745 implement this specification. Please address the information to the OASIS Executive Director.

2746

2747 Copyright (C) OASIS Open (2004-2006). All Rights Reserved.

2748

2749 This document and translations of it may be copied and furnished to others, and derivative works
2750 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
2751 published and distributed, in whole or in part, without restriction of any kind, provided that the
2752 above copyright notice and this paragraph are included on all such copies and derivative works.
2753 However, this document itself may not be modified in any way, such as by removing the copyright
2754 notice or references to OASIS, except as needed for the purpose of developing OASIS
2755 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
2756 Property Rights document must be followed, or as required to translate it into languages other
2757 than English.

2758

2759 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
2760 successors or assigns.

2761

2762 This document and the information contained herein is provided on an "AS IS" basis and OASIS
2763 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
2764 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
2765 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
2766 PARTICULAR PURPOSE.

2767