



Web Services Base Notification 1.3 (WS-BaseNotification)

Public Review Draft 02, 28 November 2005

Document identifier

wsn-ws_base_notification-1.3-spec-pr-02

Location:

http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-02.pdf

Editors:

Steve Graham, IBM <sggraham@us.ibm.com>
David Hull, Tibco <dmh@tibco.com>
Bryan Murray, Hewlett-Packard Company <bryan.murray@hp.com>

Abstract:

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

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

Status:

On November 28th, 2005, the OASIS WS-Notification Technical Committee approved this document for publication as a Public Review Draft. Committee members should send comments on this specification to the wsn@lists.oasis-open.org list. Others may submit comments to the TC via the web form found on the TC's web page at <http://www.oasis-open.org/committees/wsn>. Click the button for "Send A Comment" at the top of the page. Submitted comments (for this work as well as other works of the TC) are publicly

Deleted: [
Deleted:]
Deleted: .

37 archived and can be viewed at:
38 <http://lists.oasis-open.org/archives/wsn-comment/>.

39 For information on whether any patents have been disclosed that may be essential to
40 implementing this specification, and any offers of patent licensing terms, please refer to
41 the Intellectual Property Rights section of the WSN TC web page ([http://www.oasis-](http://www.oasis-open.org/committees/wsn/)
42 [open.org/committees/wsn/](http://www.oasis-open.org/committees/wsn/)).

Table of Contents

44	1	Introduction	5
45	1.1	Goals and Requirements	5
46	1.1.1	Requirements.....	5
47	1.1.2	Non-Goals.....	6
48	1.2	Terminology.....	6
49	1.3	Namespaces	7
50	1.4	Fault Definitions.....	7
51	2	Terminology and Concepts	8
52	2.1	Definitions.....	8
53	2.2	Production vs. Delivery	9
54	3	NotificationConsumer Interface	10
55	3.1	Notification Metadata	10
56	3.2	Notify	11
57	3.2.1	Example SOAP Encoding of the Notify Message.....	11
58	4	NotificationProducer Interface	13
59	4.1	NotificationProducer Resource Properties.....	13
60	4.2	Subscribe	14
61	4.2.1	Example SOAP Encoding of the Subscribe Message Exchange.....	20
62	4.3	GetCurrentMessage.....	21
63	4.3.1	Example SOAP Encoding of the GetCurrentMessage Message Exchange	22
64	5	Pull-Style Notification.....	24
65	5.1	PullPoint Interface	24
66	5.1.1	Accumulating Notification Messages.....	24
67	5.1.2	GetMessages.....	24
68	5.1.3	Example SOAP Encoding of the GetMessages Message Exchange.....	26
69	5.1.4	DestroyPullPoint	26
70	5.1.5	Example SOAP Encoding of the DestroyPullPoint Message Exchange	27
71	5.2	Create PullPoint Interface	28
72	5.2.1	Example SOAP Encoding of the CreatePullPoint Message Exchange.....	28
73	6	SubscriptionManager Interface.....	30
74	6.1	Base Subscription Manager	30
75	6.1.1	Renew.....	30
76	6.1.2	Unsubscribe	32
77	6.2	Pausable Subscription Manager	33
78	6.2.1	PauseSubscription.....	33
79	6.2.2	Example SOAP Encoding of the PauseSubscription Message Exchange.....	34
80	6.3	ResumeSubscription	35
81	6.3.1	Example SOAP Encoding of the ResumeSubscription Message Exchange.....	36
82	6.4	Subscriptions as WS-Resources.....	36
83	6.4.1	Subscription Resource Properties	37

84	7	Security Considerations.....	38
85	7.1	Securing the Message Exchanges.....	38
86	7.2	Securing Subscriptions and Notifications.....	38
87	8	References.....	40
88	8.1	Normative.....	40
89	8.2	Non-Normative.....	40
90		Appendix A. Acknowledgments.....	41
91		Appendix B. XML Schema.....	42
92		Appendix C. WSDL 1.1.....	53
93		Appendix D. Revision History.....	61
94		Appendix E. Notices.....	64
95			

96

1 Introduction

97

The Event-driven, or Notification-based, interaction pattern is commonly used in 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.

98

99

100

101

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

102

103

104

105

106

107

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

108

109

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

110

111

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

112

113

114

115

116

117

118

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

119

120

121

122

123

1.1 Goals and Requirements

124

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

125

126

127

1.1.1 Requirements

128

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

129

130

- **Must support resource-constrained devices.** The specifications must be factored in a way that allows resource-constrained devices to participate in the Notification pattern. Such devices will be able to send information to, and receive information from Web services, without having to implement all the features of the specifications.

131

132

133

134

- **Must provide runtime metadata:** There must be a mechanism that lets a potential Subscriber discover what elements are provided for subscription by a NotificationProducer, and in what formats the subscription for notification can be made.

135

136

137

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

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

147 1.1.2 Non-Goals

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

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

163

164 1.2 Terminology

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

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

171 This specification uses a notational convention, referred to as "Pseudo-schemas". A Pseudo-
172 schema uses a BNF-style convention to describe attributes and elements:

- 173 • '?' denotes optionality (i.e. zero or one occurrences),
- 174 • '*' denotes zero or more occurrences,
- 175 • '+' one or more occurrences,
- 176 • '[' and ']' are used to form groups,
- 177 • '|' represents choice.
- 178 • Attributes are conventionally assigned a value which corresponds to their type, as
179 defined in the normative schema.

180 <!-- sample pseudo-schema -->

Deleted:

Deleted: " in a fashion similar to the WSDL 2.0 Part 1 specification [WSDL 2.0].

```

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

```

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

194 1.3 Namespaces

195 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

Deleted: 03
Deleted: 1
Deleted: 1
Deleted: 1
Deleted: 1

196 1.4 Fault Definitions

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

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

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

202

2 Terminology and Concepts

203

2.1 Definitions

204

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

205

Situation:

206

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

207

208

- 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.

209

210

211

- 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.

212

213

Notification:

214

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

215

216

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

217

218

- 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.

219

220

221

222

- 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.

223

224

225

226

NotificationProducer:

227

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

228

229

- 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.

230

231

232

233

- 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.

234

235

236

237

- It is the factory for Subscription resources.

238

NotificationConsumer:

239

- 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.

240

241

242

- A NotificationConsumer may accept the generic Notify message, or it may be able to process one or more domain-specific Notification types.

243

244 **Subscription:**

245 • A Subscription represents the relationship between a NotificationConsumer and a
246 NotificationProducer, including any filtering parameters such as Topic and various other
247 optional filter expressions, along with any relevant policies and context information.

248 • A Subscription resource is created when a Subscriber sends the SubscribeRequest
249 message to a NotificationProducer.

250 • Subscription resources are manipulated by messages sent to the SubscriptionManager
251 Web service associated with the Subscription resource.

252 **SubscriptionManager**

253 • A SubscriptionManager is an endpoint, represented by an endpoint reference [WS-
254 Addressing] that implements message exchanges associated with the
255 SubscriptionManager interface.

256 • A SubscriptionManager provides operations that allow a service requestor to query and
257 manipulate Subscription resources that it manages

258 • A SubscriptionManager is subordinate to the NotificationProducer, and MAY be
259 implemented by the NotificationProducer service provider. However WS-Notification
260 permits it to be implemented by a separate service provider, should an implementer so
261 desire.

262 **Subscriber:**

263 • A Subscriber is any entity that sends the SubscribeRequest message to a
264 NotificationProducer.

265 • Note that a Subscriber may be a different entity from the NotificationConsumer for which
266 Notifications are actually produced.

267 **2.2 Production vs. Delivery**

268 Various statements in this document refer to a NotificationProducer *producing* a Notification.
269 Producing a Notification means: *supplying a Notification to a delivery mechanism for transmission*
270 *to a NotificationConsumer*. Depending on the actual delivery mechanism, this transmission might
271 be reliable or might be done on a best-effort basis. A Notification which is never produced will
272 definitely never be delivered, but the converse is not necessarily true: a Notification which *is*
273 produced may or may not actually be delivered, depending on the delivery mechanism, the
274 validity of the NotificationConsumer address, the state of the network, and so forth.

3 NotificationConsumer Interface

275

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

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

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

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

Deleted: .

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

3.1 Notification Metadata

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

300 wsnt:SubscriptionReference

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

Deleted: WS-Resource [WS-Resource]

302 wsnt:Topic

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

Deleted: .

307 wsnt:Topic/@Dialect

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

310 wsnt:ProducerReference

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

312

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

317 3.2 Notify

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

```
320 ...  
321 <wsnt:Notify>  
322   <wsnt:NotificationMessage>  
323     <wsnt:SubscriptionReference>  
324       wsa:EndpointReference  
325     </wsnt:SubscriptionReference> ?  
326     <wsnt:Topic Dialect="xsd:anyURI">  
327       {any} *  
328     </wsnt:Topic?>  
329     <wsnt:ProducerReference>  
330       wsa:EndpointReference  
331     </wsnt:ProducerReference> ?  
332     <wsnt:Message>  
333       {any}  
334     </wsnt:Message>  
335   <wsnt:NotificationMessage> +  
336   {any} *  
337 </wsnt:Notify>  
338 ...
```

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

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

Deleted: 1

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

342 /wsnt:Notify

343 Contains a collection of one or more Notifications.

344 /wsnt:Notify/wsnt:NotificationMessage

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

347 /wsnt:Notify/wsnt:NotificationMessage/wsnt:Message

348 A copy of the actual Notification payload.

349 /wsnt:Notify/{any}

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

353

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

355 3.2.1 Example SOAP Encoding of the Notify Message

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

```
357 <s:Envelope ... >  
358   <s:Header>  
359     <wsa:Action>  
360       http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify  
361     </wsa:Action>  
362     ...  
363   </s:Header>
```

Deleted: 1

```

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

```

Deleted: producer

Deleted: SubsMgrEndpoint

Deleted: producer

Deleted: ProducerEndpoint

388

4 NotificationProducer Interface

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

4.1 NotificationProducer Resource Properties

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

398
399
400
401
402
403
404
405
406
407
408
409

```
...  
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"/>  
...
```

Deleted:

Deleted: 1

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

411
412
413
414
415
416
417
418
419
420
421
422
423
424

```
<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" />
```

425

426 These Resource Property elements are further constrained as follows:

427 /wsnt:TopicExpression

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

434 may be referred to by multiple wsnt:TopicExpression resource property element(s), for
435 example, in different resource property elements each using a different dialect.

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

441 /wsnt:FixedTopicSet

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

448 /wsnt:TopicExpressionDialect

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

Deleted: WS-Resource

455 [/wstop:TopicSet](#).

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

458

459 4.2 Subscribe

460 A NotificationProducer is capable of producing a sequence of zero or more Notifications. A
461 Subscriber can register the interest of a NotificationConsumer to receive a subset of this
462 sequence. A Subscriber sends a Subscribe message to a NotificationProducer in order to register
463 this interest.

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

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

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

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

476 NotificationProducers MAY advertise more constrained behavior through policy assertions or
477 other means.

478 The format of the Subscribe message is:

479

...

```

480 <wsnt:Subscribe>
481   <wsnt:ConsumerReference>
482     wsa:EndpointReference
483   </wsnt:ConsumerReference>
484   <wsnt:Filter>
485     [ <wsnt:TopicExpression Dialect="xsd:anyURI">
486       {any} ?
487     </wsnt:TopicExpression> |
488     <wsnt:ProducerProperties Dialect="xsd:anyURI">
489       {any} ?
490     </wsnt:ProducerProperties> |
491     <wsnt:MessageContent Dialect="xsd:anyURI">
492       {any}
493     </wsnt:MessageContent> |
494     {any} *
495   ] *
496 </wsnt:Filter> ?
497 <wsnt:InitialTerminationTime>
498   [xsd:dateTime | xsd:duration]
499 </wsnt:InitialTerminationTime> ?
500 <wsnt:SubscriptionPolicy>
501   [ <wsnt:UseRaw/> |
502     {any}
503   ] *
504 </wsnt:SubscriptionPolicy> ?
505 {any}*
506 </wsnt:Subscribe>
507 ...

```

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

Deleted: 1

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

511 | /wsnt:Subscribe/wsnt:ConsumerReference

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

514 This component SHOULD provide all the necessary information to specify how the
515 NotificationProducer should send notifications to the NotificationConsumer. However, there
516 may be cases when the ConsumerReference EPR may not include all the details that the
517 NotificationProducer expects.

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

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

526 | /wsnt:Subscribe/wsnt:Filter

527 The Filter component is the means by which a Subscriber expresses the subset of
528 Notifications that the NotificationConsumer should receive. This subset is expressed by the
529 child elements of the Filter. The child elements are a sequence of zero or more expressions
530 evaluating to a Boolean that constrain the set of possible Notifications. Each expression is
531 evaluated in a manner specific to that kind of expression (see below); the order and timing of

532 the evaluation is determined by the NotificationProducer. All filter expressions MUST
 533 evaluate to true in order for the notification message to be sent.

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

537 The NotificationProducer MUST respond with an InvalidFilterFault message if any child
 538 expression element is not supported by the NotificationProducer. For example, if the
 539 NotificationProducer does not support the concept of Topics, it MUST respond with an
 540 InvalidFilterFault message if a Subscribe message contains a Filter component that includes
 541 a TopicExpression child. The fault MUST include the filter QNames for the filters it did not
 542 understand.

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

546 | /wsnt:Subscribe/wsnt:Filter/wsnt:TopicExpression

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

550 | /wsnt:Subscribe/wsnt:Filter/wsnt:TopicExpression/@Dialect

Deleted:

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

554 The NotificationProducer MAY refuse to process the Subscribe request if the dialect used by
 555 the Subscriber in the TopicExpression is not one of the dialects supported by the
 556 NotificationProducer.

557 The NotificationProducer MUST respond with a TopicExpressionDialectUnknownFault if it
 558 understands the TopicExpression element, but does not understand the specified
 559 TopicExpression dialect. Note that a NotificationProducer may understand the meaning of a
 560 given dialect URI without actually supporting that dialect.

561 If the TopicExpression is incompatible with or does not comply with the rules of the dialect,
 562 the NotificationProducer MUST respond with a InvalidTopicExpressionFault.

563 If the TopicExpression dialect is understood and the expression references a topic which is
 564 not supported by the NotificationProducer, the NotificationProducer MAY return a
 565 TopicNotSupportedFault.

566 | /wsnt:Subscribe/wsnt:Filter/wsnt:ProducerProperties

Deleted:

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

Deleted: .

570 | /wsnt:Subscribe/wsnt:Filter/wsnt:ProducerProperties/@Dialect

Deleted:

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

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

577 | /wsnt:Subscribe/wsnt:Filter/wsnt:MessageContent

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

581 | /wsnt:Subscribe/wsnt:Filter/wsnt:MessageContent/@Dialect

582 | This attribute contains a URI specifying the type of MessageContent filter expression
583 | contained by the element. This specification identifies the XPath 1.0 dialect corresponding to
584 | version 1.0 of the XPath language [XPATH]:

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

586 | This URI identifies the XPath 1.0 language. The contents of the MessageContent expression
587 | MUST be a string containing a valid XPath 1.0 expression.

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

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

592 | /wsnt:Subscribe/wsnt:InitialTerminationTime

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

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

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

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

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

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

613 | /wsnt:Subscribe/wsnt:SubscriptionPolicy

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

Deleted:

Deleted:

Deleted: There are two well known dialects identified by this

Deleted: ,

Deleted: two versions

Deleted: .

Deleted: <http://www.w3.org/TR/2005/WD-xpath20-20050404>
This URI identifies the Xpath 2.0 (working draft) language. The contents of the MessageContent expression MUST be a string containing a valid XPath 2.0 expression. Note: an additional URI will be added to represent the W3C Recommendation form of the XPath 2.0 language.¶

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

623 /wsnt:Subscribe/wsnt:SubscriptionPolicy/wsnt:UseRaw

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

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

- 629 • If the NotificationProducer supports this form of notification, then the NotificationProducer
630 MUST produce the Notification message in raw form.
- 631 • If the NotificationProducer does not support this form of notification, it MUST return a
632 UnsupportedPolicyRequestFault message containing a wsnt:UseRaw QName. The
633 NotificationProducer MUST NOT return an UnrecognizedPolicyRequestFault in response
634 to this element.

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

- 639 • If the NotificationProducer supports this form of notification, then the NotificationProducer
640 MUST produce Notification messages using the Notify wrapper
- 641 • If the NotificationProducer does not support this form of notification, it MUST return a
642 NotifyMessageNotSupportedFault message.

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

```
644 ...  
645 <wsnt:SubscribeResponse>  
646   <wsnt:SubscriptionReference>  
647     wsa:EndpointReference  
648   </wsnt:SubscriptionReference>  
649   <wsnt:CurrentTime>xsd:dateTime</wsnt:CurrentTime> ?  
650   <wsnt:TerminationTime>xsd:dateTime</wsnt:TerminationTime> ?  
651   {any}*  
652 </wsnt:SubscribeResponse>  
653 ...
```

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

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

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

657 /wsnt:SubscribeResponse/wsnt:SubscriptionReference

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

659 /wsnt:SubscribeResponse/wsnt:CurrentTime

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

664 /wsnt:SubscribeResponse/wsnt:TerminationTime

Deleted: This component is optional, if missing the default behavior is to use the Notify wrapper for all notification messages.

Deleted: MUST

Deleted: .

Deleted: If this element is not present then the NotificationProducer MUST use the wsnt:Notify message in Notifications produced for this subscription. In this case the

Deleted: raw

Deleted: itself

Inserted: itself.¶
If the NotificationProducer does not support this form of notification, it

Deleted: element

Inserted: element. The NotificationProducer MUST NOT return an UnrecognizedPolicyRequestFault in response to this element.¶
The absence of this element is interpreted as a request for notification using the wsnt:Notify wrapper. If notification using the Notify wrapper is requested, then the

Deleted: .

Deleted:

Formatted: Bulleted + Level: 1 + Aligned at: 18 pt + Tab after: 36 pt + Indent at: 36 pt

Deleted: element is present

Deleted: SHOULD

Deleted: the raw

Deleted: itself.

Deleted: 1

Deleted: WS-Resource

Deleted: WS-Resource

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

Deleted: .

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

672

673 ResourceUnknownFault

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

677 InvalidFilterFault

- 678 • The Subscribe message contained a filter that was not understood or supported by the
679 NotificationProducer.

680 TopicExpressionDialectUnknownFault

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

683 InvalidTopicExpressionFault

- 684 • The Subscribe message contained a TopicExpression filter where the contents of the
685 filter did not match the dialect specified.

686 TopicNotSupportedFault

- 687 • The Subscribe message contained a TopicExpression filter that referenced a topic that
688 was not supported by the NotificationProducer.

689 InvalidProducerPropertiesExpressionFault

- 690 • The Subscribe message contained a ProducerProperties filter that did not represent a
691 valid boolean expression.

692 InvalidMessageContentExpressionFault

- 693 • The Subscribe message contained a MessageContent filter that did not represent a valid
694 boolean expression.

695 UnrecognizedPolicyRequestFault

- 696 • The NotificationProducer does not recognize one or more policy requests carried in the
697 SubscriptionPolicy element. This fault SHOULD contain elements representing the
698 offending policy requests.

699 UnsupportedPolicyRequestFault

- 700 • The NotificationProducer recognizes, but does not support one or more policy requests
701 carried in the SubscriptionPolicy element. This fault SHOULD contain elements
702 representing the offending policy requests.

703 NotifyMessageNotSupportedFault

- 704 • The NotificationProducer does not support the wsnt:Notify wrapper and was not able to
705 determine an alternative.

706 UnacceptableInitialTerminationTimeFault

Deleted: InvalidUseRawValue Fault¶

Deleted: RawValue element appeared

Deleted: than once

Deleted: .

Deleted: MAY

Inserted: MAY contain elements representing the offending policy requests.¶

Deleted: MAY

Inserted: MAY contain elements representing the offending policy requests.¶
NotifyMessageNotSupportedFault ¶
The NotificationProducer does not support the wsnt:Notify wrapper and was not able to determine an alternative.¶

707 • The value of InitialTerminationTime specified in the Subscribe message was not
708 acceptable to the NotificationProducer. The NotificationProducer MAY include a hint in
709 the fault message indicating acceptable values for InitialTerminationTime.

710 SubscribeCreationFailedFault

711 • The NotificationProducer failed to process the Subscribe message. The
712 NotificationProducer SHOULD use a more specific fault message if possible. The
713 NotificationProducer MAY include a hint in the fault message indicating why it failed to
714 process the Subscribe message.

715 4.2.1 Example SOAP Encoding of the Subscribe Message Exchange

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

```
717 <s:Envelope ... >
718   <s:Header>
719     <wsa:Action>
720       http://docs.oasis-open.org/wsn/bw-
721 2/NotificationProducer/SubscribeRequest
722     </wsa:Action>
723     ...
724   </s:Header>
725   <s:Body>
726     <wsnt:Subscribe>
727       <wsnt:ConsumerReference>
728         <wsa:Address>
729           http://www.example.org/NotificationConsumer
730         </wsa:Address>
731       </wsnt:ConsumerReference>
732       <wsnt:Filter>
733         <wsnt:TopicExpression Dialect=
734           "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
735           npex:SomeTopic
736         </wsnt:TopicExpression>
737         <wsnt:MessageContent
738           Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
739           boolean(ncex:Producer="15")
740         </wsnt:MessageContent>
741       </wsnt:Filter>
742       <wsnt:InitialTerminationTime>
743         2005-12-25T00:00:00.000000Z
744       </wsnt:InitialTerminationTime>
745     </wsnt:Subscribe>
746   </s:Body>
747 </s:Envelope>
```

Deleted: 1

Deleted: consumer

Deleted: ConsumerEndpoint

748

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

```
750 <s:Envelope ... >
751   <s:Header>
752     <wsa:Action>
753       http://docs.oasis-open.org/wsn/bw-
754 2/NotificationProducer/SubscribeResponse
755     </wsa:Action>
756     ...
757   </s:Header>
758   <s:Body>
759     <wsnt:SubscribeResponse>
```

Deleted: 1/

760
761
762
763
764
765
766
767

```
<wsnt:SubscriptionReference>
  <wsa:Address>
    http://www.example.org/SubscriptionManager
  </wsa:Address>
</wsnt:SubscriptionReference>
</wsnt:SubscribeResponse>
</s:Body>
</s:Envelope>
```

Deleted: producer
Deleted: SubsMgrEndpoint
Formatted: Tabs: 186.75 pt, Left

768 4.3 GetCurrentMessage

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

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

778 The format of the GetCurrentMessage request message is:

779
780
781
782
783
784
785
786

```
...
<wsnt:GetCurrentMessage>
  <wsnt:Topic Dialect="xsd:anyURI">
    {any}
  </wsnt:Topic>
  {any} *
</wsnt:GetCurrentMessage>
...
```

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

Deleted: 1

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

790 `/wsnt:GetCurrentMessage/wsnt:Topic`

791 A TopicExpression that identifies exactly one Topic.

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

794
795
796
797
798

```
...
<wsnt:GetCurrentMessageResponse>
  {any} *
</wsnt:GetCurrentMessageResponse>
...
```

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

Deleted: 1

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

802 `/wsnt:GetCurrentMessageResponse/{any}`

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

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

807 ResourceUnknownFault

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

Deleted: (which follows the WS-Resource Access Pattern)

811 TopicExpressionDialectUnknownFault

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

814 InvalidTopicExpressionFault

- 815 • The Topic element of the GetCurrentMessage message had contents that did not match
816 the dialect specified.

817 TopicNotSupportedFault

- 818 • The Topic element of the GetCurrentMessage message referenced a topic that was not
819 supported by the NotificationProducer.

820 MultipleTopicsSpecifiedFault

- 821 • The GetCurrentMessage message referenced more than one topic.

822 NoCurrentMessageOnTopicFault

- 823 • The topic referenced in the GetCurrentMessage message does not have any pending
824 messages.

825 4.3.1 Example SOAP Encoding of the GetCurrentMessage Message Exchange

827 The following is a non-normative example of a GetCurrentMessage request message using
828 SOAP:

```
829 <s:Envelope ... >  
830 <s:Header>  
831 <wsa:Action>  
832 http://docs.oasis-open.org/wsn/bw-  
833 2/NotificationProducer/GetCurrentMessageRequest  
834 </wsa:Action>  
835 ...  
836 </s:Header>  
837 <s:Body>  
838 <wsnt:GetCurrentMessage>  
839 <wsnt:Topic Dialect=  
840 "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">  
841 npex:SomeTopic  
842 </wsnt:Topic>  
843 </wsnt:GetCurrentMessage>  
844 </s:Body>  
845 </s:Envelope>
```

Deleted: 1/

846 The following is a non-normative example of a GetCurrentMessage response message using
847 SOAP:

```
848 <s:Envelope ... >  
849 <s:Header>  
850 <wsa:Action>
```

851
852
853
854
855
856
857
858
859
860
861

```
      http://docs.oasis-open.org/wsn/bw-  
2/NotificationProducer/GetCurrentMessageResponse  
    </wsa:Action>  
    ...  
  </s:Header>  
  <s:Body>  
    <wsnt:GetCurrentMessageResponse>  
      <npex:NotifyContent>exampleNotifyContent</npex:NotifyContent>  
    </wsnt:GetCurrentMessageResponse>  
  </s:Body>  
</s:Envelope>
```

Deleted: 1/

5 Pull-Style Notification

862

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

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

5.1 PullPoint Interface

877

878 In support of the “pull-style” of Notification Message delivery, this specification describes a
879 PullPoint resource by defining a PullPoint interface. The PullPoint interface provides the means
880 by which NotificationProducers can insert Notification Messages into the PullPoint, by which
881 requestors can retrieve messages from the PullPoint and the means by which requestors can
882 destroy the PullPoint.

883 A PullPoint MAY be a WS-Resource, and if it is, the PullPoint WS-Resource MUST support the
884 immediate termination interface defined by WS-RF Resource Lifetime and it MAY support the
885 scheduled termination interface defined by WS-RF Resource Lifetime. If the PullPoint supports
886 the scheduled termination interface, the initial termination time is set by the factory that creates it.
887 This termination time may be queried and set via the standard WS-ResourceLifetime message
888 exchanges. PullPoints SHOULD NOT be destroyed if subscriptions involving them still exist.

889

Deleted: Note: the

Deleted: request messages defined in this specification MUST follow the WS-Resource Access Pattern defined by [WS-Resource] and

5.1.1 Accumulating Notification Messages

890

891 The PullPoint interface supports the NotificationConsumer interface (as defined in section 3). This
892 interface allows NotificationProducers to send Notification Messages to the PullPoint using using
893 the Notify message format. Notification Messages received by the PullPoint through its
894 NotificationConsumer interface are accumulated by the PullPoint on a best effort basis. If the
895 PullPoint is no longer capable of accumulating additional Notification Messages, it MAY ignore
896 Notification Messages sent to its NotificationConsumer interface, or it MAY dispose of any
897 previously accumulated Notification Messages and continue to accumulate incoming Notification
898 Messages. The PullPoint interface does not define additional constraints on its use of the
899 NotificationConsumer interface.

Deleted: either a “raw” or a “wrapped” approach.

5.1.2 GetMessage

900

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

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

```
906 <wsnt:GetMessages>  
907   <wsnt:MaximumNumber>xsd:nonNegativeInteger</wsnt:MaximumNumber> ?  
908   {any} *  
909 </wsnt:GetMessages>
```

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

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

Deleted: 1

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

915 /wsnt:GetMessages

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

918 /wsnt:GetMessages/wsnt:MaximumNumber

919 This optional non-negative integer indicates the maximum number of accumulated
920 Notification Messages to be returned in the response message. If the total number of
921 Notification Messages accumulated by the PullPoint is less than or equal to this number,
922 all of the Notification Messages are returned in the response message. If the total
923 number of accumulated Notification Messages exceeds this value, the PullPoint MUST
924 return exactly this number of accumulated Notification Messages; the implementation of
925 the PullPoint is free to determine which subset of messages are returned in the response
926 message in this case. If the MaximumNumber component is not present in the request
927 message, the requestor is asking to receive all messages held by the PullPoint.

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

```
929 <wsnt:GetMessagesResponse>  
930   <wsnt:NotificationMessage /> *  
931   {any} *  
932 </wsnt:GetMessagesResponse>
```

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

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

Deleted: 1

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

938 /wsnt:GetMessagesResponse

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

944 /wsnt:GetMessagesResponse/wsnt:NotificationMessage

945 The content of this component is an accumulated NotificationMessage. The
946 NotificationMessage component is described as part of the Notify message (section 3.2)
947 the GetMessagesResponse message does not define additional constraints on the
948 NotificationMessage component. The content of the NotificationMessage is exactly the
949 content of the NotificationMessage component of the accumulated Notify message.

Deleted: If the accumulated NotificationMessage was added to the PullPoint through a Notify message, the contents of the NotificationMessage is exactly the contents of the NotificationMessage component of the Notify message. If the accumulated NotificationMessage was added to the PullPoint through a "raw" message, then the contents of the NotificationMessage is formed by inserting the body of the "raw" message into the wsnt:Message component of the /wsnt:GetMessagesResponse/ wsnt:NotificationMessage.

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

953 ResourceUnknownFault

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

Deleted: (which follows the
WS-Resource Access Pattern)

956 5.1.3 Example SOAP Encoding of the GetMessages Message 957 Exchange

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

```
959 <s:Envelope ... >  
960 <s:Header>  
961 <wsa:Action>  
962 http://docs.oasis-open.org/wsn/bw-2/PullPoint/GetMessagesRequest  
963 </wsa:Action>  
964 ...  
965 </s:Header>  
966 <s:Body>  
967 <wsnt:GetMessages>  
968 <wsnt:MaximumNumber>10</wsnt:MaximumNumber>  
969 </wsnt:GetMessages>  
970 </s:Body>  
971 </s:Envelope>
```

Deleted: 1

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

```
973 <s:Envelope ... >  
974 <s:Header>  
975 <wsa:Action>  
976 http://docs.oasis-open.org/wsn/bw-2/PullPoint/GetMessagesResponse  
977 </wsa:Action>  
978 ...  
979 </s:Header>  
980 <s:Body>  
981 <wsnt:GetMessagesResponse>  
982 <wsnt:NotificationMessage>  
983 ...  
984 </wsnt:NotificationMessage>  
985 <wsnt:NotificationMessage>  
986 ...  
987 </wsnt:NotificationMessage>  
988 </wsnt:GetMessagesResponse>  
989 </s:Body>  
990 </s:Envelope>
```

Deleted: 1

991 5.1.4 DestroyPullPoint

992 The PullPoint interface provides a destroy operation, providing a means by which a requestor can
993 terminate the PullPoint resource. To terminate a PullPoint resource, a requestor MUST send a
994 DestroyPullPoint request message to the PullPoint. The DestroyPullPoint request message has
995 the following form:

Deleted: Destroy

Deleted: Destroy

```
996 <wsnt:DestroyPullPoint>  
997 {any} *  
998
```

999
1000

```
</wsnt:DestroyPullPoint>
```

1001 The WS-Addressing [action] Message Addressing Property MUST contain the URI

http://docs.oasis-open.org/wsn/bw-2/PullPoint/DestroyPullPointRequest.

1002

1003 Upon receipt of the DestroyPullPoint request, the PullPoint MUST attempt to destroy itself. If the
1004 DestroyPullPoint request message is successfully processed, the PullPoint MUST respond with
1005 the following message:

1006
1007
1008
1009
1010

```
<wsnt:DestroyPullPointResponse>  
{any} *  
</wsnt:DestroyPullPointResponse>
```

1011 The WS-Addressing [action] Message Addressing Property MUST contain the URI

http://docs.oasis-open.org/wsn/bw-2/PullPoint/DestroyPullPointResponse.

1012

1013 If the PullPoint does not respond to the DestroyPullPoint request message with the
1014 DestroyPullPointResponse message, then it MUST send a fault. This specification defines the
1015 following faults associated with failure to process the DestroyPullPoint request message:

1016 UnableToDestroyPullPointFault.

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

1017

5.1.5 Example SOAP Encoding of the DestroyPullPoint Message Exchange

1018
1019

1020 The following is a non-normative example of a DestroyPullPoint request message using SOAP:

1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032

```
<s:Envelope ... >  
<s:Header>  
<wsa:Action>  
http://docs.oasis-open.org/wsn/bw-  
2/PullPoint/DestroyPullPointRequest  
</wsa:Action>  
...  
</s:Header>  
<s:Body>  
<wsnt:DestroyPullPoint/>  
</s:Body>  
</s:Envelope>
```

1033 The following is a non-normative example of a DestroyPullPoint response message using SOAP:

1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045

```
<s:Envelope ... >  
<s:Header>  
<wsa:Action>  
http://docs.oasis-open.org/wsn/bw-  
2/PullPoint/DestroyPullPointResponse  
</wsa:Action>  
...  
</s:Header>  
<s:Body>  
<wsnt:DestroyPullPointResponse/>  
</s:Body>  
</s:Envelope>
```

Deleted: 1

Deleted: DestroyRequest.

Deleted: Destroy

Deleted: Destroy

Deleted: DestroyResponse

Deleted: DestroyResponse

Deleted: 1

Deleted: DestroyResponse.

Deleted: Destroy

Deleted: DestroyResponse

Deleted: Destroy

Deleted: Destroy

Deleted: Destroy

Deleted: 1

Deleted: DestroyRequest

Deleted: Destroy

Deleted: 1

Deleted: DestroyResponse

Deleted: DestroyResponse

1046 5.2 Create PullPoint Interface

1047 This specification describes a CreatePullPoint interface standardizing the means by which a
1048 PullPoint resource is created.

1049 If a requestor wishes to create a new PullPoint resource, it MUST send a CreatePullPoint request
1050 to an endpoint supporting the PullPoint interface. The CreatePullPoint request message has the
1051 following form:

```
1052 <wsnt:CreatePullPoint>  
1053   {any} *  
1054 </wsnt:CreatePullPoint>  
1055  
1056
```

1057 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1058 <http://docs.oasis-open.org/wsn/bw-2/CreatePullPoint/CreatePullPointRequest>

Deleted: 1

1059 Upon receipt of the CreatePullPoint request, the receiver MUST attempt to create a new PullPoint
1060 resource. If the CreatePullPoint request message is successfully processed, the receiver MUST
1061 respond with the following message:

```
1062 <wsnt:CreatePullPointResponse>  
1063   <wsnt:PullPoint>wsa:EndpointReferenceType</wsnt:PullPoint>  
1064   {any} *  
1065 </wsnt:CreatePullPointResponse>  
1066  
1067
```

1068 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1069 <http://docs.oasis-open.org/wsn/bw-2/CreatePullPoint/CreatePullPointResponse>.

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

1070 The contents of the CreatePullPoint response message are further described as follows:

1071 /wsnt:CreatePullPointResponse/wsnt:PullPoint

1072 This component is an EndpointReference, as defined by WS-Addressing, that is a
1073 reference to the PullPoint resource created during the processing of the CreatePullPoint
1074 request message.

1075 A PullPoint created by the CreatePullPoint operation MAY choose to ignore Notifications received
1076 through the NotificationConsumer interface. A Pull Point MAY accumulate Notifications through
1077 other, implementation-defined mechanisms. Pull point factories SHOULD indicate through policy
1078 assertions whether pull points they create will ignore Notifications received through the
1079 NotificationConsumer interface.

1080 If the receiver does not respond to the CreatePullPoint request message with the
1081 CreatePullPointResponse message, then it MUST send a fault. This specification defines the
1082 following fault, associated with failure to process the DestroyPullPoint request message:

Deleted: s

- 1083 • UnableToCreatePullPointFault.

Deleted: Destroy

1084

1085 5.2.1 Example SOAP Encoding of the CreatePullPoint Message 1086 Exchange

1087 The following is a non-normative example of a CreatePullPoint request message using SOAP:

```
1088 <s:Envelope ... >  
1089   <s:Header>
```

1090
1091
1092
1093
1094
1095
1096
1097
1098

```
<wsa:Action>  
http://docs.oasis-open.org/wsn/bw-2/PullPoint/CreatePullPointRequest  
</wsa:Action>  
...  
</s:Header>  
<s:Body>  
  <wsnt:CreatePullPoint/>  
</s:Body>  
</s:Envelope>
```

Deleted: 1

1099 The following is a non-normative example of a CreatePullPoint response message using SOAP:

1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115

```
<s:Envelope ... >  
  <s:Header>  
    <wsa:Action>  
      http://docs.oasis-open.org/wsn/bw-2/PullPoint/CreatePullPointResponse  
    </wsa:Action>  
    ...  
  </s:Header>  
  <s:Body>  
    <wsnt:CreatePullPointResponse>  
      <wsnt:PullPoint>  
        <wsa:Address>...</wsa:Address>  
        ...  
      </wsnt:PullPoint>  
    </wsnt:CreatePullPointResponse>  
  </s:Body>  
</s:Envelope>
```

Deleted: 1

6 SubscriptionManager Interface

1116

1117 The SubscriptionManager interface defines message exchanges to manipulate Subscription
1118 resources. There are two styles of SubscriptionManager interface: base and pausable. All
1119 SubscriptionManagers MUST implement the message exchanges described in the Base
1120 Subscription Manager section.

1121 For those environments that choose to implement a Subscription as a WS-Resource, the
1122 SubscriptionManager MUST implement the resource properties and message exchanges defined
1123 in Subscriptions as WS-Resources section. Both the Base Subscription Manager and the
1124 Pausable Subscription Manager MAY expose Subscriptions as WS-Resources.

6.1 Base Subscription Manager

1125
1126 The basic behavior of a SubscriptionManager is to renew the duration of a Subscription resource
1127 and terminate a Subscription.

6.1.1 Renew

1128
1129 To modify the current lifetime of a Subscription, a requestor sends a Renew request message to
1130 the SubscriptionManager. The Renew request message MUST have the following form:

```
1131 ...  
1132 <wsnt:Renew>  
1133   <wsnt:TerminationTime>  
1134     [xsd:dateTime | xsd:duration]  
1135   </wsnt:TerminationTime>  
1136   {any}  
1137 </wsnt:Renew>  
1138 ...
```

1139 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1140 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewRequest>.

Deleted: 1

1141 The components of the Renew request message are further described as follows:

1142 [/wsnt:Renew/wsnt:TerminationTime](#)

1143 This component contains the service requestor's suggestion for change in expiration, or
1144 termination time of the Subscription resource. There are two forms of this component,
1145 absolute time and duration. If the type of this component is xsd:dateTime, the value of the
1146 component is to be interpreted as an "absolute time". If the type of this component is
1147 xsd:duration, the value of the component is to be interpreted as a "duration".

1148 The duration form is used to "compute" the "absolute time" form in the following fashion. The
1149 value of this component in "absolute time" form is computed by adding the xsd:duration value
1150 to the current time value of the SubscriptionManager.

1151 The "absolute time" form (whether computed from a duration, or contained within the request
1152 message) is used to reset the value of the expiration or termination time component of the
1153 Subscription resource.

1154 If the SubscriptionManager is unable or unwilling to reset the termination time component of
1155 the Subscription resource to the given value of the "absolute time" form or a value greater,

1156 then the Renew request MUST fault. If the value is not "in the future" relative to the current
1157 time as known by the SubscriptionManager, the Renew request MUST fault.
1158 The use of the xsi:nil attribute with value "true" indicates there is no scheduled termination
1159 time requested for the Subscription, implying that the Subscription should be renewed to
1160 have an infinite duration. If the element does not include the time zone designation, the value
1161 of the element MUST be interpreted as universal time (UTC) time.

1162 If the SubscriptionManager successfully processes the Renew request message, it MUST
1163 respond with a RenewResponse message. This response has the following form:

```
1164 ...  
1165 <wsnt:RenewResponse>  
1166   <wsnt:TerminationTime>xsd:dateTime</wsnt:TerminationTime>  
1167   <wsnt:CurrentTime>xsd:dateTime</wsnt:CurrentTime?>  
1168   {any}  
1169 </wsnt:RenewResponse>  
1170 ...
```

1171 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1172 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewResponse>.

Deleted: 1

1173 The contents of the RenewResponse message are further described as follows:

1174 [/wsnt:RenewResponse/wsnt:TerminationTime](#)

1175 The value of this component is the value of the termination time component of the
1176 Subscription at the time the response message is created.

1177 [/wsnt:RenewResponse/wsnt:CurrentTime](#)

1178 This OPTIONAL component contains the value of the CurrentTime according to the
1179 SubscriptionManager at the time the response message is created.

1180 If the SubscriptionManager does not respond to the Renew request message with the
1181 RenewResponse message, then it MUST send a fault. This specification defines the following
1182 faults associated with failure to process the Renew request message:

1183 ResourceUnknownFault

1184 • The SubscriptionManager is acting as a WS-Resource, and the resource identified in the
1185 message is not known to the Web service. This fault is specified by the WS-Resource
1186 [WS-Resource] specification.

Deleted: (which follows the
WS-Resource Access Pattern)

1187 UnacceptableTerminationTimeFault

1188 • The value of InitialTerminationTime specified in the Subscribe message was not
1189 acceptable to the NotificationProducer. The NotificationProducer MAY include a hint in
1190 the fault message indicating acceptable values for InitialTerminationTime.

1191 **6.1.1.1 Example SOAP Encoding of the Renew Message Exchange**

1192 The following is a non-normative example of a Renew request message using SOAP:

```
1193 <s:Envelope ... >  
1194   <s:Header>  
1195     <wsa:Action>  
1196       http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewRequest  
1197     </wsa:Action>  
1198     ...  
1199   </s:Header>  
1200   <s:Body>  
1201     <wsnt:Renew>
```

Deleted: 1/

```

1202     <wsnt:TerminationTime>PlD</wsnt:TerminationTime>
1203     </wsnt:Renew>
1204     </s:Body>
1205 </s:Envelope>

```

1206 The following is a non-normative example of a Renew response message using SOAP:

```

1207 <s:Envelope ... >
1208   <s:Header>
1209     <wsa:Action>
1210     http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewResponse
1211     </wsa:Action>
1212     ...
1213   </s:Header>
1214   <s:Body>
1215     <wsnt:RenewResponse>
1216       <wsnt:TerminationTime>
1217         2005-12-26T00:00:00.00000Z
1218       </wsnt:TerminationTime>
1219     </wsnt:RenewResponse>
1220   </s:Body>
1221 </s:Envelope>

```

Deleted: 1/

1222 6.1.2 Unsubscribe

1223 To terminate a Subscription, a requestor sends an Unsubscribe request message to the
 1224 SubscriptionManager. The Unsubscribe request message MUST have the following form:

```

1225 ...
1226 <wsnt:Unsubscribe>
1227   {any}
1228 </wsnt:Unsubscribe>
1229 ...

```

1230 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1231 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeRequest>.

Deleted: 1

1232 Upon receipt of the Unsubscribe request, the SubscriptionManager MUST attempt to destroy the
 1233 Subscription resource. If the SubscriptionManager successfully processes the Unsubscribe
 1234 request message, it MUST respond with the following message:

```

1235 ...
1236 <wsnt:UnsubscribeResponse>
1237   {any}
1238 </wsnt:UnsubscribeResponse>
1239 ...

```

1240 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1241 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeResponse>.

Deleted: 1

1242 If the SubscriptionManager does not respond to the Unsubscribe request message with the
 1243 UnsubscribeResponse message, then it MUST send a fault. This specification defines the
 1244 following faults associated with failure to process the Unsubscribe request message:

1245 ResourceUnknownFault

- 1246 • The SubscriptionManager is acting as a WS-Resource, and the resource identified in the
 1247 message is not known to the Web service. This fault is specified by the WS-Resource
 1248 [WS-Resource] specification.

Deleted: (which follows the WS-Resource Access Pattern)

- 1249 UnableToDestroySubscriptionFault
1250 • The SubscriptionManager was unable to destroy the Subscription resource for some
1251 reason.

1252 6.1.2.1 Example SOAP Encoding of the Unsubscribe Message 1253 Exchange

1254 The following is a non-normative example of an Unsubscribe request message using SOAP:

```
1255 <s:Envelope ... >  
1256 <s:Header>  
1257 <wsa:Action>  
1258 http://docs.oasis-open.org/wsn/bw-  
1259 2/SubscriptionManager/UnsubscribeRequest  
1260 </wsa:Action>  
1261 ...  
1262 </s:Header>  
1263 <s:Body>  
1264 <wsnt:Unsubscribe />  
1265 </s:Body>  
1266 </s:Envelope>
```

Deleted: 1/

1267 The following is a non-normative example of an Unsubscribe response message using SOAP:

```
1268 <s:Envelope ... >  
1269 <s:Header>  
1270 <wsa:Action>  
1271 http://docs.oasis-open.org/wsn/bw-  
1272 2/SubscriptionManager/UnsubscribeResponse  
1273 </wsa:Action>  
1274 ...  
1275 </s:Header>  
1276 <s:Body>  
1277 <wsnt:UnsubscribeResponse />  
1278 </s:Body>  
1279 </s:Envelope>
```

Deleted: 1/

1280 6.2 Pausable Subscription Manager

1281 A Pausable Subscription Manager implements all the message exchanges defined in the Base
1282 Subscription Manager and augments this with a set of advanced capabilities to allow third parties
1283 to pause and resume subscriptions.

1284 6.2.1 PauseSubscription

1285 To temporarily suspend the production of Notifications on the given Subscription, a requestor
1286 MAY send a PauseSubscription request message to the SubscriptionManager. The
1287 PauseSubscription request message has the following form:

```
1288 ...  
1289 <wsnt:PauseSubscription>{any}</wsnt:PauseSubscription>  
1290 ...
```

1291 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1292 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/PauseSubscriptionRequest>.

Deleted: 1

1293 Upon successful processing of this message the Subscription is in the *paused* state. Production
1294 of Notifications can be resumed by sending the ResumeSubscription request message (see 6.3).

1295 In an asynchronous implementation, a NotificationProducer may continue producing notifications
1296 while the PauseSubscription request is in transit. In general, there is no way of knowing exactly
1297 when the pause will take effect.

1298 Note: transitioning between the paused state and the resumed state has no effect on the
1299 TerminationTime of the Subscription resource. A Subscription's TerminationTime may elapse
1300 while it is in a paused state, initiating the scheduled destruction of the Subscription resource.

1301 When a Subscription is resumed after having been paused, the NotificationProducer MAY choose
1302 from one of the courses of action listed below, or any other action. NotificationProducers MAY
1303 advertise their behavior in this situation via policy assertions. In the absence of a specific policy
1304 assertion, Subscribers SHOULD NOT assume any particular behavior on the part of the
1305 NotificationProducer. Possible actions include

- 1306 1. Produce Notifications for all the Situations related to the Subscription that occurred while
1307 the Subscription was paused (as well as for any new Situations that occur after the
1308 Subscription has been resumed).
- 1309 2. Produce a Notification for the last Situation that occurred while the Subscription was
1310 paused (as well as for any new situations that occur after the topic has been resumed).
- 1311 3. Produce no Notifications until a Situation occurs after the Subscription has been
1312 resumed.

1313 If the SubscriptionManager accepts the PauseSubscription request message, it MUST respond
1314 with a message of the following form:

```
1315 ...  
1316 <wsnt:PauseSubscriptionResponse>{any}</wsnt:PauseSubscriptionResponse>  
1317 ...
```

1318 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1319 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/PauseSubscriptionResponse>.

Deleted: 1

1320 If the SubscriptionManager does not respond to the PauseSubscription request message with the
1321 PauseSubscription response message, then it MUST send a fault. This specification defines the
1322 following faults associated with failure to process the PauseSubscription request message:

1323 ResourceUnknownFault

- 1324 • The resource identified in the message is not known to the Web service. This fault is
1325 specified by the WS-Resource [WS-Resource] specification.

1326 PauseFailedFault

- 1327 • The Pause operation could not be performed on the Subscription.

1328 6.2.2 Example SOAP Encoding of the PauseSubscription Message 1329 Exchange

1330 The following is a non-normative example of a PauseSubscription request message using SOAP:

```
1331 <s:Envelope ... >  
1332 <s:Header>  
1333 <wsa:Action>  
1334 http://docs.oasis-open.org/wsn/bw-  
1335 2/SubscriptionManager/PauseSubscriptionRequest  
1336 </wsa:Action>  
1337 ...  
1338 </s:Header>
```

Deleted: 1/

1339
1340
1341
1342

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

1343 The following is a non-normative example of a PauseSubscription response message using
1344 SOAP:

1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356

```
<s:Envelope ... >
  <s:Header>
    <wsa:Action>
      http://docs.oasis-open.org/wsn/bw-
2/SubscriptionManager/PauseSubscriptionResponse
    </wsa:Action>
    ...
  </s:Header>
  <s:Body>
    <wsnt:PauseSubscriptionResponse/>
  </s:Body>
</s:Envelope>
```

Deleted: 1/

1357 6.3 ResumeSubscription

1358 If a requestor wishes to resume the production of Notifications on the given Subscription, it must
1359 send a ResumeSubscription request message. The ResumeSubscription request message has
1360 the following form:

1361
1362
1363
1364
1365

```
...
<wsnt:ResumeSubscription>
  {any}
</wsnt:ResumeSubscription>
...
```

1366 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1367 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/ResumeSubscriptionRequest>.

Deleted: 1

1368 Upon successful processing of this message the Subscription is no longer in the *paused* state.

1369 If a Subscription is not in the *paused* state, and it is the subject of a ResumeSubscription request
1370 message, no change occurs.

1371 If the SubscriptionManager accepts the ResumeSubscription request message, it MUST respond
1372 with a message of the following form:

1373
1374
1375
1376
1377

```
...
<wsnt:ResumeSubscriptionResponse>
  {any}
</wsnt:ResumeSubscriptionResponse>
...
```

1378 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1379 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/ResumeSubscriptionResponse>.

Deleted: 1

1381 If the SubscriptionManager does not respond to the ResumeSubscriptionMessage request
1382 message with the ResumeSubscriptionResponse message, then it MUST send a fault. This
1383 specification defines the following faults associated with failure to process the
1384 ResumeSubscriptionMessage request message:

1385 ResourceUnknownFault

- 1386 • The resource identified in the message is not known to the Web service. This fault is
1387 specified by the WS-Resource [WS-Resource] specification.

1388 ResumeFailedFault

- 1389 • The Resume operation could not be performed on the Subscription.

1390 6.3.1 Example SOAP Encoding of the ResumeSubscription Message 1391 Exchange

1392 The following is a non-normative example of a ResumeSubscription request message using
1393 SOAP:

```
1394 <s:Envelope ... >  
1395   <s:Header>  
1396     <wsa:Action>  
1397       http://docs.oasis-open.org/wsn/bw-  
1398 2/SubscriptionManager/ResumeSubscriptionRequest  
1399     </wsa:Action>  
1400     ...  
1401   </s:Header>  
1402   <s:Body>  
1403     <wsnt:ResumeSubscription/>  
1404   </s:Body>  
1405 </s:Envelope>
```

Deleted: 1/

1406 The following is a non-normative example of a ResumeSubscription response message using
1407 SOAP:

```
1408 <s:Envelope ... >  
1409   <s:Header>  
1410     <wsa:Action>  
1411       http://docs.oasis-open.org/wsn/bw-  
1412 2/SubscriptionManager/ResumeSubscriptionResponse  
1413     </wsa:Action>  
1414     ...  
1415   </s:Header>  
1416   <s:Body>  
1417     <wsnt:ResumeSubscriptionResponse/>  
1418   </s:Body>  
1419 </s:Envelope>
```

Deleted: 1/

1420 6.4 Subscriptions as WS-Resources

1421 Either Base Subscription Managers or Pausable Subscription Managers MAY expose
1422 subscriptions as WS-Resources. A Subscription Manager that exposes Subscriptions as WS-
1423 Resources MUST support the required message exchanges associated with the WS-
1424 ResourceProperties specification [WS-ResourceProperties] and MAY support the optional
1425 message exchanges defined by WS-ResourceProperties.

1426 It is RECOMMENDED that Subscription Managers that expose Subscriptions as WS-Resources
1427 support the message exchanges and resource properties defined for both forms of resource
1428 lifetime (immediate and scheduled destruction) by WS-ResourceLifetime [WS-ResourceLifetime].
1429 These message exchanges MAY be used to define the means by which Subscription resources
1430 can be explicitly destroyed, or destroyed using a scheduled (time-based) mechanism. Such
1431 Subscription Managers MAY support other means of destroying subscriptions.

1432 Subscription Managers MUST (for example in its WSDL definition) declare to requestors which
1433 means of Subscription destruction is supported.

1434 6.4.1 Subscription Resource Properties

1435 In cases where subscriptions are exposed as WS-Resources, this specification defines several
1436 resource properties, which MUST conform to the following schema fragment for content:

```
1437 ...
1438 | targetNamespace="http://docs.oasis-open.org/wsn/b-2"
1439 | ...
1440 |
1441 | <xsd:element name="ConsumerReference"
1442 |             type="wsa:EndpointReferenceType" />
1443 | <xsd:element name="Filter"
1444 |             type="wsnt:FilterType" />
1445 | <xsd:element name="SubscriptionPolicy"
1446 |             type="wsnt:SubscriptionPolicyType" />
1447 | <xsd:element name="CreationTime"
1448 |             type="xsd:dateTime" />
```

Deleted: 1

1449 These properties must also conform to the following schema fragment for cardinality:

```
1450 <xsd:element ref="wsnt:ConsumerReference"
1451             minOccurs="1" maxOccurs="1" />
1452 <xsd:element ref="wsnt:Filter"
1453             minOccurs="0" maxOccurs="1" />
1454 <xsd:element ref="wsnt:SubscriptionPolicy"
1455             minOccurs="0" maxOccurs="1" />
1456 <xsd:element ref="wsnt:CreationTime"
1457             minOccurs="0" maxOccurs="1" />
```

1458 These Resource Property elements are further described as follows:

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

1460 These elements are as defined in the description of the Subscribe request message (see
1461 4.2). SubscriptionManagers MAY prohibit any combination of these properties from being set.
1462 If setting a given property is disallowed, the SubscriptionManager MUST fault on receiving a
1463 request to change it. SubscriptionManagers SHOULD describe using metadata techniques
1464 defined by WS-ResourceFramework which properties may be set.

1465 If any of these properties is settable, it is implementation dependent whether production of
1466 notifications will be attempted without interruption when a change is made. The
1467 SubscriptionManager may choose to treat the change in properties as equivalent to a
1468 cancellation followed by the creation of a new subscription. In this case, messages may
1469 potentially be lost in the interim.

1470 A SubscriptionManager MAY choose to provide a stronger guarantee of continuity of
1471 notification production, and if so it SHOULD advertise this capability, for example in a policy
1472 assertion. In the absence of any such policy assertion, a subscriber SHOULD NOT make
1473 any assumption of continuity.

1474 /wsnt:CreationTime

1475 Indicates the date and time at which the Subscription was created. This is an optional
1476 component, supporting resource constrained devices which cannot associate a creation time
1477 with subscription resources they create. This property is read-only. SubscriptionManagers
1478 MUST fault on receiving a request to change it.

1479 If the SubscriptionManager also accepts the SetResourceProperties request message as defined
1480 in WS-ResourceProperties, the following properties MAY be modified by the requestor:

1481 /wsnt:ConsumerReference and /wsnt:Filter and /wsnt:SubscriptionPolicy

1482

7 Security Considerations

1483 This section deals with the security aspects of WS-BaseNotification. It deals with (a) securing the
1484 message exchanges defined in this specification, and (b) authorization and denial of service
1485 considerations specific to the Notification pattern.

7.1 Securing the Message Exchanges

1487 In the Notification pattern, Notifications are sent to a NotificationConsumer, Subscribers
1488 exchange SubscriptionRequest and SubscriptionResponse messages with NotificationProducers,
1489 and any party with access to a SubscriptionManager endpoint may perform operations on the
1490 underlying Subscription resources. In cases where this communication must be secured it is
1491 RECOMMENDED that this be done using the mechanisms described in WS-Security.

1492 Communication between a NotificationProducer and NotificationConsumer will typically comprise
1493 a number of Notification messages. In cases where this communication must be secure, it is
1494 RECOMMENDED that a security context be established, allowing for potentially more efficient
1495 means of authentication. Note that the keys used to secure this channel may differ from any keys
1496 used in the Subscribe request/response operation that created the associated Subscription.

7.2 Securing Subscriptions and Notifications

1497
1498 As WS-BaseNotification provides mechanisms for subscribing to topics, security policies should
1499 be established such that

- 1500 1. only authorized principals can subscribe to receive Notifications
- 1501 2. only authorized principals can modify or delete Subscriptions

1502 It is recommended that the authorization policies be specified at the granularity of the Topic, if
1503 Topics are supported. It should be noted that even though Subscriptions may be done by
1504 authorized principals, the Notifications may be produced for NotificationConsumers whose
1505 identity may be different from the Subscriber. Message protection policies as outlined in the
1506 previous section can be used to ensure that sensitive Notifications are not delivered to malicious
1507 endpoints. For example, a key may need to be specified or generated during the process of
1508 Subscription, so that the Notifications can be encrypted using the key to ensure confidentiality of
1509 the messages. The mechanism by which the key is specified is governed by the Subscription
1510 policy.

1511 Given that WS-BaseNotification may use WS-ResourceProperties and WS-ResourceLifetime, the
1512 security considerations outlined in WS- those specifications need to be taken into account where
1513 appropriate. Authorization policies for those Resource Properties should be put in place so that
1514 the implications of providing the state information (through GetResourceProperty request
1515 messages) or through notification of state change and modification of the resource properties
1516 (through SetResourceProperty request messages), are taken into account.

1517 This specification provides a mechanism by which Subscribers can specify a subscription policy.
1518 Such a policy may contain security policy about protecting the message exchanges resulting from
1519 the Subscription. Security policy for Subscription message exchanges needs to take this into
1520 consideration so that the Subscription policies are protected. Also, given this policy may be
1521 contained in the resource properties of the subscription maintained by the SubscriptionManager,
1522 the resource properties must be appropriately secured.

1523 In addition to the usual concerns of authorization and message integrity which apply to all web
1524 services, notification presents issues all its own due to the third-party nature of subscription.

1525 Since the NotificationProducer is agreeing to produce Notifications for a consumer based on the
1526 requests of a Subscriber, it must assure itself that there is no harm in producing these
1527 Notifications. A malicious Subscriber may request Notifications be sent to a party that is not
1528 authorized to receive them. It may also mount DOS attacks by requesting large volumes of
1529 Notifications be sent to parties that cannot handle them.

1530
1531 The NotificationProducer may address these risks in many different ways, including but not
1532 limited to:

- 1533 • Simply trusting all Subscribers, perhaps because all parties are known to be on a closed,
1534 trusted, network, or because the consequences of unauthorized Subscriptions are
1535 otherwise known to be negligible.
- 1536 • Requiring all Subscribers to provide secure credentials proving that they are trusted to
1537 make subscriptions.
- 1538 • Refusing to produce notifications for NotificationConsumers that are not known to be
1539 authorized.
- 1540 • Explicitly confirming with NotificationConsumers whether they wish to receive the
1541 Notifications that the Subscriber has requested.
- 1542 • Some combination of the above, depending on the identity of the Subscriber and
1543 NotificationProducer

1544 NotificationProducers SHOULD advertise, whether through policy assertions or other means,
1545 what security measures they take.

1546

8 References

1547

8.1 Normative

1548

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.

1549

1550

[URI] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.

Deleted:

1551

1552

1553

[WS-Addressing] <http://www.w3.org/TR/ws-addr-core>

1554

[WS-BrokeredNotification] http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-pr-02.pdf

Deleted:

1555

1556

[WS-Resource] http://docs.oasis-open.org/wsr/wsrf-ws_resource-1.2-spec-pr-02.pdf

Deleted: cd

1557

[WS-ResourceLifetime] http://docs.oasis-open.org/wsr/wsrf-ws_resource_lifetime-1.2-spec-pr-02.pdf

Deleted: 1

1558

1559

[WS-ResourceProperties] http://docs.oasis-open.org/wsr/wsrf-ws_resource_properties-1.2-spec-pr-02.pdf

Deleted: cd-01

1560

1561

[WS-BaseFaults] http://docs.oasis-open.org/wsr/wsrf-ws_base_faults-1.2-spec-pr-02.pdf

Deleted: [WS-ResourceProperties] http://docs.oasis-open.org/wsr/wsrf-ws_resource_properties-1.2-spec-cd-01.pdf

1562

[WS-Topics] http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-pr-01.pdf

1563

[XML-InfoSet] <http://www.w3.org/TR/xml-infoset/>

1564

[XPath] <http://www.w3.org/TR/xpath>

Deleted: <http://docs.oasis-open.org/wsn/wsn-WS-Topics-1.3-cd-01.pdf>

1565

8.2 Non-Normative

1566

[SOAP 1.1] <http://www.w3.org/TR/SOAP>

1567

[SOAP 1.2] <http://www.w3.org/TR/soap12-part1/>

1568

[WS-Security] <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

Deleted:

1569

1570

Appendix A. Acknowledgments

1571 Special thanks to the Global Grid Forum's Open Grid Services Infrastructure working group,
1572 which defined the OGSi v1.0 [OGSI] specification which was a large inspiration for the ideas
1573 expressed in this specification.

1574 The following individuals were members of the committee during the development of this
1575 specification:

1576

1577 Sid Askary, Fred Carter (AmberPoint), Martin Chapman (Oracle), Dave Chappell (Sonic
1578 Software), Rick Cobb (KnowNow), Ugo Corda (SeeBeyond Technology Corporation), John Fuller,
1579 Stephen Graham (IBM), David Hull (Tibco), Hideharu Kato (Hitachi), Lily Liu (webMethods, Inc.),
1580 Tom Maguire (IBM), Susan Malaika (IBM), Samuel Meder (Argonne National Laboratory), Bryan
1581 Murray (Hewlett-Packard), Peter Niblett (IBM), Sanjay Patil (SAP), Mark Peel (Novell), Matt
1582 Roberts (IBM), Igor Sedukhin (Computer Associates), David Snelling (Fujitsu), Latha Srinivasan
1583 (Hewlett-Packard), William Vambenepe (Hewlett-Packard) and Kirk Wilson (Computer
1584 Associates).

1585

1586 In addition, the following people made contributions to this specification:

1587 Tim Banks (IBM), Nick Butler (IBM), Doug Davis (IBM), John Dinger (IBM), Don Ferguson (IBM),
1588 Jeff Frey (IBM), Andreas Koeppel (SAP), Heather Kreger (IBM), Amy Lewis (TIBCO Software),
1589 Kevin Liu (SAP), Nataraj Nagaratnam (IBM), Martin Nally (IBM), Jeff Nick (IBM), Jay Parikh
1590 (Akamai Technologies), Claus von Riegen (SAP), Rick Rineholt (IBM), John Rofrano (IBM),
1591 Shivajee Samdarshi (TIBCO Software), Eugène Sindambiwe (SAP), Jay Unger (IBM), Bill Weihl
1592 (Akamai Technologies), Mark Weitzel (IBM), and Dan Wolfson (IBM).

Appendix B. XML Schema

1593

1594 The XML types and elements used in this specification are defined in the following XML Schema:

```
1595 <?xml version="1.0" encoding="UTF-8"?>
1596 <!--
1597
1598 OASIS takes no position regarding the validity or scope of any
1599 intellectual property or other rights that might be claimed to pertain
1600 to the implementation or use of the technology described in this
1601 document or the extent to which any license under such rights might or
1602 might not be available; neither does it represent that it has made any
1603 effort to identify any such rights. Information on OASIS's procedures
1604 with respect to rights in OASIS specifications can be found at the
1605 OASIS website. Copies of claims of rights made available for
1606 publication and any assurances of licenses to be made available, or the
1607 result of an attempt made to obtain a general license or permission for
1608 the use of such proprietary rights by implementors or users of this
1609 specification, can be obtained from the OASIS Executive Director.
1610
1611 OASIS invites any interested party to bring to its attention any
1612 copyrights, patents or patent applications, or other proprietary rights
1613 which may cover technology that may be required to implement this
1614 specification. Please address the information to the OASIS Executive
1615 Director.
1616
1617 Copyright (C) OASIS Open (2004-2005). All Rights Reserved.
1618
1619 This document and translations of it may be copied and furnished to
1620 others, and derivative works that comment on or otherwise explain it or
1621 assist in its implementation may be prepared, copied, published and
1622 distributed, in whole or in part, without restriction of any kind,
1623 provided that the above copyright notice and this paragraph are
1624 included on all such copies and derivative works. However, this
1625 document itself may not be modified in any way, such as by removing the
1626 copyright notice or references to OASIS, except as needed for the
1627 purpose of developing OASIS specifications, in which case the
1628 procedures for copyrights defined in the OASIS Intellectual Property
1629 Rights document must be followed, or as required to translate it into
1630 languages other than English.
1631
1632 The limited permissions granted above are perpetual and will not be
1633 revoked by OASIS or its successors or assigns.
1634
1635 This document and the information contained herein is provided on an
1636 "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
1637 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
1638 INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
1639 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
1640
1641 -->
1642
1643 <xsd:schema
1644   targetNamespace="http://docs.oasis-open.org/wsn/b-2"
1645   xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
1646   xmlns:wsa="http://www.w3.org/2005/08/addressing"
1647   xmlns:wsrif-bf="http://docs.oasis-open.org/wsrif/bf-2"
```

Deleted: 1 "

Deleted: 1 "

Deleted: 03

Deleted: 1 "

1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707

```
xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">

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

<xsd:import namespace="http://www.w3.org/2005/08/addressing"
schemaLocation="http://www.w3.org/2005/08/addressing/ws-
addr.xsd"
/>

<xsd:import namespace="http://docs.oasis-open.org/wsrf/bf-2"
schemaLocation="http://docs.oasis-open.org/wsrf/bf-2.xsd"
/>

<xsd:import namespace="http://docs.oasis-open.org/wsn/t-1"
schemaLocation="http://docs.oasis-open.org/wsn/t-1.xsd"
/>

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

<xsd:complexType name="QueryExpressionType" mixed="true">
<xsd:sequence>
<xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
</xsd:sequence>
<xsd:attribute name="Dialect" type="xsd:anyURI" use="required"/>
</xsd:complexType>

<xsd:complexType name="TopicExpressionType" mixed="true">
<xsd:sequence>
<xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
</xsd:sequence>
<xsd:attribute name="Dialect" type="xsd:anyURI" use="required" />
<xsd:anyAttribute/>
</xsd:complexType>

<xsd:complexType name="FilterType">
<xsd:sequence>
<xsd:any minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SubscriptionPolicyType">
<xsd:sequence>
<xsd:any minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
</xsd:sequence>
</xsd:complexType>

<!-- ===== Resource Property Related ===== -->
<!-- ===== Resource Properties for NotificationProducer ===== -->
<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="NotificationProducerRP">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="wsnt:TopicExpression"
minOccurs="0" maxOccurs="unbounded" />
<xsd:element ref="wsnt:FixedTopicSet"
```

Deleted: 03
Deleted: 03
Deleted: 1"
Deleted: 1

1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767

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

<!-- ===== Resource Properties for SubscriptionManager ===== -->
<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" />

<xsd:element name="SubscriptionManagerRP" >
    <xsd:complexType>
        <xsd:sequence>
            <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" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<!-- ===== Notification Metadata ===== -->
<xsd:element name="SubscriptionReference"
    type="wsa:EndpointReferenceType" />
<xsd:element name="Topic"
    type="wsnt:TopicExpressionType" />
<xsd:element name="ProducerReference"
    type="wsa:EndpointReferenceType" />

<!-- ===== Message Helper Types ===== -->
<xsd:complexType name="NotificationMessageHolderType" >
    <xsd:sequence>
        <xsd:element ref="wsnt:SubscriptionReference"
            minOccurs="0" maxOccurs="1" />
        <xsd:element ref="wsnt:Topic"
            minOccurs="0" maxOccurs="1" />
        <xsd:element ref="wsnt:ProducerReference"
            minOccurs="0" maxOccurs="1" />
        <xsd:element name="Message">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:any namespace="##any" processContents="lax"
                        minOccurs="1" maxOccurs="1"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
```

Deleted: unbounded

Inserted: unbounded" />

Deleted: ">

Deleted:

Deleted: t

Inserted: t

Deleted:
<xs:complexType>¶
 <xs:sequence>¶
 <s:any
minOccurs="0"
maxOccurs="unbounded"
processContents="lax"/>¶
 </xs:sequence>¶
</xs:complexType>¶
</xs:element>¶

1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827

```
</xsd:complexType>
<xsd:element name="NotificationMessage"
  type="wsnt:NotificationMessageHolderType" />
<!-- ===== Message Types for NotificationConsumer ===== -->
<xsd:element name="Notify" >
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="wsnt:NotificationMessage"
        minOccurs="1" maxOccurs="unbounded" />
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!-- ===== Message Types for NotificationProducer ===== -->

<xsd:simpleType name="AbsoluteOrRelativeTimeType">
  <xsd:union memberTypes="xsd:dateTime xsd:duration" />
</xsd:simpleType>

<xsd:element name="CurrentTime" type="xsd:dateTime" />

<xsd:element name="TerminationTime"
  nillable="true" type="xsd:dateTime" />

<xsd:element name="ProducerProperties"
  type="wsnt:QueryExpressionType" />

<xsd:element name="MessageContent"
  type="wsnt:QueryExpressionType" />

<xsd:element name="UseRaw"><xsd:complexType/></xsd:element>

<xsd:element name="Subscribe" >
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="ConsumerReference"
        type="wsa:EndpointReferenceType"
        minOccurs="1" maxOccurs="1" />
      <xsd:element name="Filter"
        type="wsnt:FilterType"
        minOccurs="0" maxOccurs="1" />
      <xsd:element name="InitialTerminationTime"
        type="wsnt:AbsoluteOrRelativeTimeType"
        nillable="true"
        minOccurs="0" maxOccurs="1" />
      <xsd:element name="SubscriptionPolicy"
        minOccurs="0" maxOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:any namespace="##any" processContents="lax"
              minOccurs="0" maxOccurs="unbounded" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:any namespace="##other" processContents="lax"
    minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
```

Deleted: any

Formatted: English (U.S.)

1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887

```
</xsd:complexType>
</xsd:element>

<xsd:element name="SubscribeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="SubscriptionReference"
        type="wsa:EndpointReferenceType"
        minOccurs="1" maxOccurs="1" />
      <xsd:element ref="wsnt:CurrentTime"
        minOccurs="0" maxOccurs="1" />
      <xsd:element ref="wsnt:TerminationTime"
        minOccurs="0" maxOccurs="1" />
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="GetCurrentMessage">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Topic"
        type="wsnt:TopicExpressionType" />
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="GetCurrentMessageResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="SubscribeCreationFailedFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType" />
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="SubscribeCreationFailedFault"
  type="wsnt:SubscribeCreationFailedFaultType" />

<xsd:complexType name="InvalidFilterFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType">
      <xsd:sequence>
        <xsd:element name="UnknownFilter" type="xsd:QName"
          minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="InvalidFilterFault"
  type="wsnt:InvalidFilterFaultType" />
```

Deleted: 0

1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946

```
<xsd:complexType name="TopicExpressionDialectUnknownFaultType">  
  <xsd:complexContent>  
    <xsd:extension base="wsrf-bf:BaseFaultType"/>  
  </xsd:complexContent>  
</xsd:complexType>  
<xsd:element name="TopicExpressionDialectUnknownFault"  
  type="wsnt:TopicExpressionDialectUnknownFaultType"/>  
  
<xsd:complexType name="InvalidTopicExpressionFaultType">  
  <xsd:complexContent>  
    <xsd:extension base="wsrf-bf:BaseFaultType"/>  
  </xsd:complexContent>  
</xsd:complexType>  
<xsd:element name="InvalidTopicExpressionFault"  
  type="wsnt:InvalidTopicExpressionFaultType"/>  
  
<xsd:complexType name="TopicNotSupportedFaultType">  
  <xsd:complexContent>  
    <xsd:extension base="wsrf-bf:BaseFaultType"/>  
  </xsd:complexContent>  
</xsd:complexType>  
<xsd:element name="TopicNotSupportedFault"  
  type="wsnt:TopicNotSupportedFaultType"/>  
  
<xsd:complexType name="MultipleTopicsSpecifiedFaultType">  
  <xsd:complexContent>  
    <xsd:extension base="wsrf-bf:BaseFaultType"/>  
  </xsd:complexContent>  
</xsd:complexType>  
<xsd:element name="MultipleTopicsSpecifiedFault"  
  type="wsnt:MultipleTopicsSpecifiedFaultType"/>  
  
<xsd:complexType name="InvalidProducerPropertiesExpressionFaultType">  
  <xsd:complexContent>  
    <xsd:extension base="wsrf-bf:BaseFaultType"/>  
  </xsd:complexContent>  
</xsd:complexType>  
<xsd:element name="InvalidProducerPropertiesExpressionFault"  
  type="wsnt:InvalidProducerPropertiesExpressionFaultType"/>  
  
<xsd:complexType name="InvalidMessageContentExpressionFaultType">  
  <xsd:complexContent>  
    <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"
```

Deleted: InvalidUseRawValueFaultType

Deleted: /

Deleted: 1

Inserted: 1 "maxOccurs="unbounded" />

</xsd:sequence>

</xsd:extension>

Deleted: InvalidUseRawValueFault "

1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
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

```
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" />
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="NoCurrentMessageOnTopicFault"
  type="wsnt:NoCurrentMessageOnTopicFaultType" />
<!-- ===== Message Types for PullPoint ===== -->
<xsd:element name="GetMessages">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="MaximumNumber"
        type="xsd:nonNegativeInteger" />
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="GetMessagesResponse">
```

Deleted: InvalidUseRawValueFaultType" />

Deleted: 1

Inserted: 1"
maxOccurs="unbounded" />
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:element
name="UnsupportedPolicy
RequestFault" ¶
type="wsnt:UnsupportedP
olicyRequestFaultType" /
>¶
¶
<xsd:complexType
name="NotifyMessageNotS
upportedFaultType">¶
<xsd:complexContent>¶
<xsd:extension
base="wsrf-
bf:BaseFaultType" />¶
</xsd:complexContent>¶
</xsd:complexType>¶
<xsd:element
name="NotifyMessageNotS
upportedFault" ¶
type="wsnt:NotifyMessag
eNotSupportedFaultType"
>¶
¶

2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="wsnt:NotificationMessage"
      minOccurs="0" maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute/>
</xsd:complexType>
</xsd:element>

<xsd:element name="DestroyPullPoint">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="DestroyPullPointResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute/>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="UnableToDestroyPullPointFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType" />
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="UnableToDestroyPullPointFault"
  type="wsnt:UnableToDestroyPullPointFaultType"/>

<!-- ===== Message Types for Create PullPoint ===== -->
<xsd:element name="CreatePullPoint">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="CreatePullPointResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="PullPoint"
        type="wsa:EndpointReferenceType" />
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute/>
  </xsd:complexType>
```

Deleted: DestroyResponse
>

Deleted: UnableToDestroy
PullPointType">

Deleted: "

Deleted: UnableToDestroy
PullPointType"/>

Deleted: ¶

2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124

```
</xsd:element>
<xsd:complexType name="UnableToCreatePullPointFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType" />
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="UnableToCreatePullPointFault"
  type="wsnt:UnableToCreatePullPointFaultType" />
<!-- ===== Message Types for Base SubscriptionManager ===== -->
<xsd:element name="Renew">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="TerminationTime"
        type="wsnt:AbsoluteOrRelativeTimeType"
        nillable="true"
        minOccurs="1" maxOccurs="1" />
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="RenewResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="wsnt:TerminationTime"
        minOccurs="1" maxOccurs="1" />
      <xsd:element ref="wsnt:CurrentTime"
        minOccurs="0" maxOccurs="1" />
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="UnacceptableTerminationTimeFaultType">
  <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="UnacceptableTerminationTimeFault"
  type="wsnt:UnacceptableTerminationTimeFaultType" />
<xsd:element name="Unsubscribe">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Deleted: UnableToCreateP
ullPointType">

Deleted: "

Deleted: UnableToCreateP
ullPointType" />

Deleted: ¶

```

2125
2126     <xsd:element name="UnsubscribeResponse">
2127         <xsd:complexType>
2128             <xsd:sequence>
2129                 <xsd:any namespace="##other" processContents="lax"
2130                     minOccurs="0" maxOccurs="unbounded"/>
2131             </xsd:sequence>
2132         </xsd:complexType>
2133     </xsd:element>
2134
2135     <xsd:complexType name="UnableToDestroySubscriptionFaultType">
2136         <xsd:complexContent>
2137             <xsd:extension base="wsrf-bf:BaseFaultType"/>
2138         </xsd:complexContent>
2139     </xsd:complexType>
2140     <xsd:element name="UnableToDestroySubscriptionFault"
2141         type="wsnt:UnableToDestroySubscriptionFaultType"/>
2142
2143 <!-- ===== Message Types for Pausable SubscriptionManager ===== -->
2144
2145     <xsd:element name="PauseSubscription">
2146         <xsd:complexType>
2147             <xsd:sequence>
2148                 <xsd:any namespace="##other" processContents="lax"
2149                     minOccurs="0" maxOccurs="unbounded"/>
2150             </xsd:sequence>
2151         </xsd:complexType>
2152     </xsd:element>
2153
2154     <xsd:element name="PauseSubscriptionResponse" >
2155         <xsd:complexType>
2156             <xsd:sequence>
2157                 <xsd:any namespace="##other" processContents="lax"
2158                     minOccurs="0" maxOccurs="unbounded"/>
2159             </xsd:sequence>
2160         </xsd:complexType>
2161     </xsd:element>
2162
2163     <xsd:element name="ResumeSubscription">
2164         <xsd:complexType>
2165             <xsd:sequence>
2166                 <xsd:any namespace="##other" processContents="lax"
2167                     minOccurs="0" maxOccurs="unbounded"/>
2168             </xsd:sequence>
2169         </xsd:complexType>
2170     </xsd:element>
2171
2172     <xsd:element name="ResumeSubscriptionResponse">
2173         <xsd:complexType>
2174             <xsd:sequence>
2175                 <xsd:any namespace="##other" processContents="lax"
2176                     minOccurs="0" maxOccurs="unbounded"/>
2177             </xsd:sequence>
2178         </xsd:complexType>
2179     </xsd:element>
2180
2181     <xsd:complexType name="PauseFailedFaultType">
2182         <xsd:complexContent>
2183             <xsd:extension base="wsrf-bf:BaseFaultType"/>
2184         </xsd:complexContent>

```

```
2185 </xsd:complexType>
2186 <xsd:element name="PauseFailedFault"
2187           type="wsnt:PauseFailedFaultType" />
2188
2189 <xsd:complexType name="ResumeFailedFaultType">
2190   <xsd:complexContent>
2191     <xsd:extension base="wsrf-bf:BaseFaultType" />
2192   </xsd:complexContent>
2193 </xsd:complexType>
2194 <xsd:element name="ResumeFailedFault"
2195           type="wsnt:ResumeFailedFaultType" />
2196
2197 </xsd:schema>
```

2198

Appendix C. WSDL 1.1

2199 The following illustrates the WSDL 1.1 for the Web service methods described in this
2200 specification:

2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251

```

<?xml version="1.0" encoding="utf-8"?>
<!--
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.

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.

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

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.

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

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

-->
<wsdl:definitions name="WS-BaseNotification"
  targetNamespace="http://docs.oasis-open.org/wsn/bw-2"
  xmlns:wsntw="http://docs.oasis-open.org/wsn/bw-2"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"

```

Deleted: 1 "
 Deleted: 1 "
 Deleted: 1 "

2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310

```
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:wsrf-rw="http://docs.oasis-open.org/wsrf/rw-2"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">
<!-- ===== Imports ===== -->
<wsdl:import
  namespace="http://docs.oasis-open.org/wsrf/rw-2"
  location="http://docs.oasis-open.org/wsrf/rw-2.wsdl"/>
<!-- ===== Types Definitions ===== -->
<wsdl:types>
  <xsd:schema>
    <xsd:import
      namespace="http://docs.oasis-open.org/wsn/b-2"
      schemaLocation="http://docs.oasis-open.org/wsn/b-2.xsd"/>
    </xsd:schema>
  </wsdl:types>
<!-- ===== NotificationConsumer::Notify =====
Notify(
  NotificationMessage
  (SubscriptionReference, TopicExpression, ProducerReference,
  Message)*
  returns: n/a (one way)
-->
<wsdl:message name="Notify">
  <wsdl:part name="Notify" element="wsnt:Notify"/>
</wsdl:message>
<!-- ===== NotificationProducer::Subscribe =====
Subscribe(
  (ConsumerEndpointReference, [Filter], [SubscriptionPolicy],
  [InitialTerminationTime])
  returns: WS-Resource qualified EPR to a Subscription
-->
<wsdl:message name="SubscribeRequest" >
  <wsdl:part name="SubscribeRequest"
    element="wsnt:Subscribe"/>
</wsdl:message>
<wsdl:message name="SubscribeResponse">
  <wsdl:part name="SubscribeResponse"
    element="wsnt:SubscribeResponse"/>
</wsdl:message>
<wsdl:message name="SubscribeCreationFailedFault">
  <wsdl:part name="SubscribeCreationFailedFault"
    element="wsnt:SubscribeCreationFailedFault" />
</wsdl:message>
<wsdl:message name="TopicExpressionDialectUnknownFault">
  <wsdl:part name="TopicExpressionDialectUnknownFault"
    element="wsnt:TopicExpressionDialectUnknownFault" />
</wsdl:message>
<wsdl:message name="InvalidFilterFault">
  <wsdl:part name="InvalidFilterFault"
    element="wsnt:InvalidFilterFault" />
```

Deleted: 03
Deleted: 1"
Deleted:
xmlns:xsi="http://www.w
3.org/2001/XMLSchema-
instance"¶
Deleted: 1"
Deleted: 1
Deleted: 1"
Deleted: 1

2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370

```
</wsdl:message>  
  
<wsdl:message name="InvalidProducerPropertiesExpressionFault">  
  <wsdl:part name="InvalidProducerPropertiesExpressionFault"  
    element="wsnt:InvalidProducerPropertiesExpressionFault" />  
</wsdl:message>  
  
<wsdl:message name="InvalidMessageContentExpressionFault">  
  <wsdl:part name="InvalidMessageContentExpressionFault"  
    element="wsnt:InvalidMessageContentExpressionFault" />  
</wsdl:message>  
  
<wsdl:message name="UnrecognizedPolicyRequestFault">  
  <wsdl:part name="UnrecognizedPolicyRequestFault"  
    element="wsnt:UnrecognizedPolicyRequestFault" />  
</wsdl:message>  
  
<wsdl:message name="UnsupportedPolicyRequestFault">  
  <wsdl:part name="UnsupportedPolicyRequestFault"  
    element="wsnt:UnsupportedPolicyRequestFault" />  
</wsdl:message>  
  
<wsdl:message name="NotifyMessageNotSupportedFault">  
  <wsdl:part name="NotifyMessageNotSupportedFault"  
    element="wsnt:NotifyMessageNotSupportedFault" />  
</wsdl:message>  
  
<wsdl:message name="UnacceptableInitialTerminationTimeFault">  
  <wsdl:part name="UnacceptableInitialTerminationTimeFault"  
    element="wsnt:UnacceptableInitialTerminationTimeFault" />  
</wsdl:message>  
  
<!-- ===== NotificationProducer::GetCurrentMessage =====  
GetCurrentMessage(topicExpression)  
returns: a NotificationMessage (xsd:any)  
-->  
<wsdl:message name="GetCurrentMessageRequest">  
  <wsdl:part name="GetCurrentMessageRequest"  
    element="wsnt:GetCurrentMessage" />  
</wsdl:message>  
  
<wsdl:message name="GetCurrentMessageResponse">  
  <wsdl:part name="GetCurrentMessageResponse"  
    element="wsnt:GetCurrentMessageResponse" />  
</wsdl:message>  
  
<wsdl:message name="InvalidTopicExpressionFault">  
  <wsdl:part name="InvalidTopicExpressionFault"  
    element="wsnt:InvalidTopicExpressionFault" />  
</wsdl:message>  
  
<wsdl:message name="TopicNotSupportedFault">  
  <wsdl:part name="TopicNotSupportedFault"  
    element="wsnt:TopicNotSupportedFault" />  
</wsdl:message>  
  
<wsdl:message name="MultipleTopicsSpecifiedFault">  
  <wsdl:part name="MultipleTopicsSpecifiedFault"  
    element="wsnt:MultipleTopicsSpecifiedFault" />  
</wsdl:message>
```

Deleted: InvalidUseRawValueFault">
Deleted: InvalidUseRawValueFault "
Deleted: InvalidUseRawValueFault

2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430

```
<wsdl:message name="NoCurrentMessageOnTopicFault">
  <wsdl:part name="NoCurrentMessageOnTopicFault"
    element="wsnt:NoCurrentMessageOnTopicFault" />
</wsdl:message>

<!-- ===== PullPoint::GetMessages =====
GetMessages(MaximumNumber)
returns: NotificationMessage list
-->
<wsdl:message name="GetMessagesRequest">
  <wsdl:part name="GetMessagesRequest"
    element="wsnt:GetMessages" />
</wsdl:message>

<wsdl:message name="GetMessagesResponse">
  <wsdl:part name="GetMessagesResponse"
    element="wsnt:GetMessagesResponse" />
</wsdl:message>

<!-- ===== PullPoint::DestroyPullPoint =====
DestroyPullPoint()
returns: void
-->
<wsdl:message name="DestroyPullPointRequest">
  <wsdl:part name="DestroyPullPointRequest"
    element="wsnt:DestroyPullPoint" />
</wsdl:message>

<wsdl:message name="DestroyPullPointResponse">
  <wsdl:part name="DestroyPullPointResponse"
    element="wsnt:DestroyPullPointResponse" />
</wsdl:message>

<wsdl:message name="UnableToDestroyPullPointFault">
  <wsdl:part name="UnableToDestroyPullPointFault"
    element="wsnt:UnableToDestroyPullPointFault" />
</wsdl:message>

<!-- ===== PullPoint::CreatePullPoint =====
CreatePullPoint()
returns: PullPoint (wsa:EndpointReference)
-->
<wsdl:message name="CreatePullPointRequest">
  <wsdl:part name="CreatePullPointRequest"
    element="wsnt:CreatePullPoint" />
</wsdl:message>

<wsdl:message name="CreatePullPointResponse">
  <wsdl:part name="CreatePullPointResponse"
    element="wsnt:CreatePullPointResponse" />
</wsdl:message>

<wsdl:message name="UnableToCreatePullPointFault">
  <wsdl:part name="UnableToCreatePullPointFault"
    element="wsnt:UnableToCreatePullPointFault" />
</wsdl:message>

<!-- ===== SubscriptionManager::Renew =====
Renew( Duration | AbsoluteTime)
```

Deleted: Destroy

Deleted: DestroyRequest ">

Deleted: DestroyRequest "

Deleted: DestroyResponse ">

Deleted: DestroyResponse "

Deleted: DestroyResponse "

Deleted: "

Deleted: "


```

2431     returns: (New Termination Time [CurrentTime])
2432 -->
2433     <wsdl:message name="RenewRequest">
2434         <wsdl:part name="RenewRequest"
2435             element="wsnt:Renew"/>
2436     </wsdl:message>
2437
2438     <wsdl:message name="RenewResponse">
2439         <wsdl:part name="RenewResponse"
2440             element="wsnt:RenewResponse"/>
2441     </wsdl:message>
2442
2443     <wsdl:message name="UnacceptableTerminationTimeFault">
2444         <wsdl:part name="UnacceptableTerminationTimeFault"
2445             element="wsnt:UnacceptableTerminationTimeFault" />
2446     </wsdl:message>
2447
2448 <!-- ===== SubscriptionManager::Unsubscribe =====
2449     Unsubscribe()
2450     returns: empty
2451 -->
2452     <wsdl:message name="UnsubscribeRequest">
2453         <wsdl:part name="UnsubscribeRequest"
2454             element="wsnt:Unsubscribe"/>
2455     </wsdl:message>
2456
2457     <wsdl:message name="UnsubscribeResponse">
2458         <wsdl:part name="UnsubscribeResponse"
2459             element="wsnt:UnsubscribeResponse"/>
2460     </wsdl:message>
2461
2462     <wsdl:message name="UnableToDestroySubscriptionFault">
2463         <wsdl:part name="UnableToDestroySubscriptionFault"
2464             element="wsnt:UnableToDestroySubscriptionFault" />
2465     </wsdl:message>
2466
2467 <!-- ===== SubscriptionManager::PauseSubscription =====
2468     PauseSubscription()
2469     returns: empty
2470 -->
2471     <wsdl:message name="PauseSubscriptionRequest">
2472         <wsdl:part name="PauseSubscriptionRequest"
2473             element="wsnt:PauseSubscription"/>
2474     </wsdl:message>
2475
2476     <wsdl:message name="PauseSubscriptionResponse">
2477         <wsdl:part name="PauseSubscriptionResponse"
2478             element="wsnt:PauseSubscriptionResponse"/>
2479     </wsdl:message>
2480
2481     <wsdl:message name="PauseFailedFault">
2482         <wsdl:part name="PauseFailedFault"
2483             element="wsnt:PauseFailedFault" />
2484     </wsdl:message>
2485
2486 <!-- ===== SubscriptionManager::ResumeSubscription =====
2487     ResumeSubscription()
2488     returns: empty
2489 -->
2490     <wsdl:message name="ResumeSubscriptionRequest">

```

2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550

```
<wsdl:part name="ResumeSubscriptionRequest"
  element="wsnt:ResumeSubscription" />
</wsdl:message>

<wsdl:message name="ResumeSubscriptionResponse">
  <wsdl:part name="ResumeSubscriptionResponse"
    element="wsnt:ResumeSubscriptionResponse" />
</wsdl:message>

<wsdl:message name="ResumeFailedFault">
  <wsdl:part name="ResumeFailedFault"
    element="wsnt:ResumeFailedFault" />
</wsdl:message>

<!-- ===== PortType Definitions ===== -->
<!-- ===== NotificationConsumer PortType Definition ===== -->
<wsdl:portType name="NotificationConsumer">
  <wsdl:operation name="Notify">
    <wsdl:input message="wsntw:Notify" />
  </wsdl:operation>
</wsdl:portType>

<!-- ===== NotificationProducer PortType Definition ===== -->
<wsdl:portType name="NotificationProducer">
  <wsdl:operation name="Subscribe">
    <wsdl:input message="wsntw:SubscribeRequest" />
    <wsdl:output message="wsntw:SubscribeResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrf-rw:ResourceUnknownFault" />
    <wsdl:fault name="InvalidFilterFault"
      message="wsntw:InvalidFilterFault" />
    <wsdl:fault name="TopicExpressionDialectUnknownFault"
      message="wsntw:TopicExpressionDialectUnknownFault" />
    <wsdl:fault name="InvalidTopicExpressionFault"
      message="wsntw:InvalidTopicExpressionFault" />
    <wsdl:fault name="TopicNotSupportedFault"
      message="wsntw:TopicNotSupportedFault" />
    <wsdl:fault name="InvalidProducerPropertiesExpressionFault"
      message="wsntw:InvalidProducerPropertiesExpressionFault" />
    <wsdl:fault name="InvalidMessageContentExpressionFault"
      message="wsntw:InvalidMessageContentExpressionFault" />
    <wsdl:fault name="UnacceptableInitialTerminationTimeFault"
      message="wsntw:UnacceptableInitialTerminationTimeFault" />
    <wsdl:fault name="UnrecognizedPolicyRequestFault"
      message="wsntw:UnrecognizedPolicyRequestFault" />
    <wsdl:fault name="UnsupportedPolicyRequestFault"
      message="wsntw:UnsupportedPolicyRequestFault" />
    <wsdl:fault name="NotifyMessageNotSupportedFault"
      message="wsntw:NotifyMessageNotSupportedFault" />
    <wsdl:fault name="SubscribeCreationFailedFault"
      message="wsntw:SubscribeCreationFailedFault" />
  </wsdl:operation>

  <wsdl:operation name="GetCurrentMessage">
    <wsdl:input message="wsntw:GetCurrentMessageRequest" />
    <wsdl:output message="wsntw:GetCurrentMessageResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrf-rw:ResourceUnknownFault" />
    <wsdl:fault name="TopicExpressionDialectUnknownFault"
      message="wsntw:TopicExpressionDialectUnknownFault" />
  </wsdl:operation>
</wsdl:portType>
</wsdl:portType>
```

Deleted:
<wsdl:fault
name="InvalidUseRawValueFault" />

message="wsntw:InvalidUseRawValueFault" />

2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610

```
<wsdl:fault name="InvalidTopicExpressionFault"
message="wsntw:InvalidTopicExpressionFault" />
<wsdl:fault name="TopicNotSupportedFault"
message="wsntw:TopicNotSupportedFault" />
<wsdl:fault name="NoCurrentMessageOnTopicFault"
message="wsntw:NoCurrentMessageOnTopicFault" />
<wsdl:fault name="MultipleTopicsSpecifiedFault"
message="wsntw:MultipleTopicsSpecifiedFault" />
</wsdl:operation>
</wsdl:portType>
<!-- ===== PullPoint PortType Definition ===== -->
<wsdl:portType name="PullPoint">
  <wsdl:operation name="GetMessages">
    <wsdl:input name="GetMessagesRequest"
message="wsntw:GetMessagesRequest" />
    <wsdl:output name="GetMessagesResponse"
message="wsntw:GetMessagesResponse" />
    <wsdl:fault name="ResourceUnknownFault"
message="wsrf-rw:ResourceUnknownFault" />
  </wsdl:operation>

  <wsdl:operation name="DestroyPullPoint">
    <wsdl:input name="DestroyPullPointRequest"
message="wsntw:DestroyPullPointRequest" />
    <wsdl:output name="DestroyPullPointResponse"
message="wsntw:DestroyPullPointResponse" />
    <wsdl:fault name="UnableToDestroyPullPointFault"
message="wsntw:UnableToDestroyPullPointFault" />
  </wsdl:operation>

  <wsdl:operation name="Notify">
    <wsdl:input message="wsntw:Notify" />
  </wsdl:operation>
</wsdl:portType>
<!-- ===== CreatePullPoint PortType Definition ===== -->
<wsdl:portType name="CreatePullPoint">
  <wsdl:operation name="CreatePullPoint">
    <wsdl:input name="CreatePullPointRequest"
message="wsntw:CreatePullPointRequest" />
    <wsdl:output name="CreatePullPointResponse"
message="wsntw:CreatePullPointResponse" />
    <wsdl:fault name="UnableToCreatePullPointFault"
message="wsntw:UnableToCreatePullPointFault" />
  </wsdl:operation>
</wsdl:portType>
<!-- ===== SubscriptionManager PortType Definition ===== -->
<wsdl:portType name="SubscriptionManager">
  <wsdl:operation name="Renew">
    <wsdl:input name="RenewRequest"
message="wsntw:RenewRequest" />
    <wsdl:output name="RenewResponse"
message="wsntw:RenewResponse" />
    <wsdl:fault name="ResourceUnknownFault"
message="wsrf-rw:ResourceUnknownFault" />
    <wsdl:fault name="UnacceptableTerminationTimeFault"
message="
wsntw:UnacceptableTerminationTimeFault" />
```

Deleted: DestroyRequest "
Deleted: DestroyRequest "
Deleted: DestroyResponse "
Deleted: DestroyResponse "
Deleted: "

Deleted: "

2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669

```
</wsdl:operation>
<wsdl:operation name="Unsubscribe">
  <wsdl:input name="UnsubscribeRequest"
    message="wsntw:UnsubscribeRequest" />
  <wsdl:output name="UnsubscribeResponse"
    message="wsntw:UnsubscribeResponse" />
  <wsdl:fault name="ResourceUnknownFault"
    message="wsrf-rw:ResourceUnknownFault" />
  <wsdl:fault name="UnableToDestroySubscriptionFault"
    message="
      wsntw:UnableToDestroySubscriptionFault" />
</wsdl:operation>
</wsdl:portType>

<!-- ===== PausableSubscriptionManager PortType Definition ===== -->
<wsdl:portType name="PausableSubscriptionManager">
  <!-- ===== Extends: SubscriptionManager ===== -->
  <wsdl:operation name="Renew">
    <wsdl:input name="RenewRequest"
      message="wsntw:RenewRequest" />
    <wsdl:output name="RenewResponse"
      message="wsntw:RenewResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrf-rw:ResourceUnknownFault" />
    <wsdl:fault name="UnacceptableTerminationTimeFault"
      message="
        wsntw:UnacceptableTerminationTimeFault" />
  </wsdl:operation>
  <wsdl:operation name="Unsubscribe">
    <wsdl:input name="UnsubscribeRequest"
      message="wsntw:UnsubscribeRequest" />
    <wsdl:output name="UnsubscribeResponse"
      message="wsntw:UnsubscribeResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrf-rw:ResourceUnknownFault" />
    <wsdl:fault name="UnableToDestroySubscriptionFault"
      message="
        wsntw:UnableToDestroySubscriptionFault" />
  </wsdl:operation>

  <!-- === PausableSubscriptionManager specific operations === -->
  <wsdl:operation name="PauseSubscription">
    <wsdl:input message="wsntw:PauseSubscriptionRequest"/>
    <wsdl:output message="wsntw:PauseSubscriptionResponse"/>
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrf-rw:ResourceUnknownFault" />
    <wsdl:fault name="PauseFailedFault"
      message="wsntw:PauseFailedFault" />
  </wsdl:operation>
  <wsdl:operation name="ResumeSubscription">
    <wsdl:input message="wsntw:ResumeSubscriptionRequest"/>
    <wsdl:output message="wsntw:ResumeSubscriptionResponse"/>
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrf-rw:ResourceUnknownFault" />
    <wsdl:fault name="ResumeFailedFault"
      message="wsntw:ResumeFailedFault" />
  </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>
```

Deleted: UnsubscribeRequest

Deleted: UnsubscribeRequest

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. WSN 2.48 (but see comment to list from dmh)

Rev	Date	By Whom	What
			(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.
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

Rev	Date	By Whom	What
1.3.02f	2005-11-21	David Hull	Issue 2.62 and various errata

2671 **Appendix E. Notices**

2672 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
2673 that might be claimed to pertain to the implementation or use of the technology described in this
2674 document or the extent to which any license under such rights might or might not be available;
2675 neither does it represent that it has made any effort to identify any such rights. Information on
2676 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
2677 website. Copies of claims of rights made available for publication and any assurances of licenses
2678 to be made available, or the result of an attempt made to obtain a general license or permission
2679 for the use of such proprietary rights by implementors or users of this specification, can be
2680 obtained from the OASIS Executive Director.

2681

2682 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
2683 applications, or other proprietary rights which may cover technology that may be required to
2684 implement this specification. Please address the information to the OASIS Executive Director.

2685

2686 Copyright (C) OASIS Open (2004-2005). All Rights Reserved.

2687

2688 This document and translations of it may be copied and furnished to others, and derivative works
2689 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
2690 published and distributed, in whole or in part, without restriction of any kind, provided that the
2691 above copyright notice and this paragraph are included on all such copies and derivative works.
2692 However, this document itself may not be modified in any way, such as by removing the copyright
2693 notice or references to OASIS, except as needed for the purpose of developing OASIS
2694 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
2695 Property Rights document must be followed, or as required to translate it into languages other
2696 than English.

2697

2698 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
2699 successors or assigns.

2700

2701 This document and the information contained herein is provided on an "AS IS" basis and OASIS
2702 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
2703 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
2704 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
2705 PARTICULAR PURPOSE.

2706