



1

2

---

# Web Services Brokered Notification 1.3 (WS-BrokeredNotification)

3

4

## Public Review Draft 02, 28 November 2005

5

6

7 **Document identifier:**

8 wsn-ws-brokered-notification-1.3-spec-pr-02

9 **Location:**

10 [http://docs.oasis-open.org/wsn/wsn-ws-brokered\\_notification-1.3-spec-pr-02.pdf](http://docs.oasis-open.org/wsn/wsn-ws-brokered_notification-1.3-spec-pr-02.pdf)

11 **Editors:**

12 Dave Chappell, Sonic Software <chappell@sonicsoftware.com>

13 Lily Liu, webMethods <lily.liu@webmethods.com>

14 **Abstract:**

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

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

27 family of documents includes three normative specifications: [WS-BaseNotification], WS-  
28 BrokeredNotification, and [WS-Topics].

29 This document defines the Web services interface for the NotificationBroker. A  
30 NotificationBroker is an intermediary, which, among other things, allows publication of  
31 messages from entities that are not themselves service providers. It includes standard  
32 message exchanges to be implemented by NotificationBroker service providers along  
33 with operational requirements expected of service providers and requestors that  
34 participate in brokered notifications. This work relies upon WS-BaseNotification.

35 **Status:**

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

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

---

## Table of Contents

48	1	Introduction .....	5
49	1.1	Goals and Requirements .....	5
50	1.1.1	Requirements.....	5
51	1.1.2	Non-Goals.....	6
52	1.2	Notational Conventions .....	6
53	1.3	Namespaces .....	7
54	1.4	Fault Definitions.....	8
55	2	Relationship to Other Specifications.....	9
56	3	Terminology and Concepts .....	10
57	4	Publishing .....	13
58	5	NotificationBroker Interface.....	16
59	5.1	NotificationBroker Resource Properties .....	16
60	5.2	Notify .....	17
61	5.3	Subscribe .....	17
62	5.4	GetCurrentMessage .....	17
63	5.5	RegisterPublisher .....	18
64	5.6	CreatePullPoint .....	18
65	6	RegisterPublisher Interface.....	19
66	6.1	RegisterPublisher .....	19
67	6.1.1	Example SOAP Encoding of the RegisterPublisher Message Exchange .....	22
68	7	PublisherRegistrationManager Interface .....	25
69	7.1	PublisherRegistration Resource Properties .....	25
70	7.2	DestroyRegistration.....	26
71	7.2.1	Example SOAP Encoding of the DestroyRegistration Message Exchange .....	27
72	8	Security Considerations .....	29
73	8.1	Securing PublisherRegistration.....	29
74	9	References.....	31
75	9.1	Normative .....	31
76	9.2	Non-Normative .....	31
77		Appendix A. Acknowledgments .....	32
78		Appendix B. XML Schema.....	33
79		Appendix C. WSDL 1.1.....	38
80		Appendix D. Revision History .....	43



---

## 82 1 Introduction

83 The Event-driven, or Notification-based, interaction pattern is a commonly used pattern for inter-  
84 object communications. Examples exist in many domains, for example, in publish/subscribe  
85 systems or in system and device management domains. Message brokers are involved in many  
86 of these systems, such as the ones provided by Message Oriented Middleware vendors.

87 This specification defines the Web services interface for the NotificationBroker. A  
88 NotificationBroker is an intermediary between message Publishers and message Subscribers.  
89 Common functions of Publishers and Subscribers, such as messaging dissemination and security  
90 measurements, can be implemented at the NotificationBroker to produce lightweight Producers  
91 and Consumers. A NotificationBroker decouples NotificationProducers and Notification  
92 Consumers and can provide advanced messaging features such as demand-based publishing  
93 and load-balancing. A NotificationBroker also allows publication of messages from entities that  
94 are not themselves service providers. This is very similar to a traditional Message Oriented  
95 Middleware model.

96 The NotificationBroker interface includes standard message exchanges to be implemented by  
97 NotificationBroker service providers along with operational requirements expected of service  
98 providers and requestors that participate in brokered notifications.

### 99 1.1 Goals and Requirements

100 The goal of WS-BrokeredNotification is to standardize message exchanges involved in Web  
101 services publish and subscribe of a message broker. The overall objectives of WS-Notification  
102 are presented in [\[WS-BaseNotification\]](#). The following section lists the specific subset of those  
103 objectives realized by WS-BrokeredNotification.

#### 104 1.1.1 Requirements

105 In meeting this goal, the WS-BrokeredNotification specification must explicitly address the  
106 following requirements:

- 107 • **Must allow for a notification broker as an intermediary.** A NotificationBroker is an  
108 intermediary Web service that decouples NotificationConsumers from Publishers. A  
109 notification broker can relieve a Publisher from having to implement message exchanges  
110 associated with NotificationProducer; the NotificationBroker takes on the duties of  
111 subscription management and distributing Notifications on behalf of the Publisher. It  
112 implements NotificationProducer interface. It may implement SubscriptionManager or may  
113 delegate the subscription management work to another component.
- 114 • **Must allow for federation of brokers.** It must be possible to build configurations with  
115 multiple intermediary broker services in a dynamic fashion. This specification must allow for

- 116 a variety of broker topology usage patterns. Among other things, these allow for greater  
117 scalability and permit sharing of administrative workload.
- 118 • **Must provide runtime metadata:** There must be a mechanism that lets a potential  
119 Subscriber discover what elements available for a subscription are provided by a  
120 NotificationBroker, and in what formats the subscription for a notification can be made.
  - 121 • **Must conform to WS-BaseNotification:** A NotificationBroker must support required  
122 message exchanges defined by the [WS-BaseNotification] specification. It must conform to  
123 the NotificationProducer and the NotificationConsumer interfaces defined in WS-  
124 BaseNotification.
  - 125 • **WS-BrokeredNotification must be independent of binding-level details:** Transport  
126 protocol details must be orthogonal to the subscription and the delivery of the notifications, so  
127 that the specification can be used over a variety of different transports.
  - 128 • **Must not exclude non-service producers and subscribers:** WS-BrokeredNotification  
129 design must not exclude a non-service entity to deliver a notification message to a  
130 NotificationBroker. It must not exclude a NotificationBroker to send a notification message to  
131 a non-service consumer.
  - 132 • **Must provide publisher registration:** WS-BrokeredNotification must define standard  
133 message exchanges for registering a NotificationPublisher with a NotificationBroker.

### 134 1.1.2 Non-Goals

135 The following topics are outside the scope of the WS-BrokeredNotification specification:

- 136 • **Defining the format of notification payloads:** The data carried in Notification payloads is  
137 application-domain specific, and WS-BrokeredNotification does not prescribe any particular  
138 format for this data.
- 139 • **Defining any Events or Notifications:** The WS-BrokeredNotification specification does not  
140 define any “standard” or “built-in” notification situations, events, or messages.
- 141 • **Defining the means by which NotificationBrokers are discovered by subscribers:** It is  
142 beyond the scope of this specification to define the mechanisms for runtime discovery of  
143 NotificationBrokers.

## 144 1.2 Notational Conventions

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

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

Deleted:

151 This specification uses a notational convention, referred to as "Pseudo-schemas" in a fashion  
152 similar to the WSDL 2.0 Part 1 specification. A Pseudo-schema uses a BNF-style convention to  
153 describe attributes and elements:

Deleted: [WSDL 2.0]

- 154 • '?' denotes optionality (i.e. zero or one occurrences),
- 155 • '\*' denotes zero or more occurrences,
- 156 • '+' one or more occurrences,
- 157 • '[' and ']' are used to form groups,
- 158 • '|' represents choice.
- 159 • Attributes are conventionally assigned a value which corresponds to their type, as  
160 defined in the normative schema.

```
161 <!-- sample pseudo-schema -->  
162 <element  
163     required_attribute_of_type_QName="xs:QName"  
164     optional_attribute_of_type_string="xs:string"?>  
165   <required_element />  
166   <optional_element /> ?  
167   <one_or_more_of_these_elements /> +  
168   [ <choice_1 /> | <choice_2 /> ] *  
169 </element>
```

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

### 175 1.3 Namespaces

176 The following namespaces are used in this document:

Prefix	Namespace
s	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a> OR <a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
wsa	<a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a>
wsn-b	<a href="http://docs.oasis-open.org/wsn/b-2">http://docs.oasis-open.org/wsn/b-2</a>

Deleted: 3

Deleted: -1

wsn-br	<a href="http://docs.oasis-open.org/wsn/br-2">http://docs.oasis-open.org/wsn/br-2</a>
wsn-bw	<a href="http://docs.oasis-open.org/wsn/bw-2">http://docs.oasis-open.org/wsn/bw-2</a>
wsn-brw	<a href="http://docs.oasis-open.org/wsn/brw-2">http://docs.oasis-open.org/wsn/brw-2</a>
wsrf-bf	<a href="http://docs.oasis-open.org/wsrf/bf-2">http://docs.oasis-open.org/wsrf/bf-2</a>
wsrf-bfw	<a href="http://docs.oasis-open.org/wsrf/bfw-2">http://docs.oasis-open.org/wsrf/bfw-2</a>

Deleted: -1

Deleted: -1

Deleted: -1

177

## 1.4 Fault Definitions

178 All faults generated by a NotificationBroker, RegisterPublisher, or PublisherRegistrationManager  
 179 SHOULD be compliant with the WS-BaseFaults [WS-BaseFaults] specification.

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

182 `http://docs.oasis-open.org/wsn/fault.`



183

---

## 2 Relationship to Other Specifications

184 This specification builds on the basic notification mechanism defined in [\[WS-BaseNotification\]](#), by  
185 adding the concept of an intermediary NotificationBroker, and describing additional variants on  
186 the publisher role. A NotificationBroker takes on the role of both NotificationProducer and  
187 NotificationConsumer (as defined in [\[WS-BaseNotification\]](#)), and its interactions with other  
188 NotificationProducers and NotificationConsumers are largely defined by the WS-BaseNotification  
189 specification.

190 This means that a NotificationBroker, implemented to conform to this specification, must also  
191 conform to [\[WS-BaseNotification\]](#). Such a NotificationBroker can deliver notifications to  
192 NotificationConsumers that are implemented to conform to [\[WS-BaseNotification\]](#), and can  
193 subscribe to Notifications distributed by NotificationProducers as defined in [\[WS-  
194 BaseNotification\]](#).

195 A NotificationBroker may support hierarchical topics as defined in [\[WS-Topics\]](#). By supporting  
196 topics, NotificationBroker can manage enterprise messaging systems more efficiently.

197 WS-BrokeredNotification must be composable with other Web services specifications.

198

## 3 Terminology and Concepts

199

In addition to the terminology and usage described in the WS-BaseNotification specification, the following are the terms defined in this specification:

200

201

### **Publisher:**

202

- A Publisher is an entity that creates Notifications, based upon Situation(s) that it is capable of detecting and translating into Notification artifacts. It does not need to be a Web service.

203

204

205

- A Publisher can register what topics it wishes to publish with a NotificationBroker.

206

- A Publisher MAY be a Web service that implements the message exchanges associated with the NotificationProducer interface, in which case it also distributes the Notifications to the relevant NotificationConsumers.

207

208

209

- If a Publisher does not implement the message exchanges associated with NotificationProducer, then it is not required to support the Subscribe request message and does not have to maintain knowledge of the NotificationConsumers that are subscribed to it; a NotificationBroker takes care of this on its behalf.

210

211

212

213

### **NotificationBroker:**

214

- A NotificationBroker is an intermediary Web service that decouples NotificationConsumers from Publishers. A NotificationBroker is capable of subscribing to notifications, either on behalf of NotificationConsumers, or for the purpose of messaging management. It is capable disseminating notifications on behalf of Publishers to NotificationConsumers.

215

216

217

218

219

- A NotificationBroker aggregates NotificationProducer, NotificationConsumer, [and RegisterPublisher](#) interfaces.

220

221

- Acting as an intermediary, a NotificationBroker provides additional capabilities to the base NotificationProducer interface:

222

223

- It can relieve a Publisher from having to implement message exchanges associated with NotificationProducer; the NotificationBroker takes on the duties of a SubscriptionManager (managing subscriptions) and NotificationProducer (distributing Notifications) on behalf of the Publisher.

224

225

226

227

- It can reduce the number of inter-service connections and references, if there are many Publishers and many NotificationConsumers.

228

229

- It can act as a finder service. Potential Publishers and Subscribers can in effect find each other by utilizing a common NotificationBroker.

230

231

- It can provide anonymous Notification, so that the Publishers and the NotificationConsumers need not be aware of each other's identity.

232

233

Deleted: , and  
CreatePullPoint

- 233 • An implementation of a NotificationBroker may provide additional added-value function  
234 that is beyond the scope of this specification, for example, logging Notifications, or  
235 transforming Topics and/or Notification content. Additional function provided by a  
236 NotificationBroker can apply to all Publishers that utilize it.
- 237 • It may be the factory for Subscription resources or it may delegate the subscription  
238 factory to another component.
- 239 • A NotificationBroker provides publisher registration functions.
- 240 • A NotificationBroker may subscribe and disseminate messages that are not WS-  
241 Notification conforming.
- 242 **PublisherRegistration:**
- 243 • PublisherRegistration is a resource. A PublisherRegistration represents the relationship  
244 between a Publisher and a NotificationBroker, in particular, which topic(s) the publisher is  
245 permitted to publish to.
- 246 • A PublisherRegistration resource is created when a Publisher sends the  
247 RegisterPublisher request message to a NotificationBroker and the NotificationBroker  
248 succeeds in processing the registration.
- 249 • PublisherRegistration resources can be manipulated by messages sent to a  
250 PublisherRegistrationManager Web service.
- 251 **RegisterPublisher:**
- 252 • A RegisterPublisher is a Web service that implements the message exchanges  
253 associated with the RegisterPublisher interface. A PublisherRegistration resource is  
254 created as a result of a RegisterPublisher request to a NotificationBroker.
- 255 **PublisherRegistrationManager:**
- 256 • A PublisherRegistrationManager is a Web service that implements the message  
257 exchanges associated with the PublisherRegistrationManager interface.
- 258 • A publisher registration resource can be manipulated through  
259 PublisherRegistrationManager message exchanges.
- 260 • A PublisherRegistrationManager provides services that allow a service requestor to query  
261 and manipulate PublisherRegistration resources that it manages.
- 262 • A PublisherRegistrationManager is subordinate to the NotificationBroker, and MAY be  
263 implemented by the NotificationBroker service provider. However WS-  
264 BrokeredNotification permits it to be implemented by a separate service provider, should  
265 an implementer so desire.
- 266 **Demand-Based Publishing:**
- 267 • Some Publishers may be interested in knowing whether they have any Subscribers or  
268 not, since producing a Notification may be a costly process. Such Publishers can register  
269 with the NotificationBroker as a Demand-Based Publisher.

- 270 • Demand-Based Publishers implement message exchanges associated with the  
271 NotificationProducer interface.
- 272 • The NotificationBroker subscribes to the Demand-Based Publisher. When the  
273 NotificationBroker knows that there are no Subscribers for the Notifications from a  
274 Demand-Based Publisher it pauses its Subscription with that Publisher; when it knows  
275 that there are some Subscribers, it resumes the Subscription.
- 276 • This way the Demand-Based Publisher does not need to produce messages when there  
277 are no Subscribers, however a Demand-Based Publisher is only required to support a  
278 single Subscriber on any given Topic, and so can delegate the management of multiple  
279 Subscribers, the delivery to multiple NotificationConsumers, and other related issues (for  
280 example security) to the NotificationBroker.

---

## 4 Publishing

281

282 There are three distinct stages in the Notification process

- 283
- Observation of the Situation and its noteworthy characteristics;
  - 284 • Creation of the Notification artifact that captures the noteworthy characteristics of the  
285 Situation; and
  - 286 • Distribution of copies of the Notification to zero or more interested parties.

287 Stages 1 and 2 happen largely outside of the scope of the WS-Notification architecture; this  
288 specification does not restrict the means by which these stages must occur. We refer to an entity  
289 that performs stages 1 and 2 as a Publisher,

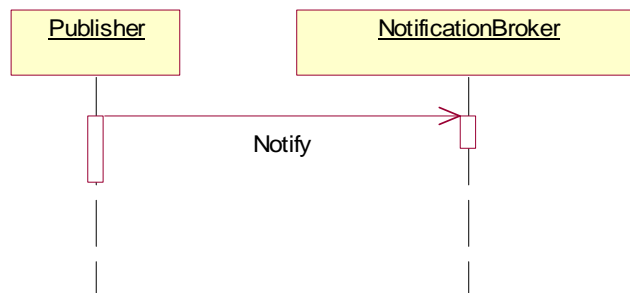
290 However, the WS-Notification family of specifications does specify how dissemination of  
291 messages SHOULD occur. There are two dominant patterns by which Notifications are  
292 disseminated in WS-Notification: direct and brokered.

293 In the direct case, the publishing Web service implements message exchanges associated with  
294 the NotificationProducer interface; it is responsible for accepting Subscribe messages and  
295 sending Notifications to interested parties. The implementer of this Web service can choose to  
296 program this behavior or delegate to specialized implementations of the Subscribe and  
297 Notification delivery behavior. This case is addressed by the WS-BaseNotification specification  
298 [[WS-BaseNotification](#)].

299 In the brokered case, an intermediary - a NotificationBroker - is responsible for disseminating  
300 messages produced by one or more Publishers to zero or more NotificationConsumers.

301 There are three patterns associated with the relationship between the Publisher and the  
302 NotificationBroker: simple publishing, broker initiated publishing, and demand-based publishing.

303 The following figure illustrates simple publishing:



304

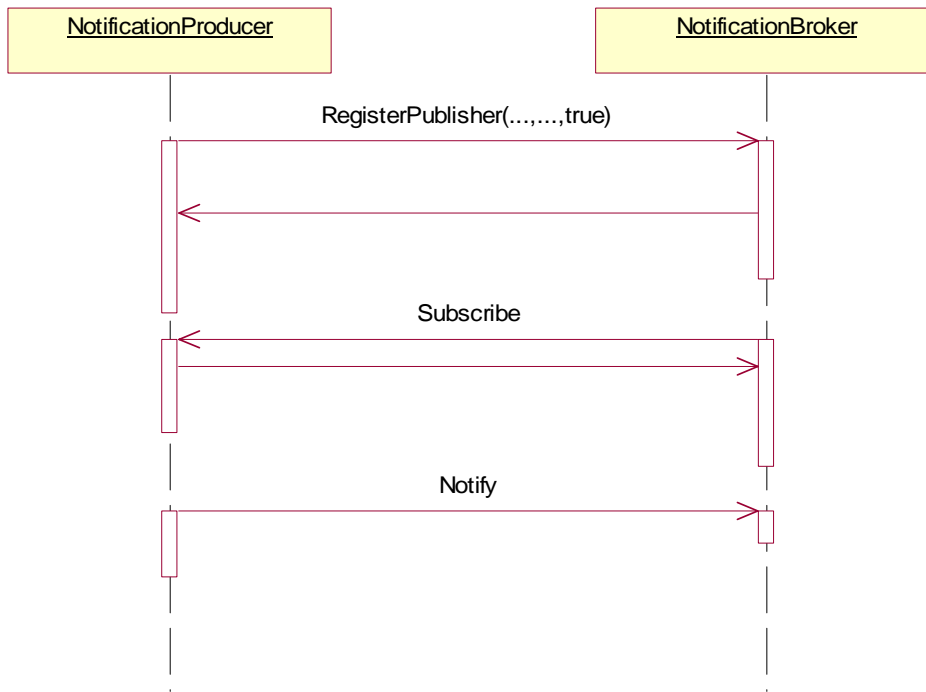
305 In the simple publishing scenario, the Publisher entity is responsible only for the core Publisher  
 306 functions - observing the Situation and formatting the Notification artifact that describes the  
 307 Situation. The dissemination step occurs when the Publisher sends the Notify message to the  
 308 NotificationBroker.

309 In the broker initiated publishing pattern, the role of the Publisher is played by a Web service that  
 310 implements NotificationProducer. The act of observing the Situation and formatting the  
 311 Notification happens within the implementation logic of the NotificationProducer itself. The  
 312 Notification is disseminated by the NotificationProducer sending the Notify message to a  
 313 NotificationBroker. The Notification may also be disseminated by sending the Notify message to  
 314 any NotificationConsumer that are subscribing to the NotificationProducer.

315 Note: in either of the above two cases, the NotificationBroker MAY require the Publisher to  
 316 register with it prior to sending the Notify message. For example, if the broker wishes to control  
 317 who can publish to a given Topic, it can perform an access control check during this registration.  
 318 However a NotificationBroker MAY choose to allow Publishers to publish without pre-registration,  
 319 if it so chooses.

320 The last pattern, the demand-based pattern, requires the Publisher to be a NotificationProducer,  
 321 and thereby accept the Subscribe message. Demand-based publication is intended for use in  
 322 cases where the act of observing the Situation or the act of formatting the Notification artifact  
 323 might be expensive to perform, and therefore should be avoided if there are no interested parties  
 324 for that Notification. To use this pattern, the Publisher must register with the NotificationBroker,  
 325 using the registration to express the intent to provide demand-based publishing only. Based upon  
 326 this style of registration, the NotificationBroker sends the Subscribe message to the Publisher  
 327 (recall: in this situation the Publisher must implement the message exchanges associated with  
 328 the NotificationProducer interface).

329 Furthermore, the NotificationBroker is expected to pause its Subscription whenever it has no



330 active Subscribers for the information provided by the Publisher. When the NotificationBroker  
331 does have active Subscribers, it is obliged to resume its Subscription to the Publisher.

332

## 5 NotificationBroker Interface

333

The NotificationBroker interface defines a standard set of message exchanges to describe a message broker, providing an intermediary between Publishers and Subscribers on a collection of Topics, similar to a traditional Message Oriented Middleware model.

335

336

NotificationBroker MAY be a WS-Resource, and if it is, it MUST support the required message exchanges defined by the [\[WS-ResourceProperties\]](#) specification, and MAY support the optional message exchanges defined by WS-ResourceProperties.

Deleted: [WS-ResourceProperties]

339

A NotificationBroker MUST also support message exchanges and Resource Property elements defined by the following interfaces:

340

341

- NotificationProducer

342

- NotificationConsumer

343

- RegisterPublisher

344

The NotificationBroker portType aggregates the [three](#) portTypes and is not the only way to implement a broker. A distributed broker implementation can be achieved by hosting NotificationProducer, NotificationConsumer, or RegisterPublisher portTypes at one or more physical endpoints.

Deleted: <#>CreatePullPoint

Deleted: four

Deleted: CreatePullPoint,

345

346

347

348

The NotificationBroker does not specify any subscription durability or continuity.

349

NotificationBrokers SHOULD advertise their durability or reliability features, either through

350

policies or other means.

351

NotificationBrokers MAY offer flow control and MAY implement Pull-Style notifications. If so,

352

NotificationBrokers SHOULD advertise these features, either through policies or other means.

353

354

### 5.1 NotificationBroker Resource Properties

355

In addition to the message exchanges described in this specification, a NotificationBroker MAY also support the required message exchanges defined in the WS-ResourceProperties specification and MAY support the optional message exchanges defined in the WS-ResourceProperties specification. If it does so, the Resource Properties document defined by the NotificationBroker MUST include references to resource properties defined in NotificationProducer and NotificationConsumer, and also MUST include a reference to the following resource property element:

356

357

358

359

360

361

362

363

364

365

366

```

...
targetNamespace="http://docs.oasis-open.org/wsn/br-2>
...
<xsd:element name="RequiresRegistration" type="xsd:boolean"/>
...

```

Deleted: -1

367

Furthermore, this reference MUST reflect the minOccurs and maxOccurs properties as follows:



368  
369

```
<xsd:element ref="wsn-br:RequiresRegistration"
  minOccurs="1" maxOccurs="1" />
```

370 This resource property element is further constrained as follows:

371 /wsn-br:RequiresRegistration

372 The value is "true" if the NotificationBroker requires a publisher to register (see 6.1)  
373 before sending it a Notify (i.e. publish) message on a Topic. The default is "false".

374

## 5.2 Notify

375 The NotificationBroker MUST support the Notify message exchange from the  
376 NotificationConsumer interface [WS-BaseNotification], with the following clarifications/restrictions:

377 A Publisher sends a Notify message to a NotificationBroker in order to publish a Notification on a  
378 given Topic. As a result of the Publisher sending this message, Notifications are delivered to all  
379 NotificationConsumers subscribed on the given Topic. For some Topics (those that require a  
380 Publisher to pre-register), the sender must be a registered Publisher in order to successfully  
381 publish a Notification on the given Topic (see 6.1).

382

## 5.3 Subscribe

383 The NotificationBroker MUST support the Subscribe message exchange from the  
384 NotificationProducer interface [WS-BaseNotification]. A NotificationBroker MAY support any  
385 TopicExpression dialect.

386 A NotificationBroker is capable of routing or producing a sequence of zero or more Notifications.  
387 A Subscriber can register the interest of a NotificationConsumer to receive a subset of this  
388 sequence. A Subscriber sends a Subscribe message to a NotificationBroker in order to register  
389 this interest.

390 If the processing of a Subscribe message is successful, the NotificationBroker MUST produce a  
391 response message, as described in WS-BaseNotification, containing an endpoint reference to a  
392 Subscription resource representing a Subscription created as a result of processing the  
393 Subscribe request. Otherwise, the NotificationBroker must fault. WS-BaseNotification defines a  
394 set of these faults.

395

## 5.4 GetCurrentMessage

396 The NotificationBroker MUST support the GetCurrentMessage message exchange from the  
397 NotificationProducer interface [WS-BaseNotification].

398 As defined in WS-BaseNotification, in response to a GetCurrentMessage message, the  
399 NotificationBroker MAY return the last Notification published on a given Topic. This is a non-  
400 destructive read, allowing a newly-subscribed NotificationConsumer to get the last Notification  
401 that other NotificationConsumers have received.

402

## 5.5 RegisterPublisher

403 The NotificationBroker MUST support the RegisterPublisher message exchange from the  
404 RegisterPublisher interface.

405 A Publisher can register its interest to publish messages through the NotificationBroker by  
406 sending a RegisterPublisherRequest. The NotificationBroker is responsible for managing the  
407 registration, and sending a RegisterPublisherResponse to the Publisher if the registration process  
408 succeeds. Otherwise, the NotificationBroker MUST fault. These message exchanges are further  
409 specified in the following Section 6.

410

## 5.6 CreatePullPoint

411 The NotificationBroker MAY support pull-style notification [as defined in WS-BaseNotification](#) and  
412 attempt to create a PullPoint resource upon receiving a CreatePullPoint request. The  
413 NotificationBroker does not define additional constraints to its usage of the CreatePullPoint  
414 [operation](#).

415

**Deleted:** The NotificationBroker MUST support the CreatePullPoint interface. The CreatePullPoint interface standardizing the means by which a PullPoint resource is created. If a requestor wishes to create a new PullPoint resource, it MUST send a CreatePullPoint request to the NotificationBroker. ¶

**Deleted:** interface

**Deleted:** If the NotificationBroker does not support pull-style notification, it MUST response with the following fault upon receiving a CreatePullPoint request: ¶ PullNotificationNotSupportedFault ¶  
<#>The NotificationBroker does not support pull-style notification. ¶

**Formatted:** Bullets and Numbering

416

## 6 RegisterPublisher Interface

417

The RegisterPublisher interface contains message exchanges for publisher registration.

418

NotificationBroker implements the RegisterPublisher interface and is responsible for publisher

419

registration. A NotificationBroker may reject processing certain publisher registrations for reasons

420

such as lacking of authorization.

421

### 6.1 RegisterPublisher

422

The RegisterPublisher message is used by the Publisher to confirm its ability to publish on a

423

given Topic or set of Topics. If an entity wishes to register a publisher, it MUST send a

424

RegisterPublisher request message to the NotificationBroker. The format of the RegisterPublisher

425

request message is:

426

```

...
427 <wsn-br:RegisterPublisher>
428   <wsn-br:PublisherReference>
429     wsa:EndpointReference
430   </wsn-br:PublisherReference?>
431   <wsn-br:Topic Dialect = "xsd:anyURI">
432     {any}
433   </wsn-br:Topic>*
434   <wsn-br:Demand>
435     xsd:Boolean
436   </wsn-br:Demand?>
437   <wsn-br:InitialTerminationTime>
438     xsd:dateTime
439   </wsn-br:InitialTerminationTime?>
440   {any} *
441 </wsn-br:RegisterPublisher>
442 ...

```

Deleted: Request

Deleted: d

Deleted: "

Deleted: ">

Deleted: Request

443

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

444

<http://docs.oasis-open.org/wsn/brw-2/RegisterPublisher/RegisterPublisherRequest>.

Deleted: -1

445

446

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

447

[/wsn-br:RegisterPublisher/PublisherReference](#)

Deleted: wsn-br:

448

An OPTIONAL endpoint reference element from WS-Addressing [[WS-Addressing](#)], used

449

to identify an entity that wishes to become a Publisher. This component MUST appear if

450

the [/wsn-br:RegisterPublisher/Demand](#) component has value "true". If this component is

Deleted: /wsn-br:D

451

missing, the Publisher is either not a Web service, or does not wish to receive messages

452

from the NotificationBroker.

453

[/wsn-br:RegisterPublisher/Topic](#)

Deleted: wsn-br:

454 A set of TopicExpressions that identifies one or more Topics. If included, the given  
455 Publisher is registered to publish only on the set of Topics identified by this component. If  
456 this is missing the Publisher is registered to publish on any Topic supported by the  
457 NotificationBroker.

458 [/wsn-br:RegisterPublisher/Demand](#)

Deleted: wsn-br:

459 A Boolean element with the default value “false”. If its value is “true”, then the intent of the  
460 Publisher is to use a demand-based model from the NotificationBroker (see Section 4). In  
461 this case, the NotificationBroker must observe the rules associated with demand-based  
462 publishing, including establishing a Subscription with the Publisher on those Topics and  
463 pausing/resuming those Subscriptions as the NotificationBroker receives Subscriptions  
464 for those Topics.

465 [/wsn-br:RegisterPublisher/InitialTerminationTime](#)

Deleted: wsn-br:

466 This component contains the service requestor’s suggestion for the initial termination  
467 time of the PublisherRegistration resource being created. This time is relative to the time  
468 source used by the NotificationBroker. If the NotificationBroker is unable or unwilling to  
469 set the TerminationTime to the given value or greater, the RegisterPublisher request  
470 MUST [return an UnacceptableInitialTerminationTimeFault message](#). If the value is not “in  
471 the future” relative to the current time as known by the NotificationBroker, the  
472 RegisterPublisher request [MUST also return an UnacceptableInitialTerminationTimeFault](#)  
473 [message](#).

Deleted: r, then

Deleted: fault

Deleted: MUST fault

474 The use of the xsi:nil attribute with value “true” indicates there is no scheduled  
475 termination time requested for the RegisterPublisher. If the element does not include the  
476 time zone designation, the value of the element MUST be interpreted as universal time  
477 (UTC).

478 The publisher should take care when choosing a value for InitialTerminationTime, and  
479 any subsequent values that modify the TerminationTime property of the publisher  
480 registration. It is RECOMMENDED that the publisher choose termination time values that  
481 are significantly (several magnitude) greater than the network latency expected in the  
482 interaction between the publisher and the broker. In so doing, the designer avoids  
483 undesirable results, such as the termination time having expired prior to the receipt of the  
484 published message. The [\[WS-ResourceLifetime\]](#) specification, (Section 5.1 Regarding  
485 time) contains further suggestions on how designers should reason about time values in  
486 a WS-Resource Lifetime application.

Deleted:

Deleted: [WS-ResourceLifetime]

Deleted: 6.1

487 If this component is not included, the initial value of the TerminationTime resource  
488 property is dependent on the implementation of the NotificationBroker.

489 [/wsn-br:RegisterPublisher/{any}](#)

Deleted: wsn-br:RegisterPublisherRequest/

490 The RegisterPublisher [request](#) message allows for open content, in order to  
491 accommodate elements that may be needed by extensions built on WS-  
492 BrokeredNotification.

Deleted: Request

493 If a [/wsn-br:RegisterPublisher/Topic](#) component is included in the message, the  
494 NotificationBroker MUST register the Web service specified by the [/wsn-](#)  
495 [br:RegisterPublisher/PublisherReference](#) component as a Publisher on the set of Topics

496 identified by the /wsn-br:RegisterPublisher/Topic component. If for any reason the registration  
497 fails, the NotificationBroker MUST fault.

Deleted: T

498 As part of the processing of a RegisterPublisher request, the NotificationBroker creates a  
499 PublisherRegistration resource representing the registration. A new resource is created  
500 regardless of whether the same Publisher has previously registered with the NotificationBroker.  
501 The NotificationBroker **MUST** return a PublisherRegistrationReference **and may return a**  
502 **ConsumerReference** in the response to the RegisterPublisher request.

Deleted: s

503 **PublisherRegistrationReference** is a WS-Addressing endpoint reference and includes the address  
504 of a PublisherRegistrationManager service. **ConsumerReference** is a WS-Addressing endpoint  
505 reference to a NotificationConsumer that subscribes to notifications published by this registered  
506 Publisher. **If Demand value is false in the RegisterPublisher request, the NotificationBroker MUST**  
507 **include a ConsumerReference in the response.**

Deleted: This P

Deleted: and a reference  
property identifying the  
PublisherRegistration resource

Deleted: .

508 If the NotificationBroker accepts the RegisterPublisher request message, it must respond with a  
509 message of the following form:

```
510 ...  
511 <wsn-br:RegisterPublisherResponse>  
512   <wsn-br:PublisherRegistrationReference>  
513     <wsa:Address>  
514       Address of PublisherRegistration Manager  
515     </wsa:Address>  
516     ...  
517   </wsn-br:PublisherRegistrationReference>  
518   <wsn-br:ConsumerReference>  
519     <wsa:Address>  
520       Address of a NotificationConsumer with which the  
521       Publisher is registered  
522     </wsa:Address>  
523     ...  
524   </wsn-br:ConsumerReference?>  
525 </wsn-br:RegisterPublisherResponse>  
526 ...
```

Deleted:  
<wsa:ReferenceParameter  
s>¶  
  
PublisherRegistration  
Identifier¶  
  
</wsa:ReferenceParamete  
rs>¶

Deleted: ¶

527 The WS-Addressing [action] Message Addressing Property MUST contain the URI  
528 <http://docs.oasis-open.org/wsn/brw-2/RegisterPublisher/RegisterPublisherResponse>

Deleted: -1

529 The components of the RegisterPublisher response message are further described as follows:

530 [/wsn-br:RegisterPublisherResponse/PublisherRegistrationReference](#)

Deleted: /

531 A WS-Addressing endpoint reference to the PublisherRegistration resource created by  
532 the RegisterPublisher request message. **This element MUST present in the**  
533 **RegisterPublisher response message. The NotificationBroker may choose to include**  
534 **extra information such as reference parameters in this reference.**

Deleted: wsn-br:

535 [/wsn-br:RegisterPublisherResponse/ConsumerReference](#)

536 A WS-Addressing endpoint reference to a NotificationConsumer resource that accepts  
537 notifications for this publisher registration.

538 [Any Notification Messages sent by the Publisher \(and considered to take place under this](#)  
539 [registration\) MUST be sent to this endpoint reference.](#)

540 [The NotificationBroker MAY use this as a mechanism for identifying the Publisher as](#)  
541 [having registered.](#)

542 If the NotificationBroker does not succeed in responding to the RegisterPublisher request  
543 message with the RegisterPublisherResponse message, then it MUST send a fault. The  
544 NotificationBroker MUST fault if it rejects the publisher registration. This specification defines the  
545 following faults associated with failure to process the RegisterPublisher request message:

Deleted: ¶

546 [ResourceUnknownFault](#)

Deleted: ¶

547 ResourceUnknownFault

- 548 • The NotificationBroker is acting as a WS-Resource, and the resource identified in the  
549 message is not known to the Web service. This fault is specified by the WS-Resource  
550 [WS-Resource] specification.

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

551 InvalidTopicExpressionFault

- 552 • The TopicExpression presented in the request message is invalid. [This fault is specified](#)  
553 [in WS-BaseNotification.](#)

554 TopicNotSupportedFault

- 555 • The TopicExpression does not match any Topic supported by the NotificationBroker. [This](#)  
556 [fault is specified in WS-BaseNotification.](#)

557 PublisherRegistrationRejectedFault

- 558 • The publisher registration is rejected by the NotificationBroker. [The NotificationBroker](#)  
559 [MAY provide a hint in the fault message indicating why the registration is rejected.](#)

560 PublisherRegistrationFailedFault

- 561 • The publisher registration process has failed. [The NotificationBroker MAY include a hint](#)  
562 [in the fault message indicating why the registration is failed.](#)

563 [UnacceptableInitialTerminationTimeFault](#)

- 564 • [The value of InitialTerminationTime specified in the RegisterPublisher request message](#)  
565 [is not acceptable to the NotificationBroker. The NotificationBroker MAY include a hint in](#)  
566 [the fault message indicating why the value is unacceptable.](#)

Formatted: Bullets and  
Numbering

## 567 6.1.1 Example SOAP Encoding of the RegisterPublisher Message 568 Exchange

569 The following is a non-normative example of a RegisterPublisher request message using SOAP:

```
570 <s:Envelope ... >  
571   <s:Header>  
572     <wsa:Action>
```

573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602

```
    http://docs.oasis-open.org/wsn/brw-
2/RegisterPublisher/RegisterPublisherRequest
  </wsa:Action>
  ...
</s:Header>
<s:Body>
  <wsn-br:RegisterPublisher>
    <wsn-br:PublisherReference>
      <wsa:Address>
        http://www.example.org/PublisherEndpoint
      </wsa:Address>
      <wsa:ReferenceParameters>
        <npex:NPRResourceDisambiguator>
          uuid:84dec55-7d3f-65ad-ac44-675d9fce5d22
        </npex:NPRResourceDisambiguator>
      </wsa:ReferenceParameters>
    </wsn-br:PublisherReference>
    <wsn-br:Topic Dialect="http://docs.oasis-open.org/wsn/t-
1/TopicExpression/Simple">
      npex:SomeTopic
    </wsn-br:Topic>
    <wsn-br:Demand>
      true
    </wsn-br:Demand>
    <wsn-br:InitialTerminationTime>
      2003-12-25T00:00:00.000000Z
    </wsn-br:InitialTerminationTime>
  </wsn-br:RegisterPublisher>
</s:Body>
</s:Envelope>
```

Deleted: -1  
Deleted:  
Deleted: www.producer.org  
Deleted: http://docs.oasis-open.org/wsn/t-1/SimpleTopicExpression

603 The following is a non-normative example of a RegisterPublisher response message using  
604 SOAP:

605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621

```
<s:Envelope ... >
  <s:Header>
    <wsa:Action>
      http://docs.oasis-open.org/wsn/brw-
2/RegisterPublisher/RegisterPublisherResponse
    </wsa:Action>
    ...
  </s:Header>
  <s:Body>
    <wsn-br:RegisterPublisherResponse>
      <wsn-br:PublisherRegistrationReference>
        <wsa:Address>
          http://www.example.org/PublisherRegistrationManager
        </wsa:Address>
        <wsa:ReferenceParameters>
          <npex:NPubResourceId>
            uuid:95fefeb3-f37d-5dfe-44fe-221d9fceed99
          </npex:NPubResourceId>
        </wsa:ReferenceParameters>
      </wsn-br:PublisherRegistrationReference>
    </wsn-br:RegisterPublisherResponse>
  </s:Body>
</s:Envelope>
```

Deleted: -1  
Deleted: www.producer.org  
Deleted: e  
Deleted: Endpoint

622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633

```
</npex:NPubResourceId>  
</wsa:ReferenceParameters>  
</wsn-br:PublisherRegistrationReference>  
<wsn-br:ConsumerReference>  
  <wsa:Address>  
    http://www.example.org/NotificationConsumer  
  </wsa:Address>  
  ...  
</wsn-br:ConsumerReference>  
</wsn-br:RegisterPublisherResponse>  
</s:Body>  
</s:Envelope>
```



634

## 7 PublisherRegistrationManager Interface

635 The PublisherRegistrationManager interface defines message exchanges to manipulate  
636 PublisherRegistration resources. The PublisherRegistrationManager MAY be a WS-Resource,  
637 and if it is, the PublisherRegistrationManager WS-Resource MUST support the immediate  
638 termination interface defined by WS-RF Resource Lifetime and it MAY support the scheduled  
639 termination interface defined by WS-RF Resource Lifetime.

640 If the PublisherRegistrationManager does not respond to a request message with a respond  
641 message defined in this specification, then it MUST send a fault. The WS-ResourceProperties  
642 and WS-ResourceLifetime define some of these fault messages.

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

643

### 7.1 PublisherRegistration Resource Properties

644 In addition to the message exchanges described in this specification, a  
645 PublisherRegistrationManager MAY also support the required message exchanges defined in the  
646 WS-ResourceProperties specification and MAY support the optional message exchanges defined  
647 in the WS-ResourceProperties specification. If it does so, the Resource Properties document  
648 defined by the PublisherRegistrationManager MUST also include references to the following  
649 resource property elements:

650  
651  
652  
653  
654  
655  
656  
657  
658

```
.....  
targetNamespace="http://docs.oasis-open.org/wsn/br-2"  
...  
<xsd:element name="PublisherReference"  
  type="wsa:EndpointReference" />  
<xsd:element name="Topic" type="wsn-b:TopicExpressionType" />  
<xsd:element name="Demand" type="xsd:boolean" />  
<xsd:element name="CreationTime" type="xsd:dateTime" />  
...
```

Deleted: -1

659 Furthermore, these references MUST reflect the minOccurs and maxOccurs properties as  
660 follows:

661  
662  
663  
664  
665  
666  
667  
668

```
<xsd:element ref="wsn-br:PublisherReference"  
  minOccurs="0" maxOccurs="1" />  
<xsd:element ref="wsn-br:Topic"  
  minOccurs="0" maxOccurs="unbounded" />  
<xsd:element ref="wsn-br:Demand"  
  minOccurs="1" maxOccurs="1" />  
<xsd:element ref="wsn-br:CreationTime"  
  minOccurs="0" maxOccurs="1" />
```

669 These resource property elements are further constrained as follows:

670 /wsn-br:PublisherReference, /wsn-br:Topic, and /wsn-br:Demand

Deleted: ,

671 These elements are defined in the description of the RegisterPublisher request message  
672 (see 6.1).

673 /wsn-br:CreationTime

674 Indicates the date and time at which the PublisherRegistration was created. This is an  
675 optional component, supporting resource constrained devices which cannot associate a  
676 creation time with PublisherRegistration resources they create.

677 If PublisherRegistrationManager is a WS-Resource, the following resource properties MAY be  
678 modified by the requestor, by sending a SetResourceProperties request message as defined in  
679 the WS-ResourceProperties specification:

- 680 • /wsn-br:TopicExpression and /wsn-br:Demand
  - 681 ○ Note: /wsn-br:Demand may not take the value “true” if there is no /wsn-  
682 br:PublisherReference resource property element in the resource property  
683 document.

## 684 **7.2 DestroyRegistration**

685 The PublisherRegistrationManager interface provides a destroy operation, providing a means by  
686 which a requestor can terminate the publisher registration manager resource. To terminate  
687 PublisherRegistrationManager resource, a requestor MUST send a DestroyRegistration request  
688 message to the PublisherRegistrationManager. The DestroyRegistration request message has  
689 the following form:

690  
691  
692  
693  
694

```
<wsn-br:DestroyRegistration>  
  {any} *  
</wsn-br:DestroyRegistration>
```

Deleted: Destroy

Deleted: Request

Deleted: DestroyRequest

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

696 [http://docs.oasis-open.org/wsn/brw2/  
697 2/PublisherRegistrationManager/DestroyRegistrationRequest](http://docs.oasis-open.org/wsn/brw2/PublisherRegistrationManager/DestroyRegistrationRequest).

698 The DestroyRegistration request message allows for open content and contains an extension  
699 component

700 /wsn-br:DestroyRegistration/{any}.

701 Upon receipt of the DestroyRegistration request, the PublisherRegistrationManager MUST  
702 attempt to destroy itself. If the DestroyRegistration request message is successfully processed,  
703 the PublisherRegistrationManager MUST respond with the following message:

704  
705  
706

```
<wsn-br:DestroyRegistrationResponse />
```

Deleted: -1

Deleted:

Deleted: Destroy

Deleted: Destroy

Deleted: wsn-  
br:DestroyRequest/

Deleted: Destroy

Deleted: Destroy

Deleted: Destroy

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

708 [http://docs.oasis-open.org/wsn/brw-](http://docs.oasis-open.org/wsn/brw-2/PublisherRegistrationManager/DestroyRegistrationResponse)  
 709 [2/PublisherRegistrationManager/DestroyRegistrationResponse](http://docs.oasis-open.org/wsn/brw-2/PublisherRegistrationManager/DestroyRegistrationResponse). Deleted: -1  
 710 If the PublisherRegistrationManager does not respond to the [DestroyRegistration](#) request Deleted: Destroy  
 711 message with the [DestroyRegistrationResponse](#) message, then it MUST send a fault. This Deleted: Destroy  
 712 specification defines the following faults associated with failure to process the Deleted: Destroy  
 713 [DestroyRegistration](#) request message: Deleted: Destroy

714 ResourceUnknownFault

- 715 • The PublisherRegistrationManager is a WS-Resource, and the resource identified in the  
 716 message is not known to the Web service. This fault is specified by the WS-Resource  
 717 [WS-Resource] specification.

718 ResourceNotDestroyedFault

- 719 • The PublisherRegistrationManager was unable to destroy the  
 720 PublisherRegistrationManager resource for some reason.

## 721 7.2.1 Example SOAP Encoding of the [DestroyRegistration](#) Message 722 Exchange

723 The following is a non-normative example of a [DestroyRegistration](#) request message using  
 724 SOAP: Deleted: Destroy

```
725 <s:Envelope ... >
726   <s:Header>
727     <wsa:Action>
728       http://docs.oasis-open.org/wsn/brw-
729       2/PublisherRegistrationManager/DestroyRegistrationRequest Deleted: -1
730     </wsa:Action> Deleted: /
731     ... Deleted: Destroy
732   </s:Header>
733   <s:Body>
734     <wsn-br:DestroyRegistration/ > Deleted: DestroyRequest
735   </s:Body>
736 </s:Envelope>
```

737 The following is a non-normative example of a [DestroyRegistration](#) response message using  
 738 SOAP: Deleted: Destroy

```
739 <s:Envelope ... >
740   <s:Header>
741     <wsa:Action>
742       http://docs.oasis-open.org/wsn/brw-
743       2/PublisherRegistrationManager/DestroyRegistrationResponse Deleted: -1
744     </wsa:Action> Deleted: Destroy
745     ...
746   </s:Header>
747   <s:Body>
748     <wsn-br:DestroyRegistrationResponse/ > Deleted: Destroy
```

749  
750

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

---

## 751 8 Security Considerations

752 Baseline security considerations for WS-Notification are discussed in WS-BaseNotification  
753 specification. This section only covers additional broker specific security measurements.

### 754 8.1 Securing PublisherRegistration

755

756 In addition to the security policies for Notification process and Subscription process, WS-  
757 BrokeredNotification should provide policies such that:

- 758 1. only authorized Publishers can register with a NotificationBroker
- 759 2. only messages of the authorized Publishers and of registered topics, can be accepted by  
760 a NotificationBroker
- 761 3. only authorized principals can modify or delete PublisherRegistration resource

762 Given that WS-BrokeredNotification may implement WS-ResourceProperties and WS-  
763 ResourceLifetime, the security considerations outlined in these specifications need to be taken  
764 into account where appropriate. Authorization policies for those Resource Properties should be  
765 put in place so that the implications of providing the state information (through  
766 GetResourceProperty request messages) or through notification of state change and modification  
767 of the resource properties (through SetResourceProperty request messages), are taken into  
768 account.

769 A NotificationBroker can support the security measurements of NotificationProducers and  
770 NotificationConsumers mentioned in WS-BaseNotification. Acting as an intermediary,  
771 NotificationBroker MAY also provide convenience to security management, including but not  
772 limited to:

- 773 • Controlling who can publish on a topic at publisher registration time
- 774 • Refusing to relay messages from unauthorized publishers
- 775 • Imposing security measurements on all messaging routing through the broker
- 776 • Providing convenience in messaging security management based on topics.

777 NotificationBrokers SHOULD advertise, whether through policy assertions or other means, what  
778 security measures they take.

779



780

## 9 References

781

### 9.1 Normative

782

[RFC2119] S. Bradner, Key words for use in RFCs to Indicate Requirement Levels,

783

<http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.

784

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

785

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

786

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

787

[WS-BaseNotification] [http://docs.oasis-open.org/wsn/wsn-ws\\_base\\_notification-1.3-pr-02.pdf](http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-pr-02.pdf)

788

02.pdf

789

[WS-Topics] [http://docs.oasis-open.org/wsn/wsn\\_ws\\_topics-1.3-pr-01.pdf](http://docs.oasis-open.org/wsn/wsn_ws_topics-1.3-pr-01.pdf)

790

[WS-Resource] [http://docs.oasis-open.org/wsrf/wsrf-ws\\_resource-1.2-spec-pr-02.pdf](http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-pr-02.pdf)

791

[WS-ResourceLifetime] [http://docs.oasis-open.org/wsrf/wsrf-ws\\_resource\\_lifetime-1.2-spec-pr-02.pdf](http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-pr-02.pdf)

792

spec-pr-02.pdf

793

[WS-ResourceProperties] [http://docs.oasis-open.org/wsrf/wsrf-ws\\_resource\\_properties-1.2-spec-pr-02.pdf](http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-pr-02.pdf)

794

1.2-spec-pr-02.pdf

795

[WS-BaseFaults] [http://docs.oasis-open.org/wsrf/wsrf-ws\\_base\\_faults-1.2-spec-pr-02.pdf](http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-pr-02.pdf)

796

797

### 9.2 Non-Normative

798

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

799

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

800

801

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

Deleted: cd

Formatted: Indent: Left: 17.85 pt, Hanging: 122.45 pt, Don't adjust space between Latin and Asian text

Field Code Changed

Deleted: 1

Deleted: -

Deleted: WS-Topics

Deleted: cd

Deleted: cd

Deleted: 01

Deleted: cd-01

Deleted: cd-01

Deleted: cd

Deleted: 01

---

## 802 Appendix A. Acknowledgments

803 The following individuals were members of the committee during the development of this  
804 specification:

805 Sid Askary, Individual, Fred Carter, AmberPoint, Martin Chapman, Oracle, Dave Chappell, Sonic  
806 Software, Rick Cobb, KnowNow, Ugo Corda, SeeBeyond Technology Corporation, John Fuller,  
807 Individual, Stephen Graham, IBM, David Hull, Tibco, Hideharu Kato, Hitachi, Lily Liu,  
808 webMethods, Tom Maguire, IBM, Susan Malaika, IBM, Samuel Meder, Argonne National  
809 Laboratory, Bryan Murray, Hewlett-Packard, Peter Niblett, IBM, Sanjay Patil, SAP, Mark Peel,  
810 Novell, Matt Roberts, IBM, Igor Sedukhin, Computer Associates, David Snelling, Fujitsu, Latha  
811 Srinivasan, Hewlett-Packard, William Vambenepe, Hewlett-Packard, Kirk Wilson, Computer  
812 Associates.

813 Special thanks to the Global Grid Forum's Open Grid Services Infrastructure working group,  
814 which defined the OGSi v1.0 specification which was a large inspiration for the ideas expressed  
815 in this specification.

816 In addition, the following people who are not members of the committee made contributions to  
817 this specification:

818 Tim Banks (IBM), Nick Butler (IBM), Doug Davis (IBM), John Dinger (IBM), Don Ferguson (IBM),  
819 Jeff Frey (IBM), Andreas Koepfel (SAP), Heather Kreger (IBM), Amy Lewis (TIBCO Software),  
820 Kevin Liu (SAP), Nataraj Nagaratnam (IBM), Martin Nally (IBM), Jeff Nick (IBM), Jay Parikh  
821 (Akamai Technologies), Claus von Riegen (SAP), Rick Rineholt (IBM), John Rofrano (IBM),  
822 Shivajee Samdarshi (TIBCO Software), Igor Sedukhin (Computer Associates), Eugène  
823 Sindambiwe (SAP), Jay Unger (IBM), Bill Wehl (Akamai Technologies), Mark Weitzel (IBM), Dan  
824 Wolfson (IBM).



825

## Appendix B. XML Schema

826  
827

The XML types and elements used in WS-BrokeredNotification are defined in the following XML Schema

828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866

```
<?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 (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.
```

867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917

```
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.
-->

<xsd:schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsn-br="http://docs.oasis-open.org/wsn/br-2"
  xmlns:wsn-b="http://docs.oasis-open.org/wsn/b-2"
  xmlns:wsrf-bf="http://docs.oasis-open.org/wsrf/bf-2"
  xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"
  targetNamespace="http://docs.oasis-open.org/wsn/br-2"
  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/b-2"
    schemaLocation="http://docs.oasis-open.org/wsn/b-2.xsd"/>

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

<!-- ===== Resource Properties for NotificationBroker ===== -->
  <xsd:element name="RequiresRegistration" type="xsd:boolean"/>

<!-- ===== Resource Properties for PublisherRegistration ===== -->
  <xsd:element name="PublisherReference"
    type="wsa:EndpointReferenceType"/>
  <xsd:element name="ConsumerReference"
    type="wsa:EndpointReferenceType"/>
  <xsd:element name="Topic"
    type="wsn-b:TopicExpressionType"/>
  <xsd:element name="Demand"
    type="xsd:boolean"/>
  <xsd:element name="CreationTime"
    type="xsd:dateTime"/>
  <xsd:element name="NotificationBrokerRP">
    <xsd:complexType>
```

- Deleted: 3
- Deleted: -1
- Deleted: -1
- Deleted: 1
- Deleted: 1
- Deleted: 3
- Deleted: 03
- Deleted: 1
- Deleted: 1
- Deleted: -1
- Deleted: -1
- Deleted: ¶

918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968

```
<xsd:sequence>
  <!-- From NotificationProducer -->
  <xsd:element ref="wsn-b:TopicExpression"
    minOccurs="0" maxOccurs="unbounded" />
  <xsd:element ref="wsn-b:FixedTopicSet"
    minOccurs="0" maxOccurs="1" />
  <xsd:element ref="wsn-b:TopicExpressionDialect"
    minOccurs="0" maxOccurs="unbounded" />
  <xsd:element ref="wstop:TopicSet"
    minOccurs="0" maxOccurs="1" />
  <!-- NotificationBroker specific -->
  <xsd:element ref="wsn-br:RequiresRegistration"
    minOccurs="1" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<!-- ===== Resource Properties for PublisherRegistration ===== -->
<xsd:element name="PublisherRegistrationRP">
  <xsd:complexType>
    <xsd:sequence>
      <!-- From WS-ResourceLifetime ScheduledResourceTermination -->
      <xsd:element ref="wsn-b:CurrentTime"
        minOccurs="0" maxOccurs="1" />
      <xsd:element ref="wsn-b:TerminationTime"
        minOccurs="1" maxOccurs="1" />

      <!-- PublisherRegistration specific -->
      <xsd:element ref="wsn-br:PublisherReference"
        minOccurs="0" maxOccurs="1" />
      <xsd:element ref="wsn-br:Topic"
        minOccurs="0" maxOccurs="unbounded" />
      <xsd:element ref="wsn-br:Demand"
        minOccurs="1" maxOccurs="1" />
      <xsd:element ref="wsn-br:CreationTime"
        minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!-- ===== Message Types for NotificationBroker ===== -->
<xsd:element name="RegisterPublisher">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="PublisherReference"
        type="wsa:EndpointReferenceType"
        minOccurs="0" maxOccurs="1" />
      <xsd:element name="Topic"
        type="wsn-b:TopicExpressionType"
        minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="Demand"
```

Formatted: English (U.K.)

Formatted: English (U.K.)

```

969         type="xsd:boolean" default="false"
970         minOccurs="0" maxOccurs="1" />
971     <xsd:element name="InitialTerminationTime"
972         type="xsd:dateTime"
973         minOccurs="0" maxOccurs="1" />
974     <xsd:any namespace="##other" processContents="lax"
975         minOccurs="0" maxOccurs="unbounded" />
976     </xsd:sequence>
977 </xsd:complexType>
978 </xsd:element>
979
980 <xsd:element name="RegisterPublisherResponse">
981     <xsd:complexType>
982         <xsd:sequence>
983             <xsd:element name="PublisherRegistrationReference"
984                 type="wsa:EndpointReferenceType"
985                 minOccurs="1" maxOccurs="1" />
986             <xsd:element name="ConsumerReference"
987                 type="wsa:EndpointReferenceType"
988                 minOccurs="0" maxOccurs="1" />
989         </xsd:sequence>
990     </xsd:complexType>
991 </xsd:element>
992
993 <xsd:complexType name="PublisherRegistrationRejectedFaultType">
994     <xsd:complexContent>
995         <xsd:extension base="wsrf-bf:BaseFaultType" />
996     </xsd:complexContent>
997 </xsd:complexType>
998
999 <xsd:element name="PublisherRegistrationRejectedFault"
1000     type="wsn-br:PublisherRegistrationRejectedFaultType" />
1001
1002 <xsd:complexType name="PublisherRegistrationFailedFaultType">
1003     <xsd:complexContent>
1004         <xsd:extension base="wsrf-bf:BaseFaultType" />
1005     </xsd:complexContent>
1006 </xsd:complexType>
1007
1008 <xsd:element name="PublisherRegistrationFailedFault"
1009     type="wsn-br:PublisherRegistrationFailedFaultType" />
1010
1011
1012 <xsd:element name="DestroyRegistration">
1013     <xsd:complexType>
1014         <xsd:sequence>
1015             <xsd:any namespace="##other" processContents="lax"
1016                 minOccurs="0" maxOccurs="unbounded" />
1017         </xsd:sequence>
1018         <xsd:anyAttribute />
1019     </xsd:complexType>

```

Deleted: 0

Formatted: English (U.K.)

Formatted: English (U.K.)

Deleted: ¶

Deleted: ¶

```

<xsd:complexType
name="InvalidTopicExpressi
onFaultType">¶
<xsd:complexContent>¶
<xsd:extension
base="wsrf-
bf:BaseFaultType" />¶
</xsd:complexContent>¶
</xsd:complexType>¶
<xsd:element
name="InvalidTopicExpre
ssionFault" ¶
type="wsn-
br:InvalidTopicExpressi
onFaultType" />¶
¶
<xsd:complexType
name="TopicNotSupported
FaultType">¶
<xsd:complexContent>¶
<xsd:extension
base="wsrf-
bf:BaseFaultType" />¶
</xsd:complexContent>¶
</xsd:complexType>¶
<xsd:element
name="TopicNotSupported
Fault" ¶
type="wsn-
br:TopicNotSupportedFau
ltType" />¶

```

Deleted: <xsd:complexTyp  
e  
name="PullNotificationN  
otSupportedType">¶

```

<xsd:complexContent>¶
<xsd:extension
base="wsrf-
bf:BaseFaultType" />¶

```

[1]

1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040

```
</xsd:element>  
<xsd:element name="DestroyRegistrationResponse">  
  <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="ResourceNotDestroyedFaultType">  
  <xsd:complexContent>  
    <xsd:extension base="wsrf-bf:BaseFaultType"/>  
  </xsd:complexContent>  
</xsd:complexType>  
<xsd:element name="ResourceNotDestroyedFault"  
  type="wsn-br:ResourceNotDestroyedFaultType"/>  
  
</xsd:schema>
```

Deleted: Destroy

---

1041 **Appendix C. WSDL 1.1**

1042 The following illustrates the WSDL 1.1 for the Web service methods described in this  
1043 specification:

```
1044 <?xml version="1.0" encoding="utf-8"?>  
1045 <!--  
1046 OASIS takes no position regarding the validity or scope of any  
1047 intellectual property or other rights that might be claimed to pertain  
1048 to the implementation or use of the technology described in this  
1049 document or the extent to which any license under such rights might or  
1050 might not be available; neither does it represent that it has made any  
1051 effort to identify any such rights. Information on OASIS's procedures  
1052 with respect to rights in OASIS specifications can be found at the  
1053 OASIS website. Copies of claims of rights made available for  
1054 publication and any assurances of licenses to be made available, or the  
1055 result of an attempt made to obtain a general license or permission for  
1056 the use of such proprietary rights by implementors or users of this  
1057 specification, can be obtained from the OASIS Executive Director.  
1058  
1059 OASIS invites any interested party to bring to its attention any  
1060 copyrights, patents or patent applications, or other proprietary rights  
1061 which may cover technology that may be required to implement this  
1062 specification. Please address the information to the OASIS Executive  
1063 Director.  
1064  
1065 Copyright (C) OASIS Open (2005). All Rights Reserved.  
1066  
1067 This document and translations of it may be copied and furnished to  
1068 others, and derivative works that comment on or otherwise explain it or  
1069 assist in its implementation may be prepared, copied, published and  
1070 distributed, in whole or in part, without restriction of any kind,  
1071 provided that the above copyright notice and this paragraph are  
1072 included on all such copies and derivative works. However, this  
1073 document itself may not be modified in any way, such as by removing the  
1074 copyright notice or references to OASIS, except as needed for the  
1075 purpose of developing OASIS specifications, in which case the  
1076 procedures for copyrights defined in the OASIS Intellectual Property  
1077 Rights document must be followed, or as required to translate it into  
1078 languages other than English.  
1079  
1080 The limited permissions granted above are perpetual and will not be  
1081 revoked by OASIS or its successors or assigns.  
1082
```

1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133

```
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-BrokeredNotification"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsn-br="http://docs.oasis-open.org/wsn/br-2"
  xmlns:wsn-brw="http://docs.oasis-open.org/wsn/brw-2"
  xmlns:wsn-b="http://docs.oasis-open.org/wsn/b-2"
  xmlns:wsn-bw="http://docs.oasis-open.org/wsn/bw-2"
  xmlns:wsrf-bf="http://docs.oasis-open.org/wsr/bf-2"
  xmlns:wsrf-rw="http://docs.oasis-open.org/wsr/rw-2"
  targetNamespace="http://docs.oasis-open.org/wsn/brw-2">

<!-- ===== Imports ===== -->
  <wsdl:import namespace="http://docs.oasis-open.org/wsr/rw-2"
    location="http://docs.oasis-open.org/wsr/rw-2.wsdl"/>

  <wsdl:import namespace="http://docs.oasis-open.org/wsn/bw-2"
    location="http://docs.oasis-open.org/wsn/bw-2.wsdl"/>

<!-- ===== Types Definitions ===== -->
  <wsdl:types>
    <xsd:schema>
      <xsd:import
        namespace="http://docs.oasis-open.org/wsn/br-2"
        schemaLocation="http://docs.oasis-open.org/wsn/br-2.xsd"/>
      </xsd:schema>
    </wsdl:types>

<!-- ===== NotificationBroker::RegisterPublisher =====
  RegisterPublisher(PublisherReference, TopicExpression* ,
    [Demand], [InitialTerminationTime])
  returns: WS-Resource qualified EPR to a PublisherRegistration -->
  <wsdl:message name="RegisterPublisherRequest">
    <wsdl:part name="RegisterPublisherRequest"
      element="wsn-br:RegisterPublisher"/>
  </wsdl:message>

  <wsdl:message name="RegisterPublisherResponse">
    <wsdl:part name="RegisterPublisherResponse"
      element="wsn-br:RegisterPublisherResponse"/>
  </wsdl:message>
```

- Deleted: 3
- Deleted: -1
- Deleted: -1
- Deleted: -1
- Deleted: -1
- Formatted: Tabs: 297 pt, Left
- Deleted: 1
- Deleted: 1
- Deleted: 1
- Deleted: 1
- Deleted: 1
- Deleted: -1
- Deleted: -1

```

1134 <wsdl:message name="PublisherRegistrationRejectedFault">
1135 <wsdl:part name="PublisherRegistrationRejectedFault"
1136     element="wsn-br:PublisherRegistrationRejectedFault" />
1137 </wsdl:message>
1138
1139 <wsdl:message name="PublisherRegistrationFailedFault">
1140 <wsdl:part name="PublisherRegistrationFailedFault"
1141     element="wsn-br:PublisherRegistrationFailedFault" />
1142 </wsdl:message>
1143
1144 <wsdl:message name="DestroyRegistrationRequest">
1145 <wsdl:part name="DestroyRegistrationRequest"
1146     element="wsn-br:DestroyRegistration" />
1147 </wsdl:message>
1148
1149 <wsdl:message name="DestroyRegistrationResponse">
1150 <wsdl:part name="DestroyRegistrationResponse"
1151     element="wsn-br:DestroyRegistrationResponse" />
1152 </wsdl:message>
1153
1154 <wsdl:message name="ResourceNotDestroyedFault">
1155 <wsdl:part name="ResourceNotDestroyedFault"
1156     element="wsn-br:ResourceNotDestroyedFault" />
1157 </wsdl:message>
1158
1159 <!-- ===== PortType Definitions ===== -->
1160
1161 <!-- ===== RegisterPublisher ===== -->
1162 <wsdl:portType name="RegisterPublisher">
1163 <wsdl:operation name="RegisterPublisher">
1164 <wsdl:input message="wsn-brw:RegisterPublisherRequest" />
1165 <wsdl:output message="wsn-brw:RegisterPublisherResponse" />
1166 <wsdl:fault name="ResourceUnknownFault"
1167     message="wsrf-rw:ResourceUnknownFault" />
1168 <wsdl:fault name="InvalidTopicExpressionFault"
1169     message="wsn-bw:InvalidTopicExpressionFault" />
1170 <wsdl:fault name="TopicNotSupportedFault"
1171     message="wsn-bw:TopicNotSupportedFault" />
1172 <wsdl:fault name="PublisherRegistrationRejectedFault"
1173     message="wsn-brw:PublisherRegistrationRejectedFault" />
1174 <wsdl:fault name="PublisherRegistrationFailedFault"
1175     message="wsn-brw:PublisherRegistrationFailedFault" />
1176 <wsdl:fault name="UnacceptableInitialTerminationTimeFault"
1177     message="wsn-bw:UnacceptableInitialTerminationTimeFault" />
1178 </wsdl:operation>
1179 </wsdl:portType>
1180
1181 <!-- ===== NotificationBroker PortType Definition ===== -->
1182 <wsdl:portType name="NotificationBroker">
1183 <!-- ===== extends NotificationConsumer ===== -->
1184 <wsdl:operation name="Notify">

```

**Deleted:**

```

<wsdl:message
name="InvalidTopicExpres
sionFault">¶
    <wsdl:part
name="InvalidTopicExpres
sionFault"¶
    </wsdl:message> ¶
¶
<wsdl:message
name="TopicNotSupported
Fault">¶
    <wsdl:part
name="TopicNotSupported
Fault"¶
    element="wsn-
br:InvalidTopicExpres
sionFault" />¶
</wsdl:message> ¶
¶
<wsdl:message
name="TopicNotSupported
Fault">¶
    <wsdl:part
name="TopicNotSupported
Fault"¶
    element="wsn-
br:TopicNotSupportedFau
lt" />¶
</wsdl:message> ¶
¶

```

**Deleted: ¶**

```

<wsdl:message
name="PullNotificationN
otSupportedFault">¶
    <wsdl:part
name="PullNotificationN
otSupportedFault"¶
    element="wsn-
br:PullNotificationNotS
upportedFault" />¶ ... [2]

```

**Deleted: Destroy**

**Deleted: Destroy**

**Deleted: Destroy**

**Deleted: Destroy**

**Deleted: Destroy**

**Deleted: Destroy**

**Deleted: r**

**Deleted: r**

**Deleted: ¶**



```

1185     <wsdl:input message="wsn-bw:Notify" />
1186 </wsdl:operation>
1187
1188     <!-- ===== extends NotificationProducer ===== -->
1189     <wsdl:operation name="Subscribe">
1190         <wsdl:input message="wsn-bw:SubscribeRequest" />
1191         <wsdl:output message="wsn-bw:SubscribeResponse" />
1192         <wsdl:fault name="ResourceUnknownFault"
1193             message="wsrf-rw:ResourceUnknownFault" />
1194         <wsdl:fault name="InvalidFilterFault"
1195             message="wsn-bw:InvalidFilterFault" />
1196         <wsdl:fault name="TopicExpressionDialectUnknownFault"
1197             message="wsn-
1198 bw:TopicExpressionDialectUnknownFault" />
1199         <wsdl:fault name="InvalidTopicExpressionFault"
1200             message="wsn-bw:InvalidTopicExpressionFault" />
1201         <wsdl:fault name="TopicNotSupportedFault"
1202             message="wsn-bw:TopicNotSupportedFault" />
1203         <wsdl:fault name="InvalidProducerPropertiesExpressionFault"
1204             message="wsn-
1205 bw:InvalidProducerPropertiesExpressionFault" />
1206         <wsdl:fault name="InvalidMessageContentExpressionFault"
1207             message="wsn-bw:InvalidMessageContentExpressionFault" />
1208         <wsdl:fault name="UnacceptableInitialTerminationTimeFault"
1209             message="wsn-bw:UnacceptableInitialTerminationTimeFault" />
1210         <wsdl:fault name="UnrecognizedPolicyRequestFault"
1211             message="wsn-bw:UnrecognizedPolicyRequestFault" />
1212         <wsdl:fault name="UnsupportedPolicyRequestFault"
1213             message="wsn-bw:UnsupportedPolicyRequestFault" />
1214         <wsdl:fault name="NotifyMessageNotSupportedFault"
1215             message="wsn-bw:NotifyMessageNotSupportedFault" />
1216         <wsdl:fault name="SubscribeCreationFailedFault"
1217             message="wsn-bw:SubscribeCreationFailedFault" />
1218     </wsdl:operation>
1219     <wsdl:operation name="GetCurrentMessage">
1220         <wsdl:input message="wsn-bw:GetCurrentMessageRequest" />
1221         <wsdl:output message="wsn-bw:GetCurrentMessageResponse" />
1222         <wsdl:fault name="ResourceUnknownFault"
1223             message="wsrf-rw:ResourceUnknownFault" />
1224         <wsdl:fault name="TopicExpressionDialectUnknownFault"
1225             message="wsn-bw:TopicExpressionDialectUnknownFault" />
1226         <wsdl:fault name="InvalidTopicExpressionFault"
1227             message="wsn-bw:InvalidTopicExpressionFault" />
1228         <wsdl:fault name="TopicNotSupportedFault"
1229             message="wsn-bw:TopicNotSupportedFault" />
1230         <wsdl:fault name="NoCurrentMessageOnTopicFault"
1231             message="wsn-bw:NoCurrentMessageOnTopicFault" />
1232         <wsdl:fault name="MultipleTopicsSpecifiedFault"
1233             message="wsn-bw:MultipleTopicsSpecifiedFault" />
1234     </wsdl:operation>
1235

```

**Deleted:**  
<wsdl:fault  
name="InvalidUseRawValue  
Fault" ¶  
message="wsn-  
bw:InvalidUseRawValueFa  
ult" />¶

**Deleted:** ¶

1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271

```

<!-- ===== extends RegisterPublisher ===== -->
<wsdl:operation name="RegisterPublisher">
  <wsdl:input message="wsn-brw:RegisterPublisherRequest" />
  <wsdl:output message="wsn-brw:RegisterPublisherResponse" />
  <wsdl:fault name="ResourceUnknownFault"
    message="wsrf-rw:ResourceUnknownFault" />
  <wsdl:fault name="InvalidTopicExpressionFault"
    message="wsn-bw:InvalidTopicExpressionFault" />
  <wsdl:fault name="TopicNotSupportedFault"
    message="wsn-bw:TopicNotSupportedFault" />
  <wsdl:fault name="PublisherRegistrationRejectedFault"
    message="wsn-brw:PublisherRegistrationRejectedFault" />
  <wsdl:fault name="PublisherRegistrationFailedFault"
    message="wsn-brw:PublisherRegistrationFailedFault" />
  <wsdl:fault name="UnacceptableInitialTerminationTimeFault"
    message="wsn-bw:UnacceptableInitialTerminationTimeFault" />
</wsdl:operation>
</wsdl:portType>

<!-- ===== PublisherRegistrationManager PortType Definition ===== -->
<wsdl:portType name="PublisherRegistrationManager">
  <!--==== DestroyRegistration: ImmediateResourceTermination====-->
  <wsdl:operation name="DestroyRegistration">
    <wsdl:input name="DestroyRegistrationRequest"
      message="wsn-brw:DestroyRegistrationRequest" />
    <wsdl:output name="DestroyRegistrationResponse"
      message="wsn-brw:DestroyRegistrationResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrf-rw:ResourceUnknownFault" />
    <wsdl:fault name="ResourceNotDestroyedFault"
      message="wsn-brw:ResourceNotDestroyedFault" />
  </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>

```

Deleted: r

Deleted: r

Deleted: ¶

Deleted: <!--  
===== extends  
CreatePullPoint =====  
--> ¶  
<wsdl:operation  
name="CreatePullPoint">¶  
<wsdl:input  
name="CreatePullPointRe  
quest" ¶  
message="wsn-  
bw:CreatePullPointReque  
st" />¶  
<wsdl:output  
name="CreatePullPointRe  
sponse" ¶  
message="wsn-  
bw:CreatePullPointRespo  
nse" />¶  
<wsdl:fault  
name="UnableToCreatePul  
lPoint" ¶  
message="wsn-  
bw:UnableToCreatePullPo  
int" />¶  
<wsdl:fault  
name="PullNotificationN  
otSupportedFault" ¶  
message="wsn-  
brw:PullNotificationNot  
SupportedFault" />¶  
</wsdl:operation>¶

Deleted:

Deleted:

Deleted: Destroy

Deleted:

Deleted: Destroy

Deleted: Destroy

Deleted: Destroy

Deleted: Destroy

Deleted: Destroy

Deleted: Destroy

Deleted: Destroy

## Appendix D. Revision History

<u>Rev</u>	<u>Date</u>	<u>By Whom</u>	<u>What</u>
<u>1.2 01</u>	<u>2004-05-12</u>	<u>Lily Liu</u>	<u>Initial version</u>
<u>1.2 02</u>	<u>2004-06-07</u>	<u>Dave Chappell</u>	<u>Updates and consistency check w/ other WS-N specs</u>
<u>1.2 03</u>	<u>2004-06-24</u>	<u>Lily Liu, Dave Chappell</u>	<u>Addition of a Goals and Requirements section and minor format changes</u>
<u>1.2 03</u>	<u>2004-07-12</u>	<u>Lily Liu</u>	<u>Addition of a status paragraph</u>
<u>1.3 01a – 1.3 01e</u>	<u>2005-02-01</u>	<u>Dave Chappell, Lily Liu</u>	<u>Series of issue resolution and consistency reviews with WS-BaseNotification</u>
<u>1.3 01f</u>	<u>2005-06-10</u>	<u>Lily Liu</u>	<u>Issues: 3.1, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.20</u> <u>Updated the Terminology, Introduction, and Security sections.</u> <u>Updated sections about NotificationBroker and PublisherRegistrationManager resource properties.</u>
<u>1.3 01g</u>	<u>2005-07-01</u>	<u>Lily Liu</u>	<u>Updated the status section.</u> <u>Changed term NotificationMessage to Notification.</u> <u>Added CreatePullPoint portType to NotificationBroker.</u> <u>Completed issue resolutions.</u> <u>Replaced the Abstract section.</u>
<u>1.3 02d</u>	<u>2005-11-04</u>	<u>Lily Liu</u>	<u>Included changes to address: WSN 2.62, WSN 3.23, WSN 3.24, WSN 3.25, WSN 3.26, WSN 3.28, and WSN 3.29</u> <u>Resolved AI 137, AI 138, AI 141, AI 142 AI 144, and AI 145.</u>

<u>Rev</u>	<u>Date</u>	<u>By Whom</u>	<u>What</u>
			<u>Updated references.</u>

1273



**Deleted:** ¶

**Formatted:** Bullets and Numbering

1274

---

## Appendix E. Notices

1275 OASIS takes no position regarding the validity or scope of any intellectual property or other rights  
1276 that might be claimed to pertain to the implementation or use of the technology described in this  
1277 document or the extent to which any license under such rights might or might not be available;  
1278 neither does it represent that it has made any effort to identify any such rights. Information on  
1279 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS  
1280 website. Copies of claims of rights made available for publication and any assurances of licenses  
1281 to be made available, or the result of an attempt made to obtain a general license or permission  
1282 for the use of such proprietary rights by implementers or users of this specification, can be  
1283 obtained from the OASIS Executive Director.

1284 OASIS invites any interested party to bring to its attention any copyrights, patents or patent  
1285 applications, or other proprietary rights which may cover technology that may be required to  
1286 implement this specification. Please address the information to the OASIS Executive Director.

1287 Copyright © OASIS Open 2004. All Rights Reserved.

1288 This document and translations of it may be copied and furnished to others, and derivative works  
1289 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,  
1290 published and distributed, in whole or in part, without restriction of any kind, provided that the  
1291 above copyright notice and this paragraph are included on all such copies and derivative works.  
1292 However, this document itself does not be modified in any way, such as by removing the  
1293 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS  
1294 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual  
1295 Property Rights document must be followed, or as required to translate it into languages other  
1296 than English.

1297 The limited permissions granted above are perpetual and will not be revoked by OASIS or its  
1298 successors or assigns.

1299 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
1300 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO  
1301 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE  
1302 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
1303 PARTICULAR PURPOSE.

```
<xsd:complexType name="PullNotificationNotSupportedType">
  <xsd:complexContent>
    <xsd:extension base="wsrf-bf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="PullNotificationNotSupportedFault"
  type="wsn-br:PullNotificationNotSupportedType"/>
```

```
<wsdl:message name="PullNotificationNotSupportedFault">
  <wsdl:part name="PullNotificationNotSupportedFault"
    element="wsn-br:PullNotificationNotSupportedFault"/>
</wsdl:message>
```