

# Web Services Brokered Notification 1.3(WS-BrokeredNotification)

# ■ Public Review Draft 03, 31 May 2006

#### Document identifier:

wsn-ws\_brokered\_notification-1.3-spec-pr-03

#### 9 Location:

http://docs.oasis-open.org/wsn/wsn-ws\_brokered\_notification-1.3-spec-pr-03.pdf

#### **Editors:**

Dave Chappell, Sonic Software <chappell@sonicsoftware.com> Lily Liu, webMethods <lily.liu@webmethods.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].

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 1 of 43

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

#### Status:

 On May 31<sup>st</sup>, 2006, 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 archived and can be viewed at http://lists.oasis-open.org/archives/wsn-comment/.

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

The errata document for this specification is maintained at: http://docs.oasis-open.org/wsn/wsn-ws\_brokered\_notification-1.3-errata.pdf

# **Table of Contents**

51	1	Introduction	4
52		1.1 Goals and Requirements	4
53		1.1.1 Requirements	4
54		1.1.2 Non-Goals	5
55		1.2 Notational Conventions	5
56		1.3 Namespaces	7
57		1.4 Fault Definitions	7
58	2	Relationship to Other Specifications	8
59	3	Terminology and Concepts	9
60	4	Publishing	12
61	5	NotificationBroker Interface	15
62		5.1 NotificationBroker Resource Properties	16
63		5.2 Notify	16
64		5.3 Subscribe	16
65		5.4 GetCurrentMessage	17
66		5.5 RegisterPublisher	17
67		5.6 CreatePullPoint	17
86	6	RegisterPublisher Interface	18
69		6.1 RegisterPublisher	18
70		6.1.1 Example SOAP Encoding of the RegisterPublisher Message Exchange	22
71	7	PublisherRegistrationManager Interface	24
72		7.1 PublisherRegistration Resource Properties	24
73		7.2 DestroyRegistration	25
74		7.2.1 Example SOAP Encoding of the DestroyRegistration Message Exchange	26
75	8	Security Considerations	27
76		8.1 Securing PublisherRegistration	27
77	9	References	28
78		9.1 Normative	28
79		9.2 Non-Normative	28
80	Αp	pendix A. Acknowledgments	30
81	Αŗ	pendix B. XML Schema	31
82	Αp	pendix C. WSDL 1.1	36
83	Αŗ	ppendix D. Revision History	41
84	Ap	ppendix E. Notices	43
	WS	sn-ws brokered notification-1.3-spec-pr-03	5/31/2006

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

## 1 Introduction

85

99

104

105

106 107

108

109

110

111 112

113

114

115

116

117

- 86 The Event-driven, or Notification-based, interaction pattern is a commonly used pattern for inter-
- 87 object communications. Examples exist in many domains, for example, in publish/subscribe
- 88 systems or in system and device management domains. Message brokers are involved in many
- 89 of these systems, such as the ones provided by Message Oriented Middleware vendors.
- 90 This specification defines the Web services interface for the NotificationBroker, A
- 91 NotificationBroker is an intermediary between message Publishers and message Subscribers. A
- 92 NotificationBroker decouples NotificationProducers and Notification Consumers and can provide
- 93 advanced messaging features such as demand-based publishing and load-balancing. A
- 94 NotificationBroker also allows publication of messages from entities that are not themselves
- 95 service providers. This is very similar to a traditional Message Oriented 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.

Deleted: Common functions of Publishers and Subscribers, such as messaging dissemination and security measurements, can be implemented at the NotificationBroker to produce lightweight Producers and Consumers.

#### 1.1 Goals and Requirements

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

#### 1.1.1 Requirements

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

- Must allow for a notification broker as an intermediary. A NotificationBroker is an intermediary Web service that decouples NotificationConsumers from Publishers. A notification broker can relieve a Publisher from having to implement message exchanges associated with NotificationProducer; the NotificationBroker takes on the duties of subscription management and distributing Notifications on behalf of the Publisher. It implements NotificationProducer interface. It may implement SubscriptionManager or may delegate the subscription management work to another component.
- Must allow for federation of brokers. It must be possible to build configurations with
  multiple intermediary broker services in a dynamic fashion. This specification must allow for
  a variety of broker topology usage patterns. Among other things, these allow for greater
  scalability and permit sharing of administrative workload.
- Must provide runtime metadata: There must be a mechanism that lets a potential
   Subscriber discover what elements available for a subscription are provided by a
   NotificationBroker, and in what formats the subscription for a notification can be made.

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 4 of 43

**Deleted:** objectives

- Must conform to WS-BaseNotification: A NotificationBroker must support required
   message exchanges defined by the [WS-BaseNotification] specification. It must conform to
   the NotificationProducer and the NotificationConsumer interfaces defined in WS BaseNotification.
- WS-BrokeredNotification must be independent of binding-level details: Transport
   protocol details must be orthogonal to the subscription and the delivery of the notifications, so
   that the specification can be used over a variety of different transports.
- Must not exclude non-service producers and subscribers: WS-BrokeredNotification
   design must not exclude a non-service entity to deliver a notification message to a
   NotificationBroker. It must not exclude a NotificationBroker to send a notification message to
   a non-service consumer.
- Must provide publisher registration: WS-BrokeredNotification must define standard
   message exchanges for registering a NotificationPublisher with a NotificationBroker.

1.1.2 Non-Goals

134

136

137

138

144

154

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

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

#### 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
- the [XML-Infoset]. Specifically, abstract property names always appear in square brackets (e.g.,
- 150 [some property]).
- 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
- describe attributes and elements:
  - `?' denotes optionality (i.e. zero or one occurrences),
- `\*' denotes zero or more occurrences,
- 156 `+' one or more occurrences.

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Deleted:

Page 5 of 43

**Deleted:**, so that the specification can be used over a variety of different transports.

- `[' and `]' are used to form groups,
  - `|' represents choice.
  - Attributes are conventionally assigned a value which corresponds to their type, as defined in the normative schema.
  - Elements with simple content are conventionally assigned a value which corresponds to the type of their content, as defined in the normative schema.
  - The use of {any} indicates the presence of an element wildcard (<xs:any/>).
  - The use of @{any} indicates the presence of an attribute wildcard (<xs:anyAttribute/>).

```
165
166
167
```

168

169

170

171

172

173

174

158

159

160

161

162

163

164

```
<!-- sample pseudo-schema -->
<element
    required_attribute_of_type_QName="xs:QName"
    optional_attribute_of_type_string="xs:string"?>
    <required_element />
    <optional_element /> ?
    <one_or_more_of_these_elements /> +
    [ <choice_1 /> | <choice_2 /> ] *
</element>
```

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

## 180 1.3 Namespaces

181 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		
wsn-b http://docs.oasis-open.org/wsn/b-2			
wsn-br http://docs.oasis-open.org/wsn/br-2			
wsn-bw http://docs.oasis-open.org/wsn/bw-2			
wsn-brw	http://docs.oasis-open.org/wsn/brw-2		
wsrf-bf	http://docs.oasis-open.org/wsrf/bf-2		
wsrf-bfw http://docs.oasis-open.org/wsrf/bfw-2			

#### 1.4 Fault Definitions

182

- All faults generated by a NotificationBroker, RegisterPublisher, or PublisherRegistrationManager
- 184 SHOULD be compliant with the WS-BaseFaults [WS-BaseFaults] specification.
- All faults defined by this specification MUST use the following URI for the WS-Addressing [action]
  Message Addressing Property:
- 187 http://docs.oasis-open.org/wsn/fault.

# 2 Relationship to Other Specifications

- This specification builds on the basic notification mechanism defined in [WS-BaseNotification] by \_\_\_\_\_\_ Deleted: ,
- adding the concept of an intermediary NotificationBroker, and describing additional variants on
- the publisher role. A NotificationBroker takes on the role of both NotificationProducer and
- 192 NotificationConsumer (as defined in [WS-BaseNotification]), and its interactions with other
- 193 NotificationProducers and NotificationConsumers are largely defined by the WS-BaseNotification
- 194 specification.

188

- 195 This means that a NotificationBroker, implemented to conform to this specification, must also
- 196 conform to [WS-BaseNotification]. Such a NotificationBroker can deliver notifications to
- 197 NotificationConsumers that are implemented to conform to [WS-BaseNotification], and can
- 198 subscribe to Notifications distributed by NotificationProducers as defined in [WS-
- 199 BaseNotification].
- 200 A NotificationBroker may support hierarchical topics as defined in [WS-Topics]. By supporting
- 201 topics, NotificationBroker can manage enterprise messaging systems more efficiently.
- 202 WS-BrokeredNotification must be composable with other Web services specifications.

# 3 Terminology and Concepts

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

#### Publisher:

203

206

207

208

209

210

211

212 213

214

215

216 217

218

219

220

221

222

223

224

225 226

227

228

229

230

231232

233

234

235

236

237

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

#### NotificationBroker:

- 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.
- A NotificationBroker aggregates NotificationProducer, NotificationConsumer, and RegisterPublisher interfaces.
- Acting as an intermediary, a NotificationBroker provides additional capabilities to the base NotificationProducer interface:
  - 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.
  - It can reduce the number of inter-service connections and references, if there are many Publishers and many NotificationConsumers.
  - It can act as a finder service. Potential Publishers and Subscribers can in effect find each other by utilizing a common NotificationBroker.
  - It can provide anonymous Notification, so that the Publishers and the NotificationConsumers need not be aware of each other's identity.

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 9 of 43

- An implementation of a NotificationBroker may provide additional added-value function
   that is beyond the scope of this specification, for example, logging Notifications, or
   transforming Topics and/or Notification content. Additional function provided by a
   NotificationBroker can apply to all Publishers that utilize it.
  - It may be the factory for Subscription resources or it may delegate the subscription factory to another component.
  - A NotificationBroker provides publisher registration functions.
  - A NotificationBroker may <u>bridge between WS-Notification and other publish/subscribe</u> systems.

#### PublisherRegistration:

242 243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

- PublisherRegistration is a resource. A PublisherRegistration represents the relationship between a Publisher and a NotificationBroker, in particular, which topic(s) the publisher is permitted to publish to.
- A PublisherRegistration resource is created when a Publisher sends the RegisterPublisher request message to a NotificationBroker and the NotificationBroker succeeds in processing the registration.
- PublisherRegistration resources can be manipulated by messages sent to a PublisherRegistrationManager Web service.

#### RegisterPublisher:

 A RegisterPublisher is a Web service that implements the message exchanges associated with the RegisterPublisher interface. A PublisherRegistration resource is created as a result of a RegisterPublisher request to a NotificationBroker.

#### PublisherRegistrationManager:

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

A PublisherRegistration resource can be manipulated through

- PublisherRegistrationManager message exchanges.
   A PublisherRegistrationManager provides services that allow a service requestor to query
- A PublisherRegistration Manager provides services that allow a service requestor to querand manipulate PublisherRegistration resources that it manages.
   A PublisherRegistrationManager is subordinate to the NotificationBroker, and MAY be
- A PublisherRegistrationManager is subordinate to the NotificationBroker, and MAY be implemented by the NotificationBroker service provider. However WS-BrokeredNotification permits it to be implemented by a separate service provider, should an implementer so desire.

#### **Demand-Based Publishing:**

 Some Publishers may be interested in knowing whether they have any Subscribers or not, since producing a Notification may be a costly process. Such Publishers can register with the NotificationBroker as a Demand-Based Publisher.

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 10 of 43

**Deleted:** subscribe and disseminate messages that are not WS-Notification conforming.

Deleted: p

Demand-Based Publishers implement message exchanges associated with the
 NotificationProducer interface.

277

278 279

280

281

282

283 284

285

- The NotificationBroker subscribes to the Demand-Based Publisher. When the
  NotificationBroker knows that there are no Subscribers for the Notifications from a
  Demand-Based Publisher, it pauses its Subscription with that Publisher; when it knows
  that there are some Subscribers, it resumes the Subscription.
  - This way the Demand-Based Publisher does not need to produce messages when there
    are no Subscribers, however a Demand-Based Publisher is only required to support a
    single Subscriber on any given Topic, and so can delegate the management of multiple
    Subscribers, the delivery to multiple NotificationConsumers, and other related issues (for
    example security) to the NotificationBroker.

# 4 Publishing

286

288

289

290 291

292

293

294 295

296

297 298

299 300

301

302

303 304

305 306

307

308

309

287 There are three distinct stages in the Notification process

- · Observation of the Situation and its noteworthy characteristics;
- Creation of the Notification artifact that captures the noteworthy characteristics of the Situation: and
- Distribution of copies of the Notification to zero or more interested parties.

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

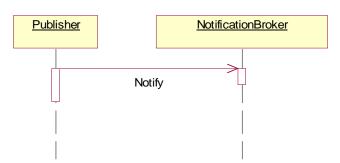
However, the WS-Notification family of specifications does specify how dissemination of messages SHOULD occur. There are two dominant patterns by which Notifications are disseminated in WS-Notification: direct and brokered.

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

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

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

The following figure illustrates simple publishing:



In the simple publishing scenario, the Publisher entity is responsible only for the core Publisher functions - observing the Situation and formatting the Notification artifact that describes the

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Deleted:

Page 12 of 43

312 Situation. The dissemination step occurs when the Publisher sends the Notify message to the 313 NotificationBroker.

314 In the broker-initiated publishing pattern, the role of the Publisher is played by a Web service that 315

implements NotificationProducer. The act of observing the Situation and formatting the

316 Notification happens within the implementation logic of the NotificationProducer itself. The 317 Notification is disseminated by the NotificationProducer sending the Notify message to a

318 NotificationBroker. The Notification may also be disseminated by sending the Notify message to

319 any NotificationConsumers that are subscribing to the NotificationProducer.

Note: in either of the above two cases, the NotificationBroker MAY require the Publisher to register with it prior to sending the Notify message. For example, if the broker wishes to control who can publish to a given Topic, it can perform an access control check during this registration.

However a NotificationBroker MAY allow Publishers to publish without pre-registration, if it so

324 chooses.

320

321

322

323

325

326 327

328

329

330

331

332

333

334

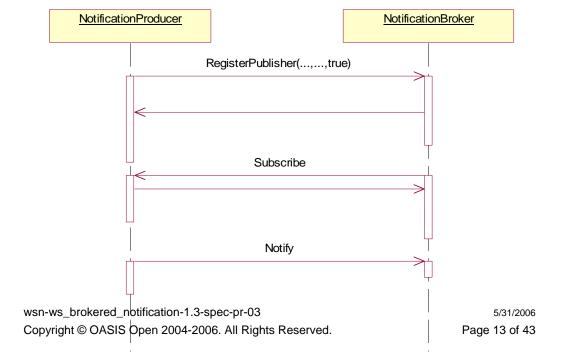
The last pattern, the demand-based pattern, requires the Publisher to be a NotificationProducer, and thereby accept the Subscribe message. Demand-based publication is intended for use in cases where the act of observing the Situation or the act of formatting the Notification artifact might be expensive to perform, and therefore should be avoided if there are no interested parties for that Notification. A Publisher indicates its intention to use this pattern by registering with the NotificationProducer and setting the Demand component of the RegisterPublisher request message to "true". Based upon this style of registration, the NotificationBroker sends the Subscribe message to the Publisher (recall: in this situation the Publisher must implement the message exchanges associated with the NotificationProducer interface).

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

Deleted:

Deleted: choose to

Deleted: . To use this pattern, the Publisher must register with the NotificationBroker, using the registration to express the intent to provide demand-based publishing only.



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

335 336

#### 5 NotificationBroker Interface 337 338 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 339 of Topics, similar to a traditional Message Oriented Middleware model. 340 A NotificationBroker MAY be a WS-Resource, and if it is, it MUST support the required message 341 342 exchanges defined by the [WS-ResourceProperties] specification and MAY support the optional message exchanges defined by WS-ResourceProperties. 343 A NotificationBroker MUST also support message exchanges and MAY support Resource 344 345 Property elements defined by the following interfaces: 346 NotificationProducer 347 NotificationConsumer 348 RegisterPublisher The NotificationBroker portType aggregates the three portTypes and is not the only way to 349 350 implement a broker. A distributed broker implementation can be achieved by hosting NotificationProducer, NotificationConsumer, or RegisterPublisher portTypes at one or more 351 352 physical endpoints. 353 The NotificationBroker is not required to provide any specific subscription durability or continuity. Deleted: does not specify 354 NotificationBrokers SHOULD advertise their durability or reliability features, either through 355 policies or other means.

NotificationBrokers MAY offer flow control and MAY implement Pull-Style notifications. If they do

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

356 357

358

#### **5.1 NotificationBroker Resource Properties**

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-

Resource Properties specification. In such cases, the Resource Properties document defined by

364 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:

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

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

```
373
374

<pre
```

This resource property element is further constrained as follows:

/wsn-br:RequiresRegistration

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

**Deleted:** The default is "false".

Deleted: If it does so

#### 5.2 Notify

359 360

361

362

367 368

369 370

371 372

375

376

377

378

379

386

387

388

389

390

393

380 The NotificationBroker MUST support the Notify message exchange from the

381 NotificationConsumer interface [WS-BaseNotification], with the following clarifications/restrictions:

A Publisher sends a Notify message to a NotificationBroker in order to publish a Notification on a given Topic. As a result of the Publisher sending this message, Notifications are delivered to all NotificationConsumers subscribed on the given Topic. The NotificationBroker may require that a Publisher be registered before the Publisher sends it a Notification (see 6.1).

5.3 Subscribe

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

The NotificationBroker MUST support the Subscribe message exchange from the NotificationProducer interface [WS-BaseNotification]. A NotificationBroker MAY suit

NotificationProducer interface [WS-BaseNotification]. A NotificationBroker MAY support any TopicExpression dialect.

Jf the processing of a Subscribe message is successful, the NotificationBroker MUST produce a
 response message, as described in WS-BaseNotification, containing an endpoint reference to a
 Subscription resource representing a Subscription created as a result of processing the

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 16 of 43

**Deleted:** For some Topics (those that require a Publisher to pre-register), the sender

must be a registered Publisher

in order to successfully publish

a Notification on the given

Topic

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

397 Subscribe request. Otherwise, the NotificationBroker must fault. WS-BaseNotification defines a 398 set of these faults.

#### 5.4 GetCurrentMessage

- 400 The NotificationBroker MUST support the GetCurrentMessage message exchange from the
- 401 NotificationProducer interface [WS-BaseNotification].
- 402 As defined in WS-BaseNotification, in response to a GetCurrentMessage message, the
- 403 NotificationBroker MAY return the last Notification published on a given Topic. This is a non-
- 404 destructive read, allowing a newly-subscribed NotificationConsumer to get the last Notification
- 405 that other NotificationConsumers have received.

#### 5.5 RegisterPublisher

- 407 The NotificationBroker MUST support the RegisterPublisher message exchange from the
- 408 RegisterPublisher interface.
- 409 A Publisher can register its interest to publish messages through the NotificationBroker by
- 410 sending a RegisterPublisherRequest. The NotificationBroker is responsible for managing the
- 411 registration, and sending a RegisterPublisherResponse to the Publisher if the registration process
- 412 succeeds. Otherwise, the NotificationBroker MUST fault. These message exchanges are further
- 413 specified in the following Section 6.

#### 5.6 CreatePullPoint

- 415 The NotificationBroker MAY support pull-style notification as defined in WS-BaseNotification and
- 416 attempt to create a PullPoint resource upon receiving a CreatePullPoint request. The
- 417 NotificationBroker does not define additional constraints to its usage of the CreatePullPoint
- 418 operation.

399

406

414

# 6 RegisterPublisher Interface

- 420 The RegisterPublisher interface contains message exchanges for publisher registration.
- 421 NotificationBroker implements the RegisterPublisher interface and is responsible for publisher
- 422 registration. A NotificationBroker may reject processing certain publisher registrations for reasons
- 423 such as lacking of authorization.

419

424 425

426

427 428

446

447 448

449 450

#### 6.1 RegisterPublisher

The RegisterPublisher message is used by the Publisher to confirm its ability to publish on a given Topic or set of Topics. If an entity wishes to register a publisher, it <u>must send a</u>
RegisterPublisher request message to the NotificationBroker. The format of the RegisterPublisher request message is:

Deleted: MUST

```
429
430
          <wsn-br:RegisterPublisher>
431
             <wsn-br:PublisherReference>
432
                wsa:EndpointReferenceType
433
             </wsn-br:PublisherReference>?
434
              <wsn-br:Topic Dialect = "xsd:anyURI">
435
                 \{any\}?
436
              </wsn-br:Topic>*
437
             <wsn-br:Demand>
438
                 xsd:boolean
439
              </wsn-br:Demand>?
440
              <wsn-br:InitialTerminationTime>
441
                xsd:dateTime
442
              </wsn-br:InitialTerminationTime>?
443
444
          </wsn-br:RegisterPublisher>
445
```

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

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

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

451 /wsn-br:RegisterPublisher/PublisherReference

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 18 of 43

452 An endpoint reference element from WS-Addressing [WS-Addressing], used to identify 453 an entity that wishes to become a Publisher. This component MUST appear if the /wsn-454 br:RegisterPublisher/Demand component has value "true". If this component is missing, the Publisher is either not a Web service, or does not wish to receive messages from the 455 NotificationBroker. 456 /wsn-br:RegisterPublisher/Topic 457 458 A set of TopicExpressions that identifies one or more Topics. If included, the given 459 Publisher is registered to publish only on the set of Topics identified by this component. If 460 this is missing the Publisher is registered to publish on any Topic supported by the NotificationBroker. 461 462 /wsn-br:RegisterPublisher/Demand 463 A Boolean element with the default value "false". If its value is "true", then the intent of the Publisher is to use a demand-based model from the NotificationBroker (see Section 4). In 464 465 this case, the NotificationBroker MUST observe the rules associated with demand-based publishing, including establishing a Subscription with the Publisher on those Topics and 466 pausing/resuming those Subscriptions as the NotificationBroker receives Subscriptions 467 468 for those Topics. 469 /wsn-br:RegisterPublisher/InitialTerminationTime 470 This component contains the service requestor's suggestion for the initial termination 471 time of the PublisherRegistration resource being created. This time is relative to the time 472 source used by the NotificationBroker. If the NotificationBroker is unable or unwilling to 473 set the TerminationTime to the given value or greater, the RegisterPublisher request MUST return an UnacceptableInitialTerminationTimeFault message. If the value is not "in 474 475 the future" relative to the current time as known by the NotificationBroker, the RegisterPublisher request MUST also return an UnacceptableInitialTerminationTimeFault 476 477 message. 478 The use of the xsi:nil attribute with value "true" indicates there is no scheduled termination time requested for the RegisterPublisher. If the element does not include the 479 480 time zone designation, the value of the element MUST be interpreted as universal time 481 482 The publisher should take care when choosing a value for InitialTerminationTime, and 483 any subsequent values that modify the TerminationTime property of the publisher 484 registration. It is RECOMMENDED that the publisher choose termination time values that 485 are significantly (several orders of magnitude) greater than the network latency expected in the interaction between the publisher and the broker. In so doing, the designer avoids 486 487 undesirable results, such as the termination time having expired prior to the receipt of the 488 published message. The [WS-ResourceLifetime] specification (Section 5.1 Regarding 489 time) contains further suggestions on how designers should reason about time values in 490 a WS-Resource Lifetime application. 491 If this component is not included, the initial value of the TerminationTime resource 492 property is dependent on the implementation of the NotificationBroker. 493 /wsn-br:RegisterPublisher/{any}

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Deleted: OPTIONAL

Deleted: must

Page 19 of 43

494 The RegisterPublisher request message allows for open content, in order to 495 accommodate elements that may be needed by extensions built on WS-496 BrokeredNotification. 497 If a /wsn-br:RegisterPublisher/Topic component is included in the message, the 498 NotificationBroker MUST register the Web service specified by the /wsnbr:RegisterPublisher/PublisherReference component as a Publisher on the set of Topics 499 500 identified by the /wsn-br:RegisterPublisher/Topic component. If for any reason the registration 501 fails, the NotificationBroker MUST fault. 502 As part of the processing of a RegisterPublisher request, the NotificationBroker creates a 503 PublisherRegistration resource representing the registration. A new resource is created 504 regardless of whether the same Publisher has previously registered with the NotificationBroker. 505 The NotificationBroker MUST return a PublisherRegistrationReference in the response to the Deleted: and may return a 506 RegisterPublisher request. ConsumerReference PublisherRegistrationReference is a WS-Addressing endpoint reference and includes the address 507 Deleted: of a PublisherRegistrationManager service. 508 509 ConsumerReference is a WS-Addressing endpoint reference to a NotificationConsumer that 510 accepts notifications from this registered Publisher. If Demand value is false in the Deleted: subscribes 511 RegisterPublisher request, the NotificationBroker MUST include a ConsumerReference in the Deleted: to response. 512 Deleted: published by 513 If the NotificationBroker accepts the RegisterPublisher request message, it must respond with a 514 message of the following form: 515 516

```
<wsn-br:RegisterPublisherResponse>
517
             <wsn-br:PublisherRegistrationReference>
518
                 <wsa:Address>
                   Address of PublisherRegistration Manager
519
520
                 </wsa:Address>
521
522
             </wsn-br:PublisherRegistrationReference>
523
             <wsn-br:ConsumerReference>
524
                <wsa:Address>
525
                    Address of a NotificationConsumer with which the
526
                    Publisher is registered
527
                 </wsa:Address>
528
529
             </wsn-br:ConsumerReference>?
530
          </wsn-br:RegisterPublisherResponse>
531
```

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

http://docs.oasis-open.org/wsn/brw-2/RegisterPublisher/RegisterPublisherResponse

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

/wsn-br:RegisterPublisherResponse/PublisherRegistrationReference

A WS-Addressing endpoint reference to the PublisherRegistration resource created by the RegisterPublisher request message. This element MUST be present in the

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

532

533

534

535

536

537

5/31/2006

Page 20 of 43

538 RegisterPublisher response message. The NotificationBroker may choose to include 539 extra information such as reference parameters in this reference. 540 /wsn-br:RegisterPublisherResponse/ConsumerReference 541 A WS-Addressing endpoint reference to a NotificationConsumer resource that accepts 542 notifications for this publisher registration. 543 Any Notification Messages sent by the Publisher (and considered to take place under this 544 registration) MUST be sent to this endpoint reference. 545 The NotificationBroker MAY use this as a mechanism for identifying the Publisher as 546 having registered. 547 If the NotificationBroker does not respond to the RegisterPublisher request message with the RegisterPublisherResponse message, then it MUST send a fault. The NotificationBroker MUST 548 549 fault if it rejects the publisher registration. This specification defines the following faults associated with failure to process the RegisterPublisher request message: 550 551 552 ResourceUnknownFault 553 The NotificationBroker is acting as a WS-Resource, and the resource identified in the message is not known to the Web service. This fault is specified by the WS-Resource 554 555 [WS-Resource] specification. 556 InvalidTopicExpressionFault 557 The TopicExpression presented in the request message is invalid. This fault is specified in WS-BaseNotification. 558 559 **TopicNotSupportedFault** 560 The TopicExpression does not match any Topic supported by the NotificationBroker. This 561 fault is specified in WS-BaseNotification. 562 PublisherRegistrationRejectedFault 563 The publisher registration is rejected by the NotificationBroker. The NotificationBroker 564 MAY provide a hint in the fault message indicating why the registration is rejected. 565 PublisherRegistrationFailedFault 566 The publisher registration process has failed. The NotificationBroker MAY include a hint in the fault message indicating why the registration is failed. 567 UnacceptableInitialTerminationTimeFault 568 569 The value of InitialTerminationTime specified in the RegisterPublisher request message 570 is not acceptable to the NotificationBroker. The NotificationBroker MAY include a hint in the fault message indicating why the value is unacceptable. 571 wsn-ws\_brokered\_notification-1.3-spec-pr-03 5/31/2006

Deleted: succeed in

Deleted: ing

# 6.1.1 Example SOAP Encoding of the RegisterPublisher Message Exchange

572

573 574

575

610

611 612

613

614

615

616

617

618 619 The following is a non-normative example of a RegisterPublisher request message using SOAP 1.1 [SOAP 1.1] or SOAP 1.2 [SOAP 1.2]:

```
576
           <s:Envelope ... >
577
             <s:Header>
578
               <wsa:Action>
579
                   http://docs.oasis-open.org/wsn/brw-
580
          2/RegisterPublisher/RegisterPublisherRequest
581
               </wsa:Action>
582
583
             </s:Header>
584
             <s:Body>
585
               <wsn-br:RegisterPublisher>
586
                 <wsn-br:PublisherReference>
587
                    <wsa:Address>
588
                      http://www.example.org/PublisherEndpoint
589
                    </wsa:Address>
590
                    <wsa:ReferenceParameters>
591
                      <npex: NPResourceDisambiguator>
592
                          uuid:84decd55-7d3f-65ad-ac44-675d9fce5d22
593
                      </npex: NPResourceDisambiguator>
594
                    </wsa:ReferenceParameters>
595
                 </wsn-br:PublisherReference>
596
                 <wsn-br:Topic Dialect="http://docs.oasis-open.org/wsn/t-</pre>
597
          1/TopicExpression/Simple">
598
                    npex:SomeTopic
599
                </wsn-br:Topic>
600
                 <wsn-br:Demand>
601
                    true
602
                </wsn-br:Demand>
603
                 <wsn-br:InitialTerminationTime>
604
                    2003-12-25T00:00:00.00000Z
605
                 </wsn-br:InitialTerminationTime>
606
               </wsn-br:RegisterPublisher>
607
             </s:Body>
608
          </s:Envelope>
609
```

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

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 22 of 43

```
620
621
622
             <s:Body>
               <wsn-br:RegisterPublisherResponse>
                 <wsn-br:PublisherRegistrationReference>
623
                   <wsa:Address>
624
625
626
                      http://www.example.org/PublisherRegistrationManager
                   </wsa:Address>
                   <wsa:ReferenceParameters>
627
                     <npex:NPubResourceId>
628
629
                        uuid:95fefeb3-f37d-5dfe-44fe-221d9fceec99
                     </npex:NPubResourceId>
630
                   </wsa:ReferenceParameters>
631
                 </wsn-br:PublisherRegistrationReference>
632
633
                 <wsn-br:ConsumerReference>
                   <wsa:Address>
                     http://www.example.org/NotificationConsumer
634
635
                   </wsa:Address>
636
637
                 </wsn-br:ConsumerReference>
638
               </wsn-br:RegisterPublisherResponse>
639
             </s:Body>
640
           </s:Envelope>
```

# 7 PublisherRegistrationManager Interface

The PublisherRegistrationManager interface defines message exchanges to manipulate PublisherRegistration resources. The PublisherRegistrationManager MAY <a href="expose-publisherRegistrations-as-ws-Resources">expose-publisherRegistrations-as-ws-Resources</a>, and if <a href="mailto:it.does-then-the-publisherRegistrationManager">it.does-then-the-publisherRegistrationManager</a>—<a href="mailto:ws-Resources-then-the-publisherRegistrationManager">ws-Resources-then-the-publisherRegistrationManager</a>—<a href="mailto:ws-Resources-then-the-publisher-ws-mailto:ws-Resources-then-the-publisher-ws-mailto:ws-

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

#### 7.1 PublisherRegistration Resource Properties

In addition to the message exchanges described in this specification, a PublisherRegistrationManager 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. In such cases, the Resource Properties document defined by the PublisherRegistrationManager MUST also include references to the following resource property elements:

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

These resource property elements are further constrained as follows:

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

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

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 24 of 43

Deleted: be a

Deleted: it is

Deleted: ,

Deleted: WS-Resource

Deleted: RF

Deleted:

Deleted: RF

Deleted: RF

Deleted: Colleted: RF

Deleted: D

Deleted: If it does so

680 /wsn-br:CreationTime 681 Indicates the date and time at which the PublisherRegistration was created. This Deleted: This is an optional 682 component MAY be omitted, for example by resource-constrained devices that cannot component, supporting resource constrained devices associate a creation time with PublisherRegistration resources they create. 683 which cannot associate a 684 If PublisherRegistration is exposed as a WS-Resource, the PublisherRegistrationManager MAY creation time with PublisherRegistration 685 permit the following resource properties to be modified by a requestor, by sending a resources SetResourceProperties request message as defined in the WS-ResourceProperties specification: 686 Comment: WSN 3.31 687 /wsn-br:Topic and /wsn-br:Demand resolution 688 Deleted: Manager 689 Note: /wsn-br:Demand may not take the value "true" if there is no /wsn-Deleted: MAY 690 br:PublisherReference resource property element in the resource property document. Deleted: the 7.2 DestroyRegistration 691 Deleted: Expression 692 The PublisherRegistrationManager interface provides a destroy operation, providing a means by Deleted: ¶ which a requestor can terminate the <u>PublisherRegistration</u> resource. The DestroyRegistration 693 Comment: WSN 2.65 694 request message has the following form: resolution 695 Deleted: publisher registration manager 696 <wsn-br:DestroyRegistration>  $\{any\} *$ 697 Deleted: To terminate 698 </wsn-br:DestroyRegistration> PublisherRegistrationManager 699 resource, a requestor MUST send a DestrovRegistration 700 The WS-Addressing [action] Message Addressing Property MUST contain the URI request message to the PublisherRegistrationManager. 701 http://docs.oasis-open.org/wsn/brw-2/PublisherRegistrationManager/DestroyRegistrationReguest Deleted: 702 The DestroyRegistration request message allows for open content and contains an extension 703 component 704 /wsn-br:DestroyRegistration/{any}. 705 706 Upon receipt of the DestroyRegistration request, the PublisherRegistrationManager MUST attempt to destroy the PublisherRegistration resource. If the DestroyRegistration request 707 Deleted: itself 708 message is successfully processed, the PublisherRegistrationManager MUST respond with the 709 following message: 710 711 <wsn-br:DestroyRegistrationResponse> Deleted: / 712 {any} \* 713 </wsn-br:DestroyRegistrationResponse> 714 715 The WS-Addressing [action] Message Addressing Property MUST contain the URI 716 http://docs.oasis-open.org/wsn/brw-Deleted: ¶ 717 2/PublisherRegistrationManager/DestroyRegistrationResponse\_ 718 If the PublisherRegistrationManager does not respond to the DestroyRegistration request message with the DestroyRegistrationResponse message, then it MUST send a fault. This 719 wsn-ws\_brokered\_notification-1.3-spec-pr-03 5/31/2006

Page 25 of 43

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

- 720 specification defines the following faults associated with failure to process the
   721 DestroyRegistration request message:
- 722 ResourceUnknownFault
- The PublisherRegistration is a WS-Resource, and the resource identified in the message is not known to the Web service. This fault is specified by the WS-Resource [WS-Resource] specification.
- 726 ResourceNotDestroyedFault

729

730 731

732

733

734

735

736

737

738

739 740

741

742

743

744

745

746

The PublisherRegistrationManager was unable to destroy the PublisherRegistration Deleted: Manager resource for some reason.

# 7.2.1 Example SOAP Encoding of the DestroyRegistration Message Exchange

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

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

```
747
          <s:Envelope ... >
748
            <s:Header>
749
               <wsa:Action>
750
                http://docs.oasis-open.org/wsn/brw-
751
          2/PublisherRegistrationManager/DestroyRegistrationResponse
752
              </wsa:Action>
753
754
            </s:Header>
755
            <s:Body>
756
              <wsn-br:DestroyRegistrationResponse/>
757
            </s:Body>
758
          </s:Envelope>
```

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 26 of 43

# **8 Security Considerations**

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

#### 8.1 Securing PublisherRegistration

762 763 764

759 760

761

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

765 766 767

768

769

770

771

772

773

774 775

776

777

778

781

782

783

784

785

786

- 1. only authorized Publishers can register with a NotificationBroker
- only messages of authorized Publishers and of registered topics, can be accepted by a NotificationBroker

Deleted: the

3. only authorized principals can modify or delete PublisherRegistration resource

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

account.

A NotificationBroker can support the security measurements of NotificationProducers and

NotificationConsumers mentioned in WS-BaseNotification. Acting as an intermediary, A

NotificationBroker MAY also provide convenience to security management, including but not

779 Notificatio 780 limited to:

Controlling who can publish on a topic at publisher registration time

- Refusing to relay messages from unauthorized publishers
- · Imposing security measurements on all messaging routing through the broker
- Providing convenience in messaging security management based on topics.

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

Deleted: ¶

787

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 27 of 43

# 9 References

## 9.1 Normative

788

789

790	[RFC2119]		
791 792	S. Bradner, <u>"Key words for use in RFCs to Indicate Requirement Levels", IETF RFC 2119</u> , March 1997_http://www.ietf.org/rfc/rfc2119.txt		Deleted: ,
793	[XML],	~~;	Deleted:
794	"Extensible Markup Language (XML) 1.0", W3C Recommendation.		Deleted: , IETF RFC 2119,
795	http://www.w3.org/TR/REC-xml		March 1997.
796	[XML-Infoset]	\	Deleted:
797	"XML Information Set", W3C Recommendation. http://www.w3.org/TR/xml-infoset/		
798	[WS-Addressing]		
799	"Web Services Addressing 1.0 - Core", W3C Recommendation.		Deleted 6
800	http://www.w3.org/TR/ws-addr-core		Deleted: ¶
801	[WS-BaseNotification],		Deleted:
802 803	"Web Services Base Notification 1.3". http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-03.pdf		Deleted: d
804	· · · · · · · · · · · · · · · · · · ·	<b>~</b> <[[	
805	[WS-Topics], "Web Services Topics 1.3".		Deleted: 01
806	http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-pr-02.pdf		Deleted:
807	[WS-Resource]	7	Deleted: WS-
808	"Web Services Resource 1.2", OASIS Standard.	11/1	Deleted: T
809	http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-os.pdf	','	Deleted: cd
810	[WS-ResourceLifetime]		Deleted: 1
811	"Web Services Resource Lifetime 1.2", OASIS Standard.	1	
812	http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-os.pdf	<b>-</b> -	Deleted: pr-02
813	[WS-ResourceProperties]		Deleted:
814	"Web Services Resource Properties 1.2", OASIS Standard.		Deleted: pr-02
815	http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-os.pdf		Deleted:
816	[WS-BaseFaults]		Deleted: pr-02
817	"Web Services Base Faults 1.2", OASIS Standard.		
818	http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf	-<[	Deleted:
819			Deleted: pr-02
820	9.2 Non-Normative		

#### 9.2 Non-Normative

821 [SOAP 1.1] "Simple Object Access Protocol (SOAP) 1.1" http://www.w3.org/TR/2000/NOTE-SOAP-20000508/ 822 823

wsn-ws\_brokered\_notification-1.3-spec-pr-03 Copyright © OASIS Open 2004-2006. All Rights Reserved. 5/31/2006

Page 28 of 43

824	[SOAP 1.2]	
825	"SOAP Version 1.2 Part 1:Messaging Framework", W3C Recommendation.	Deleted:
826	http://www.w3.org/TR/soap12-part1/	
827	[WS-Security]	
828	"Web Services Security: SOAP Message Security 1.0", OASIS Standard.	
829	\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-	Deleted:
830	1.0.pdf,	Deleted: ¶

# **Appendix A. Acknowledgments**

- The following individuals were members of the committee during the development of this specification:
- 834 Sid Askary, Individual, Fred Carter, AmberPoint, Martin Chapman, Oracle, Dave Chappell, Sonic
- 835 Software, Rick Cobb, KnowNow, Ugo Corda, SeeBeyond Technology Corporation, John Fuller,
- 836 Individual, Stephen Graham, IBM, David Hull, Tibco, Hideharu Kato, Hitachi, Lily Liu,
- 837 webMethods, Tom Maguire, IBM, Susan Malaika, IBM, Samuel Meder, Argonne National
- 838 Laboratory, Bryan Murray, Hewlett-Packard, Peter Niblett, IBM, Sanjay Patil, SAP, Mark Peel,
- 839 Novell, Matt Roberts, IBM, Igor Sedukhin, Computer Associates, David Snelling, Fujitsu, Latha
- 840 Srinivasan, Hewlett-Packard, William Vambenepe, Hewlett-Packard, Kirk Wilson, Computer
- 841 Associates.

831

- 842 Special thanks to the Global Grid Forum's Open Grid Services Infrastructure working group,
- 843 which defined the OGSI v1.0 specification which was a large inspiration for the ideas expressed
- 844 in this specification.
- 845 In addition, the following people who are not members of the committee made contributions to
- 846 this specification:
- Tim Banks (IBM), Nick Butler (IBM), Doug Davis (IBM), John Dinger (IBM), Don Ferguson (IBM),
- 848 Jeff Frey (IBM), Andreas Koeppel (SAP), Heather Kreger (IBM), Amy Lewis (TIBCO Software),
- 849 Kevin Liu (SAP), Nataraj Nagaratnam (IBM), Martin Nally (IBM), Jeff Nick (IBM), Jay Parikh
- 850 (Akamai Technologies), Claus von Riegen (SAP), Rick Rineholt (IBM), John Rofrano (IBM),
- 851 Shivajee Samdarshi (TIBCO Software), Igor Sedukhin (Computer Associates), Eugène
- 852 Sindambiwe (SAP), Jay Unger (IBM), Bill Weihl (Akamai Technologies), Mark Weitzel (IBM), Dan
- 853 Wolfson (IBM).

# Appendix B. XML Schema

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

<?xml version="1.0" encoding="UTF-8"?>
<!-<!-<!-<!--</pre>

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 (2004-2006). All Rights Reserved.

Deleted: 5

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.

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Page 31 of 43

```
902
903
          <xsd:schema
904
            xmlns="http://www.w3.org/2001/XMLSchema"
905
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
906
            xmlns:wsa="http://www.w3.org/2005/08/addressing"
907
            xmlns:wsn-br="http://docs.oasis-open.org/wsn/br-2"
908
            xmlns:wsn-b="http://docs.oasis-open.org/wsn/b-2"
909
            xmlns:wsrf-bf="http://docs.oasis-open.org/wsrf/bf-2"
910
            xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"
911
            targetNamespace="http://docs.oasis-open.org/wsn/br-2"
912
            elementFormDefault="qualified"
913
            attributeFormDefault="unqualified">
914
915
          916
917
             <xsd:import namespace="http://www.w3.org/2005/08/addressing"</pre>
918
                        schemaLocation="http://www.w3.org/2005/08/addressing/ws-
919
          addr.xsd"/>
920
921
             <xsd:import namespace="http://docs.oasis-open.org/wsrf/bf-2"</pre>
922
                        schemaLocation="http://docs.oasis-open.org/wsrf/bf-
923
          2.xsd"/>
924
925
             <xsd:import namespace="http://docs.oasis-open.org/wsn/b-2"</pre>
926
                        schemaLocation="http://docs.oasis-open.org/wsn/b-2.xsd"/>
927
928
             <xsd:import namespace="http://docs.oasis-open.org/wsn/t-1"</pre>
929
                        schemaLocation="http://docs.oasis-open.org/wsn/t-1.xsd"/>
930
931
          <!-- ===== Resource Properties for NotificationBroker ======= -->
932
             <xsd:element name="RequiresRegistration" type="xsd:boolean"/>
933
934
          <!-- ===== Resource Properties for PublisherRegistration ======= -->
935
             <xsd:element name="PublisherReference"</pre>
936
                                 type="wsa:EndpointReferenceType"/>
937
             <xsd:element name="ConsumerReference"</pre>
938
                                type="wsa:EndpointReferenceType"/>
939
             <xsd:element name="Topic"</pre>
940
                                 type="wsn-b:TopicExpressionType"/>
941
             <xsd:element name="Demand"</pre>
942
                                type="xsd:boolean"/>
943
             <xsd:element name="CreationTime"</pre>
944
                                type="xsd:dateTime"/>
945
             <xsd:element name="NotificationBrokerRP">
946
                      <xsd:complexType>
947
                          <xsd:sequence>
948
                   <!-- From NotificationProducer -->
949
                             <xsd:element ref="wsn-b:TopicExpression"</pre>
950
                                minOccurs="0" maxOccurs="unbounded" />
951
                             <xsd:element ref="wsn-b:FixedTopicSet"</pre>
952
                                minOccurs="0" maxOccurs="1" />
953
                             <xsd:element ref="wsn-b:TopicExpressionDialect"</pre>
```

```
954
                                   minOccurs="0" maxOccurs="unbounded" />
 955
                               <xsd:element ref="wstop:TopicSet"</pre>
 956
                                  minOccurs="0" maxOccurs="1" />
 957
                      <!-- NotificationBroker specific -->
 958
                               <xsd:element ref="wsn-br:RequiresRegistration"</pre>
 959
                                             minOccurs="1" maxOccurs="1" />
 960
                            </xsd:sequence>
 961
                         </xsd:complexType>
 962
               </xsd:element>
 963
 964
            <!-- ===== Resource Properties for PublisherRegistration ======= -->
 965
               <xsd:element name="PublisherRegistrationRP">
 966
                         <xsd:complexType>
 967
                            <xsd:sequence>
 968
                               <xsd:element ref="wsn-br:PublisherReference"</pre>
                                                                                                     Deleted:
 969
                                              minOccurs="0" maxOccurs="1" />
                                                                                                     From WS-
 970
                               <xsd:element ref="wsn-br:Topic"</pre>
                                                                                                     ResourceLifetime
 971
                                             minOccurs="0" maxOccurs="unbounded" />
                                                                                                     ScheduledResourceTermin
 972
                                                                                                     ation -->¶
                               <xsd:element ref="wsn-br:Demand"</pre>
 973
                                             minOccurs="1" maxOccurs="1" />
                                                                                                     <xsd:element ref="wsn-</pre>
 974
                               <xsd:element ref="wsn-br:CreationTime"</pre>
                                                                                                     b:CurrentTime"¶
 975
                                             minOccurs="0" maxOccurs="1" />
 976
                            </xsd:sequence>
                                                                                                     minOccurs="0"
 977
                         </xsd:complexType>
                                                                                                     maxOccurs="1" /> ¶
 978
               </xsd:element>
 979
                                                                                                     <xsd:element ref="wsn-</pre>
 980
                                                                                                     b:TerminationTime" ¶
            <!-- ====== Message Types for NotificationBroker ======== -->
 981
               <xsd:element name="RegisterPublisher">
                                                                                                     minOccurs="1"
 982
                         <xsd:complexType>
                                                                                                     maxOccurs="1" />¶
 983
                            <xsd:sequence>
 984
                               <xsd:element name="PublisherReference"</pre>
 985
                                             type="wsa:EndpointReferenceType"
                                                                                                     PublisherRegistration
                                             minOccurs="0" maxOccurs="1" />
 986
                                                                                                     specific -->¶
 987
                               <xsd:element name="Topic"</pre>
 988
                                             type="wsn-b:TopicExpressionType"
 989
                                             minOccurs="0" maxOccurs="unbounded" />
 990
                                <xsd:element name="Demand"</pre>
 991
                                             type="xsd:boolean" default="false"
 992
                                             minOccurs="0" maxOccurs="1" />
 993
                               <xsd:element name="InitialTerminationTime"</pre>
 994
                                              type="xsd:dateTime"
 995
                                             minOccurs="0" maxOccurs="1" />
 996
                               <xsd:any namespace="##other" processContents="lax"</pre>
 997
                                     minOccurs="0" maxOccurs="unbounded"/>
 998
                            </xsd:sequence>
 999
                         </xsd:complexType>
1000
               </xsd:element>
1001
1002
               <xsd:element name="RegisterPublisherResponse">
1003
                         <xsd:complexType>
1004
```

<xsd:element name="PublisherRegistrationReference"</pre>

<!--

<!--

wsn-ws\_brokered\_notification-1.3-spec-pr-03 Copyright © OASIS Open 2004-2006. All Rights Reserved.

1005

<xsd:sequence>

Page 33 of 43

```
1006
                                            type="wsa:EndpointReferenceType"
1007
                                            minOccurs="1" maxOccurs="1" />
1008
                               <xsd:element name="ConsumerReference"</pre>
1009
                                            type="wsa:EndpointReferenceType"
1010
                                            minOccurs="0" maxOccurs="1" />
1011
1012
                            </xsd:sequence>
1013
                        </xsd:complexType>
1014
               </xsd:element>
1015
1016
               <xsd:complexType name="PublisherRegistrationRejectedFaultType">
1017
                        <xsd:complexContent>
1018
                            <xsd:extension base="wsrf-bf:BaseFaultType"/>
1019
                        </xsd:complexContent>
1020
                     </xsd:complexType>
1021
               <xsd:element name="PublisherRegistrationRejectedFault"</pre>
1022
                             type="wsn-br:PublisherRegistrationRejectedFaultType"/>
1023
1024
               <xsd:complexType name="PublisherRegistrationFailedFaultType">
1025
                        <xsd:complexContent>
1026
                            <xsd:extension base="wsrf-bf:BaseFaultType"/>
1027
                        </xsd:complexContent>
1028
                     </xsd:complexType>
1029
               <xsd:element name="PublisherRegistrationFailedFault"</pre>
1030
                             type="wsn-br:PublisherRegistrationFailedFaultType"/>
1031
1032
1033
1034
               <xsd:element name="DestroyRegistration">
1035
                <xsd:complexType>
1036
                  <xsd:sequence>
1037
                    <xsd:any namespace="##other" processContents="lax"</pre>
1038
                             minOccurs="0" maxOccurs="unbounded"/>
1039
                  </xsd:sequence>
1040
                  <xsd:anyAttribute/>
1041
                </xsd:complexType>
1042
              </xsd:element>
1043
1044
              <xsd:element name="DestroyRegistrationResponse">
1045
                <xsd:complexType>
1046
                  <xsd:sequence>
1047
                    <xsd:any namespace="##other" processContents="lax"</pre>
1048
                             minOccurs="0" maxOccurs="unbounded"/>
1049
                  </xsd:sequence>
1050
                  <xsd:anyAttribute/>
1051
                </xsd:complexType>
1052
              </xsd:element>
1053
1054
              <xsd:complexType name="ResourceNotDestroyedFaultType">
1055
                <xsd:complexContent>
1056
                  <xsd:extension base="wsrf-bf:BaseFaultType"/>
1057
                </xsd:complexContent>
```

| 1058<br>1059<br>1060<br>1061 | <pre> <xsd:element <="" name="ResourceNotDestroyedFault" th=""></xsd:element></pre> |
|------------------------------|---|
| 1062                         |   |

# **Appendix C. WSDL 1.1**

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

<?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.</pre>

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 (2004-2006). 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.

wsn-ws\_brokered\_notification-1.3-spec-pr-03
Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Deleted: 5

Page 36 of 43

```
1110
           -->
1111
1112
           <wsdl:definitions name="WS-BrokeredNotification"</pre>
1113
             xmlns="http://schemas.xmlsoap.org/wsdl/"
1114
             xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
1115
             xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1116
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1117
             xmlns:wsa="http://www.w3.org/2005/08/addressing"
1118
             xmlns:wsn-br="http://docs.oasis-open.org/wsn/br-2"
1119
             xmlns:wsn-brw="http://docs.oasis-open.org/wsn/brw-2"
1120
             xmlns:wsn-b="http://docs.oasis-open.org/wsn/b-2"
1121
             xmlns:wsn-bw="http://docs.oasis-open.org/wsn/bw-2"
1122
             xmlns:wsrf-bf="http://docs.oasis-open.org/wsrf/bf-2"
1123
             xmlns:wsrf-rw="http://docs.oasis-open.org/wsrf/rw-2"
1124
             targetNamespace="http://docs.oasis-open.org/wsn/brw-2">
1125
1126
           1127
              <wsdl:import namespace="http://docs.oasis-open.org/wsrf/rw-2"</pre>
1128
                  location="http://docs.oasis-open.org/wsrf/rw-2.wsdl"/>
1129
1130
              <wsdl:import namespace="http://docs.oasis-open.org/wsn/bw-2"</pre>
1131
                  location="http://docs.oasis-open.org/wsn/bw-2.wsdl"/>
1132
1133
           <!-- ========== Types Definitions ============== -->
1134
              <wsdl:types>
1135
                <xsd:schema>
1136
                  <xsd:import</pre>
1137
                    namespace="http://docs.oasis-open.org/wsn/br-2"
1138
                    schemaLocation="http://docs.oasis-open.org/wsn/br-2.xsd"/>
1139
                </xsd:schema>
1140
              </wsdl:types>
1141
1142
           <!-- ====== NotificationBroker::RegisterPublisher =========
1143
              RegisterPublisher(PublisherReference, TopicExpression* ,
1144
                                [Demand], [InitialTerminationTime])
1145
              returns: WS-Resource qualified EPR to a PublisherRegistration -->
1146
              <wsdl:message name="RegisterPublisherRequest">
1147
                 <wsdl:part name="RegisterPublisherRequest"</pre>
1148
                            element="wsn-br:RegisterPublisher"/>
1149
              </wsdl:message>
1150
1151
              <wsdl:message name="RegisterPublisherResponse">
1152
                 <wsdl:part name="RegisterPublisherResponse"</pre>
1153
                            element="wsn-br:RegisterPublisherResponse"/>
1154
              </wsdl:message>
1155
1156
              <wsdl:message name="PublisherRegistrationRejectedFault">
1157
                 <wsdl:part name="PublisherRegistrationRejectedFault"</pre>
1158
                       element="wsn-br:PublisherRegistrationRejectedFault"/>
1159
              </wsdl:message>
1160
1161
              <wsdl:message name="PublisherRegistrationFailedFault">
```

```
1162
                  <wsdl:part name="PublisherRegistrationFailedFault"</pre>
1163
                        element="wsn-br:PublisherRegistrationFailedFault"/>
1164
              </wsdl:message>
1165
1166
              <wsdl:message name="DestroyRegistrationRequest">
1167
                  <wsdl:part name="DestroyRegistrationRequest"</pre>
1168
                       element="wsn-br:DestroyRegistration"/>
1169
               </wsdl:message>
1170
1171
              <wsdl:message name="DestroyRegistrationResponse">
1172
                  <wsdl:part name="DestroyRegistrationResponse"</pre>
1173
                        element="wsn-br:DestroyRegistrationResponse"/>
1174
              </wsdl:message>
1175
1176
              <wsdl:message name="ResourceNotDestroyedFault">
1177
                  <wsdl:part name="ResourceNotDestroyedFault"</pre>
1178
                       element="wsn-br:ResourceNotDestroyedFault"/>
1179
              </wsdl:message>
1180
1181
           <!-- ============ PortType Definitions ====================== -->
1182
1183
           <!-- ====== RegisterPublisher ======== -->
1184
            <wsdl:portType name="RegisterPublisher">
1185
               <wsdl:operation name="RegisterPublisher">
1186
                     <wsdl:input message="wsn-brw:RegisterPublisherRequest"/>
1187
                     <wsdl:output message="wsn-brw:RegisterPublisherResponse"/>
1188
                     <wsdl:fault name="ResourceUnknownFault"</pre>
1189
                                  message="wsrf-rw:ResourceUnknownFault"/>
1190
                     <wsdl:fault name="InvalidTopicExpressionFault"</pre>
1191
                                  message="wsn-bw:InvalidTopicExpressionFault"/>
1192
                     <wsdl:fault name="TopicNotSupportedFault"</pre>
1193
                                  message="wsn-bw:TopicNotSupportedFault"/>
1194
                     <wsdl:fault name="PublisherRegistrationRejectedFault"</pre>
1195
                             message="wsn-brw:PublisherRegistrationRejectedFault"/>
1196
                     <wsdl:fault name="PublisherRegistrationFailedFault"</pre>
1197
                             message="wsn-brw:PublisherRegistrationFailedFault"/>
1198
                     <wsdl:fault name="UnacceptableInitialTerminationTimeFault"</pre>
1199
                        message="wsn-bw:UnacceptableInitialTerminationTimeFault"/>
1200
                </wsdl:operation>
1201
             </wsdl:portType>
1202
1203
           <!-- ====== NotificationBroker PortType Definition ======== -->
1204
              <wsdl:portType name="NotificationBroker">
1205
                      ======= extends NotificationConsumer ======== -->
1206
                  <wsdl:operation name="Notify">
1207
                     <wsdl:input message="wsn-bw:Notify" />
                 </wsdl:operation>
1208
1209
1210
                  <!-- ====== extends NotificationProducer ======== -->
1211
                  <wsdl:operation name="Subscribe">
1212
                     <wsdl:input message="wsn-bw:SubscribeRequest" />
1213
                     <wsdl:output message="wsn-bw:SubscribeResponse" />
```

```
1214
                     <wsdl:fault name="ResourceUnknownFault"</pre>
1215
                                   message="wsrf-rw:ResourceUnknownFault" />
1216
                                  name="InvalidFilterFault"
                     <wsdl:fault
1217
                                message="wsn-bw:InvalidFilterFault"/>
1218
                     <wsdl:fault name="TopicExpressionDialectUnknownFault"</pre>
1219
                                message="wsn-bw:TopicExpressionDialectUnknownFault"/>
1220
                     <wsdl:fault name="InvalidTopicExpressionFault"</pre>
1221
                                   message="wsn-bw:InvalidTopicExpressionFault" />
1222
                     <wsdl:fault name="TopicNotSupportedFault"</pre>
1223
                                   message="wsn-bw:TopicNotSupportedFault" />
1224
                     <wsdl:fault name="InvalidProducerPropertiesExpressionFault"</pre>
1225
                         message="wsn-bw:InvalidProducerPropertiesExpressionFault"/>
                     <wsdl:fault name="InvalidMessageContentExpressionFault"</pre>
1226
1227
                         message="wsn-bw:InvalidMessageContentExpressionFault"/>
1228
                     <wsdl:fault name="UnacceptableInitialTerminationTimeFault"</pre>
1229
                         message="wsn-bw:UnacceptableInitialTerminationTimeFault"/>
1230
                     <wsdl:fault name="UnrecognizedPolicyRequestFault"</pre>
1231
                         message="wsn-bw:UnrecognizedPolicyRequestFault"/>
1232
                     <wsdl:fault name="UnsupportedPolicyRequestFault"</pre>
                         message="wsn-bw:UnsupportedPolicyRequestFault"/>
1233
                     <wsdl:fault name="NotifyMessageNotSupportedFault"</pre>
1234
1235
                         message="wsn-bw:NotifyMessageNotSupportedFault"/>
1236
                     <wsdl:fault name="SubscribeCreationFailedFault"</pre>
                                   message="wsn-bw:SubscribeCreationFailedFault"/>
1237
1238
                  </wsdl:operation>
1239
                  <wsdl:operation name="GetCurrentMessage">
1240
                     <wsdl:input message="wsn-bw:GetCurrentMessageRequest"/>
1241
                     <wsdl:output message="wsn-bw:GetCurrentMessageResponse"/>
1242
                     <wsdl:fault name="ResourceUnknownFault"</pre>
1243
                                   message="wsrf-rw:ResourceUnknownFault"/>
1244
                     <wsdl:fault name="TopicExpressionDialectUnknownFault"</pre>
1245
                               message="wsn-bw:TopicExpressionDialectUnknownFault"/>
1246
                     <wsdl:fault name="InvalidTopicExpressionFault"</pre>
1247
                                   message="wsn-bw:InvalidTopicExpressionFault"/>
1248
                     <wsdl:fault name="TopicNotSupportedFault"</pre>
1249
                                   message="wsn-bw:TopicNotSupportedFault"/>
1250
                     <wsdl:fault
                                  name="NoCurrentMessageOnTopicFault"
1251
                                   message="wsn-bw:NoCurrentMessageOnTopicFault"/>
1252
                     <wsdl:fault
                                  name="MultipleTopicsSpecifiedFault"
1253
                                   message="wsn-bw:MultipleTopicsSpecifiedFault"/>
1254
                  </wsdl:operation>
1255
1256
                  <!-- ====== extends RegisterPublisher ====== -->
1257
                  <wsdl:operation name="RegisterPublisher">
1258
                     <wsdl:input message="wsn-brw:RegisterPublisherRequest"/>
1259
                     <wsdl:output message="wsn-brw:RegisterPublisherResponse"/>
1260
                     <wsdl:fault name="ResourceUnknownFault"</pre>
1261
                                   message="wsrf-rw:ResourceUnknownFault"/>
1262
                     <wsdl:fault name="InvalidTopicExpressionFault"</pre>
1263
                                   message="wsn-bw:InvalidTopicExpressionFault"/>
1264
                     <wsdl:fault
                                  name="TopicNotSupportedFault"
1265
                                   message="wsn-bw:TopicNotSupportedFault"/>
```

Page 39 of 43

```
1266
                     <wsdl:fault name="PublisherRegistrationRejectedFault"</pre>
1267
                             message="wsn-brw:PublisherRegistrationRejectedFault"/>
1268
                     <wsdl:fault name="PublisherRegistrationFailedFault"</pre>
1269
                               message="wsn-brw:PublisherRegistrationFailedFault"/>
1270
                     <wsdl:fault name="UnacceptableInitialTerminationTimeFault"</pre>
1271
                         message="wsn-bw:UnacceptableInitialTerminationTimeFault"/>
1272
                  </wsdl:operation>
1273
1274
               </wsdl:portType>
1275
1276
            <!-- ==== PublisherRegistrationManager PortType Definition ====== -->
1277
               <wsdl:portType name="PublisherRegistrationManager">
1278
1279
               <!--==DestroyRegistration:ImmediateResourceTermination=======-->
1280
                  <wsdl:operation name="DestroyRegistration">
1281
                     <wsdl:input name="DestroyRegistrationRequest"</pre>
1282
                                   message="wsn-brw:DestroyRegistrationRequest" />
1283
                     <wsdl:output name="DestroyRegistrationResponse"</pre>
1284
                                   message="wsn-brw:DestroyRegistrationResponse" />
1285
                     <wsdl:fault name="ResourceUnknownFault"</pre>
1286
                               message="wsrf-rw:ResourceUnknownFault" />
1287
                     <wsdl:fault name="ResourceNotDestroyedFault"</pre>
1288
                                   message="wsn-brw:ResourceNotDestroyedFault" />
1289
                  </wsdl:operation>
1290
               </wsdl:portType>
1291
            </wsdl:definitions>
```

| Rev                     | Date                         | By Whom                       | What  |  |
|-------------------------|------------------------------|-------------------------------|---|--|
| 1.2 01                  | 2004-<br>05-12               | Lily Liu                      | Initial version   |  |
| 1.2 02                  | 2004-<br>06-07               | Dave<br>Chappell              | Updates and consistency check w/ other WS-N specs   |  |
| 1.2 03                  | 2004-<br>06-24               | Lily Liu,<br>Dave<br>Chappell | Addition of a Goals and Requirements section and minor format changes   |  |
| 1.2 03                  | 2004-<br>07-12               | Lily Liu                      | Addition of a status paragraph  |  |
| 1.3 01a<br>-<br>1.3 01e | 2005-<br>02-01               | Dave<br>Chappell,<br>Lily Liu | Series of issue resolution and consistency reviews with WS-BaseNotification   |  |
| 1.3 01f                 | 2005-<br>06-10               | Lily Liu                      | 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   |  |
|                         |                              |                               | Updated the Terminology, Introduction, and Security sections.   |  |
|                         |                              |                               | Updated sections about NotificationBroker and PublisherRegistrationManager resource properties.   |  |
| 1.3 01g                 | 2005-<br>07-01               | Lily Liu                      | Updated the status section. Changed term NotificationMessage to Notification. Added CreatePullPoint portType to NotificationBroker. Completed issue resolutions. Replaced the Abstract section. |  |
| 1.3 02d                 | 2005-<br>11-04               | Lily Liu                      | 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 Resolved AI 137, AI 138, AI 141, AI 142 AI 144, and AI 145. Updated references.           |  |
| 1.3 pr02                | <u>2006-</u><br><u>02-17</u> | Lily Liu                      | Updated the document with changes suggested in the errata document (http://www.oasis-open.org/apps/org/workgroup/wsn/download.php/16679/wsn-  |  |

5/31/2006

Page 41 of 43

| Rev         | Date           | By Whom          | What   |
|-------------|----------------|------------------|--|
|             |                |                  | ws_brokered_notification-1.3-errata.doc).  |
| 1.3<br>cd03 | 2006-<br>03-23 | Lily Liu         | Updated the document again with more changes suggested in the errata document (http://www.oasis-open.org/apps/org/workgroup/wsn/download.php/16679/wsn-ws_brokered_notification-1.3-errata.doc).  Updated copyrights and status of the document. |
| 1.3<br>cd03 | 2006-<br>04-20 | Peter<br>Niblett | Updated the References section to point at OASIS standard versions of WSRF specifications, and moved the WS-BaseNotification and WS-Topics references on to point to the latest committee drafts   |

Deleted:

# **Appendix E. Notices**

1294 OASIS takes no position regarding the validity or scope of an intellectual property or other rights 1295 that might be claimed to pertain to the implementation or use of the technology described in this 1296 document or the extent to which any license under such rights might or might not be available; 1297 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 1298 1299 website. Copies of claims of rights made available for publication and any assurances of licenses 1300 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 implementers or users of this specification, can be 1301

1302 obtained from the OASIS Executive Director.

1303 OASIS invites any interested party to bring to its attention any copyrights, patents or patent 1304 applications, or other proprietary rights which may cover technology that may be required to 1305 implement this specification. Please address the information to the OASIS Executive Director.

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

1307 This document and translations of it may be copied and furnished to others, and derivative works

1308 that comment on or otherwise explain it or assist in its implementation may be prepared, copied. 1309 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. 1310

1311 However, this document itself does 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 1312

1313 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual

1314 Property Rights document must be followed, or as required to translate it into languages other

1315 than English.

1293

1306

1316 The limited permissions granted above are perpetual and will not be revoked by OASIS or its

1317 successors or assigns.

1318 This document and the information contained herein is provided on an "AS IS" basis and OASIS

1319 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO

ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY 1320

1321 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A

1322 PARTICULAR PURPOSE.

> wsn-ws\_brokered\_notification-1.3-spec-pr-03 Copyright © OASIS Open 2004-2006. All Rights Reserved.

5/31/2006

Deleted: 4

Page 43 of 43