



# Web Services Brokered Notification 1.2 (WS-BrokeredNotification)

Working Draft 01, 21 July 2004

## Document identifier:

wsn-WS-BrokeredNotification-1.2-draft-01

## Location:

<http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BrokeredNotification-1.2-draft-01.pdf>

## Editors:

Dave Chappell, Sonic Software <[chappell@sonicsoftware.com](mailto:chappell@sonicsoftware.com)>

Lily Liu, webMethods <[lily.liu@webmethods.com](mailto: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 white papers and 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: a white paper: *Publish-Subscribe Notification for Web services* as well as three normative specifications: WS-BaseNotification, WS-BrokeredNotification, and WS-Topics.

This specification defines the Web services interface for the NotificationBroker. A NotificationBroker is an intermediary, which, 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-Base Notification* and *WS-Topics*, as well as the *Publish-Subscribe Notification for Web Services* document.

36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50

**Status:**

On 21 July 2004, this document was approved by the OASIS WS-Notification Technical Committee for publication so that users of the specification have a stable draft version available until the TC publishes a Committee Draft. Send comments to the editor.

Committee members should send comments on this specification to the [wsn@lists.oasis-open.org](mailto:wsn@lists.oasis-open.org) list. Others should subscribe to and send comments to the [wsn-comment@lists.oasis-open.org](mailto:wsn-comment@lists.oasis-open.org) list. To subscribe, send an email message to [wsn-comment-request@lists.oasis-open.org](mailto:wsn-comment-request@lists.oasis-open.org) with the word "subscribe" as the body of the message.

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/>). General OASIS IPR information can be found at <http://www.oasis-open.org/who/intellectualproperty.shtml>.

---

## 51 Table of Contents

52	1	Introduction .....	4
53	1.1	Goals and Requirements .....	4
54	1.1.1	Requirements.....	4
55	1.1.2	Non-Goals .....	5
56	1.2	Notational Conventions .....	5
57	1.3	Namespaces .....	5
58	2	Relationship to Other Specifications.....	6
59	3	Terminology and Concepts.....	7
60	4	Publishing .....	7
61	5	NotificationBroker Interface.....	9
62	5.1	NotificationBroker Resource Properties .....	9
63	5.2	Notify .....	9
64	5.3	Subscribe .....	9
65	5.4	RegisterPublisher .....	9
66	5.4.1	Example SOAP Encoding of the RegisterPublisher Message Exchange .....	11
67	6	PublisherRegistrationManager Interface .....	13
68	6.1	PublisherRegistration Resource Properties .....	13
69	7	Security Considerations .....	14
70	8	References.....	15
71		Appendix A. Acknowledgments .....	17
72		Appendix B. Revision History .....	17
73		Appendix C. Notices .....	17
74		Appendix D. UML.....	17
75		Appendix E. XML Schema.....	18
76		Appendix F. WSDL 1.1 .....	20
77			

77

# 1 Introduction

78

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.

81

82

This specification defines the Web services interface for the NotificationBroker. A NotificationBroker is an intermediary, which, 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-Base Notification* and *WS-Topics*, as well as the *Publish-Subscribe Notification for Web Services* document.

83

84

85

86

87

88

89

90

91

## 1.1 Goals and Requirements

92

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

93

94

95

96

97

### 1.1.1 Requirements

98

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

99

100

- **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 a SubscriptionManager (managing subscriptions) and NotificationProducer (distributing NotificationMessages) on behalf of the Publisher.

101

102

103

104

105

106

107

- **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.

108

109

110

111

112

- **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.

113

114

115

116

- **Must conform to WS-BaseNotification:** A NotificationBroker must support required message exchanges defined by the [\[WS-ResourceProperties\]](#) specification. It must conform to the NotificationProducer and the NotificationConsumer interfaces defined in WS-BaseNotification.

117

118

119

- 120 • **WS-BrokeredNotification must be independent of binding-level details:**  
 121 Transport protocol details must be orthogonal to the subscription and the delivery  
 122 of the notifications, so that the specification can be used over a variety of  
 123 different transports.
- 124 • **Must not exclude non-service producers and subscribers:** WS-  
 125 BrokeredNotification design must not exclude a non-service entity to deliver a  
 126 notification message to a NotificationBroker. It must not exclude a  
 127 NotificationBroker to send a notification to a non-service consumer.
- 128 • **Must provide publisher registration:** WS-BrokeredNotification must define  
 129 standard message exchanges for registering a NotificationPublisher with a  
 130 NotificationBroker.

131

### 132 1.1.2 Non-Goals

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

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

### 144 1.2 Notational Conventions

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

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

151 When describing concrete XML schemas, this specification uses the notational convention of  
 152 [\[Web Services Security\]](#). Specifically, each member of an element's [children] or [attributes]  
 153 property is described using an XPath-like notation (e.g., /x:MyHeader/x:SomeProperty/@value1).  
 154 The use of {any} indicates the presence of an element wildcard (<xsd:any/>). The use of @{any}  
 155 indicates the presence of an attribute wildcard (<xsd:anyAttribute/>).

### 156 1.3 Namespaces

157 The following namespaces are used in this document:

Prefix	Namespace
s12	http://www.w3.org/2003/05/soap-envelope

xsd	http://www.w3.org/2001/XMLSchema
wsa	http://schemas.xmlsoap.org/ws/2003/02/addressing
wsbn	http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BrokeredNotification-1.2-draft-01.xsd
wsnt	http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd
wsrl	http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.xsd
wsrp	http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd
wstop	http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.xsd
wsbf	http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-BaseFaults-1.2-draft-01.xsd
wsbfw	http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-BaseFaults-1.2-draft-01.wsdl

---

## 2 Relationship to Other Specifications

158

159 This specification builds on the basic notification mechanism defined in [[WS-BaseNotification](#)], by  
 160 adding the concept of an intermediary NotificationBroker, and describing additional variants on  
 161 the publisher role. A NotificationBroker takes on the role of both NotificationProducer and  
 162 NotificationConsumer (as defined in [[WS-BaseNotification](#)]), and its interactions with other  
 163 NotificationProducers and NotificationConsumers are largely defined by the WS-BaseNotification  
 164 specification.

165 This means that a NotificationBroker, implemented to conform to this specification, must also  
 166 conform to [[WS-BaseNotification](#)]. Such a NotificationBroker can deliver notification messages to  
 167 NotificationConsumers that are implemented to conform to [[WS-BaseNotification](#)], and can  
 168 subscribe to Notifications distributed by NotificationProducers as defined in [[WS-  
 169 BaseNotification](#)].

170 In addition a NotificationBroker MUST support hierarchical topics, and the ConcreteTopicPath  
 171 topic expression dialects defined in [[WS-Topics](#)].

172 Please refer to [[WS-Notification Whitepaper](#)] for a description of how WS-Notification relates to  
 173 other specifications, in particular to the WS-Resource Framework family of specifications.

174 WS-BrokeredNotification must be composable with other Web services specifications, in  
 175 particular WS-Security, WS-Policy, WS-Federation, WS-Addressing, WS-Coordination, WS-  
 176 ResourceProperties, WS-ResourceLifetime, WS-ReliableMessaging [[WS-ReliableMessaging](#)]  
 177 and the WS-Resource framework [[State Paper](#)].

178

179

## 3 Terminology and Concepts

180

Please refer to [\[WS-Notification Whitepaper\]](#) for a list of terms and their definitions.

181

## 4 Publishing

182

There are three distinct stages in the Notification process

183

- Observation of the Situation and its noteworthy characteristics;

184

- Creation of the NotificationMessage artifact that captures the noteworthy characteristics of the Situation; and

185

186

- Distribution of copies of the NotificationMessage to zero or more interested parties.

187

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,

188

189

190

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

191

192

193

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 NotificationMessages to interested parties. The implementer of this Web service can choose to program this behavior or delegate to specialized implementations of the Subscribe and NotificationMessage delivery behavior. This case is addressed by the WS-BaseNotification specification [\[WS-BaseNotification\]](#).

194

195

196

197

198

199

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

200

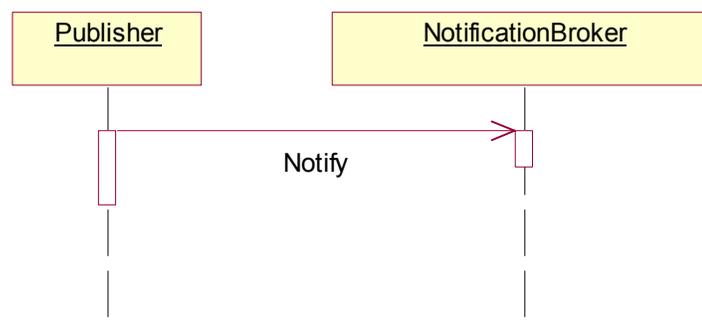
201

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

202

203

The following figure illustrates simple publishing:



204

205

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

206

207

208

209

In the composable publishing pattern, the role of the Publisher is played by a Web service that implements NotificationProducer. The act of observing the Situation and formatting the NotificationMessage happens within the implementation logic of the NotificationProducer itself.

210

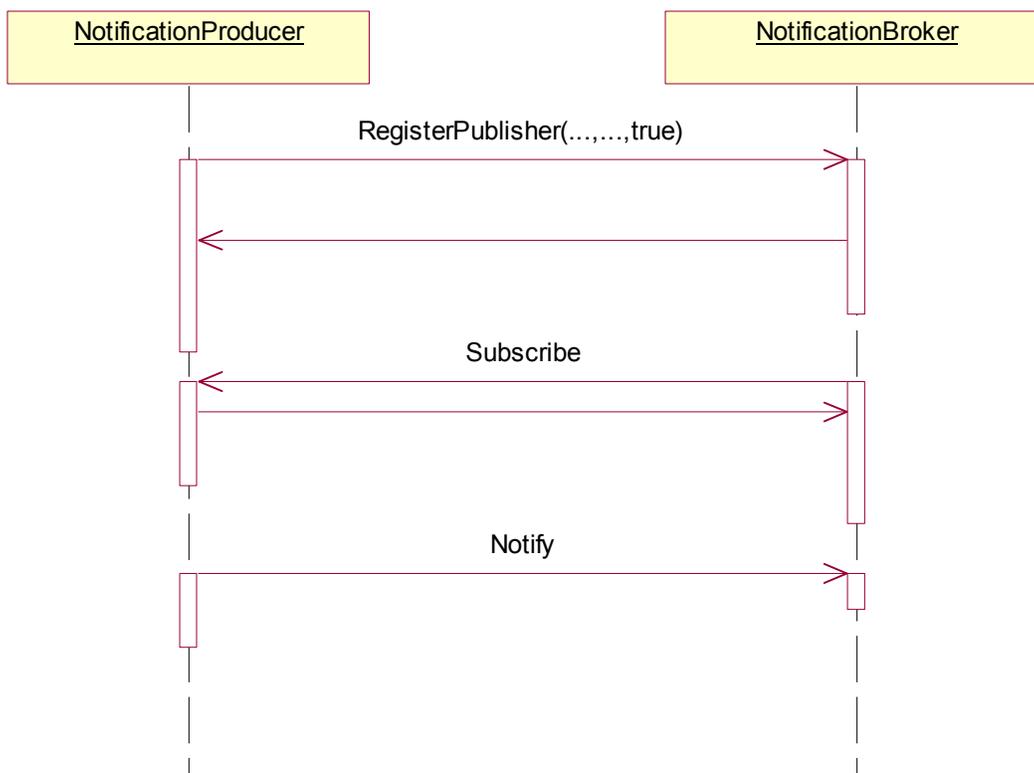
211

212 The NotificationMessage is disseminated by the NotificationProducer sending the Notify message  
213 to a NotificationBroker. The NotificationMessage may also be disseminated by sending the Notify  
214 message to any NotificationConsumer that had been previously subscribed to the  
215 NotificationProducer.

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

221 The last pattern, the *demand-based* pattern, requires the Publisher to be a NotificationProducer,  
222 and thereby accept the Subscribe message. Demand-based publication is intended for use in  
223 cases where the act of observing the Situation or the act of formatting the NotificationMessage  
224 artifact might be expensive to perform, and therefore should be avoided if there are no interested  
225 parties for that NotificationMessage. To use this pattern, the Publisher must register with the  
226 NotificationBroker, using the registration to express the intent to provide demand-based  
227 publishing only. Based upon this style of registration, the NotificationBroker sends the Subscribe  
228 message to the Publisher (recall: in this situation the Publisher must implement the message  
229 exchanges associated with the NotificationProducer interface).

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



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

---

## 233 5 NotificationBroker Interface

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

237 A NotificationBroker MUST support the required message exchanges defined by the WS-  
238 ResourceProperties specification [[WS-ResourceProperties](#)] and MAY support the optional  
239 message exchanges defined by WS-ResourceProperties.

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

- 242 • NotificationProducer
- 243 • NotificationConsumer

### 244 5.1 NotificationBroker Resource Properties

245 In addition to the Resource Property elements associated with the interfaces the Notification  
246 Broker extends, the Notification Broker must also include the following reference property  
247 element:

```
248 <xsd:element name="RequiresRegistration" type="xsd:boolean"/>
```

249 This resource property element is further constrained as follows:

250 /wsbn:RequiresRegistration

251 The value is "true" if the NotificationBroker requires a publisher to register (see 5.4)  
252 before sending it a Notify (i.e. publish) message on this Topic. The default is "false".

### 253 5.2 Notify

254 The NotificationBroker MUST support the Notify message exchange from  
255 NotificationConsumer interface [[WS-BaseNotification](#)], with the following  
256 clarifications/restrictions:

257 A Publisher sends a Notify message to a NotificationBroker in order to publish a  
258 NotificationMessage on a given Topic. As a result of the Publisher sending this message,  
259 NotificationMessages are delivered to all NotificationConsumers subscribed on the given Topic.  
260 For some Topics (those that require a Publisher to pre-register), the requestor must be a  
261 registered Publisher in order to successfully publish a NotificationMessage to the given Topic  
262 (see 5.4).

### 263 5.3 Subscribe

264 The NotificationBroker MUST support the Subscribe message exchange from the  
265 NotificationProducer interface [[WS-BaseNotification](#)]. Although a NotificationProducer MAY  
266 support any TopicExpression dialect, a NotificationBroker is further constrained in that it MUST  
267 accept use of the ConcreteTopicPath dialect (defined in [[WS-Topics](#)]) in the Subscribe message's  
268 TopicExpression. It MAY also accept use of the FullTopicPath dialect or any other  
269 TopicExpression dialects.

### 270 5.4 RegisterPublisher

271 The RegisterPublisher message is used by the Publisher to confirm its ability to publish on a  
272 given Topic or set of Topics.

273 If an entity wishes to register a publisher, it MUST send a RegisterPublisher request message to  
274 the NotificationBroker. The format of the RegisterPublisher request message is:

```
275 ...  
276 <wsbn:RegisterPublisher>  
277   <wsbn:PublisherReference>  
278     wsa:EndpointReference  
279   </wsbn:PublisherReference?>  
280   <wsbn:Topic dialect = "xsd:anyURI">  
281     {any}  
282   </wsbn:Topic>*<br>  
283   <wsbn:Demand>  
284     xsd:Boolean  
285   </wsbn:Demand?>  
286   <wsbn:InitialTerminationTime>  
287     xsd:dateTime  
288   </wsbn:InitialTerminationTime?>  
289 </wsbn:RegisterPublisher>  
290 ...
```

291 This request message MUST follow the implied resource pattern as outlined in [State Paper]. The  
292 components of the RegisterPublisher request message are further described as follows:

293 /wsbn:PublisherReference

294 An OPTIONAL EndpointReference to an entity that wishes to become a Publisher on one  
295 or more Topics supported by the Notification Broker. This component MUST appear if the  
296 /wsbn:Demand component has value "true". If this component is missing, the Publisher is  
297 either not a Web service, or does not wish to receive messages from the  
298 NotificationBroker.

299 /wsbn:Topic

300 A set of TopicExpressions that identifies one or more Topics. If included, the given  
301 Publisher is registered to publish only on the set of Topics identified by this component. If  
302 this is missing the Publisher is registered to publish on any topic supported by the  
303 NotificationBroker.

304 /wsbn:Demand

305 A boolean, default value is "false". If its value is "true", then the intent of the Publisher is  
306 to use a demand-based model from the NotificationBroker (see 4). In this case, the  
307 NotificationBroker must observe the rules associated with demand-based publishing,  
308 including establishing a Subscription with the Publisher on those Topics and  
309 pausing/resuming those Subscriptions as the NotificationBroker receives Subscriptions  
310 for those Topics.

311 /wsbn:InitialTerminationTime

312 This component contains the service requestor's suggestion for the initial termination  
313 time of the PublisherRegistration resource being created. This time is relative to the time  
314 source used by the NotificationBroker. If the NotificationBroker is unable or unwilling to  
315 set the TerminationTime to the given value or greater, then the RegisterPublisher request  
316 MUST fault. If the value is not "in the future" relative to the current time as known by the  
317 NotificationBroker, the RegisterPublisher request MUST fault. The use of the xsi:nil  
318 attribute with value "true" indicates there is no scheduled termination time requested for  
319 the RegisterPublisher. If the element does not include the time zone designation, the  
320 value of the element MUST be interpreted as universal time (UTC) time.

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

323 If a /wsbn:Topic component is included in the message, the NotificationBroker MUST register the  
324 Web service specified by the /wsbn:PublisherReference component as a Publisher on the set of  
325 Topics identified by the /wsbn:Topic component.

326 As part of the processing of a RegisterPublisher request, the NotificationBroker creates a  
327 PublisherRegistration resource representing the registration. A new resource is created  
328 regardless of whether the same Publisher has previously registered with the NotificationBroker.  
329 The NotificationBroker returns an EndpointReference in the response to the RegisterPublisher  
330 request. This EndpointReference is a WS-Resource-Qualified EndpointReference as defined in  
331 [State Paper] and includes the address of a PublisherRegistrationManager service and a  
332 reference property identifying the PublisherRegistration resource.

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

```
335 ...  
336 <wsbn:RegisterPublisherResponse>  
337   <wsbn:PublisherRegistrationReference>  
338     <wsa:Address>  
339       Address of PublisherRegistration Manager  
340     </wsa:Address>  
341     <wsa:ReferenceProperties>  
342       PublisherRegistration Identifier  
343     </wsa:ReferenceProperties>  
344     ...  
345   </wsbn:PublisherRegistrationReference>  
346 </wsbn:RegisterPublisherResponse>  
347 ...
```

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

349 /wsbn:PublisherRegistrationReference

350 A WS-Resource-Qualified EndpointReference to the PublisherRegistration WS-Resource  
351 created by the RegisterPublisher request message.

352 Instead of the RegisterPublisherResponse message, the NotificationBroker may also send the  
353 following faults in response to a RegisterPublisher request message:

- 354 • Invalid TopicExpression
- 355 • Given TopicExpression did not match any Topic supported by the NotificationBroker
- 356 • Publisher registration failed
- 357 • Others, TBD

## 358 5.4.1 Example SOAP Encoding of the RegisterPublisher Message 359 Exchange

360 The following is a non-normative example of a RegisterPublisher response message using SOAP  
361 1.2 [SOAP 1.2]:

```
362 <s12:Envelope  
363   xmlns:s12="http://www.w3.org/2003/05/soap-envelope"  
364   xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"  
365   xmlns:wsbn=  
366     "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-  
367   BrokeredNotification-1.2-draft-01.xsd"  
368   xmlns:npex="http://www.producer.org/RefProp">  
369   <s12:Header>  
370     <wsa:Action>  
371       http://docs.oasis-open.org/wsn/2004/06/WS-BrokeredNotification  
372   /RegisterPublisher  
373     </wsa:Action>  
374     <wsa:To s12:mustUnderstand="1">  
375       http://www.producer.org/NotificationBrokerEndpoint  
376     </wsa:To>  
377   </s12:Header>  
378   <s12:Body>
```

```

379     <wsbn:RegisterPublisher>
380       <wsbn:PublisherReference>
381         <wsa:Address>
382           http://www.producer.org/ProducerEndpoint
383         </wsa:Address>
384         <wsa:ReferenceProperties>
385           <npex: NPResourceDisambiguator>
386             uuid:84decd55-7d3f-65ad-ac44-675d9fce5d22
387           </npex: NPResourceDisambiguator>
388         </wsa:ReferenceProperties>
389       </wsbn:PublisherReference>
390     <wsbn:Topic dialect="http://docs.oasis-open.org/wsn/2004/06/WS-
391 Topics/SimpleTopicExpression">
392       npex:SomeTopic
393     </wsbn:Topic>
394     <wsbn:Demand>
395       true
396     </wsbn:Demand>
397     <wsbn:InitialTerminationTime>
398       2003-12-25T00:00:00.00000Z
399     </wsbn:InitialTerminationTime>
400   </wsbn:RegisterPublisher>
401 </s12:Body>
402 </s12:Envelope>

```

403 The following is a non-normative example of a RegisterPublisher response message using SOAP  
404 1.2 [[SOAP 1.2](#)]:

```

405 <s12:Envelope
406   xmlns:s12="http://www.w3.org/2003/06/soap-envelope"
407   xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
408   xmlns:wsbn=
409     "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
410 BrokeredNotification-1.2-draft-01.xsd"
411   xmlns:npex="http://www.consumer.org/RefProp">
412   <s12:Header>
413     <wsa:Action>
414       http://docs.oasis-open.org/wsn/2004/06/WS-BrokeredNotification
415 /RegisterPublisherResponse
416     </wsa:Action>
417     <wsa:To s12:mustUnderstand="1">
418       http://www.publisher.org/PublisherEndpoint
419     </wsa:To>
420     <pubex:PublisherId xmlns:pubex="http://www.publisher.org/RP">
421       Fred
422     </pubex:PublisherId>
423   </s12:Header>
424   <s12:Body>
425     <wsbn:RegisterPublisherResponse>
426       <wsbn:PublisherRegistrationReference>
427         <wsa:Address>
428           http://www.producer.org/PublisherEndpoint
429         </wsa:Address>
430         <wsa:ReferenceProperties>
431           <npex:NPubResourceId>
432             uuid:95fefeb3-f37d-5dfe-44fe-221d9fcee99
433           </npex:NPubResourceId>
434         </wsa:ReferenceProperties>
435       </wsbn:PublisherRegistrationReference>
436     </wsbn:RegisterPublisherResponse>
437   </s12:Body>
438 </s12:Envelope>

```

439

## 6 PublisherRegistrationManager Interface

440 The PublisherRegistrationManager interface defines message exchanges to manipulate  
441 PublisherRegistration resources. The PublisherRegistrationManager follows the implied resource  
442 pattern as described in [State Paper].

443 A PublisherRegistrationManager MUST support the required message exchanges associated  
444 with the WS-ResourceProperties specification [WS-ResourceProperties] and MAY support the  
445 optional message exchanges defined by WS-ResourceProperties.

446 The PublisherRegistrationManager MUST support the message exchanges defined for both  
447 forms of resource lifetime (immediate and scheduled destruction) by WS-ResourceLifetime [WS-  
448 ResourceLifetime]. These message exchanges define the means by which PublisherRegistration  
449 resources can be explicitly destroyed, or destroyed using a scheduled (time-based) mechanism.

### 6.1 PublisherRegistration Resource Properties

451 In addition to the resource properties required by WS-ResourceLifetime, the  
452 PublisherRegistration resource MUST include a reference to each of the following resource  
453 property elements in its resource properties document:

454

```
...  
455 <xsd:element name="PublisherReference"  
456             type="wsa:EndpointReference"  
457             minOccurs="0" maxOccurs="1" />  
458 <xsd:element name="Topic"  
459             type="wsnt:TopicExpressionType"  
460             minOccurs="0" maxOccurs="unbounded" />  
461 <xsd:element name="Demand"  
462             type="xsd:boolean"  
463             minOccurs="1" maxOccurs="1" />  
464 <xsd:element name="CreationTime"  
465             type="xsd:dateTime"  
466             minOccurs="0" maxOccurs="1" />  
467
```

468 These resource property elements are further described as follows:

469 /wsbn:PublisherReference and /wsbn:Topic and /wsbn:Demand

470 These elements are as defined in the description of the RegisterPublisher request  
471 message (see 5.4).

472 /wsbn:CreationTime

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

476 The following resource properties MAY be modified by the requestor, by sending a  
477 SetResourceProperties request message as defined in the WS-ResourceProperties specification:

- 478 • /wsbn:TopicPathExpression and /wsbn:Demand
  - 479 ○ Note: /wsbn:Demand may not take the value "true" if there is no
  - 480 /wsbn:PublisherReference resource property element in the resource property
  - 481 document.

482

---

## 7 Security Considerations

483

A non-normative discussion of the security scenarios and considerations associated with the entire family of WS-Notification specifications is contained in [\[WS-Notification Whitepaper\]](#).

484

485

486

---

## 8 References

- 486
- 487     **[RFC2119]**             S. Bradner, *Key words for use in RFCs to Indicate Requirement*  
488                             *Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March  
489                             1997.
- 490     **[WS-BaseNotification]** [http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf)  
491                             [1.2-draft-03.pdf](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf)
- 492     **[WS-Notification WhitePaper]**     [http://www.oasis-](http://www.oasis-open.org/committees/download.php/6661/WSNpubsub-1-0.pdf)  
493                             [open.org/committees/download.php/6661/WSNpubsub-1-0.pdf](http://www.oasis-open.org/committees/download.php/6661/WSNpubsub-1-0.pdf)
- 494     **[WS-Security]**             [http://www.oasis-open.org/committees/download.php/5531/oasis-](http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wss-soap-message-security-1.0.pdf)  
495                             [200401-wss-soap-message-security-1.0.pdf](http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wss-soap-message-security-1.0.pdf)
- 496     **[XML]**                     <http://www.w3.org/TR/REC-xml>
- 497     **[XML-Infoset]**            <http://www.w3.org/TR/xml-infoset/>
- 498     **[XPath]**                   <http://www.w3.org/TR/xpath>
- 499     **[SOAP 1.2]**               <http://www.w3.org/TR/soap12-part1/>
- 500     **[WS-Addressing]**         [http://www-106.ibm.com/developerworks/library/specification/ws-](http://www-106.ibm.com/developerworks/library/specification/ws-add/)  
501                             [add/](http://www-106.ibm.com/developerworks/library/specification/ws-add/)
- 502     **[WS-Topics]**             [http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf)  
503                             [01.pdf](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf)
- 504     **[State Paper]**            [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/wsrp/download.php/6795/ws-modelingresources.pdf)  
505                             [open.org/apps/org/workgroup/wsrp/download.php/6795/ws-](http://www.oasis-open.org/apps/org/workgroup/wsrp/download.php/6795/ws-modelingresources.pdf)  
506                             [modelingresources.pdf](http://www.oasis-open.org/apps/org/workgroup/wsrp/download.php/6795/ws-modelingresources.pdf)
- 507     **[WS-ResourceLifetime]** [http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-](http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceLifetime-1.2-draft-03.pdf)  
508                             [ResourceLifetime-1.2-draft-03.pdf](http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceLifetime-1.2-draft-03.pdf)
- 509     **[WS-ResourceProperties]** [http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-](http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-04.pdf)  
510                             [ResourceProperties-1.2-draft-04.pdf](http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-04.pdf)

---

## 511 **Appendix A. Acknowledgments**

512 The following individuals were members of the committee during the development of this  
513 specification:

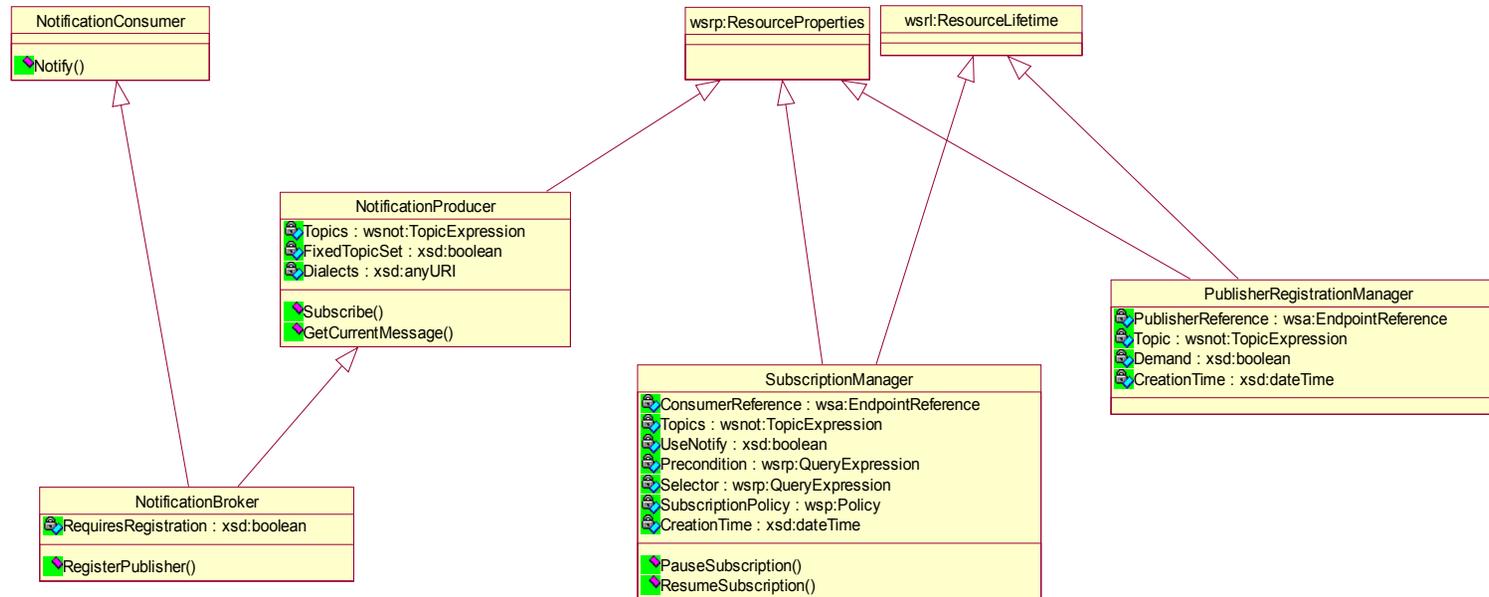
514 Geoff Acton, LSC Group Ltd, Sid Askary, Individual, Jeff Bohren, OpenNetwork, Fred Carter,  
515 AmberPoint, Martin Chapman, Oracle, Dave Chappell, Sonic Software, Ugo Corda, SeeBeyond  
516 Technology Corporation, Glen Daniels, Sonic Software, John Fuller, Individual, Stephen Graham,  
517 IBM, David Hull, Tibco, Anish Karmarkar, Oracle, John Kemp, Nokia, Paul Lipton, Computer  
518 Associates, Lily Liu, webMethods, Tom Maguire, IBM, Susan Malaika, IBM, David Martin, IBM,  
519 Samuel Meder, Argonne National Laboratory, Jeff Mischkinsky, Oracle, Bryan Murray, Hewlett-  
520 Packard, Peter Niblett, IBM, Dave Orchard, BEA Systems, Sanjay Patil, SAP, Mark Piller,  
521 webMethods, Ian Robinson, IBM, David Snelling, Fujitsu, Latha Srinivasan, Hewlett-Packard,  
522 John Tollefsrud, Sun Microsystems, Steve Tuecke, Globus / Argonne National Laboratory, Jem  
523 Treadwell, Hewlett-Packard, William Vambenepe, Hewlett-Packard, Alan Weissberger, NEC

524 Special thanks to the Global Grid Forum's Open Grid Services Infrastructure working group,  
525 which defined the OGSF v1.0 specification which was a large inspiration for the ideas expressed  
526 in this specification.

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

529 Tim Banks (IBM), Nick Butler (IBM), Doug Davis (IBM), John Dinger (IBM), Don Ferguson (IBM),  
530 Jeff Frey (IBM), Andreas Koeppel (SAP), Heather Kreger (IBM), Amy Lewis (TIBCO Software),  
531 Kevin Liu (SAP), Nataraj Nagaratnam (IBM), Martin Nally (IBM), Jeff Nick (IBM), Jay Parikh  
532 (Akamai Technologies), Claus von Riegen (SAP), Rick Rineholt (IBM), John Rofrano (IBM),  
533 Shivajee Samdarshi (TIBCO Software), Igor Sedukhin (Computer Associates), Eugène  
534 Sindambiwe (SAP), Jay Unger (IBM), Bill Wehl (Akamai Technologies), Mark Weitzel (IBM), Dan  
535 Wolfson (IBM).

## Appendix B. UML



540

## Appendix C. XML Schema

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

```
543 <?xml version="1.0" encoding="UTF-8"?>
544 <!--
545 OASIS takes no position regarding the validity or scope of any
546 intellectual property or other rights that might be claimed to pertain
547 to the implementation or use of the technology described in this
548 document or the extent to which any license under such rights might or
549 might not be available; neither does it represent that it has made any
550 effort to identify any such rights. Information on OASIS's procedures
551 with respect to rights in OASIS specifications can be found at the
552 OASIS website. Copies of claims of rights made available for
553 publication and any assurances of licenses to be made available, or the
554 result of an attempt made to obtain a general license or permission for
555 the use of such proprietary rights by implementors or users of this
556 specification, can be obtained from the OASIS Executive Director.
557
558 OASIS invites any interested party to bring to its attention any
559 copyrights, patents or patent applications, or other proprietary rights
560 which may cover technology that may be required to implement this
561 specification. Please address the information to the OASIS Executive
562 Director.
563
564 Copyright (C) OASIS Open (2004). All Rights Reserved.
565
566 This document and translations of it may be copied and furnished to
567 others, and derivative works that comment on or otherwise explain it or
568 assist in its implementation may be prepared, copied, published and
569 distributed, in whole or in part, without restriction of any kind,
570 provided that the above copyright notice and this paragraph are
571 included on all such copies and derivative works. However, this
572 document itself may not be modified in any way, such as by removing the
573 copyright notice or references to OASIS, except as needed for the
574 purpose of developing OASIS specifications, in which case the
575 procedures for copyrights defined in the OASIS Intellectual Property
576 Rights document must be followed, or as required to translate it into
577 languages other than English.
578
579 The limited permissions granted above are perpetual and will not be
580 revoked by OASIS or its successors or assigns.
581
582 This document and the information contained herein is provided on an
583 "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
584 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
585 INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
586 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
587 -->
588
589 <xsd:schema
590   xmlns="http://www.w3.org/2001/XMLSchema"
591   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
592   xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
593   xmlns:wsbn=
594     "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
595 BrokeredNotification-1.2-draft-01.xsd"
596   xmlns:wsnt=
```

```

597     "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-
598 1.2-draft-01.xsd"
599     targetNamespace=
600     "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
601 BrokeredNotification-1.2-draft-01.xsd"
602     elementFormDefault="qualified" attributeFormDefault="unqualified">
603
604 <!-- ===== Imports ===== -->
605
606     <xsd:import namespace=
607     "http://schemas.xmlsoap.org/ws/2003/03/addressing"
608     schemaLocation=
609     "http://schemas.xmlsoap.org/ws/2003/03/addressing"
610     />
611
612     <xsd:import namespace=
613     "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-
614 draft-01.xsd"
615     schemaLocation=
616     "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-
617 draft-01.xsd"
618     />
619
620 <!-- ===== Resource Properties for NotificationBroker ===== -->
621     <xsd:element name="RequiresRegistration" type="xsd:boolean"/>
622
623 <!-- ===== Resource Properties for PublisherRegistration ===== -->
624     <xsd:element name="PublisherReference"
625     type="wsa:EndpointReferenceType" />
626     <xsd:element name="Topic"
627     type="wsnt:TopicExpressionType" />
628     <xsd:element name="Demand"
629     type="xsd:boolean" />
630     <xsd:element name="CreationTime"
631     type="xsd:dateTime" />
632 </xsd:schema>

```

## Appendix D. WSDL 1.1

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

636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688

```
<?xml version="1.0" encoding="utf-8"?>
<!--
OASIS takes no position regarding the validity or scope of any
intellectual property or other rights that might be claimed to pertain
to the implementation or use of the technology described in this
document or the extent to which any license under such rights might or
might not be available; neither does it represent that it has made any
effort to identify any such rights. Information on OASIS's procedures
with respect to rights in OASIS specifications can be found at the
OASIS website. Copies of claims of rights made available for
publication and any assurances of licenses to be made available, or the
result of an attempt made to obtain a general license or permission for
the use of such proprietary rights by implementors or users of this
specification, can be obtained from the OASIS Executive Director.

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

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

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

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

This document and the information contained herein is provided on an
"AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
-->
<wsdl:definitions name="WS-BrokeredNotification"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsbn=
```

```

689     "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
690 BrokeredNotification-1.2-draft-01.xsd"
691     xmlns:wsnt=
692     "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
693 BaseNotification-1.2-draft-01.xsd"
694     xmlns:wsbf=
695     "http://docs.oasis-open.org/wsn/2004/06/wsrf-WS-BaseFaults-1.2-
696 draft-01.xsd"
697     xmlns:wsrp=
698     "http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
699 ResourceProperties-1.2-draft-01.xsd"
700     xmlns:wsl=
701     "http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
702 ResourceLifetime-1.2-draft-01.xsd"
703     targetNamespace=
704     "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
705 BrokeredNotification-1.2-draft-01.wsdl">
706
707 <!-- ===== Imports ===== -->
708     <wsdl:import
709         namespace=
710         "http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
711 ResourceProperties-1.2-draft-01.wsdl"
712         location=
713         "http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
714 ResourceProperties-1.2-draft-01.wsdl" />
715
716     <wsdl:import
717         namespace=
718         "http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
719 ResourceLifetime-1.2-draft-01.wsdl"
720         location=
721         "http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
722 ResourceLifetime-1.2-draft-01.wsdl" />
723
724     <wsdl:import
725         namespace=
726         "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
727 BaseNotification-1.2-draft-01.wsdl"
728         location=
729         "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-
730 1.2-draft-01.wsdl" />
731
732 <!-- ===== Types Definitions ===== -->
733     <wsdl:types>
734         <xsd:schema
735             targetNamespace=
736             "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
737 BrokeredNotification-1.2-draft-01.xsd">
738
739             <xsd:include schemaLocation=
740                 "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
741 BrokeredNotification-1.2-draft-01.xsd" />
742
743             <xsd:import
744                 namespace=
745                 "http://schemas.xmlsoap.org/ws/2003/03/addressing"
746                 schemaLocation=
747                 "http://schemas.xmlsoap.org/ws/2003/03/addressing"
748             />

```

```

749
750         <xsd:import namespace=
751         "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
752 BaseNotification-1.2-draft-01.xsd"
753         schemaLocation=
754         "http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
755 BaseNotification-1.2-draft-01.xsd"
756         />
757
758     <xsd:import
759     namespace=
760     "http://docs.oasis-open.org/wsn/2004/06/wsrp-WS-BaseFaults-1.2-
761 draft-01.xsd"
762     schemaLocation=
763     "http://docs.oasis-open.org/wsn/2004/06/wsrp-WS-BaseFaults-1.2-draft-
764 01.xsd"
765     />
766
767     <xsd:import namespace=
768     "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-
769 ResourceProperties-1.2-draft-01.xsd"
770     schemaLocation=
771     "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-
772 ResourceProperties-1.2-draft-01.xsd"
773     />
774
775     <xsd:import
776     namespace=
777     "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-
778 ResourceLifetime-1.2-draft-01.xsd"
779     schemaLocation=
780     "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-
781 ResourceLifetime-1.2-draft-01.xsd"
782     />
783
784 <!-- ===== Resource Property Related ===== -->
785
786 <!-- ===== Resource Properties for NotificationBroker ===== -->
787     <xsd:element name="NotificationBrokerRP" >
788         <xsd:complexType>
789             <xsd:sequence>
790                 <!-- From NotificationProducer -->
791                 <xsd:element ref="wsnt:Topic"
792                     minOccurs="1" maxOccurs="unbounded" />
793                 <xsd:element ref="wsnt:FixedTopicSet"
794                     minOccurs="1" maxOccurs="1" />
795                 <xsd:element ref="wsnt:TopicExpressionDialects"
796                     minOccurs="1" maxOccurs="unbounded" />
797                 <!-- NotificationBroker specific -->
798                 <xsd:element ref="wsbn:RequiresRegistration"
799                     minOccurs="1" maxOccurs="1" />
800             </xsd:sequence>
801         </xsd:complexType>
802     </xsd:element>
803
804 <!-- ===== Resource Properties for PublisherRegistration ===== -->
805     <xsd:element name="PublisherRegistrationRP" >
806         <xsd:complexType>
807             <xsd:sequence>
808                 <!-- From WS-ResourceLifetime ScheduledResourceTermination -->
809                 <xsd:element ref="wsrl:CurrentTime"

```

```

810         minOccurs="1" maxOccurs="1" />
811     <xsd:element ref="wsrl:TerminationTime"
812         minOccurs="1" maxOccurs="1" />
813
814     <!-- PublisherRegistration specific -->
815     <xsd:element ref="wsbn:PublisherReference"
816         minOccurs="0" maxOccurs="1" />
817     <xsd:element ref="wsbn:Topic"
818         minOccurs="0" maxOccurs="unbounded" />
819     <xsd:element ref="wsbn:Demand"
820         minOccurs="1" maxOccurs="1" />
821     <xsd:element ref="wsbn:CreationTime"
822         minOccurs="0" maxOccurs="1" />
823     </xsd:sequence>
824 </xsd:complexType>
825 </xsd:element>
826
827 <!-- ===== Message Types for NotificationBroker ===== -->
828 <xsd:element name="RegisterPublisher">
829     <xsd:complexType>
830         <xsd:sequence>
831             <xsd:element name="PublisherReference"
832                 type="wsa:EndpointReferenceType"
833                 minOccurs="0" maxOccurs="1" />
834             <xsd:element name="Topic"
835                 type="wsnt:TopicExpressionType"
836                 minOccurs="0" maxOccurs="unbounded" />
837             <xsd:element name="Demand"
838                 type="xsd:boolean" default="false"
839                 minOccurs="0" maxOccurs="1" />
840             <xsd:element name="InitialTerminationTime"
841                 type="xsd:dateTime"
842                 minOccurs="0" maxOccurs="1" />
843         </xsd:sequence>
844     </xsd:complexType>
845 </xsd:element>
846
847 <xsd:element name="RegisterPublisherResponse">
848     <xsd:complexType>
849         <xsd:sequence>
850             <xsd:element name="PublisherRegistrationReference"
851                 type="wsa:EndpointReferenceType"
852                 minOccurs="0" maxOccurs="1" />
853         </xsd:sequence>
854     </xsd:complexType>
855 </xsd:element>
856
857 <xsd:complexType name="ResourceUnknownFaultType">
858     <xsd:complexContent>
859         <xsd:extension base="wsbf:BaseFaultType"/>
860     </xsd:complexContent>
861 </xsd:complexType>
862 <xsd:element name="ResourceUnknownFault"
863     type="wsbn:ResourceUnknownFaultType"/>
864
865 <xsd:complexType name="InvalidTopicExpressionFaultType">
866     <xsd:complexContent>
867         <xsd:extension base="wsbf:BaseFaultType"/>
868     </xsd:complexContent>
869 </xsd:complexType>
870 <xsd:element name="InvalidTopicExpressionFault"
871     type="wsbn:InvalidTopicExpressionFaultType"/>

```

872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934

```
<xsd:complexType name="TopicNotSupportedFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsbf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="TopicNotSupportedFault"
  type="wsbn:TopicNotSupportedFaultType"/>

<xsd:complexType name="PublisherRegistrationFailedFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsbf:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="PublisherRegistrationFailedFault"
  type="wsbn:PublisherRegistrationFailedFaultType"/>

</xsd:schema>
</wsdl:types>

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

<wsdl:message name="RegisterPublisherResponse">
  <wsdl:part name="RegisterPublisherResponse"
    element="wsbn:RegisterPublisherResponse"/>
</wsdl:message>

<wsdl:message name="ResourceUnknownFault">
  <part name="ResourceUnknownFault"
    element="wsbn:ResourceUnknownFault" />
</wsdl:message>

<wsdl:message name="InvalidTopicExpressionFault">
  <part name="InvalidTopicExpressionFault"
    element="wsbn:InvalidTopicExpressionFault" />
</wsdl:message>

<wsdl:message name="TopicNotSupportedFault">
  <part name="TopicNotSupportedFault"
    element="wsbn:TopicNotSupportedFault" />
</wsdl:message>

<wsdl:message name="PublisherRegistrationFailedFault">
  <part name="PublisherRegistrationFailedFault"
    element="wsbn:PublisherRegistrationFailedFault" />
</wsdl:message>

<!-- ===== PortType Definitions ===== -->
<!-- ===== NotificationBroker PortType Definition ===== -->
<wsdl:portType name="NotificationBroker"
  wsrp:ResourceProperties ="wsbn:NotificationBrokerRP">
  <!-- ===== extends NotificationConsumer ===== -->
  <wsdl:operation name="Notify">
    <wsdl:input message="wsnt:Notify" />
  </wsdl:operation>
```

```

935
936 <!-- ===== extends wsrp:ResourceProperties ===== -->
937 <wsdl:operation name="GetResourceProperty">
938     <wsdl:input name="GetResourcePropertyRequest"
939               message="wsrp:GetResourcePropertyRequest" />
940     <wsdl:output name="GetResourcePropertyResponse"
941               message="wsrp:GetResourcePropertyResponse" />
942     <wsdl:fault name="ResourceUnknownFault"
943               message="wsrp:ResourceUnknownFault" />
944     <wsdl:fault name="InvalidResourcePropertyQNameFault"
945               message="wsrp:InvalidResourcePropertyQNameFault"
946 />
947 </wsdl:operation>
948
949 <!-- ===== extends NotificationProducer ===== -->
950 <wsdl:operation name="Subscribe">
951     <wsdl:input message="wsnt:SubscribeRequest" />
952     <wsdl:output message="wsnt:SubscribeResponse" />
953     <wsdl:fault name="ResourceUnknownFault"
954               message="wsnt:ResourceUnknownFault" />
955     <wsdl:fault name="SubscribeCreationFailedFault"
956               message="wsnt:SubscribeCreationFailedFault"/>
957 </wsdl:operation>
958
959 <wsdl:operation name="GetCurrentMessage">
960     <wsdl:input message="wsnt:GetCurrentMessageRequest"/>
961     <wsdl:output message="wsnt:GetCurrentMessageResponse"/>
962     <wsdl:fault name="ResourceUnknownFault"
963               message="wsnt:ResourceUnknownFault" />
964     <wsdl:fault name="InvalidTopicExpressionFault"
965               message="wsnt:InvalidTopicExpressionFault" />
966     <wsdl:fault name="TopicNotSupportedFault"
967               message="wsnt:TopicNotSupportedFault" />
968     <wsdl:fault name="NoCurrentMessageOnTopicFault"
969               message="wsnt:NoCurrentMessageOnTopicFault" />
970 </wsdl:operation>
971
972 <!-- ===== NotificationBroker specific operations ===== -->
973 <wsdl:operation name="RegisterPublisher">
974     <wsdl:input message="wsbn:RegisterPublisherRequest"/>
975     <wsdl:output message="wsbn:RegisterPublisherResponse"/>
976     <wsdl:fault name="ResourceUnknownFault"
977               message="wsbn:ResourceUnknownFault" />
978     <wsdl:fault name="InvalidTopicExpressionFault"
979               message="wsbn:InvalidTopicExpressionFault"/>
980     <wsdl:fault name="TopicNotSupportedFault"
981               message="wsbn:TopicNotSupportedFault"/>
982     <wsdl:fault name="PublisherRegistrationFailedFault"
983               message="wsbn:PublisherRegistrationFailedFault"/>
984 </wsdl:operation>
985 </wsdl:portType>
986
987 <!-- ===== PublisherRegistrationManager PortType Definition ===== -->
988 <wsdl:portType name="PublisherRegistrationManager"
989               wsrp:ResourceProperties ="wsbn:PublisherRegistrationManagerRP">
990     <!-- ===== extends wsrp:ResourceProperties ===== -->
991     <wsdl:operation name="GetResourceProperty">
992         <wsdl:input name="GetResourcePropertyRequest"
993                   message="wsrp:GetResourcePropertyRequest" />
994         <wsdl:output name="GetResourcePropertyResponse"
995                   message="wsrp:GetResourcePropertyResponse" />
996         <wsdl:fault name="ResourceUnknownFault"
997                   message="wsrp:ResourceUnknownFault" />

```

```

998         <wsdl:fault name="InvalidResourcePropertyQNameFault"
999             message="wsrp:InvalidResourcePropertyQNameFault"
1000     />
1001     </wsdl:operation>
1002
1003     <!-- === extends wsrl:ImmediateResourceTermination ===== -->
1004     <wsdl:operation name="Destroy">
1005         <wsdl:input message="wsrl:DestroyRequest" />
1006         <wsdl:output message="wsrl:DestroyResponse" />
1007         <wsdl:fault name="ResourceUnknownFault"
1008             message="wsrl:ResourceUnknownFault" />
1009         <wsdl:fault name="ResourceNotDestroyedFault"
1010             message="wsrl:ResourceNotDestroyedFault" />
1011     </wsdl:operation>
1012
1013     <!-- === extends wsrl:ScheduledResourceTermination ===== -->
1014     <wsdl:operation name="SetTerminationTime">
1015         <wsdl:input message="wsrl:SetTerminationTimeRequest" />
1016         <wsdl:output message="wsrl:SetTerminationTimeResponse" />
1017         <wsdl:fault name="ResourceUnknownFault"
1018             message="wsrl:ResourceUnknownFault" />
1019         <wsdl:fault name="UnableToSetTerminationTimeFault"
1020             message="wsrl:UnableToSetTerminationTimeFault" />
1021         <wsdl:fault name="TerminationTimeChangeRejectedFault"
1022             message="wsrl:TerminationTimeChangeRejectedFault"
1023     />
1024     </wsdl:operation>
1025     <!-- ===== PublisherRegistrationManager specific operations == -->
1026 </wsdl:portType>
1027 </wsdl:definitions>

```

---

## Appendix E. Revision History

Rev	Date	By Whom	What
Ws-bn-01	2004-05-12	Lily Liu	Initial version
Ws-bn-02	2004-06-07	Dave Chappell	Updates and consistency check w/ other WS-N specs
Ws-bn-03	2004-06-24	Lily Liu, Dave Chappell	Addition of a Goals and Requirements section and minor format changes
Ws-bn-03	2004-07-12	Lily Liu	Addition of a status paragraph
Ws-bn-01	2004-07-20	Lily Liu	Changed the version back to draft 1. Fixed the URLs and references.

---

## Appendix F. Notices

1030 OASIS takes no position regarding the validity or scope of any intellectual property or other rights  
1031 that might be claimed to pertain to the implementation or use of the technology described in this  
1032 document or the extent to which any license under such rights might or might not be available;  
1033 neither does it represent that it has made any effort to identify any such rights. Information on  
1034 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS  
1035 website. Copies of claims of rights made available for publication and any assurances of licenses  
1036 to be made available, or the result of an attempt made to obtain a general license or permission  
1037 for the use of such proprietary rights by implementers or users of this specification, can be  
1038 obtained from the OASIS Executive Director.

1039 OASIS invites any interested party to bring to its attention any copyrights, patents or patent  
1040 applications, or other proprietary rights which may cover technology that may be required to  
1041 implement this specification. Please address the information to the OASIS Executive Director.

1042 Copyright © OASIS Open 2004. *All Rights Reserved.*

1043 This document and translations of it may be copied and furnished to others, and derivative works  
1044 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,  
1045 published and distributed, in whole or in part, without restriction of any kind, provided that the  
1046 above copyright notice and this paragraph are included on all such copies and derivative works.  
1047 However, this document itself does not be modified in any way, such as by removing the  
1048 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS  
1049 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual  
1050 Property Rights document must be followed, or as required to translate it into languages other  
1051 than English.

1052 The limited permissions granted above are perpetual and will not be revoked by OASIS or its  
1053 successors or assigns.

1054 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
1055 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO  
1056 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE  
1057 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
1058 PARTICULAR PURPOSE.