



Web Services Base Notification 1.3 (WS-BaseNotification)

OASIS Standard, 1 October 2006

Document identifier

wsn-ws_base_notification-1.3-spec-os

Location:

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

Editors:

Steve Graham, IBM <sggraham@us.ibm.com>

David Hull, Tibco <dmh@tibco.com>

Bryan Murray, Hewlett-Packard Company <bryan.murray@hp.com>

Abstract:

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

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

Status:

This document is an OASIS standard.

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

35 works of the TC) are publicly archived and can be viewed at:
36 <http://lists.oasis-open.org/archives/wsn-comment/>.
37 For information on whether any patents have been disclosed that may be essential to
38 implementing this specification, and any offers of patent licensing terms, please refer to
39 the Intellectual Property Rights section of the WSN TC web page ([http://www.oasis-
open.org/committees/wsn/](http://www.oasis-
40 open.org/committees/wsn/)).
41 The errata document for this specification is maintained at:
42 http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-errata.pdf

Table of Contents

44	1	Introduction	5
45	1.1	Goals and Requirements	5
46	1.1.1	Requirements.....	5
47	1.1.2	Non-Goals.....	6
48	1.2	Terminology.....	6
49	1.3	Namespaces	7
50	1.4	Fault Definitions.....	8
51	2	Terminology and Concepts	9
52	2.1	Definitions.....	9
53	2.2	Production vs. Delivery	10
54	3	NotificationConsumer Interface	11
55	3.1	Notification Metadata	11
56	3.2	Notify	12
57	3.2.1	Example SOAP Encoding of the Notify Message.....	13
58	4	NotificationProducer Interface	14
59	4.1	NotificationProducer Resource Properties.....	14
60	4.2	Subscribe	15
61	4.2.1	Example SOAP Encoding of the Subscribe Message Exchange.....	21
62	4.3	GetCurrentMessage.....	22
63	4.3.1	Example SOAP Encoding of the GetCurrentMessage Message Exchange	24
64	5	Pull-Style Notification	25
65	5.1	PullPoint Interface	25
66	5.1.1	Accumulating Notification Messages.....	25
67	5.1.2	GetMessages.....	26
68	5.1.3	Example SOAP Encoding of the GetMessages Message Exchange.....	27
69	5.1.4	DestroyPullPoint	28
70	5.1.5	Example SOAP Encoding of the DestroyPullPoint Message Exchange	28
71	5.2	Create PullPoint Interface	29
72	5.2.1	Example SOAP Encoding of the CreatePullPoint Message Exchange.....	30
73	6	SubscriptionManager Interface.....	31
74	6.1	Base Subscription Manager	31
75	6.1.1	Renew.....	31
76	6.1.2	Unsubscribe	33
77	6.2	Pausable Subscription Manager	34
78	6.2.1	PauseSubscription	35
79	6.2.2	Example SOAP Encoding of the PauseSubscription Message Exchange.....	36
80	6.3	ResumeSubscription	36

81	6.3.1 Example SOAP Encoding of the ResumeSubscription Message Exchange.....	37
82	6.4 Subscriptions as WS-Resources.....	38
83	6.4.1 Subscription Resource Properties	38
84	7 Security Considerations	40
85	7.1 Securing the Message Exchanges.....	40
86	7.2 Securing Subscriptions and Notifications.....	40
87	8 References.....	42
88	8.1 Normative	42
89	8.2 Non-Normative	42
90	Appendix A. Acknowledgments	44
91	Appendix B. XML Schema.....	45
92	Appendix C. WSDL 1.1.....	56
93	Appendix D. Revision History	65
94	Appendix E. Notices	68
95		

1 Introduction

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

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

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

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

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

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

1.1 Goals and Requirements

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

1.1.1 Requirements

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

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

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

- **WS-BaseNotification 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 allow for Message Oriented Middleware implementations.** The design of the WS-BaseNotification must allow a service that is acting as a NotificationProducer to delegate its implementation of WS-BaseNotification semantics to a Message Oriented Middleware provider.
- **Relationship to other WS-* specifications:** WS-BaseNotification must be composable with other Web services specifications.

1.1.2 Non-Goals

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

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

1.2 Terminology

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

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., [some property]).

This specification uses a notational convention, referred to as “Pseudo-schemas”. 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,
- ‘+’ one or more occurrences,
- ‘[’ 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/>).
- In the interest of brevity, some extensibility points have been omitted from the Pseudo-schemas.

```

<!-- 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 pseudo-schema) 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.

1.3 Namespaces

The following namespaces are used in this document:

Prefix	Namespace
s	http://schemas.xmlsoap.org/soap/envelope/ OR http://www.w3.org/2003/05/soap-envelope
xsd	http://www.w3.org/2001/XMLSchema
wsa	http://www.w3.org/2005/08/addressing
wsrf-rp	http://docs.oasis-open.org/wsrf/rp-2
wsrf-bf	http://docs.oasis-open.org/wsrf/bf-2
wsnt	http://docs.oasis-open.org/wsn/b-2
wsntw	http://docs.oasis-open.org/wsn/bw-2
wstop	http://docs.oasis-open.org/wsn/t-1

201 **1.4 Fault Definitions**

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

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

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

2 Terminology and Concepts

2.1 Definitions

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

Situation:

- A Situation is some occurrence known to a NotificationProducer and of potential interest to third parties.
- A Situation could be a change of the internal state of a resource or could be environmental, such as a timer event. It could also be an external event, such as a piece of news that has been supplied by a news-feed service.
- WS-Notification does not specify what a Situation is or is not, nor does it define the relationship between a Situation and the Notification(s) that are used to describe it.

Notification:

- A Notification is an artifact of a Situation containing information about that Situation that some entity wishes to communicate to other entities.
- A Notification is represented as an XML element with a Namespace qualified QName and a type defined using XML Schema.
- A typical usage pattern is to define a single Notification type (to be precise, its defining XML element) for each kind of Situation, containing information pertinent to that kind of Situation; in this case one can think of a Notification instance as in some sense *being* (or at least representing) the Situation.
- A designer could choose to associate several different Notification types with a Situation, for example, describing different aspects of the Situation, destined for different target recipients, etc. Conversely it is possible that several essentially different Situations give rise to Notifications of the same type.

NotificationProducer:

- A NotificationProducer is a Web service that implements the message exchanges associated with the NotificationProducer interface.
- A NotificationProducer is capable of producing Notifications for those NotificationConsumers for which Subscriptions have been registered, based on Situations that occur and on the parameters supplied with the requests from which the Subscriptions were created.
- A Web Service that implements the message exchanges associated with NotificationProducer may directly produce Notifications itself, or it may be a NotificationBroker, reproducing Notifications that were produced by separate Publisher and/or NotificationProducer entities.
- It is the factory for Subscription resources.

NotificationConsumer:

244 • A NotificationConsumer is an endpoint, represented by a WS-Addressing endpoint
245 reference, designated to receive Notifications produced by a NotificationProducer as a
246 result of a subscription.

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

249 **Subscription:**

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

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

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

257 **SubscriptionManager**

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

261 • A SubscriptionManager provides operations that allow a service requestor to query and
262 manipulate Subscription resources that it manages.

263 • A SubscriptionManager is subordinate to the NotificationProducer, and MAY be
264 implemented by the NotificationProducer service provider. However WS-Notification
265 permits it to be implemented by a separate service provider, should an implementer so
266 desire.

267 **Subscriber:**

268 • A Subscriber is any entity that sends the SubscribeRequest message to a
269 NotificationProducer.

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

272 **2.2 Production vs. Delivery**

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

3 NotificationConsumer Interface

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

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

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

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

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

3.1 Notification Metadata

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

wsnt:SubscriptionReference

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

wsnt:Topic

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

wsnt:Topic/@Dialect

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

wsnt:ProducerReference

An EndpointReference to the NotificationProducer that produced the Notification artifact.

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

wsn-ws_base_notification-1.3-spec-os

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

1 October 2006

Page 11 of 68

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

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

3.2 Notify

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

```
...
<wsnt:Notify>
  <wsnt:NotificationMessage>
    <wsnt:SubscriptionReference>
      wsa:EndpointReferenceType
    </wsnt:SubscriptionReference> ?
    <wsnt:Topic Dialect="xsd:anyURI">
      {any} ?
    </wsnt:Topic>?
    <wsnt:ProducerReference>
      wsa:EndpointReferenceType
    </wsnt:ProducerReference> ?
    <wsnt:Message>
      {any}
    </wsnt:Message>
  </wsnt:NotificationMessage> +
  {any} *
</wsnt:Notify>
...
```

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

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

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

/wsnt:Notify

Contains a collection of one or more Notifications.

/wsnt:Notify/wsnt:NotificationMessage

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

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

A copy of the actual Notification payload.

/wsnt:Notify/{any}

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

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

3.2.1 Example SOAP Encoding of the Notify Message

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

```
<s:Envelope ... >
  <s:Header>
    <wsa:Action>
      http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify
    </wsa:Action>
    ...
  </s:Header>
  <s:Body>
    <wsnt:Notify>
      <wsnt:NotificationMessage>
        <wsnt:SubscriptionReference>
          <wsa:Address>
            http://www.example.org/SubscriptionManager
          </wsa:Address>
        </wsnt:SubscriptionReference>
        <wsnt:Topic Dialect=
          "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
          npex:SomeTopic
        </wsnt:Topic>
        <wsnt:ProducerReference>
          <wsa:Address>
            http://www.example.org/NotificationProducer
          </wsa:Address>
        </wsnt:ProducerReference>
        <wsnt:Message>
          <npex:NotifyContent>exampleNotifyContent</npex:NotifyContent>
        </wsnt:Message>
      </wsnt:NotificationMessage>
    </wsnt:Notify>
  </s:Body>
</s:Envelope>
```

4 NotificationProducer Interface

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

4.1 NotificationProducer Resource Properties

In addition to the message exchanges described in this specification, a NotificationProducer 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 a case, this specification defines several resource properties, which MUST conform to the following schema fragment for content:

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

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

```
<xsd:element ref="wsnt:TopicExpression"
  minOccurs="0" maxOccurs="unbounded" />

<xsd:element ref="wsnt:FixedTopicSet"
  minOccurs="0" maxOccurs="1" />

<xsd:element ref="wsnt:TopicExpressionDialect"
  minOccurs="0" maxOccurs="unbounded" />

<xsd:element ref="wstop:TopicSet"
  minOccurs="0" maxOccurs="1" />
```

These Resource Property elements are further constrained as follows:

/wsnt:TopicExpression

This resource property contains a collection of topics supported by the NotificationProducer. The set of topics is expressed using one or more wsnt:TopicExpression resource property element(s). Each wsnt:TopicExpression resource property element is a TopicExpression. The dialect of TopicExpression used can be any dialect. It is RECOMMENDED to use one of the

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

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

/wsnt:FixedTopicSet

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

/wsnt:TopicExpressionDialect

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

/wstop:TopicSet

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

4.2 Subscribe

A NotificationProducer is capable of 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 NotificationProducer in order to register this interest.

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

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

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

485 NotificationProducers MAY advertise more constrained behavior through policy assertions or
486 other means.

487 The format of the Subscribe message is:

```
488 ...  
489 <wsnt:Subscribe>  
490   <wsnt:ConsumerReference>  
491     wsa:EndpointReferenceType  
492   </wsnt:ConsumerReference>  
493   <wsnt:Filter>  
494     [ <wsnt:TopicExpression Dialect="xsd:anyURI">  
495       {any} ?  
496     </wsnt:TopicExpression> |  
497     <wsnt:ProducerProperties Dialect="xsd:anyURI">  
498       {any} ?  
499     </wsnt:ProducerProperties> |  
500     <wsnt:MessageContent Dialect="xsd:anyURI">  
501       {any} ?  
502     </wsnt:MessageContent> |  
503     {any} *  
504   ] *  
505 </wsnt:Filter> ?  
506 <wsnt:InitialTerminationTime>  
507   [xsd:dateTime | xsd:duration]  
508 </wsnt:InitialTerminationTime> ?  
509 <wsnt:SubscriptionPolicy>  
510   [ <wsnt:UseRaw/> |  
511     {any}  
512   ] *  
513 </wsnt:SubscriptionPolicy> ?  
514 {any}*  
515 </wsnt:Subscribe>  
516 ...
```

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

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

520 /wsnt:Subscribe/wsnt:ConsumerReference

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

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

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

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

535 /wsnt:Subscribe/wsnt:Filter

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

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

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

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

555 /wsnt:Subscribe/wsnt:Filter/wsnt:TopicExpression

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

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

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

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

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

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

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

575 /wsnt:Subscribe/wsnt:Filter/wsnt:ProducerProperties

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

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

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

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

586 /wsnt:Subscribe/wsnt:Filter/wsnt:MessageContent

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

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

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

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

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

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

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

601 /wsnt:Subscribe/wsnt:InitialTerminationTime

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

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

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

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

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

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

622 /wsnt:Subscribe/wsnt:SubscriptionPolicy

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

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

632 /wsnt:Subscribe/wsnt:SubscriptionPolicy/wsnt:UseRaw

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

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

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

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

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

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

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

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

```
653 ...
654 <wsnt:SubscribeResponse>
655   <wsnt:SubscriptionReference>
656     wsa:EndpointReferenceType
657   </wsnt:SubscriptionReference>
658   <wsnt:CurrentTime>xsd:dateTime</wsnt:CurrentTime> ?
659   <wsnt:TerminationTime>xsd:dateTime</wsnt:TerminationTime> ?
660   {any}*
661 </wsnt:SubscribeResponse>
```

662 ...

663 The WS-Addressing [action] Message Addressing Property MUST contain the URI
664 <http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/SubscribeResponse>.
665 The contents of the SubscribeResponse message are further described as follows:
666 /wsnt:SubscribeResponse/wsnt:SubscriptionReference
667 A reference to the Subscription created as a result of the Subscribe message.
668 /wsnt:SubscribeResponse/wsnt:CurrentTime
669 This OPTIONAL component SHOULD be returned if the SubscriptionManager uses
670 scheduled termination from WS-ResourceLifetime. The value of this component is the value
671 of the CurrentTime resource property of the Subscription at the time the response message is
672 created.
673 /wsnt:SubscribeResponse/wsnt:TerminationTime
674 This OPTIONAL component SHOULD be returned if the SubscriptionManager uses
675 scheduled termination from [WS-ResourceLifetime]. The value of this component is the value
676 of the TerminationTime resource property of the Subscription at the time the response
677 message is created.
678 If the NotificationProducer does not respond to the Subscribe request message with the
679 SubscribeResponse message, then it MUST send a fault. This specification defines the following
680 faults associated with failure to process the Subscribe request message:
681
682 ResourceUnknownFault
683 • The NotificationProducer is a WS-Resource, and the resource identified in the message
684 is not known to the Web service. This fault is specified by the WS-Resource [WS-
685 Resource] specification.
686 InvalidFilterFault
687 • The Subscribe message contained a filter that was not understood or supported by the
688 NotificationProducer.
689 TopicExpressionDialectUnknownFault
690 • The Subscribe message contained a TopicExpression filter having a dialect that was not
691 understood or supported by the NotificationProducer.
692 InvalidTopicExpressionFault
693 • The Subscribe message contained a TopicExpression filter where the contents of the
694 filter did not match the dialect specified.
695 TopicNotSupportedFault
696 • The Subscribe message contained a TopicExpression filter that referenced a topic that
697 was not supported by the NotificationProducer.
698 InvalidProducerPropertiesExpressionFault
699 • The Subscribe message contained a ProducerProperties filter that did not represent a
700 valid boolean expression.
701 InvalidMessageContentExpressionFault

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

724 4.2.1 Example SOAP Encoding of the Subscribe Message Exchange

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

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

```

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

```

757

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

```

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

```

777 4.3 GetCurrentMessage

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

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

787 The format of the GetCurrentMessage request message is:

```

788 ...
789 <wsnt:GetCurrentMessage>
790   <wsnt:Topic Dialect="xsd:anyURI">
791     {any} ?
792   </wsnt:Topic>
793   {any} *
794 </wsnt:GetCurrentMessage>
795 ...

```

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

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

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

799 /wsnt:GetCurrentMessage/wsnt:Topic

800 A TopicExpression that identifies exactly one Topic.

801 If the NotificationProducer successfully processes the GetCurrentMessage request, it MUST

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

```
803 ...  
804 <wsnt:GetCurrentMessageResponse>  
805   {any}  
806 </wsnt:GetCurrentMessageResponse>  
807 ...
```

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

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

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

811 /wsnt:GetCurrentMessageResponse/{any}

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

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

816 ResourceUnknownFault

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

820 TopicExpressionDialectUnknownFault

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

823 InvalidTopicExpressionFault

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

826 TopicNotSupportedFault

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

829 MultipleTopicsSpecifiedFault

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

831 NoCurrentMessageOnTopicFault

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

4.3.1 Example SOAP Encoding of the GetCurrentMessage Message Exchange

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

```
<s:Envelope ... >
  <s:Header>
    <wsa:Action>
      http://docs.oasis-open.org/wsn/bw-
2/NotificationProducer/GetCurrentMessageRequest
    </wsa:Action>
    ...
  </s:Header>
  <s:Body>
    <wsnt:GetCurrentMessage>
      <wsnt:Topic Dialect=
        "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
        npex:SomeTopic
      </wsnt:Topic>
    </wsnt:GetCurrentMessage>
  </s:Body>
</s:Envelope>
```

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

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

5 Pull-Style Notification

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

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

5.1 PullPoint Interface

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

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

5.1.1 Accumulating Notification Messages

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

5.1.2 GetMessages

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

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

```
<wsnt:GetMessages>
  <wsnt:MaximumNumber>xsd:nonNegativeInteger</wsnt:MaximumNumber> ?
  {any} *
</wsnt:GetMessages>
```

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

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

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

/wsnt:GetMessages

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

/wsnt:GetMessages/wsnt:MaximumNumber

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

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

```
<wsnt:GetMessagesResponse>
  <wsnt:NotificationMessage /> *
  {any} *
</wsnt:GetMessagesResponse>
```

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

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

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

/wsnt:GetMessagesResponse

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

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

954 /wsnt:GetMessagesResponse/wsnt:NotificationMessage

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

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

963 ResourceUnknownFault

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

967 UnableToGetMessagesFault

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

969 5.1.3 Example SOAP Encoding of the GetMessages Message 970 Exchange

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

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

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

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

```

998     <wsnt:NotificationMessage>
999         ...
1000     </wsnt:NotificationMessage>
1001 </wsnt:GetMessagesResponse>
1002 </s:Body>
1003 </s:Envelope>

```

1004 **5.1.4 DestroyPullPoint**

1005 The PullPoint interface provides a destroy operation, providing a means by which a requestor can
1006 terminate the PullPoint resource. The DestroyPullPoint request message has the following form:

```

1007
1008 <wsnt:DestroyPullPoint>
1009     {any} *
1010 </wsnt:DestroyPullPoint>
1011

```

1012 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1013 <http://docs.oasis-open.org/wsn/bw-2/PullPoint/DestroyPullPointRequest>.

1014 Upon receipt of the DestroyPullPoint request, the PullPoint MUST attempt to destroy itself. If the
1015 DestroyPullPoint request message is successfully processed, the PullPoint MUST respond with
1016 the following message:

```

1017
1018 <wsnt:DestroyPullPointResponse>
1019     {any} *
1020 </wsnt:DestroyPullPointResponse>
1021

```

1022 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1023 <http://docs.oasis-open.org/wsn/bw-2/PullPoint/DestroyPullPointResponse>.

1024 If the PullPoint does not respond to the DestroyPullPoint request message with the
1025 DestroyPullPointResponse message, then it MUST send a fault. This specification defines the
1026 following faults associated with failure to process the DestroyPullPoint request message:

1027 ResourceUnknownFault

- 1028 • The PullPoint is acting as a WS-Resource, and the resource identified in the request
1029 message is not known to the Web service. This fault is specified by the WS-Resource
1030 [WS-Resource] specification.

1031

1032 UnableToDestroyPullPointFault.

- 1033 • The PullPoint was unable to destroy the PullPoint resource for some reason.

1034 **5.1.5 Example SOAP Encoding of the DestroyPullPoint Message** 1035 **Exchange**

1036 The following is a non-normative example of a DestroyPullPoint request message using SOAP:

```

1037 <s:Envelope ... >
1038     <s:Header>

```

```

1039     <wsa:Action>
1040         http://docs.oasis-open.org/wsn/bw-
1041 2/PullPoint/DestroyPullPointRequest
1042     </wsa:Action>
1043     ...
1044 </s:Header>
1045 <s:Body>
1046     <wsnt:DestroyPullPoint/>
1047 </s:Body>
1048 </s:Envelope>

```

1049 The following is a non-normative example of a DestroyPullPoint response message using SOAP:

```

1050 <s:Envelope ... >
1051   <s:Header>
1052     <wsa:Action>
1053       http://docs.oasis-open.org/wsn/bw-
1054 2/PullPoint/DestroyPullPointResponse
1055     </wsa:Action>
1056     ...
1057   </s:Header>
1058   <s:Body>
1059     <wsnt:DestroyPullPointResponse/>
1060   </s:Body>
1061 </s:Envelope>

```

1062 5.2 Create PullPoint Interface

1063 This specification describes a CreatePullPoint interface standardizing the means by which a
 1064 PullPoint resource is created.

1065 If a requestor wishes to create a new PullPoint resource, it MUST send a CreatePullPoint request
 1066 to an endpoint supporting the PullPoint interface. The CreatePullPoint request message has the
 1067 following form:

```

1068 <wsnt:CreatePullPoint>
1069   {any} *
1070 </wsnt:CreatePullPoint>
1072

```

1073 The WS-Addressing [action] Message Addressing Property MUST contain the URI
 1074 http://docs.oasis-open.org/wsn/bw-2/CreatePullPoint/CreatePullPointRequest

1075 Upon receipt of the CreatePullPoint request, the receiver MUST attempt to create a new PullPoint
 1076 resource. If the CreatePullPoint request message is successfully processed, the receiver MUST
 1077 respond with the following message:

```

1078 <wsnt:CreatePullPointResponse>
1079   <wsnt:PullPoint>wsa:EndpointReferenceType</wsnt:PullPoint>
1080   {any} *
1081 </wsnt:CreatePullPointResponse>
1083

```

1084 The WS-Addressing [action] Message Addressing Property MUST contain the URI

11085 <http://docs.oasis-open.org/wsn/bw-2/CreatePullPoint/CreatePullPointResponse>.
11086 The contents of the CreatePullPoint response message are further described as follows:
11087 /wsnt:CreatePullPointResponse/wsnt:PullPoint
11088 This component is an EndpointReference, as defined by WS-Addressing, that is a
11089 reference to the PullPoint resource created during the processing of the CreatePullPoint
11090 request message.

11091 A PullPoint created by the CreatePullPoint operation MAY choose to ignore Notifications received
11092 through the NotificationConsumer interface. A Pull Point MAY accumulate Notifications through
11093 other, implementation-defined mechanisms. Pull point factories SHOULD indicate through policy
11094 assertions whether pull points they create will ignore Notifications received through the
11095 NotificationConsumer interface.

11096 If the receiver does not respond to the CreatePullPoint request message with the
11097 CreatePullPointResponse message, then it MUST send a fault. This specification defines the
11098 following fault associated with failure to process the DestroyPullPoint request message:

- 11099 • UnableToCreatePullPointFault.

11100

11101 **5.2.1 Example SOAP Encoding of the CreatePullPoint Message** 11102 **Exchange**

11103 The following is a non-normative example of a CreatePullPoint request message using SOAP:

```
11104 <s:Envelope ... >  
11105   <s:Header>  
11106     <wsa:Action>  
11107       http://docs.oasis-open.org/wsn/bw-2/PullPoint/CreatePullPointRequest  
11108     </wsa:Action>  
11109     ...  
11110   </s:Header>  
11111   <s:Body>  
11112     <wsnt:CreatePullPoint/>  
11113   </s:Body>  
11114 </s:Envelope>
```

11115 The following is a non-normative example of a CreatePullPoint response message using SOAP:

```
11116 <s:Envelope ... >  
11117   <s:Header>  
11118     <wsa:Action>  
11119       http://docs.oasis-open.org/wsn/bw-2/PullPoint/CreatePullPointResponse  
11120     </wsa:Action>  
11121     ...  
11122   </s:Header>  
11123   <s:Body>  
11124     <wsnt:CreatePullPointResponse>  
11125       <wsnt:PullPoint>  
11126         <wsa:Address>...</wsa:Address>  
11127         ...  
11128       </wsnt:PullPoint>  
11129     </wsnt:CreatePullPointResponse>  
11130   </s:Body>  
11131 </s:Envelope>
```

6 SubscriptionManager Interface

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

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

6.1 Base Subscription Manager

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

6.1.1 Renew

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

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

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

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

/wsnt:Renew/wsnt:TerminationTime

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

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

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

1170 If the SubscriptionManager is unable or unwilling to reset the termination time component of
1171 the Subscription resource to the given value of the “absolute time” form or a value greater,
1172 then the Renew request MUST fault. If the value is not “in the future” relative to the current
1173 time as known by the SubscriptionManager, the Renew request MUST fault.

1174 The use of the xsi:nil attribute with value “true” indicates there is no scheduled termination
1175 time requested for the Subscription, implying that the Subscription should be renewed to
1176 have an infinite duration. If the element does not include the time zone designation, the value
1177 of the element MUST be interpreted as universal time (UTC) time.

1178 If the SubscriptionManager successfully processes the Renew request message, it MUST
1179 respond with a RenewResponse message. This response has the following form:

```
1180 ...  
1181 <wsnt:RenewResponse>  
1182   <wsnt:TerminationTime>xsd:dateTime</wsnt:TerminationTime>  
1183   <wsnt:CurrentTime>xsd:dateTime</wsnt:CurrentTime>?  
1184   {any} *  
1185 </wsnt:RenewResponse>  
1186 ...
```

1187 The WS-Addressing [action] Message Addressing Property MUST contain the URI
1188 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewResponse>.

1189 The contents of the RenewResponse message are further described as follows:

1190 /wsnt:RenewResponse/wsnt:TerminationTime

1191 The value of this component is the value of the termination time component of the
1192 Subscription at the time the response message is created.

1193 /wsnt:RenewResponse/wsnt:CurrentTime

1194 This OPTIONAL component contains the value of the CurrentTime according to the
1195 SubscriptionManager at the time the response message is created.

1196 If the SubscriptionManager does not respond to the Renew request message with the
1197 RenewResponse message, then it MUST send a fault. This specification defines the following
1198 faults associated with failure to process the Renew request message:

1199 ResourceUnknownFault

- 1200 • The SubscriptionManager is acting as a WS-Resource, and the resource identified in the
1201 message is not known to the Web service. This fault is specified by the WS-Resource
1202 [WS-Resource] specification.

1203 UnacceptableTerminationTimeFault

- 1204 • The value of TerminationTime specified in the Renew message was not acceptable to the
1205 NotificationProducer. The NotificationProducer MAY include a hint in the fault message
1206 indicating acceptable values for InitialTerminationTime.

1207 **6.1.1.1 Example SOAP Encoding of the Renew Message Exchange**

1208 The following is a non-normative example of a Renew request message using SOAP:

```
1209 <s:Envelope ... >  
1210   <s:Header>  
1211     <wsa:Action>
```

```

1212 http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewRequest
1213 </wsa:Action>
1214 ...
1215 </s:Header>
1216 <s:Body>
1217   <wsnt:Renew>
1218     <wsnt:TerminationTime>P1D</wsnt:TerminationTime>
1219   </wsnt:Renew>
1220 </s:Body>
1221 </s:Envelope>

```

1222 The following is a non-normative example of a Renew response message using SOAP:

```

1223 <s:Envelope ... >
1224   <s:Header>
1225     <wsa:Action>
1226       http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewResponse
1227     </wsa:Action>
1228     ...
1229   </s:Header>
1230   <s:Body>
1231     <wsnt:RenewResponse>
1232       <wsnt:TerminationTime>
1233         2005-12-26T00:00:00.000000Z
1234       </wsnt:TerminationTime>
1235     </wsnt:RenewResponse>
1236   </s:Body>
1237 </s:Envelope>

```

1238 6.1.2 Unsubscribe

1239 To terminate a Subscription, a requestor sends an Unsubscribe request message to the
 1240 SubscriptionManager. The Unsubscribe request message MUST have the following form:

```

1241 ...
1242 <wsnt:Unsubscribe>
1243   {any} *
1244 </wsnt:Unsubscribe>
1245 ...

```

1246 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1247 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeRequest>.

1248 Upon receipt of the Unsubscribe request, the SubscriptionManager MUST attempt to destroy the
 1249 Subscription resource. If the SubscriptionManager successfully processes the Unsubscribe
 1250 request message, it MUST respond with the following message:

```

1251 ...
1252 <wsnt:UnsubscribeResponse>
1253   {any} *
1254 </wsnt:UnsubscribeResponse>
1255 ...

```

1256 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1257 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeResponse>.

1258 If the SubscriptionManager does not respond to the Unsubscribe request message with the
1259 UnsubscribeResponse message, then it MUST send a fault. This specification defines the
1260 following faults associated with failure to process the Unsubscribe request message:
1261 ResourceUnknownFault

- 1262 • The SubscriptionManager is acting as a WS-Resource, and the resource identified in the
1263 message is not known to the Web service. This fault is specified by the WS-Resource
1264 [WS-Resource] specification.

1265 UnableToDestroySubscriptionFault

- 1266 • The SubscriptionManager was unable to destroy the Subscription resource for some
1267 reason.

6.1.2.1 Example SOAP Encoding of the Unsubscribe Message Exchange

1268
1269 The following is a non-normative example of an Unsubscribe request message using SOAP:

```
1271 <s:Envelope ... >  
1272   <s:Header>  
1273     <wsa:Action>  
1274     http://docs.oasis-open.org/wsn/bw-  
1275     2/SubscriptionManager/UnsubscribeRequest  
1276     </wsa:Action>  
1277     ...  
1278   </s:Header>  
1279   <s:Body>  
1280     <wsnt:Unsubscribe />  
1281   </s:Body>  
1282 </s:Envelope>
```

1283 The following is a non-normative example of an Unsubscribe response message using SOAP:

```
1284 <s:Envelope ... >  
1285   <s:Header>  
1286     <wsa:Action>  
1287     http://docs.oasis-open.org/wsn/bw-  
1288     2/SubscriptionManager/UnsubscribeResponse  
1289     </wsa:Action>  
1290     ...  
1291   </s:Header>  
1292   <s:Body>  
1293     <wsnt:UnsubscribeResponse />  
1294   </s:Body>  
1295 </s:Envelope>
```

6.2 Pausable Subscription Manager

1296
1297 A Pausable Subscription Manager implements all the message exchanges defined in the Base
1298 Subscription Manager and augments this with a set of advanced capabilities to allow third parties
1299 to pause and resume subscriptions.

6.2.1 PauseSubscription

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

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

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

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

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

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

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

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

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

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

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

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

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

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

ResourceUnknownFault

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

PauseFailedFault

- 1343
- The Pause operation could not be performed on the Subscription.

1344 **6.2.2 Example SOAP Encoding of the PauseSubscription Message**
1345 **Exchange**

1346 The following is a non-normative example of a PauseSubscription request message using SOAP:

```
1347 <s:Envelope ... >
1348   <s:Header>
1349     <wsa:Action>
1350       http://docs.oasis-open.org/wsn/bw-
1351 2/SubscriptionManager/PauseSubscriptionRequest
1352     </wsa:Action>
1353     ...
1354   </s:Header>
1355   <s:Body>
1356     <wsnt:PauseSubscription/>
1357   </s:Body>
1358 </s:Envelope>
```

1359 The following is a non-normative example of a PauseSubscription response message using
1360 SOAP:

```
1361 <s:Envelope ... >
1362   <s:Header>
1363     <wsa:Action>
1364       http://docs.oasis-open.org/wsn/bw-
1365 2/SubscriptionManager/PauseSubscriptionResponse
1366     </wsa:Action>
1367     ...
1368   </s:Header>
1369   <s:Body>
1370     <wsnt:PauseSubscriptionResponse/>
1371   </s:Body>
1372 </s:Envelope>
```

1373 **6.3 ResumeSubscription**

1374 If a requestor wishes to resume the production of Notifications on the given Subscription, it must
1375 send a ResumeSubscription request message. The ResumeSubscription request message has
1376 the following form:

```
1377 ...
1378 <wsnt:ResumeSubscription>
1379   {any} *
1380 </wsnt:ResumeSubscription>
1381 ...
```

1382 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1383 <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/ResumeSubscriptionRequest>.

1384 Upon successful processing of this message the Subscription is no longer in the *paused* state.

1385 If a Subscription is not in the *paused* state, and it is the subject of a ResumeSubscription request
1386 message, no change occurs.

1387 If the SubscriptionManager accepts the ResumeSubscription request message, it MUST respond
1388 with a message of the following form:

```
1389 ...  
1390 <wsnt:ResumeSubscriptionResponse>  
1391 {any} *  
1392 </wsnt:ResumeSubscriptionResponse>  
1393 ...
```

1394 The WS-Addressing [action] Message Addressing Property MUST contain the URI

1395 [http://docs.oasis-open.org/wsn/bw-](http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/ResumeSubscriptionResponse)
1396 [2/SubscriptionManager/ResumeSubscriptionResponse](http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/ResumeSubscriptionResponse).

1397 If the SubscriptionManager does not respond to the ResumeSubscriptionMessage request
1398 message with the ResumeSubscriptionResponse message, then it MUST send a fault. This
1399 specification defines the following faults associated with failure to process the
1400 ResumeSubscriptionMessage request message:

1401 ResourceUnknownFault

- 1402 • The resource identified in the message is not known to the Web service. This fault is
1403 specified by the WS-Resource [WS-Resource] specification.

1404 ResumeFailedFault

- 1405 • The Resume operation could not be performed on the Subscription.

1406 **6.3.1 Example SOAP Encoding of the ResumeSubscription Message** 1407 **Exchange**

1408 The following is a non-normative example of a ResumeSubscription request message using
1409 SOAP:

```
1410 <s:Envelope ... >  
1411 <s:Header>  
1412 <wsa:Action>  
1413 http://docs.oasis-open.org/wsn/bw-  
1414 2/SubscriptionManager/ResumeSubscriptionRequest  
1415 </wsa:Action>  
1416 ...  
1417 </s:Header>  
1418 <s:Body>  
1419 <wsnt:ResumeSubscription/>  
1420 </s:Body>  
1421 </s:Envelope>
```

1422 The following is a non-normative example of a ResumeSubscription response message using
1423 SOAP:

```
1424 <s:Envelope ... >  
1425 <s:Header>  
1426 <wsa:Action>  
1427 http://docs.oasis-open.org/wsn/bw-  
1428 2/SubscriptionManager/ResumeSubscriptionResponse  
1429 </wsa:Action>  
1430 ...  
1431 </s:Header>  
1432 <s:Body>
```

```
1433     <wsnt:ResumeSubscriptionResponse/>
1434   </s:Body>
1435 </s:Envelope>
```

1436 6.4 Subscriptions as WS-Resources

1437 Either Base Subscription Managers or Pausable Subscription Managers MAY expose
1438 subscriptions as WS-Resources. A Subscription Manager that exposes Subscriptions as WS-
1439 Resources MUST support the required message exchanges associated with the WS-
1440 ResourceProperties specification [WS-ResourceProperties] and MAY support the OPTIONAL
1441 message exchanges defined by WS-ResourceProperties.

1442 If a SubscriptionManager exposes Subscriptions as WS-Resources then these WS-Resources
1443 MUST support the immediate termination interface defined by Web Services Resource Lifetime
1444 [WS-ResourceLifetime] and MAY support the scheduled termination interface defined by Web
1445 Services Resource Lifetime. Such Subscription Managers MAY support other means of
1446 destroying Subscriptions

1447 A Subscription Manager MUST declare to requestors (for example in a WSDL definition) which
1448 means of Subscription destruction are supported.

1449 6.4.1 Subscription Resource Properties

1450 In cases where subscriptions are exposed as WS-Resources, this specification defines several
1451 resource properties, which MUST conform to the following schema fragment for content:.

```
1452 ...
1453     targetNamespace="http://docs.oasis-open.org/wsn/b-2"
1454 ...
1455
1456     <xsd:element name="ConsumerReference"
1457                 type="wsa:EndpointReferenceType" />
1458     <xsd:element name="Filter"
1459                 type="wsnt:FilterType" />
1460     <xsd:element name="SubscriptionPolicy"
1461                 type="wsnt:SubscriptionPolicyType" />
1462     <xsd:element name="CreationTime"
1463                 type="xsd:dateTime" />
```

1464 These properties must also conform to the following schema fragment for cardinality:

```
1465     <xsd:element ref="wsnt:ConsumerReference"
1466                 minOccurs="1" maxOccurs="1" />
1467     <xsd:element ref="wsnt:Filter"
1468                 minOccurs="0" maxOccurs="1" />
1469     <xsd:element ref="wsnt:SubscriptionPolicy"
1470                 minOccurs="0" maxOccurs="1" />
1471     <xsd:element ref="wsnt:CreationTime"
1472                 minOccurs="0" maxOccurs="1" />
```

1473 These Resource Property elements are further described as follows:

1474 /wsnt:ConsumerReference and /wsnt:Filter and /wsnt:SubscriptionPolicy

1475 These elements are as defined in the description of the Subscribe request message (see
1476 4.2). SubscriptionManagers MAY prohibit any combination of these properties from being set.

1477 If setting a given property is disallowed, the SubscriptionManager MUST fault on receiving a
1478 request to change it. SubscriptionManagers SHOULD describe using metadata techniques
1479 defined by WS-ResourceFramework which properties may be set.

1480 If any of these properties is settable, it is implementation dependent whether production of
1481 notifications will be attempted without interruption when a change is made. The
1482 SubscriptionManager may choose to treat the change in properties as equivalent to a
1483 cancellation followed by the creation of a new subscription. In this case, messages may
1484 potentially be lost in the interim.

1485 A SubscriptionManager MAY choose to provide a stronger guarantee of continuity of
1486 notification production, and if so it SHOULD advertise this capability, for example in a policy
1487 assertion. In the absence of any such policy assertion, a subscriber SHOULD NOT make
1488 any assumption of continuity.

1489 /wsnt:CreationTime

1490 Indicates the date and time at which the Subscription was created. This component MAY be
1491 omitted, for example by resource constrained devices which cannot associate a creation time
1492 with subscription resources they create. This property is read-only. SubscriptionManagers
1493 MUST fault on receiving a request to change it.

1494 If the SubscriptionManager also accepts the SetResourceProperties request message as defined
1495 in WS-ResourceProperties, the following properties MAY be modified by the requestor:
1496 /wsnt:ConsumerReference and /wsnt:Filter and /wsnt:SubscriptionPolicy

7 Security Considerations

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

7.1 Securing the Message Exchanges

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

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

7.2 Securing Subscriptions and Notifications

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

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

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

Given that WS-BaseNotification may use WS-ResourceProperties and WS-ResourceLifetime, the security considerations outlined in WS- those 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.

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

1538 In addition to the usual concerns of authorization and message integrity which apply to all web
1539 services, notification presents issues all its own due to the third-party nature of subscription.
1540 Since the NotificationProducer is agreeing to produce Notifications for a consumer based on the
1541 requests of a Subscriber, it must assure itself that there is no harm in producing these
1542 Notifications. A malicious Subscriber may request Notifications be sent to a party that is not
1543 authorized to receive them. It may also mount DOS attacks by requesting large volumes of
1544 Notifications be sent to parties that cannot handle them.
1545
1546 The NotificationProducer may address these risks in many different ways, including but not
1547 limited to:

- 1548 • Simply trusting all Subscribers, perhaps because all parties are known to be on a closed,
1549 trusted, network, or because the consequences of unauthorized Subscriptions are
1550 otherwise known to be negligible.
- 1551 • Requiring all Subscribers to provide secure credentials proving that they are trusted to
1552 make subscriptions.
- 1553 • Refusing to produce notifications for NotificationConsumers that are not known to be
1554 authorized.
- 1555 • Explicitly confirming with NotificationConsumers whether they wish to receive the
1556 Notifications that the Subscriber has requested.
- 1557 • Some combination of the above, depending on the identity of the Subscriber and
1558 NotificationProducer

1559 NotificationProducers SHOULD advertise, whether through policy assertions or other means,
1560 what security measures they take.

8 References

8.1 Normative

[RFC2119]

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

[URI]

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

[WS-Addressing]

"Web Services Addressing 1.0 - Core", W3C Recommendation.
<http://www.w3.org/TR/ws-addr-core>

[WS-BrokeredNotification]

"Web Services Brokered Notification 1.3", OASIS Standard.
http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf

[WS-Resource]

"Web Services Resource 1.2", OASIS Standard.
http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-os.pdf

[WS-ResourceLifetime]

"Web Services Resource Lifetime 1.2", OASIS Standard.
http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-os.pdf

[WS-ResourceProperties]

"Web Services Resource Properties 1.2", OASIS Standard.
http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-os.pdf

[WS-BaseFaults]

"Web Services Base Faults 1.2", OASIS Standard.
http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf

[WS-Topics]

"Web Services Topics 1.3", OASIS Standard.
http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf

[XML-InfoSet]

"XML Information Set", W3C Recommendation.
<http://www.w3.org/TR/xml-infoset/>

[XPath]

"XML Path Language (XPath) Version 1.0", W3C Recommendation.
<http://www.w3.org/TR/xpath>

8.2 Non-Normative

[SOAP 1.1]

"Simple Object Access Protocol (SOAP) 1.1"
<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

1601 **[SOAP 1.2]**
1602 “SOAP Version 1.2 Part 1: Messaging Framework”, W3C Recommendation.
1603 <http://www.w3.org/TR/soap12-part1/>
1604 **[WS-Security]**
1605 “Web Services Security: SOAP Message Security 1.0”, OASIS Standard.
1606 [http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf)
1607 [1.0.pdf](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf)

Appendix A. Acknowledgments

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

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

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

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

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

Appendix B. XML Schema

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

```
<?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 (2004-2006). 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.

-->

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

```

1683 xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
1684 xmlns:wsa="http://www.w3.org/2005/08/addressing"
1685 xmlns:wsrf-bf="http://docs.oasis-open.org/wsrf/bf-2"
1686 xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"
1687 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1688 elementFormDefault="qualified" attributeFormDefault="unqualified">
1689
1690 <!-- ===== Imports ===== -->
1691
1692 <xsd:import namespace="http://www.w3.org/2005/08/addressing"
1693           schemaLocation="http://www.w3.org/2005/08/addressing/ws-
1694 addr.xsd"
1695           />
1696
1697 <xsd:import namespace="http://docs.oasis-open.org/wsrf/bf-2"
1698           schemaLocation="http://docs.oasis-open.org/wsrf/bf-2.xsd"
1699           />
1700 <xsd:import namespace="http://docs.oasis-open.org/wsn/t-1"
1701           schemaLocation="http://docs.oasis-open.org/wsn/t-1.xsd"
1702           />
1703
1704 <!-- ===== Misc. Helper Types ===== -->
1705
1706 <xsd:complexType name="QueryExpressionType" mixed="true">
1707   <xsd:sequence>
1708     <xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
1709   </xsd:sequence>
1710   <xsd:attribute name="Dialect" type="xsd:anyURI" use="required"/>
1711 </xsd:complexType>
1712
1713 <xsd:complexType name="TopicExpressionType" mixed="true">
1714   <xsd:sequence>
1715     <xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
1716   </xsd:sequence>
1717   <xsd:attribute name="Dialect" type="xsd:anyURI" use="required" />
1718   <xsd:anyAttribute/>
1719 </xsd:complexType>
1720
1721 <xsd:complexType name="FilterType">
1722   <xsd:sequence>
1723     <xsd:any minOccurs="0" maxOccurs="unbounded"/>
1724   </xsd:sequence>
1725 </xsd:complexType>
1726
1727 <xsd:complexType name="SubscriptionPolicyType">
1728   <xsd:sequence>
1729     <xsd:any minOccurs="0" maxOccurs="unbounded"
1730 processContents="lax"/>
1731   </xsd:sequence>
1732 </xsd:complexType>
1733
1734 <!-- ===== Resource Property Related ===== -->
1735 <!-- ===== Resource Properties for NotificationProducer ===== -->
1736 <xsd:element name="TopicExpression" type="wsnt:TopicExpressionType"/>
1737 <xsd:element name="FixedTopicSet" type="xsd:boolean" default="true"/>
1738 <xsd:element name="TopicExpressionDialect" type="xsd:anyURI"/>
1739

```

```

1740 <xsd:element name="NotificationProducerRP">
1741   <xsd:complexType>
1742     <xsd:sequence>
1743       <xsd:element ref="wsnt:TopicExpression"
1744         minOccurs="0" maxOccurs="unbounded" />
1745       <xsd:element ref="wsnt:FixedTopicSet"
1746         minOccurs="0" maxOccurs="1" />
1747       <xsd:element ref="wsnt:TopicExpressionDialect"
1748         minOccurs="0" maxOccurs="unbounded" />
1749       <xsd:element ref="wstop:TopicSet"
1750         minOccurs="0" maxOccurs="1" />
1751     </xsd:sequence>
1752   </xsd:complexType>
1753 </xsd:element>
1754
1755 <!-- ===== Resource Properties for SubscriptionManager ===== -->
1756 <xsd:element name="ConsumerReference"
1757   type="wsa:EndpointReferenceType" />
1758 <xsd:element name="Filter" type="wsnt:FilterType" />
1759 <xsd:element name="SubscriptionPolicy"
1760   type="wsnt:SubscriptionPolicyType" />
1761
1762
1763 <xsd:element name="CreationTime" type="xsd:dateTime" />
1764
1765 <xsd:element name="SubscriptionManagerRP" >
1766   <xsd:complexType>
1767     <xsd:sequence>
1768       <xsd:element ref="wsnt:ConsumerReference"
1769         minOccurs="1" maxOccurs="1" />
1770       <xsd:element ref="wsnt:Filter"
1771         minOccurs="0" maxOccurs="1" />
1772       <xsd:element ref="wsnt:SubscriptionPolicy"
1773         minOccurs="0" maxOccurs="1" />
1774       <xsd:element ref="wsnt:CreationTime"
1775         minOccurs="0" maxOccurs="1" />
1776     </xsd:sequence>
1777   </xsd:complexType>
1778 </xsd:element>
1779
1780 <!-- ===== Notification Metadata ===== -->
1781 <xsd:element name="SubscriptionReference"
1782   type="wsa:EndpointReferenceType" />
1783 <xsd:element name="Topic"
1784   type="wsnt:TopicExpressionType" />
1785 <xsd:element name="ProducerReference"
1786   type="wsa:EndpointReferenceType" />
1787
1788 <!-- ===== Message Helper Types ===== -->
1789 <xsd:complexType name="NotificationMessageHolderType" >
1790   <xsd:sequence>
1791     <xsd:element ref="wsnt:SubscriptionReference"
1792       minOccurs="0" maxOccurs="1" />
1793     <xsd:element ref="wsnt:Topic"
1794       minOccurs="0" maxOccurs="1" />
1795     <xsd:element ref="wsnt:ProducerReference"
1796       minOccurs="0" maxOccurs="1" />

```

```

1797     <xsd:element name="Message">
1798         <xsd:complexType>
1799             <xsd:sequence>
1800                 <xsd:any namespace="##any" processContents="lax"
1801                     minOccurs="1" maxOccurs="1"/>
1802             </xsd:sequence>
1803         </xsd:complexType>
1804     </xsd:element>
1805 </xsd:sequence>
1806 </xsd:complexType>
1807 <xsd:element name="NotificationMessage"
1808     type="wsnt:NotificationMessageHolderType"/>
1809
1810 <!-- ===== Message Types for NotificationConsumer ===== -->
1811 <xsd:element name="Notify" >
1812     <xsd:complexType>
1813         <xsd:sequence>
1814             <xsd:element ref="wsnt:NotificationMessage"
1815                 minOccurs="1" maxOccurs="unbounded" />
1816             <xsd:any namespace="##other" processContents="lax"
1817                 minOccurs="0" maxOccurs="unbounded"/>
1818         </xsd:sequence>
1819     </xsd:complexType>
1820 </xsd:element>
1821
1822 <!-- ===== Message Types for NotificationProducer ===== -->
1823
1824     <xsd:simpleType name="AbsoluteOrRelativeTimeType">
1825         <xsd:union memberTypes="xsd:dateTime xsd:duration" />
1826     </xsd:simpleType>
1827
1828     <xsd:element name="CurrentTime" type="xsd:dateTime" />
1829
1830     <xsd:element name="TerminationTime"
1831         nillable="true" type="xsd:dateTime" />
1832
1833     <xsd:element name="ProducerProperties"
1834         type="wsnt:QueryExpressionType" />
1835
1836     <xsd:element name="MessageContent"
1837         type="wsnt:QueryExpressionType" />
1838
1839     <xsd:element name="UseRaw"><xsd:complexType/></xsd:element>
1840
1841     <xsd:element name="Subscribe" >
1842         <xsd:complexType>
1843             <xsd:sequence>
1844                 <xsd:element name="ConsumerReference"
1845                     type="wsa:EndpointReferenceType"
1846                     minOccurs="1" maxOccurs="1" />
1847                 <xsd:element name="Filter"
1848                     type="wsnt:FilterType"
1849                     minOccurs="0" maxOccurs="1" />
1850                 <xsd:element name="InitialTerminationTime"
1851                     type="wsnt:AbsoluteOrRelativeTimeType"
1852                     nillable="true"
1853                     minOccurs="0" maxOccurs="1" />

```

```

1854         <xsd:element name="SubscriptionPolicy"
1855             minOccurs="0" maxOccurs="1">
1856             <xsd:complexType>
1857                 <xsd:sequence>
1858                     <xsd:any namespace="##any" processContents="lax"
1859                         minOccurs="0" maxOccurs="unbounded"/>
1860                 </xsd:sequence>
1861             </xsd:complexType>
1862         </xsd:element>
1863         <xsd:any namespace="##other" processContents="lax"
1864             minOccurs="0" maxOccurs="unbounded"/>
1865     </xsd:sequence>
1866 </xsd:complexType>
1867 </xsd:element>
1868
1869 <xsd:element name="SubscribeResponse">
1870     <xsd:complexType>
1871         <xsd:sequence>
1872             <xsd:element name="SubscriptionReference"
1873                 type="wsa:EndpointReferenceType"
1874                 minOccurs="1" maxOccurs="1" />
1875             <xsd:element ref="wsnt:CurrentTime"
1876                 minOccurs="0" maxOccurs="1" />
1877             <xsd:element ref="wsnt:TerminationTime"
1878                 minOccurs="0" maxOccurs="1" />
1879             <xsd:any namespace="##other" processContents="lax"
1880                 minOccurs="0" maxOccurs="unbounded"/>
1881         </xsd:sequence>
1882     </xsd:complexType>
1883 </xsd:element>
1884
1885 <xsd:element name="GetCurrentMessage">
1886     <xsd:complexType>
1887         <xsd:sequence>
1888             <xsd:element name="Topic"
1889                 type="wsnt:TopicExpressionType" />
1890             <xsd:any namespace="##other" processContents="lax"
1891                 minOccurs="0" maxOccurs="unbounded"/>
1892         </xsd:sequence>
1893     </xsd:complexType>
1894 </xsd:element>
1895
1896 <xsd:element name="GetCurrentMessageResponse">
1897     <xsd:complexType>
1898         <xsd:sequence>
1899             <xsd:any namespace="##other" processContents="lax"
1900                 minOccurs="0" maxOccurs="unbounded"/>
1901         </xsd:sequence>
1902     </xsd:complexType>
1903 </xsd:element>
1904
1905 <xsd:complexType name="SubscribeCreationFailedFaultType">
1906     <xsd:complexContent>
1907         <xsd:extension base="wsrf-bf:BaseFaultType"/>
1908     </xsd:complexContent>
1909 </xsd:complexType>
1910 <xsd:element name="SubscribeCreationFailedFault"

```

```

1911         type="wsnt:SubscribeCreationFailedFaultType"/>
1912
1913     <xsd:complexType name="InvalidFilterFaultType">
1914         <xsd:complexContent>
1915             <xsd:extension base="wsrf-bf:BaseFaultType">
1916                 <xsd:sequence>
1917                     <xsd:element name="UnknownFilter" type="xsd:QName"
1918                         minOccurs="1" maxOccurs="unbounded"/>
1919                 </xsd:sequence>
1920             </xsd:extension>
1921         </xsd:complexContent>
1922     </xsd:complexType>
1923     <xsd:element name="InvalidFilterFault"
1924         type="wsnt:InvalidFilterFaultType"/>
1925
1926     <xsd:complexType name="TopicExpressionDialectUnknownFaultType">
1927         <xsd:complexContent>
1928             <xsd:extension base="wsrf-bf:BaseFaultType"/>
1929         </xsd:complexContent>
1930     </xsd:complexType>
1931     <xsd:element name="TopicExpressionDialectUnknownFault"
1932         type="wsnt:TopicExpressionDialectUnknownFaultType"/>
1933
1934     <xsd:complexType name="InvalidTopicExpressionFaultType">
1935         <xsd:complexContent>
1936             <xsd:extension base="wsrf-bf:BaseFaultType"/>
1937         </xsd:complexContent>
1938     </xsd:complexType>
1939     <xsd:element name="InvalidTopicExpressionFault"
1940         type="wsnt:InvalidTopicExpressionFaultType"/>
1941
1942     <xsd:complexType name="TopicNotSupportedFaultType">
1943         <xsd:complexContent>
1944             <xsd:extension base="wsrf-bf:BaseFaultType"/>
1945         </xsd:complexContent>
1946     </xsd:complexType>
1947     <xsd:element name="TopicNotSupportedFault"
1948         type="wsnt:TopicNotSupportedFaultType"/>
1949
1950     <xsd:complexType name="MultipleTopicsSpecifiedFaultType">
1951         <xsd:complexContent>
1952             <xsd:extension base="wsrf-bf:BaseFaultType"/>
1953         </xsd:complexContent>
1954     </xsd:complexType>
1955     <xsd:element name="MultipleTopicsSpecifiedFault"
1956         type="wsnt:MultipleTopicsSpecifiedFaultType"/>
1957
1958     <xsd:complexType name="InvalidProducerPropertiesExpressionFaultType">
1959         <xsd:complexContent>
1960             <xsd:extension base="wsrf-bf:BaseFaultType"/>
1961         </xsd:complexContent>
1962     </xsd:complexType>
1963     <xsd:element name="InvalidProducerPropertiesExpressionFault"
1964         type="wsnt:InvalidProducerPropertiesExpressionFaultType"/>
1965
1966     <xsd:complexType name="InvalidMessageContentExpressionFaultType">
1967         <xsd:complexContent>

```

```

1968     <xsd:extension base="wsrf-bf:BaseFaultType"/>
1969     </xsd:complexContent>
1970 </xsd:complexType>
1971 <xsd:element name="InvalidMessageContentExpressionFault"
1972     type="wsnt:InvalidMessageContentExpressionFaultType"/>
1973
1974 <xsd:complexType name="UnrecognizedPolicyRequestFaultType">
1975     <xsd:complexContent>
1976         <xsd:extension base="wsrf-bf:BaseFaultType">
1977             <xsd:sequence>
1978                 <xsd:element name="UnrecognizedPolicy" type="xsd:QName"
1979                     minOccurs="0" maxOccurs="unbounded"/>
1980             </xsd:sequence>
1981         </xsd:extension>
1982     </xsd:complexContent>
1983 </xsd:complexType>
1984 <xsd:element name="UnrecognizedPolicyRequestFault"
1985     type="wsnt:UnrecognizedPolicyRequestFaultType"/>
1986
1987 <xsd:complexType name="UnsupportedPolicyRequestFaultType">
1988     <xsd:complexContent>
1989         <xsd:extension base="wsrf-bf:BaseFaultType">
1990             <xsd:sequence>
1991                 <xsd:element name="UnsupportedPolicy" type="xsd:QName"
1992                     minOccurs="0" maxOccurs="unbounded"/>
1993             </xsd:sequence>
1994         </xsd:extension>
1995     </xsd:complexContent>
1996 </xsd:complexType>
1997 <xsd:element name="UnsupportedPolicyRequestFault"
1998     type="wsnt:UnsupportedPolicyRequestFaultType"/>
1999
2000 <xsd:complexType name="NotifyMessageNotSupportedFaultType">
2001     <xsd:complexContent>
2002         <xsd:extension base="wsrf-bf:BaseFaultType"/>
2003     </xsd:complexContent>
2004 </xsd:complexType>
2005 <xsd:element name="NotifyMessageNotSupportedFault"
2006     type="wsnt:NotifyMessageNotSupportedFaultType"/>
2007
2008 <xsd:complexType name="UnacceptableInitialTerminationTimeFaultType">
2009     <xsd:complexContent>
2010         <xsd:extension base="wsrf-bf:BaseFaultType">
2011             <xsd:sequence>
2012                 <xsd:element name="MinimumTime" type="xsd:dateTime"/>
2013                 <xsd:element name="MaximumTime" type="xsd:dateTime"
2014                     minOccurs="0"/>
2015             </xsd:sequence>
2016         </xsd:extension>
2017     </xsd:complexContent>
2018 </xsd:complexType>
2019 <xsd:element name="UnacceptableInitialTerminationTimeFault"
2020     type="wsnt:UnacceptableInitialTerminationTimeFaultType"/>
2021
2022 <xsd:complexType name="NoCurrentMessageOnTopicFaultType">
2023     <xsd:complexContent>
2024         <xsd:extension base="wsrf-bf:BaseFaultType"/>

```

```

2025     </xsd:complexContent>
2026 </xsd:complexType>
2027 <xsd:element name="NoCurrentMessageOnTopicFault"
2028             type="wsnt:NoCurrentMessageOnTopicFaultType"/>
2029
2030 <!-- ===== Message Types for PullPoint ===== -->
2031 <xsd:element name="GetMessages">
2032     <xsd:complexType>
2033         <xsd:sequence>
2034             <xsd:element name="MaximumNumber"
2035                         type="xsd:nonNegativeInteger"
2036                         minOccurs="0"/>
2037             <xsd:any namespace="##other" processContents="lax"
2038                 minOccurs="0" maxOccurs="unbounded"/>
2039         </xsd:sequence>
2040         <xsd:anyAttribute/>
2041     </xsd:complexType>
2042 </xsd:element>
2043
2044 <xsd:element name="GetMessagesResponse">
2045     <xsd:complexType>
2046         <xsd:sequence>
2047             <xsd:element ref="wsnt:NotificationMessage"
2048                         minOccurs="0" maxOccurs="unbounded" />
2049             <xsd:any namespace="##other" processContents="lax"
2050                 minOccurs="0" maxOccurs="unbounded"/>
2051         </xsd:sequence>
2052         <xsd:anyAttribute/>
2053     </xsd:complexType>
2054 </xsd:element>
2055
2056 <xsd:element name="DestroyPullPoint">
2057     <xsd:complexType>
2058         <xsd:sequence>
2059             <xsd:any namespace="##other" processContents="lax"
2060                 minOccurs="0" maxOccurs="unbounded"/>
2061         </xsd:sequence>
2062         <xsd:anyAttribute/>
2063     </xsd:complexType>
2064 </xsd:element>
2065
2066 <xsd:element name="DestroyPullPointResponse">
2067     <xsd:complexType>
2068         <xsd:sequence>
2069             <xsd:any namespace="##other" processContents="lax"
2070                 minOccurs="0" maxOccurs="unbounded"/>
2071         </xsd:sequence>
2072         <xsd:anyAttribute/>
2073     </xsd:complexType>
2074 </xsd:element>
2075
2076 <xsd:complexType name="UnableToGetMessagesFaultType">
2077     <xsd:complexContent>
2078         <xsd:extension base="wsrf-bf:BaseFaultType"/>
2079     </xsd:complexContent>
2080 </xsd:complexType>
2081

```

```

2082     <xsd:element name="UnableToGetMessagesFault"
2083               type="wsnt:UnableToGetMessagesFaultType"/>
2084
2085   <xsd:complexType name="UnableToDestroyPullPointFaultType">
2086     <xsd:complexContent>
2087       <xsd:extension base="wsrf-bf:BaseFaultType"/>
2088     </xsd:complexContent>
2089   </xsd:complexType>
2090
2091   <xsd:element name="UnableToDestroyPullPointFault"
2092             type="wsnt:UnableToDestroyPullPointFaultType"/>
2093
2094   <!-- ===== Message Types for Create PullPoint ===== -->
2095   <xsd:element name="CreatePullPoint">
2096     <xsd:complexType>
2097       <xsd:sequence>
2098         <xsd:any namespace="##other" processContents="lax"
2099               minOccurs="0" maxOccurs="unbounded"/>
2100       </xsd:sequence>
2101       <xsd:anyAttribute/>
2102     </xsd:complexType>
2103   </xsd:element>
2104
2105   <xsd:element name="CreatePullPointResponse">
2106     <xsd:complexType>
2107       <xsd:sequence>
2108         <xsd:element name="PullPoint"
2109                   type="wsa:EndpointReferenceType"/>
2110         <xsd:any namespace="##other" processContents="lax"
2111               minOccurs="0" maxOccurs="unbounded"/>
2112       </xsd:sequence>
2113       <xsd:anyAttribute/>
2114     </xsd:complexType>
2115   </xsd:element>
2116
2117   <xsd:complexType name="UnableToCreatePullPointFaultType">
2118     <xsd:complexContent>
2119       <xsd:extension base="wsrf-bf:BaseFaultType"/>
2120     </xsd:complexContent>
2121   </xsd:complexType>
2122   <xsd:element name="UnableToCreatePullPointFault"
2123             type="wsnt:UnableToCreatePullPointFaultType"/>
2124
2125   <!-- ===== Message Types for Base SubscriptionManager ===== -->
2126   <xsd:element name="Renew">
2127     <xsd:complexType>
2128       <xsd:sequence>
2129         <xsd:element name="TerminationTime"
2130                   type="wsnt:AbsoluteOrRelativeTimeType"
2131                   nillable="true"
2132                   minOccurs="1" maxOccurs="1" />
2133         <xsd:any namespace="##other" processContents="lax"
2134               minOccurs="0" maxOccurs="unbounded"/>
2135       </xsd:sequence>
2136     </xsd:complexType>
2137   </xsd:element>
2138

```

```

2139 <xsd:element name="RenewResponse">
2140   <xsd:complexType>
2141     <xsd:sequence>
2142       <xsd:element ref="wsnt:TerminationTime"
2143         minOccurs="1" maxOccurs="1" />
2144       <xsd:element ref="wsnt:CurrentTime"
2145         minOccurs="0" maxOccurs="1" />
2146       <xsd:any namespace="##other" processContents="lax"
2147         minOccurs="0" maxOccurs="unbounded"/>
2148     </xsd:sequence>
2149   </xsd:complexType>
2150 </xsd:element>
2151
2152 <xsd:complexType name="UnacceptableTerminationTimeFaultType">
2153   <xsd:complexContent>
2154     <xsd:extension base="wsrf-bf:BaseFaultType">
2155       <xsd:sequence>
2156         <xsd:element name="MinimumTime" type="xsd:dateTime"/>
2157         <xsd:element name="MaximumTime" type="xsd:dateTime"
2158           minOccurs="0"/>
2159       </xsd:sequence>
2160     </xsd:extension>
2161   </xsd:complexContent>
2162 </xsd:complexType>
2163 <xsd:element name="UnacceptableTerminationTimeFault"
2164   type="wsnt:UnacceptableTerminationTimeFaultType"/>
2165
2166 <xsd:element name="Unsubscribe">
2167   <xsd:complexType>
2168     <xsd:sequence>
2169       <xsd:any namespace="##other" processContents="lax"
2170         minOccurs="0" maxOccurs="unbounded"/>
2171     </xsd:sequence>
2172   </xsd:complexType>
2173 </xsd:element>
2174
2175 <xsd:element name="UnsubscribeResponse">
2176   <xsd:complexType>
2177     <xsd:sequence>
2178       <xsd:any namespace="##other" processContents="lax"
2179         minOccurs="0" maxOccurs="unbounded"/>
2180     </xsd:sequence>
2181   </xsd:complexType>
2182 </xsd:element>
2183
2184 <xsd:complexType name="UnableToDestroySubscriptionFaultType">
2185   <xsd:complexContent>
2186     <xsd:extension base="wsrf-bf:BaseFaultType"/>
2187   </xsd:complexContent>
2188 </xsd:complexType>
2189 <xsd:element name="UnableToDestroySubscriptionFault"
2190   type="wsnt:UnableToDestroySubscriptionFaultType"/>
2191
2192 <!-- ===== Message Types for Pausable SubscriptionManager ===== -->
2193
2194   <xsd:element name="PauseSubscription">
2195     <xsd:complexType>

```

```

2196     <xsd:sequence>
2197         <xsd:any namespace="##other" processContents="lax"
2198             minOccurs="0" maxOccurs="unbounded"/>
2199     </xsd:sequence>
2200 </xsd:complexType>
2201 </xsd:element>
2202
2203 <xsd:element name="PauseSubscriptionResponse" >
2204     <xsd:complexType>
2205         <xsd:sequence>
2206             <xsd:any namespace="##other" processContents="lax"
2207                 minOccurs="0" maxOccurs="unbounded"/>
2208         </xsd:sequence>
2209     </xsd:complexType>
2210 </xsd:element>
2211
2212 <xsd:element name="ResumeSubscription">
2213     <xsd:complexType>
2214         <xsd:sequence>
2215             <xsd:any namespace="##other" processContents="lax"
2216                 minOccurs="0" maxOccurs="unbounded"/>
2217         </xsd:sequence>
2218     </xsd:complexType>
2219 </xsd:element>
2220
2221 <xsd:element name="ResumeSubscriptionResponse">
2222     <xsd:complexType>
2223         <xsd:sequence>
2224             <xsd:any namespace="##other" processContents="lax"
2225                 minOccurs="0" maxOccurs="unbounded"/>
2226         </xsd:sequence>
2227     </xsd:complexType>
2228 </xsd:element>
2229
2230 <xsd:complexType name="PauseFailedFaultType">
2231     <xsd:complexContent>
2232         <xsd:extension base="wsrf-bf:BaseFaultType"/>
2233     </xsd:complexContent>
2234 </xsd:complexType>
2235 <xsd:element name="PauseFailedFault"
2236     type="wsnt:PauseFailedFaultType"/>
2237
2238 <xsd:complexType name="ResumeFailedFaultType">
2239     <xsd:complexContent>
2240         <xsd:extension base="wsrf-bf:BaseFaultType"/>
2241     </xsd:complexContent>
2242 </xsd:complexType>
2243 <xsd:element name="ResumeFailedFault"
2244     type="wsnt:ResumeFailedFaultType"/>
2245
2246 </xsd:schema>

```

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.

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.

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

```

2299   xmlns:wsntw="http://docs.oasis-open.org/wsn/bw-2"
2300   xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
2301   xmlns:wsa="http://www.w3.org/2005/08/addressing"
2302   xmlns:wsrf-rw="http://docs.oasis-open.org/wsrf/rw-2"
2303   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2304   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
2305
2306   <!-- ===== Imports ===== -->
2307   <wsdl:import
2308       namespace="http://docs.oasis-open.org/wsrf/rw-2"
2309       location="http://docs.oasis-open.org/wsrf/rw-2.wsdl"/>
2310
2311   <!-- ===== Types Definitions ===== -->
2312   <wsdl:types>
2313       <xsd:schema>
2314           <xsd:import
2315               namespace="http://docs.oasis-open.org/wsn/b-2"
2316               schemaLocation="http://docs.oasis-open.org/wsn/b-2.xsd"/>
2317           </xsd:schema>
2318       </wsdl:types>
2319
2320   <!-- ===== NotificationConsumer::Notify ===== -->
2321   Notify(
2322       NotificationMessage
2323       (SubscriptionReference, TopicExpression, ProducerReference,
2324       Message)*
2325       returns: n/a (one way)
2326   -->
2327   <wsdl:message name="Notify">
2328       <wsdl:part name="Notify" element="wsnt:Notify"/>
2329   </wsdl:message>
2330
2331   <!-- ===== NotificationProducer::Subscribe ===== -->
2332   Subscribe(
2333       (ConsumerEndpointReference, [Filter], [SubscriptionPolicy],
2334       [InitialTerminationTime])
2335       returns: WS-Resource qualified EPR to a Subscription
2336   -->
2337   <wsdl:message name="SubscribeRequest" >
2338       <wsdl:part name="SubscribeRequest"
2339           element="wsnt:Subscribe"/>
2340   </wsdl:message>
2341
2342   <wsdl:message name="SubscribeResponse">
2343       <wsdl:part name="SubscribeResponse"
2344           element="wsnt:SubscribeResponse"/>
2345   </wsdl:message>
2346
2347   <wsdl:message name="SubscribeCreationFailedFault">
2348       <wsdl:part name="SubscribeCreationFailedFault"
2349           element="wsnt:SubscribeCreationFailedFault" />
2350   </wsdl:message>
2351
2352   <wsdl:message name="TopicExpressionDialectUnknownFault">
2353       <wsdl:part name="TopicExpressionDialectUnknownFault"
2354           element="wsnt:TopicExpressionDialectUnknownFault" />
2355   </wsdl:message>

```

```

2356
2357 <wsdl:message name="InvalidFilterFault">
2358   <wsdl:part name="InvalidFilterFault"
2359     element="wsnt:InvalidFilterFault" />
2360 </wsdl:message>
2361
2362 <wsdl:message name="InvalidProducerPropertiesExpressionFault">
2363   <wsdl:part name="InvalidProducerPropertiesExpressionFault"
2364     element="wsnt:InvalidProducerPropertiesExpressionFault" />
2365 </wsdl:message>
2366
2367 <wsdl:message name="InvalidMessageContentExpressionFault">
2368   <wsdl:part name="InvalidMessageContentExpressionFault"
2369     element="wsnt:InvalidMessageContentExpressionFault" />
2370 </wsdl:message>
2371
2372 <wsdl:message name="UnrecognizedPolicyRequestFault">
2373   <wsdl:part name="UnrecognizedPolicyRequestFault"
2374     element="wsnt:UnrecognizedPolicyRequestFault" />
2375 </wsdl:message>
2376
2377 <wsdl:message name="UnsupportedPolicyRequestFault">
2378   <wsdl:part name="UnsupportedPolicyRequestFault"
2379     element="wsnt:UnsupportedPolicyRequestFault" />
2380 </wsdl:message>
2381
2382 <wsdl:message name="NotifyMessageNotSupportedFault">
2383   <wsdl:part name="NotifyMessageNotSupportedFault"
2384     element="wsnt:NotifyMessageNotSupportedFault" />
2385 </wsdl:message>
2386
2387 <wsdl:message name="UnacceptableInitialTerminationTimeFault">
2388   <wsdl:part name="UnacceptableInitialTerminationTimeFault"
2389     element="wsnt:UnacceptableInitialTerminationTimeFault"/>
2390 </wsdl:message>
2391
2392 <!-- ===== NotificationProducer::GetCurrentMessage =====
2393   GetCurrentMessage(topicExpression)
2394   returns: a NotificationMessage (xsd:any)
2395 -->
2396 <wsdl:message name="GetCurrentMessageRequest">
2397   <wsdl:part name="GetCurrentMessageRequest"
2398     element="wsnt:GetCurrentMessage"/>
2399 </wsdl:message>
2400
2401 <wsdl:message name="GetCurrentMessageResponse">
2402   <wsdl:part name="GetCurrentMessageResponse"
2403     element="wsnt:GetCurrentMessageResponse"/>
2404 </wsdl:message>
2405
2406 <wsdl:message name="InvalidTopicExpressionFault">
2407   <wsdl:part name="InvalidTopicExpressionFault"
2408     element="wsnt:InvalidTopicExpressionFault" />
2409 </wsdl:message>
2410
2411 <wsdl:message name="TopicNotSupportedFault">
2412   <wsdl:part name="TopicNotSupportedFault"

```

```

2413         element="wsnt:TopicNotSupportedFault" />
2414     </wsdl:message>
2415
2416     <wsdl:message name="MultipleTopicsSpecifiedFault">
2417         <wsdl:part name="MultipleTopicsSpecifiedFault"
2418             element="wsnt:MultipleTopicsSpecifiedFault" />
2419     </wsdl:message>
2420
2421     <wsdl:message name="NoCurrentMessageOnTopicFault">
2422         <wsdl:part name="NoCurrentMessageOnTopicFault"
2423             element="wsnt:NoCurrentMessageOnTopicFault" />
2424     </wsdl:message>
2425
2426 <!-- ===== PullPoint::GetMessages =====
2427 GetMessages(MaximumNumber)
2428 returns: NotificationMessage list
2429 -->
2430     <wsdl:message name="GetMessagesRequest">
2431         <wsdl:part name="GetMessagesRequest"
2432             element="wsnt:GetMessages"/>
2433     </wsdl:message>
2434
2435     <wsdl:message name="GetMessagesResponse">
2436         <wsdl:part name="GetMessagesResponse"
2437             element="wsnt:GetMessagesResponse"/>
2438     </wsdl:message>
2439
2440     <wsdl:message name="UnableToGetMessagesFault">
2441         <wsdl:part name="UnableToGetMessagesFault"
2442             element="wsnt:UnableToGetMessagesFault"/>
2443     </wsdl:message>
2444
2445
2446 <!-- ===== PullPoint::DestroyPullPoint =====
2447 DestroyPullPoint()
2448 returns: void
2449 -->
2450     <wsdl:message name="DestroyPullPointRequest">
2451         <wsdl:part name="DestroyPullPointRequest"
2452             element="wsnt:DestroyPullPoint"/>
2453     </wsdl:message>
2454
2455     <wsdl:message name="DestroyPullPointResponse">
2456         <wsdl:part name="DestroyPullPointResponse"
2457             element="wsnt:DestroyPullPointResponse"/>
2458     </wsdl:message>
2459
2460     <wsdl:message name="UnableToDestroyPullPointFault">
2461         <wsdl:part name="UnableToDestroyPullPointFault"
2462             element="wsnt:UnableToDestroyPullPointFault"/>
2463     </wsdl:message>
2464
2465 <!-- ===== PullPoint::CreatePullPoint =====
2466 CreatePullPoint()
2467 returns: PullPoint (wsa:EndpointReference)
2468 -->
2469     <wsdl:message name="CreatePullPointRequest">

```

```

2470     <wsdl:part name="CreatePullPointRequest"
2471         element="wsnt:CreatePullPoint"/>
2472 </wsdl:message>
2473
2474 <wsdl:message name="CreatePullPointResponse">
2475     <wsdl:part name="CreatePullPointResponse"
2476         element="wsnt:CreatePullPointResponse"/>
2477 </wsdl:message>
2478
2479 <wsdl:message name="UnableToCreatePullPointFault">
2480     <wsdl:part name="UnableToCreatePullPointFault"
2481         element="wsnt:UnableToCreatePullPointFault"/>
2482 </wsdl:message>
2483
2484 <!-- ===== SubscriptionManager::Renew =====
2485     Renew( Duration | AbsoluteTime)
2486     returns: (New Termination Time [CurrentTime])
2487 -->
2488 <wsdl:message name="RenewRequest">
2489     <wsdl:part name="RenewRequest"
2490         element="wsnt:Renew"/>
2491 </wsdl:message>
2492
2493 <wsdl:message name="RenewResponse">
2494     <wsdl:part name="RenewResponse"
2495         element="wsnt:RenewResponse"/>
2496 </wsdl:message>
2497
2498 <wsdl:message name="UnacceptableTerminationTimeFault">
2499     <wsdl:part name="UnacceptableTerminationTimeFault"
2500         element="wsnt:UnacceptableTerminationTimeFault" />
2501 </wsdl:message>
2502
2503 <!-- ===== SubscriptionManager::Unsubscribe =====
2504     Unsubscribe()
2505     returns: empty
2506 -->
2507 <wsdl:message name="UnsubscribeRequest">
2508     <wsdl:part name="UnsubscribeRequest"
2509         element="wsnt:Unsubscribe"/>
2510 </wsdl:message>
2511
2512 <wsdl:message name="UnsubscribeResponse">
2513     <wsdl:part name="UnsubscribeResponse"
2514         element="wsnt:UnsubscribeResponse"/>
2515 </wsdl:message>
2516
2517 <wsdl:message name="UnableToDestroySubscriptionFault">
2518     <wsdl:part name="UnableToDestroySubscriptionFault"
2519         element="wsnt:UnableToDestroySubscriptionFault" />
2520 </wsdl:message>
2521
2522 <!-- ===== SubscriptionManager::PauseSubscription =====
2523     PauseSubscription()
2524     returns: empty
2525 -->
2526 <wsdl:message name="PauseSubscriptionRequest">

```

```

2527     <wsdl:part name="PauseSubscriptionRequest"
2528             element="wsnt:PauseSubscription"/>
2529     </wsdl:message>
2530
2531     <wsdl:message name="PauseSubscriptionResponse">
2532         <wsdl:part name="PauseSubscriptionResponse"
2533                 element="wsnt:PauseSubscriptionResponse"/>
2534     </wsdl:message>
2535
2536     <wsdl:message name="PauseFailedFault">
2537         <wsdl:part name="PauseFailedFault"
2538                 element="wsnt:PauseFailedFault" />
2539     </wsdl:message>
2540
2541 <!-- ===== SubscriptionManager::ResumeSubscription =====
2542 ResumeSubscription()
2543 returns: empty
2544 -->
2545     <wsdl:message name="ResumeSubscriptionRequest">
2546         <wsdl:part name="ResumeSubscriptionRequest"
2547                 element="wsnt:ResumeSubscription"/>
2548     </wsdl:message>
2549
2550     <wsdl:message name="ResumeSubscriptionResponse">
2551         <wsdl:part name="ResumeSubscriptionResponse"
2552                 element="wsnt:ResumeSubscriptionResponse"/>
2553     </wsdl:message>
2554
2555     <wsdl:message name="ResumeFailedFault">
2556         <wsdl:part name="ResumeFailedFault"
2557                 element="wsnt:ResumeFailedFault" />
2558     </wsdl:message>
2559
2560 <!-- ===== PortType Definitions ===== -->
2561 <!-- ===== NotificationConsumer PortType Definition ===== -->
2562     <wsdl:portType name="NotificationConsumer">
2563         <wsdl:operation name="Notify">
2564             <wsdl:input message="wsntw:Notify" />
2565         </wsdl:operation>
2566     </wsdl:portType>
2567
2568 <!-- ===== NotificationProducer PortType Definition ===== -->
2569     <wsdl:portType name="NotificationProducer">
2570         <wsdl:operation name="Subscribe">
2571             <wsdl:input message="wsntw:SubscribeRequest" />
2572             <wsdl:output message="wsntw:SubscribeResponse" />
2573             <wsdl:fault name="ResourceUnknownFault"
2574                     message="wsrf-rw:ResourceUnknownFault" />
2575             <wsdl:fault name="InvalidFilterFault"
2576                     message="wsntw:InvalidFilterFault"/>
2577             <wsdl:fault name="TopicExpressionDialectUnknownFault"
2578                     message="wsntw:TopicExpressionDialectUnknownFault"/>
2579             <wsdl:fault name="InvalidTopicExpressionFault"
2580                     message="wsntw:InvalidTopicExpressionFault" />
2581             <wsdl:fault name="TopicNotSupportedFault"
2582                     message="wsntw:TopicNotSupportedFault" />
2583             <wsdl:fault name="InvalidProducerPropertiesExpressionFault"

```

```

2584         message="wsntw:InvalidProducerPropertiesExpressionFault"/>
2585     <wsdl:fault name="InvalidMessageContentExpressionFault"
2586         message="wsntw:InvalidMessageContentExpressionFault"/>
2587     <wsdl:fault name="UnacceptableInitialTerminationTimeFault"
2588         message="wsntw:UnacceptableInitialTerminationTimeFault"/>
2589     <wsdl:fault name="UnrecognizedPolicyRequestFault"
2590         message="wsntw:UnrecognizedPolicyRequestFault"/>
2591     <wsdl:fault name="UnsupportedPolicyRequestFault"
2592         message="wsntw:UnsupportedPolicyRequestFault"/>
2593     <wsdl:fault name="NotifyMessageNotSupportedFault"
2594         message="wsntw:NotifyMessageNotSupportedFault"/>
2595     <wsdl:fault name="SubscribeCreationFailedFault"
2596         message="wsntw:SubscribeCreationFailedFault"/>
2597 </wsdl:operation>
2598
2599     <wsdl:operation name="GetCurrentMessage">
2600         <wsdl:input message="wsntw:GetCurrentMessageRequest"/>
2601         <wsdl:output message="wsntw:GetCurrentMessageResponse"/>
2602         <wsdl:fault name="ResourceUnknownFault"
2603             message="wsrf-rw:ResourceUnknownFault" />
2604         <wsdl:fault name="TopicExpressionDialectUnknownFault"
2605             message="wsntw:TopicExpressionDialectUnknownFault"/>
2606         <wsdl:fault name="InvalidTopicExpressionFault"
2607             message="wsntw:InvalidTopicExpressionFault" />
2608         <wsdl:fault name="TopicNotSupportedFault"
2609             message="wsntw:TopicNotSupportedFault" />
2610         <wsdl:fault name="NoCurrentMessageOnTopicFault"
2611             message="wsntw:NoCurrentMessageOnTopicFault" />
2612         <wsdl:fault name="MultipleTopicsSpecifiedFault"
2613             message="wsntw:MultipleTopicsSpecifiedFault" />
2614     </wsdl:operation>
2615 </wsdl:portType>
2616
2617 <!-- ===== PullPoint PortType Definition ===== -->
2618 <wsdl:portType name="PullPoint">
2619     <wsdl:operation name="GetMessages">
2620         <wsdl:input name="GetMessagesRequest"
2621             message="wsntw:GetMessagesRequest" />
2622         <wsdl:output name="GetMessagesResponse"
2623             message="wsntw:GetMessagesResponse" />
2624         <wsdl:fault name="ResourceUnknownFault"
2625             message="wsrf-rw:ResourceUnknownFault" />
2626         <wsdl:fault name="UnableToGetMessagesFault"
2627             message="wsntw:UnableToGetMessagesFault" />
2628     </wsdl:operation>
2629
2630     <wsdl:operation name="DestroyPullPoint">
2631         <wsdl:input name="DestroyPullPointRequest"
2632             message="wsntw:DestroyPullPointRequest" />
2633         <wsdl:output name="DestroyPullPointResponse"
2634             message="wsntw:DestroyPullPointResponse" />
2635         <wsdl:fault name="ResourceUnknownFault"
2636             message="wsrf-rw:ResourceUnknownFault"/>
2637         <wsdl:fault name="UnableToDestroyPullPointFault"
2638             message="wsntw:UnableToDestroyPullPointFault" />
2639     </wsdl:operation>
2640

```

```

2641     <wsdl:operation name="Notify">
2642         <wsdl:input message="wsntw:Notify"/>
2643     </wsdl:operation>
2644 </wsdl:portType>
2645
2646 <!-- ===== CreatePullPoint PortType Definition ===== -->
2647 <wsdl:portType name="CreatePullPoint">
2648     <wsdl:operation name="CreatePullPoint">
2649         <wsdl:input name="CreatePullPointRequest"
2650             message="wsntw:CreatePullPointRequest" />
2651         <wsdl:output name="CreatePullPointResponse"
2652             message="wsntw:CreatePullPointResponse" />
2653         <wsdl:fault name="UnableToCreatePullPointFault"
2654             message="wsntw:UnableToCreatePullPointFault" />
2655     </wsdl:operation>
2656 </wsdl:portType>
2657
2658 <!-- ===== SubscriptionManager PortType Definition ===== -->
2659 <wsdl:portType name="SubscriptionManager">
2660     <wsdl:operation name="Renew">
2661         <wsdl:input name="RenewRequest"
2662             message="wsntw:RenewRequest" />
2663         <wsdl:output name="RenewResponse"
2664             message="wsntw:RenewResponse" />
2665         <wsdl:fault name="ResourceUnknownFault"
2666             message="wsrf-rw:ResourceUnknownFault" />
2667         <wsdl:fault name="UnacceptableTerminationTimeFault"
2668             message=
2669             "wsntw:UnacceptableTerminationTimeFault" />
2670     </wsdl:operation>
2671     <wsdl:operation name="Unsubscribe">
2672         <wsdl:input name="UnsubscribeRequest"
2673             message="wsntw:UnsubscribeRequest" />
2674         <wsdl:output name="UnsubscribeResponse"
2675             message="wsntw:UnsubscribeResponse" />
2676         <wsdl:fault name="ResourceUnknownFault"
2677             message="wsrf-rw:ResourceUnknownFault" />
2678         <wsdl:fault name="UnableToDestroySubscriptionFault"
2679             message=
2680             "wsntw:UnableToDestroySubscriptionFault" />
2681     </wsdl:operation>
2682 </wsdl:portType>
2683
2684 <!-- ===== PausableSubscriptionManager PortType Definition ===== -->
2685 <wsdl:portType name="PausableSubscriptionManager">
2686     <!-- ===== Extends: SubscriptionManager ===== -->
2687     <wsdl:operation name="Renew">
2688         <wsdl:input name="RenewRequest"
2689             message="wsntw:RenewRequest" />
2690         <wsdl:output name="RenewResponse"
2691             message="wsntw:RenewResponse" />
2692         <wsdl:fault name="ResourceUnknownFault"
2693             message="wsrf-rw:ResourceUnknownFault" />
2694         <wsdl:fault name="UnacceptableTerminationTimeFault"
2695             message=
2696             "wsntw:UnacceptableTerminationTimeFault" />
2697     </wsdl:operation>

```

```

2698     <wsdl:operation name="Unsubscribe">
2699         <wsdl:input name="UnsubscribeRequest"
2700             message="wsntw:UnsubscribeRequest" />
2701         <wsdl:output name="UnsubscribeResponse"
2702             message="wsntw:UnsubscribeResponse" />
2703         <wsdl:fault name="ResourceUnknownFault"
2704             message="wsrf-rw:ResourceUnknownFault" />
2705         <wsdl:fault name="UnableToDestroySubscriptionFault"
2706             message="
2707             wsntw:UnableToDestroySubscriptionFault" />
2708     </wsdl:operation>
2709
2710     <!-- === PausableSubscriptionManager specific operations === -->
2711     <wsdl:operation name="PauseSubscription">
2712         <wsdl:input message="wsntw:PauseSubscriptionRequest"/>
2713         <wsdl:output message="wsntw:PauseSubscriptionResponse"/>
2714         <wsdl:fault name="ResourceUnknownFault"
2715             message="wsrf-rw:ResourceUnknownFault" />
2716         <wsdl:fault name="PauseFailedFault"
2717             message="wsntw:PauseFailedFault" />
2718     </wsdl:operation>
2719     <wsdl:operation name="ResumeSubscription">
2720         <wsdl:input message="wsntw:ResumeSubscriptionRequest"/>
2721         <wsdl:output message="wsntw:ResumeSubscriptionResponse"/>
2722         <wsdl:fault name="ResourceUnknownFault"
2723             message="wsrf-rw:ResourceUnknownFault" />
2724         <wsdl:fault name="ResumeFailedFault"
2725             message="wsntw:ResumeFailedFault" />
2726     </wsdl:operation>
2727 </wsdl:portType>
2728 </wsdl:definitions>

```

Appendix D. Revision History

Rev	Date	By Whom	What
wd-01	2004-05-18	Steve Graham	Initial version created from submission by contributing companies. Minor modifications made to reflect OASIS formatting and namespace URI choices.
wd-02	2004-05-24	Steve Graham	Added requirements section draft and updated acknowledgments appendix.
wd-03	2004-06-07	Steve Graham	Minor modifications on wd-02 as reported by Peter N. and William V.
wd-04	2004-06-21	Steve Graham	Finalize list of acknowledgments and annotate title page with result of TC vote.
wd-03	2004-07-12	Steve Graham	"rename" 04 back to 03 to minimize reference rework in WSRF.
wd-04	2004-09-20	Steve Graham	WSN 2.3, 2.11, 2.17, 2.18, 2.19, 2.21, 2.22, 2.29
1.3, 01a	2004-09-27	Steve Graham	Rename to follow agreed upon format, add resolution to WSN 5.2
1.3, 01b	2004-11-17	Steve Graham	Post Oct f2f edits
1.3, 01c	2004-12-7	Steve Graham, David Hull	WSN 1,7, 2.7, 2.13, 2.14, 2.15, 2.24 and WSN2.17 revisited
1.3, 01d	2004-12-8	Steve Graham	Fix Action URIs and other edits per Tom Maguire's comments
1.3, 01e	2005-02-19	Bryan Murray	Editorial text and formatting, WSN2.1, WSN2.37, WSN2.38
1.3,01e	2005-02-25	Steve Graham	Mod's to Bryan's edits, WSN 2.34, WSN2.35
1.3, 01e	2005-03-02	Steve Graham	Minor typos reported by Matt Haynos' team.
1.3, 01f	2005-05-09	Bryan Murray, David Hull	WSN1.5, WSN2.1 (InvalidFilterFault), WSN2.32, WSN2.39, WSN2.43, WSN2.45, WSN2.46
1.3, 01g	2005-05-18	Bryan Murray	Per face-to-face agreement on hygiene issues.

Rev	Date	By Whom	What
			WSN 2.48 (but see comment to list from dmh) (Partially?) WSN 2.49 WSN 2.50 WSN 2.53
1.3.01g	2005-05-19	David Hull	Integrated definitions from White Paper, including updates for WSN2.39 compliance. NotificationMessage is not Notification throughout. Began integrating security section from whitepaper, along with proposed text for WSN 2.24 Fixed a few more WS 2.39 stragglers
1.3.01 h	2005-05-27	David Hull	WSN 2.51, WSN 2.52, (also it seems, 2.33, 2.48, 2.1 (pending David raising omnibus issue), 2.23 (partial), 2.25, 2.34, 2.46)
1.3.01.i	2005-06-02	Steve Graham	Modifications on 01g and 01h text, completion of schema and WSDL related items and 2.47, 2.55, 2.41, 2.23
1.3.01.j	2005-06-09	Bryan Murray	AI 106, AI 109, WSN1.4, section 7.1 cleanup, remove ref to Primer
1.3.01.k	2005-06-29	David Hull	Integrated review comments for 2.1, 2.25, 2.33, 2.39, 2.48, 2.50, 2.52. Pseudo-schema descriptions of open content.
1.3.01.l	2005-07-06	David Hull	Integrated review from chairs: 2.6, 2.34, 2.52, 2.55, typos, abstract, acknowledgements.
1.3.01.m	2005-07-07	David Hull	Amended requirement for composability, per committee vote. Formatting in section 4.2
1.3.02.a	2005-09-13	David Hull	Issues 2.57, 2.59. 2.60, 2.61
1.3.02.a	2005-09-14	Bryan Murray	Changed namespace and [action] URIs to b-2 and bw-2 Added resolution to 2.58
1.3.02b	2005-10-10	David Hull	Issues 2.63, 4.2, AI 132, 134, 136, cleanup of references.

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

Appendix E. Notices

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.

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.