

Web Services Base Notification (WS-Base Notification)

Version 1.0

3/5/2004

Authors

Steve Graham, IBM (editor)
Peter Niblett, IBM (editor)
Dave Chappell, Sonic Software
Amy Lewis, TIBCO Software
Nataraj Nagaratnam, IBM
Jay Parikh, Akamai Technologies
Sanjay Patil, SAP AG
Shivajee Samdarshi, TIBCO Software
Igor Sedukhin, Computer Associates International
David Snelling, Fujitsu Laboratories of Europe
Steve Tuecke, Globus / Argonne National Laboratory
William Vambenepe, Hewlett-Packard
Bill Weihl, Akamai Technologies

Copyright Notice

© Copyright Akamai Technologies, Computer Associates International, Inc., Fujitsu Limited, Hewlett-Packard Development Company, International Business Machines Corporation, SAP AG, Sonic Software Corporation, The University of Chicago and Tibco Software Inc. 2003, 2004. All rights reserved.

Permission to copy and display this "Web Services Base Notification" ("this Specification"), in any medium without fee or royalty is hereby granted, provided that you include the following on ALL copies of this Specification, or portions thereof, that you make:

1. A link or URL to this Specification at this location.
2. This Copyright Notice as shown in this Specification.

THIS SPECIFICATION IS PROVIDED "AS IS". AKAMAI TECHNOLOGIES, COMPUTER ASSOCIATES INTERNATIONAL, INC, FUJITSU LIMITED, HEWLETT-PACKARD DEVELOPMENT COMPANY, IBM, SAP AG, SONIC SOFTWARE, THE UNIVERSITY OF CHICAGO AND TIBCO SOFTWARE (COLLECTIVELY, THE "COMPANIES") MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS SPECIFICATION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE

IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

THE COMPANIES WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS SPECIFICATION.

The companies each agree to grant you a royalty-free license, under commercially reasonable terms and conditions, to their respective patents that they deem necessary to implement this Specification.

The names and trademarks of the Companies may NOT be used in any manner, including advertising or publicity pertaining to the Specification or its contents without specific, written prior permission. Title to copyright in this Specification will at all times remain with the Companies.

No other rights are granted by implication, estoppel or otherwise.

PORTIONS OF THIS MATERIAL WERE PREPARED AS AN ACCOUNT OF WORK SPONSORED BY IBM CORPORATION AT UNIVERSITY OF CHICAGO'S ARGONNE NATIONAL LABORATORY. NEITHER THE AUTHORS, NOR THE UNITED STATES GOVERNMENT OR ANY AGENCY THEREOF, NOR THE UNIVERSITY OF CHICAGO, NOR IBM, NOR ANY OF THEIR EMPLOYEES OR OFFICERS, NOR ANY OTHER COPYRIGHT HOLDERS OR CONTRIBUTORS, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT, OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS. REFERENCE HEREIN TO ANY SPECIFIC COMMERCIAL PRODUCT, PROCESS, OR SERVICE BY TRADE NAME, TRADEMARK, MANUFACTURER, OR OTHERWISE, DOES NOT NECESSARILY CONSTITUTE OR IMPLY ITS ENDORSEMENT, RECOMMENDATION, OR FAVORING BY IBM, THE UNITED STATES GOVERNMENT OR ANY AGENCY THEREOF OR ANY OTHER COPYRIGHT HOLDERS OR CONTRIBUTORS. THE VIEW AND OPINIONS OF AUTHORS EXPRESSED HEREIN DO NOT NECESSARILY STATE OR REFLECT THOSE OF IBM, THE UNITED STATES GOVERNMENT OR ANY AGENCY THEREOF, OR THE ENTITY BY WHICH AN AUTHOR MAY BE EMPLOYED.

This manuscript has been created in part by the University of Chicago as Operator of Argonne National Laboratory ("Argonne") under Contract No. W-31-109-ENG-38 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

Abstract

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

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

This document defines the Web services interfaces for NotificationProducers and NotificationConsumers. It includes standard message exchanges to be implemented by service providers that wish to act in these roles, along with operational requirements expected of them. This is the base specification on which the other WS-Notification specification documents depend. An implementer interested just in direct, point to point, notification need only read this *WS-Base Notification* specification, together with the *Publish-Subscribe Notification for Web Services* white paper.

Status

This WS-Notification specification is an initial draft release and is provided for review and evaluation only. The Companies hope to solicit your contributions and suggestions in the near future. The Companies make no warranties or representations regarding the specification in any manner whatsoever.

Table of Contents

1	INTRODUCTION	4
1.1	NOTATIONAL CONVENTIONS	5
1.2	NAMESPACES	5
2	TERMINOLOGY AND CONCEPTS	6
3	NOTIFICATIONCONSUMER INTERFACE	6
3.1	NOTIFY	7
3.1.1	<i>Example SOAP Encoding of the Notify Message</i>	8
4	NOTIFICATIONPRODUCER INTERFACE	8
4.1	NOTIFICATIONPRODUCER RESOURCE PROPERTIES	9
4.2	SUBSCRIBE	10
4.2.1	<i>Example SOAP Encoding of the Subscribe Message Exchange</i>	13
4.3	GETCURRENTMESSAGE	15
4.3.1	<i>Example SOAP Encoding of the GetCurrentMessage Message Exchange</i>	16
5	SUBSCRIPTIONMANAGER INTERFACE	17
5.1	SUBSCRIPTION RESOURCE PROPERTIES	17
5.2	PAUSESUBSCRIPTION	18
5.2.1	<i>Example SOAP Encoding of the PauseSubscription Message Exchange</i>	19
5.3	RESUMESUBSCRIPTION	20
5.3.1	<i>Example SOAP Encoding of the ResumeSubscription Message Exchange</i>	21
6	SECURITY CONSIDERATIONS	22
7	ACKNOWLEDGEMENTS	22
8	REFERENCES	22
9	APPENDIX I – UML	24
10	APPENDIX II – XML SCHEMA	25
11	APPENDIX III – WSDL 1.1	26

1 Introduction

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.

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.

A NotificationConsumer can be registered to receive notifications directly from a NotificationProducer, so an implementer interested just in direct, point to point, notification need only read this *WS-Base Notification* specification, together with the additional material that is to be found in *Publish-Subscribe Notification for Web Services* document [WS-Notification Whitepaper].

In the remainder of this section we will give a brief introduction to the Notification pattern, and the terms we will use in this specification. For a more complete description, the reader is encouraged to consult [WS-Notification Whitepaper].

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 accepts incoming Subscribe requests, each subscribe request identifies one or more Topics of interest and a reference to a NotificationConsumer. The NotificationProducer maintains a list of subscriptions (each subscription entry contains the Topic(s), a reference to the NotificationConsumer and possibly additional information). When it has a notification to distribute, the NotificationProducer matches the notification against the interest registered in each Subscription known to it. If it identifies a match it issues the Notification to the NotificationConsumer associated with that Subscription.

This specification allows the use of a variety of dialects in which to refer to Topics, ranging from simple QNames, to more sophisticated path-based Expression languages. Topic classifications and Topic Expressions are discussed more fully in [WS-Topics].

The configuration where a NotificationConsumer is subscribed directly to the NotificationProducer is referred to as the "direct" or "point-to-point" notification pattern. There are other variations of the Notification pattern, where the NotificationConsumer is subscribed to an intermediary NotificationBroker service. These are covered by the [WS-Brokered Notification] specification.

1.1 Notational Conventions

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]).

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

1.2 Namespaces

The following namespaces are used in this document:

Prefix	Namespace
s12	http://www.w3.org/2003/05/soap-envelope
xsd	http://www.w3.org/2001/XMLSchema
wsp	http://schemas.xmlsoap.org/ws/2002/12/policy
wsa	http://schemas.xmlsoap.org/ws/2003/02/addressing
wsnt	http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification
wsrl	http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceLifetime
wsrp	http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties

2 Terminology and Concepts

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

3 NotificationConsumer Interface

WS-Notification allows a NotificationProducer to send a NotificationMessage to a NotificationConsumer in one of two ways:

1. The NotificationProducer MAY simply send the "raw" NotificationMessage (i.e. the application-specific content) to the NotificationConsumer.
2. The NotificationProducer MAY send the NotificationMessage data using the Notify message described below

The second option (the Notify message) allows the NotificationProducer to supply additional WS-Notification-defined information (such as the Topic) in addition to the application-specific NotificationMessage content. It also means that the NotificationConsumer can receive a wide range of NotificationMessages without having to explicitly provide support for each one in its WSDL. Furthermore, this form of Notification allows a batch of multiple NotificationMessages to be delivered using a single message.

When a Subscriber sends a Subscribe request message, it indicates which form of Notification is required (the raw NotificationMessage, or the Notify Message). The NotificationProducer MUST observe this Subscription parameter, and use the form that has been requested.

This means that a NotificationConsumer MAY choose to implement the Notify Message, or to implement raw NotificationMessage(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 Notify

A NotificationProducer MAY send the Notify message to the NotificationConsumer in order to deliver one or more NotificationMessage(s). The format of the Notify message is:

```

...
<wsnt:Notify>
  <wsnt:NotificationMessage>
    <wsnt:Topic dialect="xsd:anyURI">
      {any}
    </wsnt:Topic>
    <wsnt:ProducerReference>?
      wsa:EndpointReference
    </wsnt:ProducerReference>
    <wsnt:Message>xsd:any</wsnt:Message>
  <wsnt:NotificationMessage>+
</wsnt:Notify>
...

```

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

/wsnt:Notify

Contains a collection of one or more NotificationMessages to be delivered. This message MAY follow the implied resource pattern as outlined in [State Paper].

/wsnt:Notify/wsnt:NotificationMessage

Contains a NotificationMessage to be delivered.

/wsnt:Notify/wsnt:NotificationMessage/wsnt:Topic

The TopicExpression describing exactly one Topic - the Topic that is associated with this NotificationMessage. This element describes the topic that matched to a subscription, causing the NotificationProducer to send the Notify message to the NotificationConsumer.

/wsnt:Notify/wsnt:NotificationMessage/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:Notify/wsnt:NotificationMessage/wsnt:ProducerReference

An OPTIONAL EndpointReference to the NotificationProducer that originally generated the NotificationMessage artifact.

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

A copy of the actual NotificationMessage payload.

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

3.1.1 Example SOAP Encoding of the Notify Message

The following is a non-normative example of a Notify message using SOAP 1.2 [SOAP 1.2]:

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsnt=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
  xmlns:ncex="http://www.consumer.org/RefProp"
  xmlns:npex="http://www.producer.org/RefProp">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification/Notify
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.consumer.org/ConsumerEndpoint
    </wsa:To>
    <ncex:NCResourceId>
      uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
    </ncex:NCResourceId>
  </s12:Header>
  <s12:Body>
    <wsnt:Notify>
      <wsnt:NotificationMessage>
        <wsnt:Topic
          dialect="http://www.ibm.com/xmlns/stdwip/web-services/WS-
Topics/TopicExpression/simple">
          npex:SomeTopic
        </wsnt:Topic>
        <wsnt:ProducerReference
          xmlns:npex="http://www.producer.org/RefProp">
          <wsa:Address>
            http://www.producer.org/ProducerEndpoint
          </wsa:Address>
          <wsa:ReferenceProperties>
            <npex:NPResourceId>
              uuid:84decd55-7d3f-65ad-ac44-675d9fce5d22
            </npex:NPResourceId>
          </wsa:ReferenceProperties>
        </wsnt:ProducerReference>
        <wsnt:Message>
          <npex:NotifyContent>exampleNotifyContent</npex:NotifyContent>
        </wsnt:Message>
      </wsnt:NotificationMessage>
    </wsnt:Notify>
  </s12:Body>
</s12:Envelope>
```

Note the use of the `wsa:ReferenceProperties` elements in the message. These are examples of WS-Resource qualified EndpointReferences, following the implied resource pattern as outlined in [state paper].

4 NotificationProducer Interface

This section describes the message exchanges and resource properties 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 or to control the delivery of NotificationMessages by the NotificationProducer.

4.1 NotificationProducer Resource Properties

In addition to the message exchanges described in this specification, a NotificationProducer MUST 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.

The Resource Properties document defined by the NotificationProducer MUST include a reference to the following resource property elements:

```
...
  targetNamespace=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
...

<xsd:element name="Topics" type="wsnt:TopicExpressionType"
              minOccurs="0" maxOccurs="unbounded" />

<xsd:element name="FixedTopicSet" type="xsd:boolean"
              minOccurs="1" maxOccurs="1" />

<xsd:element name="TopicExpressionDialects" type="xsd:anyURI"
              minOccurs="1" maxOccurs="unbounded" />
...
```

These Resource Property elements are further constrained as follows:

/wsnt:Topics

This resource property contains collection of Topics supported by the NotificationProducer. The set of topics is expressed using one or more wsnt:Topic resource property element(s). Each wsnt:Topic 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:Topic resource property element(s), for example, in different resource property elements each using a different dialect.

If a Topic appears in one of the wsnt:Topic resource property elements, a Subscriber can reasonably expect the NotificationProducer to deliver NotificationMessages on this Topic. It is not a guarantee that it will receive any NotificationMessages; the NotificationProducer may not actually produce any NotificationMessages on the particular Topic during the time that the Subscriber is registered.

/wsnt:FixedTopicSet

Indicates if the collection of Topics contained within the wsnt:Topic 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).

/wsnt:TopicExpressionDialects

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 TopicExpressionDialects property is a "fixed property", i.e. its value, for any given NotificationProducer WS-Resource, does not change over time.

4.2 Subscribe

A Subscriber sends a Subscribe message to a NotificationProducer in order to register the interest of a NotificationConsumer for NotificationMessages related to one or more Topics.

As part of the processing of a Subscribe request message, the NotificationProducer MUST create a Subscription resource representing the Subscription. It returns an EndpointReference in the response to the Subscribe request. This EndpointReference is a WS-Resource-Qualified EndpointReference as defined in [State Paper] and includes the address of a SubscriptionManager service and a reference property containing the identity of the Subscription resource.

Sending two identical Subscribe request messages to a NotificationProducer will result in the creation of two Subscription resources. The NotificationConsumer will be associated with both Subscription resources and so will receive two copies of any matching NotificationMessage.

The format of the Subscribe message is:

```
...
<wsnt:Subscribe>
  <wsnt:ConsumerReference>
    wsa:endpointReference
  </wsnt:ConsumerReference>
  <wsnt:TopicExpression dialect = "xsd:anyURI">
    {any}
  </wsnt:TopicExpression>
  <wsnt:UseNotify> xsd:boolean </wsnt:UseNotify?>
  <wsnt:Precondition> wsrp:QueryExpression </Precondition?>
  <wsnt:Selector> wsrp:QueryExpression </wsnt:Selector?>
  <wsnt:SubscriptionPolicy> {any} </wsnt:SubscriptionPolicy?>
  <wsnt:InitialTerminationTime>
    xsd:dateTime
  </wsnt:InitialTerminationTime?>
</wsnt:Subscribe>
```

...

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

/wsnt:ConsumerReference

An endpoint reference element from WS-Addressing [WS-Addressing], used to identify the NotificationConsumer.

In cases where the value of the wsnt:UseNotify component is "true", the Web service identified by the endpoint reference MUST implement the message exchanges defined by NotificationConsumer (e.g. the Notify message).

/wsnt:ConsumerReference/wsa:ReferenceProperties

The ReferenceProperties MAY contain an identifier of the resource to which the NotificationMessage should be delivered. This is used in the situation where the NotificationConsumer itself is implementing the implied resource pattern [State Paper].

The ReferenceProperties MAY contain other elements useful in the NotificationMessage delivery. For example the ReferenceProperties could contain a "correlationContext" - an element that is returned as part of each NotificationMessage delivered to the NotificationConsumer.

/wsnt:TopicExpression

A TopicExpression that identifies one or more Topics supported by the NotificationProducer.

/wsnt:TopicExpression/@dialect

A URI that identifies the language of the TopicExpression. WS-Topics defines some standard URIs. Designers MAY define and use other domain-specific URIs to identify the dialect of the TopicExpression.

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

/wsnt:UseNotify

A boolean value that indicates how the NotificationProducer is to deliver NotificationMessages to the NotificationConsumer. This component is optional, if missing the default value is "true".

If this value is "true" then the NotificationProducer MUST use the wsnt:Notify message to deliver NotificationMessages to the NotificationConsumer. In this case the NotificationConsumer referred to in the wsnt:ConsumerReference element MUST implement the Notify message and include a Notify operation in its portType definition.

If this value is "false" then the NotificationProducer MUST deliver the NotificationMessage using the raw NotificationMessage itself. In this case the NotificationConsumer referred to in the wsnt:ConsumerReference element MUST implement all the NotificationMessages associated with the Topic or Topics referred to by the TopicExpression, and include corresponding operations in its portType definition.

/wsnt:Precondition

This component contains a QueryExpression as defined by WS-ResourceProperties.

This component expresses a NotificationProducer-specific precondition associated with generation of NotificationMessages by the NotificationProducer. A NotificationProducer object is free to define the syntax and meaning of this expression

Non-normative examples of a possible precondition include:

- An XPath expression on the Resource Properties of the NotificationProducer that evaluates to a Boolean. The NotificationProducer must only deliver NotificationMessages to the NotificationConsumer if this expression evaluates to "true".
- A database query operation to be performed in order to generate a NotificationMessage, or periodic sequence of NotificationMessages.

/wsnt:Selector

A QueryExpression as defined by WS-ResourceProperties. A wsnt:Selector expression MUST evaluate to a Boolean. The evaluation context is NotificationMessage(s) that are Published on the Topic(s) identified by the TopicPathExpression defined by the wsnt:TopicPathExpression component of this message.

The NotificationProducer must only deliver NotificationMessages to the NotificationConsumer if this expression evaluates to "true".

/wsnt:SubscriptionPolicy

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

One non-normative example use of this component is to specify a WS-Policy [WS-Policy] element indicating the requested policies to apply to the Subscription.

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

/wsnt:InitialTerminationTime

This component contains the service requestor's suggestion for the initial termination time of the Subscription being created. This time is relative to the time source used by the NotificationProducer. If the NotificationProducer is unable or unwilling to set the TerminationTime resource property of the Subscription resource to the given value or greater, then the Subscribe request MUST fault. If the value is not "in the future" relative to the current time as known by the NotificationProducer, the Subscribe request MUST fault. The use of the xsi:nil attribute with value "true" indicates there is no scheduled termination time requested for the Subscription. If the element

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

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

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

```

...
<wsnt:SubscribeResponse>
  <wsnt:SubscriptionReference>
    <wsa:Address> Address of Subscription Manager </wsa:Address>
    <wsa:ReferenceProperties>
      Subscription Identifier
    </wsa:ReferenceProperties>
    ...
  </wsnt:SubscriptionReference>
  ...
</wsnt:SubscribeResponse>
...

```

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

`/wsnt:SubscriptionReference/wsa:Address`

The address of a Web service that implements the SubscriptionManager set of messages. This SubscriptionManager is associated with the Subscription resource created by the Subscribe message following the implied resource pattern.

`/wsnt:SubscriptionReference/wsa:ReferenceProperties`

An identifier of the Subscription resource following to the implied resource pattern.

Instead of the SubscribeResponse message, the NotificationProducer may also send the following faults in response to a Subscribe message:

- Unable to create subscription
- TopicPath Dialect not understood
- Others, TBD

4.2.1 Example SOAP Encoding of the Subscribe Message Exchange

The following is a non-normative example of a Subscribe request message using SOAP 1.2 [SOAP 1.2]:

```

<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsnt=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
  xmlns:npex="http://www.producer.org/RefProp">
  <s12:Header>

```

```

    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
BaseNotification/Subscribe
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.producer.org/ProducerEndpoint
    </wsa:To>
    <npex:NPResourceId>
      uuid:84decd55-7d3f-65ad-ac44-675d9fce5d22
    </npex:NPResourceId>
  </s12:Header>
  <s12:Body>
    <wsnt:Subscribe>
      <wsnt:ConsumerReference
        xmlns:ncex="http://www.consumer.org/RefProp">
        <wsa:Address>
          http://www.producer.org/ProducerEndpoint
        </wsa:Address>
        <wsa:ReferenceProperties>
          <ncex:NCResourceId>
            uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
          </ncex:NCResourceId>
        </wsa:ReferenceProperties>
        </wsnt:ConsumerReference>
        <wsnt:TopicExpression
          dialect="http://www.ibm.com/xmlns/stdwip/web-services/WS-
Topics/TopicExpression/simple">
          npex:SomeTopic
        </wsnt:TopicExpression>
        <wsnt:UseNotify>true</wsnt:UseNotify>
        <wsnt:Precondition
          dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
          boolean(ncex:NumberOfBlocks > 20 and ncex:BlockSize=1024)
        </Precondition>
        <wsnt:Selector
          dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
          boolean(ncex:Producer="15")
        </wsnt:Selector>
        <wsnt:InitialTerminationTime>
          2003-12-25T00:00:00.00000Z
        </wsnt:InitialTerminationTime>
        </wsnt:Subscribe>
      </s12:Body>
    </s12:Envelope>

```

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

```

<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsnt="
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
  xmlns:ncex="http://www.consumer.org/RefProp">
  <s12:Header>
    <wsa:Action>

```

```

    http://www.ibm.com/xmlns/stdwip/web-services/WS-
BaseNotification/SubscribeResponse
  </wsa:Action>
  <wsa:To s12:mustUnderstand="1">
    http://www.consumer.org/ConsumerEndpoint
  </wsa:To>
  <ncex:NCResourceId>
    uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
  </ncex:NCResourceId>
</s12:Header>
<s12:Body>
  <wsnt:SubscribeResponse>
    <wsnt:SubscriptionReference
      xmlns:npex="http://www.producer.org/RefProp">
      <wsa:Address>
        http://www.producer.org/SubsMgrEndpoint
      </wsa:Address>
      <wsa:ReferenceProperties>
        <npex:NSResourceId>
          uuid:8fefcd11-7d3d-66b344a2-ca44-9876bacd44e9
        </npex:NSResourceId>
      </wsa:ReferenceProperties>
    </wsnt:SubscriptionReference>
  </wsnt:SubscribeResponse>
</s12:Body>
</s12:Envelope>

```

4.3 *GetCurrentMessage*

In response to a *GetCurrentMessage* message, the NotificationProducer MAY return the last NotificationMessage published to a given Topic. This is a non-destructive read, and allows a newly-subscribed NotificationConsumer to get the last NotificationMessage that other NotificationConsumers have received.

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

The format of the *GetCurrentMessage* request message is:

```

...
<wsnt:GetCurrentMessage>
  <wsnt:Topic dialect="xsd:anyURI">{any}</wsnt:Topic>
</wsnt:GetCurrentMessage>
...

```

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

/wsnt:Topic

A TopicExpression that identifies exactly one Topic.

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

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

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

/wsnt:GetCurrentMessageResponse/{any}

Contains the last NotificationMessage associated with the Topic identified by the request message.

Instead of the GetCurrentMessageResponse message, the NotificationProducer may also send the following faults in response to a GetCurrentMessage message:

- o Invalid TopicExpression
- o Given TopicExpression did not match exactly one Topic supported by the NotificationProducer
- o The Topic is supported but there is no NotificationMessage currently available on that Topic
- o Others, TBD

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 1.2 [SOAP 1.2]:

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsnt="http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
  xmlns:npex="http://www.producer.org/RefProp">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification/GetCurrentMessage
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.producer.org/ProducerEndpoint
    </wsa:To>
    <npex:NPResourceId>
      uuid:84decd55-7d3f-65ad-ac44-675d9fce5d22
    </npex:NPResourceId>
  </s12:Header>
  <s12:Body>
    <wsnt:GetCurrentMessage>
      <wsnt:Topic
        dialect="http://www.ibm.com/xmlns/stdwip/web-services/WS-Topics/TopicExpression/simple">
        npex:SomeTopic
```



```

    </wsnt:Topic>
  </wsnt:GetCurrentMessage>
</s12:Body>
</s12:Envelope>

```

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

```

<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsnt=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
  xmlns:wsrp=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
  xmlns:npex="http://www.producer.org/RefProp">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
BaseNotification/GetCurrentMessageResponse
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.consumer.org/ConsumerEndpoint
    </wsa:To>
    <ncex:NCResourceId xmlns:ncex="http://www.consumer.org/RefProp">
      uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
    </ncex:NCResourceId>
  </s12:Header>
  <s12:Body>
    <wsnt:GetCurrentMessageResponse>
      <npex:NotifyContent>exampleNotifyContent</npex:NotifyContent>
    </wsnt:GetCurrentMessageResponse>
  </s12:Body>
</s12:Envelope>

```

5 SubscriptionManager Interface

The SubscriptionManager interface defines message exchanges to manipulate Subscription resources. The SubscriptionManager follows the implied resource pattern as described in [State Paper].

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

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

5.1 Subscription Resource Properties

In addition to the Resource Properties required by WS-ResourceLifetime, the Subscription resource MUST include a reference to each of the following Resource Property elements in its Resource Properties document:

```

...
  <xsd:element name="ConsumerReference"

```

```

        type="wsa:EndpointReferenceType"
        minOccurs="1" maxOccurs="1" />
<xsd:element name="TopicExpression"
        type="wsnt:TopicExpressionType"
        minOccurs="1" maxOccurs="1" />
<xsd:element name="UseNotify"
        type="xsd:boolean"
        minOccurs="1" maxOccurs="1" />
<xsd:element name="Precondition"
        type="wsrp:QueryExpressionType"
        minOccurs="0" maxOccurs="1" />
<xsd:element name="Selector"
        type="wsrp:QueryExpressionType"
        minOccurs="0" maxOccurs="1" />
<xsd:element name="SubscriptionPolicy"
        type="xsd:anyType"
        minOccurs="0" maxOccurs="1" />
<xsd:element name="CreationTime"
        type="xsd:dateTime"
        minOccurs="0" maxOccurs="1" />

```

These Resource Property elements are further described as follows:

/wsnt:ConsumerReference and /wsnt:TopicExpression and /wsnt:UseNotify and /wsnt:Precondition and /wsnt:Selector and /wsnt:SubscriptionPolicy

These elements are as defined in the description of the Subscribe request message (see 4.2).

/wsnt:CreationTime

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

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

/wsnt:ConsumerReference and /wsnt:TopicExpression and /wsnt:UseNotify and /wsnt:Precondition and /wsnt:Selector and /wsnt:SubscriptionPolicy

5.2 *PauseSubscription*

To temporarily suspend the delivery of NotificationMessages on the given Subscription, a requestor MUST send a PauseSubscription request message to the SubscriptionManager. The PauseSubscription request message has the following form:

```

...
<wsnt:PauseSubscription/>
...

```

This request message MUST follow the implied resource pattern as outlined in [State Paper].

Upon successful processing of this message the Subscription is in the *paused* state. Delivery of NotificationMessages can be resumed by sending the ResumeSubscription request message (see 5.3).

In an asynchronous implementation, there could be some NotificationMessages *in-flight* at the time the Subscription was paused. A NotificationProducer MAY attempt to deliver these in-flight NotificationMessages.

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

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

```
...
<wsnt:PauseSubscriptionResponse />
...
```

Instead of the PauseSubscriptionResponse message, the SubscriptionManager may also send the following faults in response to a PauseSubscription request message:

- Pause operation could not be performed on the Subscription
- Others, TBD

5.2.1 Example SOAP Encoding of the PauseSubscription Message Exchange

The following is a non-normative example of a PauseSubscription request message using SOAP 1.2 [SOAP 1.2]:

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsnt="
    http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
  xmlns:npex="http://www.producer.org/RefProp">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
BaseNotification/PauseSubscription
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.producer.org/SubsMgrEndpoint
    </wsa:To>
    <npex:NSResourceId>
      uuid:8fefcd11-7d3d-66b344a2-ca44-9876bacd44e9
    </npex:NSResourceId>
  </s12:Header>
  <s12:Body>
    <wsnt:PauseSubscription/>
  </s12:Body>
</s12:Envelope>
```

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

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsnt="
    http://www.ibm.com/xmlns/stdwip/web-services/WS-
```

```

BaseNotification"
  xmlns:ncex="http://www.consumer.org/RefProp">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
BaseNotification/PauseSubscriptionResponse
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.consumer.org/ConsumerEndpoint
    </wsa:To>
    <ncex:NCResourceId>
      uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
    </ncex:NCResourceId>
  </s12:Header>
  <s12:Body>
    <wsnt:PauseSubscriptionResponse/>
  </s12:Body>
</s12:Envelope>

```

5.3 ResumeSubscription

If a requestor wishes to resume the delivery of NotificationMessages on the given Subscription, it must send a ResumeSubscription request message. The ResumeSubscription request message has the following form:

```

...
<wsnt:ResumeSubscription/>
...

```

This request message MUST follow the implied resource pattern as outlined in [State Paper]. Upon successful processing of this message the Subscription is no longer in the *paused* state.

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

When a Subscription is resumed the NotificationProducer MAY choose from one of the following courses of action. The choice MAY be determined by the NotificationProducer itself, or MAY be specified by a Policy associated with the Subscription.

1. Generate NotificationMessages 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. Generate a NotificationMessage 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. Generate no NotificationMessages until a Situation occurs after the Subscription has been resumed.

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

```

...
<wsnt:ResumeSubscriptionResponse />
...

```

Instead of the ResumeSubscriptionResponse message, the SubscriptionManager may also send the following faults in response to a ResumeSubscription request message:

- Resume operation could not be performed on the Subscription
- Others, TBD

5.3.1 Example SOAP Encoding of the ResumeSubscription Message Exchange

The following is a non-normative example of a ResumeSubscription request message using SOAP 1.2 [SOAP 1.2]:

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsnt="
    http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
  xmlns:npex="http://www.producer.org/RefProp">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
BaseNotification/ResumeSubscription
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.producer.org/SubsMgrEndpoint
    </wsa:To>
    <npex:NSResourceId>
      uuid:8fefcd11-7d3d-66b344a2-ca44-9876bacd44e9
    </npex:NSResourceId>
  </s12:Header>
  <s12:Body>
    <wsnt:ResumeSubscription/>
  </s12:Body>
</s12:Envelope>
```

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

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsnt="
    http://www.ibm.com/xmlns/stdwip/web-services/WS-
BaseNotification"
  xmlns:ncex="http://www.consumer.org/RefProp">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
BaseNotification/ResumeSubscriptionResponse
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.consumer.org/ConsumerEndpoint
    </wsa:To>
    <ncex:NCRResourceId>
      uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
    </ncex:NCRResourceId>
  </s12:Header>
  <s12:Body>
    <wsnt:ResumeSubscriptionResponse/>
```

```
</s12:Body>  
</s12:Envelope>
```

6 Security Considerations

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

7 Acknowledgements

This specification has been developed as a result of joint work with many individuals and teams. The authors wish to acknowledge the contributions from many people, including:

Tim Banks (IBM), Nick Butler (IBM), Glen Daniels (Sonic Software), Doug Davis (IBM), John Dinger (IBM), Don Ferguson (IBM), Jeff Frey (IBM), David Hull (Tibco), Andreas Koeppel (SAP), Heather Kreger (IBM), Kevin Liu (SAP), Tom Maguire (IBM), Susan Malaika (IBM), David Martin (IBM), Bryan Murray (HP), Martin Nally (IBM), Jeff Nick (IBM), Claus von Riegen (SAP), Rick Rineholt (IBM), John Rofrano (IBM), Eugène Sindambiwe (SAP), Jay Unger (IBM), Mark Weitzel (IBM), Dan Wolfson (IBM).

8 References

[SOAP 1.2]

<http://www.w3.org/TR/soap12-part1/>

[State Paper]

<http://www-106.ibm.com/developerworks/webservices/library/ws-resource/ws-modelingresources.pdf>

[WS-Addressing]

<http://www.ibm.com/developerworks/webservices/library/ws-add/>

[WS-Brokered Notification]

<ftp://www6.software.ibm.com/software/developer/library/ws-notification/WS-BrokeredN.pdf>

[WS-Notification Whitepaper]

<http://www-106.ibm.com/developerworks/library/ws-pubsub/WS-PubSub.pdf>

[WS-Policy]

<http://www-106.ibm.com/developerworks/library/ws-polfram/>

[WS-ResourceLifetime]

<http://www-106.ibm.com/developerworks/webservices/library/ws-resource/ws-resourcelifetime.pdf>

[WS-ResourceProperties]

<http://www-106.ibm.com/developerworks/webservices/library/ws-resource/ws-resourceproperties.pdf>

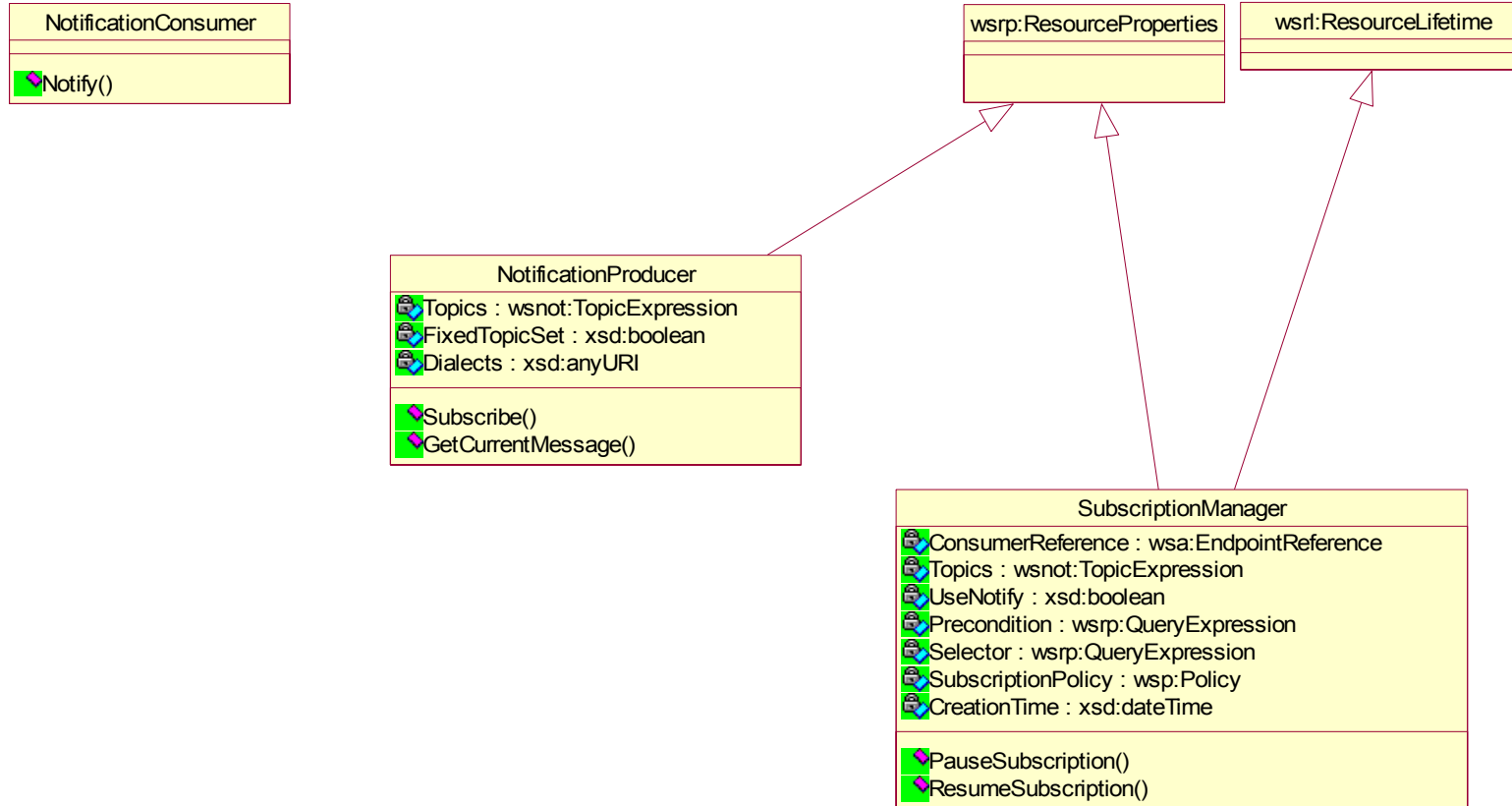
[WS-Security]

<http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wss-soap-message-security-1.0.pdf>

[WS-Topics]

<ftp://www6.software.ibm.com/software/developer/library/ws-notification/WS-Topics.pdf>

9 Appendix I – UML



10 Appendix II – XML Schema

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Legal Disclaimer

  Copyright Notice

  (c) Copyright International Business Machines Corporation,
  The University of Chicago, Sonic Software Corporation,
  SAP AG, Hewlett-Packard Development Company, Akamai Technologies
Inc.
  and Tibco Software Inc. 2003, 2004 All rights reserved.
-->

<xsd:schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsnt=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
  xmlns:wsrp=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-
ResourceProperties"
  targetNamespace=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
<!-- ===== Imports ===== -->

  <xsd:import namespace=
    "http://schemas.xmlsoap.org/ws/2003/03/addressing"
    schemaLocation=
    "http://schemas.xmlsoap.org/ws/2003/03/addressing"
  />

  <xsd:import namespace=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-
ResourceProperties"
    schemaLocation=
    "http://www-106.ibm.com/developerworks/webservices/library/ws-
resource/WS-ResourceProperties.xsd"
  />
<!-- ===== Topic Expression Container ===== -->
  <xsd:complexType name="TopicExpressionType" mixed="true">
    <xsd:sequence>
      <xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
    </xsd:sequence>
    <xsd:attribute name="dialect" type="xsd:anyURI" />
  </xsd:complexType>

  <xsd:element name="TopicExpression"
```

```

        type="wsnt:TopicExpressionType" />
<!-- ===== Resource Property Related ===== -->
<!-- ===== Resource Properties for NotificationProducer ===== -->
  <xsd:element name="Topic" type="wsnt:TopicExpressionType" />

  <xsd:element name="FixedTopicSet" type="xsd:boolean"/>

  <xsd:element name="TopicExpressionDialects" type="xsd:anyURI" />

<!-- ===== Resource Properties for SubscriptionManager ===== -->
  <xsd:element name="ConsumerReference"
    type="wsa:EndpointReferenceType" />
  <xsd:element name="UseNotify"
    type="xsd:boolean" />
  <xsd:element name="Precondition"
    type="wsrp:QueryExpressionType" />
  <xsd:element name="Selector"
    type="wsrp:QueryExpressionType" />
  <xsd:element name="SubscriptionPolicy"
    type="xsd:anyType" />
  <xsd:element name="CreationTime"
    type="xsd:dateTime" />

</xsd:schema>

```

11 Appendix III – 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"?>
<!--
  Legal Disclaimer

  Copyright Notice

  (c) Copyright Akamai Technologies,
  Computer Associates International, Inc., Fujitsu Limited,
  Hewlett-Packard Development Company,
  International Business Machines Corporation, SAP AG,
  Sonic Software Corporation, Tibco Software Inc. and
  The University of Chicago 2003, 2004 All rights reserved.
-->
<wsdl:definitions name="WS-BaseNotification"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2002/12/policy"
  xmlns:wsnt=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification"
  xmlns:wsrp=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"

```

```

xmlns:wsrl=
"http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceLifetime"
targetNamespace=
"http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification">
<!-- ===== Imports ===== -->
  <wsdl:import
    namespace=
      "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
    location=
      "http://www-106.ibm.com/developerworks/webservices/library/ws-
resource/WS-ResourceProperties.wsdl" />

  <wsdl:import
    namespace=
      "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceLifetime"
    location=
      "http://www-106.ibm.com/developerworks/webservices/library/ws-
resource/WS-ResourceLifetime.wsdl" />

<!-- ===== Types Definitions ===== -->
  <wsdl:types>
    <xsd:schema
      targetNamespace=
        "http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification" >

      <xsd:include schemaLocation=
        "http://www-106.ibm.com/developerworks/library/specification/ws-
notification/WS-BaseN.xsd" />

      <xsd:import
        namespace=
          "http://schemas.xmlsoap.org/ws/2003/03/addressing"
        schemaLocation=
          "http://schemas.xmlsoap.org/ws/2003/03/addressing"
        />
      <xsd:import namespace=
        "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
        schemaLocation=
          "http://www-106.ibm.com/developerworks/webservices/library/ws-
resource/WS-ResourceProperties.xsd"
        />
      <xsd:import
        namespace=
          "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceLifetime"
        schemaLocation=
          "http://www-106.ibm.com/developerworks/webservices/library/ws-
resource/WS-ResourceLifetime.xsd"
        />

<!-- ===== Resource Property Related ===== -->
<!-- ===== Resource Properties for NotificationProducer ===== -->
    <xsd:element name="NotificationProducerRP" >
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="wsnt:Topic"
            minOccurs="1" maxOccurs="unbounded" />

```

```

        <xsd:element ref="wsnt:FixedTopicSet"
            minOccurs="1" maxOccurs="1" />
        <xsd:element ref="wsnt:TopicExpressionDialects"
            minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
</xsd:element>

<!-- ===== Resource Properties for SubscriptionManager ===== -->
<xsd:element name="SubscriptionManagerRP" >
    <xsd:complexType>
        <xsd:sequence>
            <!-- From WS-ResourceLifetime ScheduledResourceTermination -->
            <xsd:element ref="wsrl:CurrentTime"
                minOccurs="1" maxOccurs="1" />
            <xsd:element ref="wsrl:TerminationTime"
                minOccurs="1" maxOccurs="1" />

            <!-- SubscriptionManager specific -->
            <xsd:element ref="wsnt:ConsumerReference"
                minOccurs="1" maxOccurs="1" />
            <xsd:element ref="wsnt:TopicExpression"
                minOccurs="1" maxOccurs="1" />
            <xsd:element ref="wsnt:UseNotify"
                minOccurs="1" maxOccurs="1" />
            <xsd:element ref="wsnt:Precondition"
                minOccurs="0" maxOccurs="1" />
            <xsd:element ref="wsnt:Selector"
                minOccurs="0" maxOccurs="1" />
            <xsd:element ref="wsnt:SubscriptionPolicy"
                minOccurs="0" maxOccurs="1" />
            <xsd:element ref="wsnt:CreationTime"
                minOccurs="0" maxOccurs="1" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<!-- ===== Message Helper Types ===== -->
<xsd:complexType name="NotificationMessageHolderType" >
    <xsd:sequence>
        <xsd:element name="Topic"
            type="wsnt:TopicExpressionType"
            minOccurs="1" maxOccurs="1" />
        <xsd:element name="ProducerReference"
            type="wsa:EndpointReferenceType"
            minOccurs="1" maxOccurs="1" />
        <xsd:element name="Message" type="xsd:anyType"
            minOccurs="1" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>

<!-- ===== Common fault information to carry in all fault messages ===== -->
<xsd:complexType name="BaseFaultType">
    <xsd:sequence>
        <xsd:element name="Timestamp" type="xsd:dateTime"
            minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

```

```

        <xsd:element name="Originator"
            type="wsa:EndpointReferenceType"
            minOccurs="0" maxOccurs="1"/>
        <xsd:element name="ErrorCode"
            minOccurs="0" maxOccurs="1">
            <xsd:complexType>
                <xsd:complexContent mixed="true">
                    <xsd:extension base="xsd:anyType">
                        <xsd:attribute name="dialect"
type="xsd:anyURI"
                            use="required"/>
                    </xsd:extension>
                </xsd:complexContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="Description" type="xsd:string"
            minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="FaultCause" type="wsnt:BaseFaultType"
            minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<!-- ===== Message Types for NotificationConsumer ===== -->
    <xsd:element name="Notify" >
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="NotificationMessage"
                    type="wsnt:NotificationMessageHolderType"
                    minOccurs="1" maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

<!-- ===== Message Types for NotificationProducer ===== -->
    <xsd:element name="Subscribe" >
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="ConsumerReference"
                    type="wsa:EndpointReferenceType"
                    minOccurs="1" maxOccurs="1" />
                <xsd:element name="TopicExpression"
                    type="wsnt:TopicExpressionType"
                    minOccurs="1" maxOccurs="1" />
                <xsd:element name="UseNotify"
                    type="xsd:boolean" default="true"
                    minOccurs="0" maxOccurs="1" />
                <xsd:element name="Precondition"
                    type="wsrp:QueryExpressionType"
                    minOccurs="0" maxOccurs="1" />
                <xsd:element name="Selector"
                    type="wsrp:QueryExpressionType"
                    minOccurs="0" maxOccurs="1" />
                <xsd:element name="SubscriptionPolicy"
                    type="xsd:anyType"
                    minOccurs="0" maxOccurs="1" />
                <xsd:element name="InitialTerminationTime"
                    type="xsd:dateTime"

```

```
        minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:element name="SubscribeResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="SubscriptionReference"
        type="wsa:EndpointReferenceType"
        minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="GetCurrentMessage">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Topic"
        type="wsnt:TopicExpressionType" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="GetCurrentMessageResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="ResourceUnknownFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsnt:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="ResourceUnknownFault"
  type="wsnt:ResourceUnknownFaultType"/>

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

<xsd:complexType name="TopicPathDialectUnknownFaultType">
  <xsd:complexContent>
    <xsd:extension base="wsnt:BaseFaultType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="TopicPathDialectUnknownFault"
  type="wsnt:TopicPathDialectUnknownFaultType"/>

<xsd:complexType name="InvalidTopicExpressionFaultType">
```

```

        <xsd:complexContent>
            <xsd:extension base="wsnt:BaseFaultType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="InvalidTopicExpressionFault"
        type="wsnt:InvalidTopicExpressionFaultType"/>

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

    <xsd:complexType name="NoCurrentMessageOnTopicFaultType">
        <xsd:complexContent>
            <xsd:extension base="wsnt:BaseFaultType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="NoCurrentMessageOnTopicFault"
        type="wsnt:NoCurrentMessageOnTopicFaultType"/>

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

    <xsd:element name="PauseSubscription" />

    <xsd:element name="PauseSubscriptionResponse" />

    <xsd:element name="ResumeSubscription" />

    <xsd:element name="ResumeSubscriptionResponse" />

    <xsd:complexType name="PauseFailedFaultType">
        <xsd:complexContent>
            <xsd:extension base="wsnt:BaseFaultType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="PauseFailedFault"
        type="wsnt:PauseFailedFaultType"/>

    <xsd:complexType name="ResumeFailedFaultType">
        <xsd:complexContent>
            <xsd:extension base="wsnt:BaseFaultType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="ResumeFailedFault"
        type="wsnt:ResumeFailedFaultType"/>

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

<!-- ===== NotificationConsumer::Notify =====>
    Notify(
        NotificationMessage

```

```

    (TopicExpression, ProducerReference, Message)*
    returns: n/a (one way)
-->
<wsdl:message name="Notify">
  <wsdl:part name="Notify" element="wsnt:Notify"/>
</wsdl:message>

<!-- ===== NotificationProducer::Subscribe =====>
Subscribe(
  (ConsumerEndpointReference, TopicExpression, [UseNotify],
  [Precondition], [Selector], [SubscriptionPolicy],
  [InitialTerminationTime])
  returns: WS-Resource qualified EPR to a Subscription
-->
  <wsdl:message name="SubscribeRequest" >
    <wsdl:part name="SubscribeRequest"
      element="wsnt:Subscribe"/>
  </wsdl:message>

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

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

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

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

<!-- ===== NotificationProducer::GetCurrentMessage =====>
GetCurrentMessage(topicExpression)
  returns: a NotificationMessage (xsd:any)
-->
  <wsdl:message name="GetCurrentMessageRequest">
    <wsdl:part name="GetCurrentMessageRequest"
      element="wsnt:GetCurrentMessage"/>
  </wsdl:message>

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

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

```



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

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

<!-- ===== SubscriptionManager::PauseSubscription =====
PauseSubscription()
returns: empty
-->
<wsdl:message name="PauseSubscriptionRequest">
  <wsdl:part name="PauseSubscriptionRequest"
    element="wsnt:PauseSubscription"/>
</wsdl:message>

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

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

<!-- ===== SubscriptionManager::ResumeSubscription =====
ResumeSubscription()
returns: empty
-->
<wsdl:message name="ResumeSubscriptionRequest">
  <wsdl:part name="ResumeSubscriptionRequest"
    element="wsnt:ResumeSubscription"/>
</wsdl:message>

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

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

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

```

<!-- ===== NotificationProducer PortType Definition ===== -->
<wsdl:portType name="NotificationProducer"
  wsrp:ResourceProperties ="wsnt:NotificationProducerRP">

  <!-- ===== extends wsrp:ResourceProperties ===== -->
  <wsdl:operation name="GetResourceProperty">
    <wsdl:input name="GetResourcePropertyRequest"
      message="wsrp:GetResourcePropertyRequest" />
    <wsdl:output name="GetResourcePropertyResponse"
      message="wsrp:GetResourcePropertyResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrp:ResourceUnknownFault" />
    <wsdl:fault name="InvalidResourcePropertyQNameFault"
      message="wsrp:InvalidResourcePropertyQNameFault"
    />
  />

  </wsdl:operation>

  <!-- ===== NotificationProducer Specific ===== -->
  <wsdl:operation name="Subscribe">
    <wsdl:input message="wsnt:SubscribeRequest" />
    <wsdl:output message="wsnt:SubscribeResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsnt:ResourceUnknownFault" />
    <wsdl:fault name="SubscribeCreationFailedFault"
      message="wsnt:SubscribeCreationFailedFault"/>
    <wsdl:fault name="TopicPathDialectUnknownFault"
      message="wsnt:TopicPathDialectUnknownFault"/>
  </wsdl:operation>

  <wsdl:operation name="GetCurrentMessage">
    <wsdl:input message="wsnt:GetCurrentMessageRequest"/>
    <wsdl:output message="wsnt:GetCurrentMessageResponse"/>
    <wsdl:fault name="ResourceUnknownFault"
      message="wsnt:ResourceUnknownFault" />
    <wsdl:fault name="InvalidTopicExpressionFault"
      message="wsnt:InvalidTopicExpressionFault" />
    <wsdl:fault name="TopicNotSupportedFault"
      message="wsnt:TopicNotSupportedFault" />
    <wsdl:fault name="NoCurrentMessageOnTopicFault"
      message="wsnt:NoCurrentMessageOnTopicFault" />
  </wsdl:operation>
</wsdl:portType>

<!-- ===== SubscriptionManager PortType Definition ===== -->
<wsdl:portType name="SubscriptionManager"
  wsrp:ResourceProperties ="wsnt:SubscriptionManagerRP">
  <!-- ===== extends wsrp:ResourceProperties ===== -->
  <wsdl:operation name="GetResourceProperty">
    <wsdl:input name="GetResourcePropertyRequest"
      message="wsrp:GetResourcePropertyRequest" />
    <wsdl:output name="GetResourcePropertyResponse"
      message="wsrp:GetResourcePropertyResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrp:ResourceUnknownFault" />
    <wsdl:fault name="InvalidResourcePropertyQNameFault"
      message="wsrp:InvalidResourcePropertyQNameFault"
    />
  />

```

```
</wsdl:operation>

<!-- === extends wsrl:ImmediateResourceTermination ===== -->
<wsdl:operation name="Destroy">
  <wsdl:input message="wsrl:DestroyRequest" />
  <wsdl:output message="wsrl:DestroyResponse" />
  <wsdl:fault name="ResourceUnknownFault"
    message="wsrl:ResourceUnknownFault" />
  <wsdl:fault name="ResourceNotDestroyedFault"
    message="wsrl:ResourceNotDestroyedFault" />
</wsdl:operation>

<!-- === extends wsrl:ScheduledResourceTermination ===== -->
<wsdl:operation name="SetTerminationTime">
  <wsdl:input message="wsrl:SetTerminationTimeRequest" />
  <wsdl:output message="wsrl:SetTerminationTimeResponse" />
  <wsdl:fault name="ResourceUnknownFault"
    message="wsrl:ResourceUnknownFault" />
  <wsdl:fault name="UnableToSetTerminationTimeFault"
    message="wsrl:UnableToSetTerminationTimeFault" />
  <wsdl:fault name="TerminationTimeChangeRejectedFault"
    message="wsrl:TerminationTimeChangeRejectedFault"
/>
</wsdl:operation>

<!-- ===== SubscriptionManager specific operations ===== -->
<wsdl:operation name="PauseSubscription">
  <wsdl:input message="wsnt:PauseSubscriptionRequest"/>
  <wsdl:output message="wsnt:PauseSubscriptionResponse"/>
  <wsdl:fault name="ResourceUnknownFault"
    message="wsnt:ResourceUnknownFault" />
  <wsdl:fault name="PauseFailedFault"
    message="wsnt:PauseFailedFault" />

</wsdl:operation>
<wsdl:operation name="ResumeSubscription">
  <wsdl:input message="wsnt:ResumeSubscriptionRequest"/>
  <wsdl:output message="wsnt:ResumeSubscriptionResponse"/>
  <wsdl:fault name="ResourceUnknownFault"
    message="wsnt:ResourceUnknownFault" />
  <wsdl:fault name="ResumeFailedFault"
    message="wsnt:ResumeFailedFault" />
</wsdl:operation>
</wsdl:portType>
</wsdl:definitions>
```