Open Geospatial Consortium

Submission Date: 2015-12-19

Approval Date: 2016-02-19

Publication Date: 2016-08-22

External identifier of this OGC® document: http://www.opengis.net/doc/is/pubsub-soap/1.0

Internal reference number of this OGC® document: 13-133r1

Version: 1.0

Category: OGC® Implementation Standard

Editors: Aaron Braeckel, Lorenzo Bigagli

OGC® Publish/Subscribe Interface Standard 1.0

SOAP Protocol Binding Extension

Copyright notice

Warning

Although OGC 13-133r1 is an OGC Standard. This formatted version of this document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard. The normative version is available at:
http://docs.opengeospatial.org/is/13-133r1/13-133r1.html
Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Standard
Document subtype:	Implementation
Document stage:	Approved
Document language:	English

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without soapriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

Contents

Tables

i. Abstract

Publish/Subscribe 1.0 is an interface specification that supports the core components and concepts of the Publish/Subscribe message exchange pattern with OGC Web Services. The Publish/Subscribe pattern complements the Request/Reply pattern historically specified by many OGC Web Services. This specification may be used either in concert with, or independently of, existing OGC Web Services to publish data of interest to subscribers.

Publish/Subscribe 1.0 primarily addresses subscription management capabilities such as creating a subscription, renewing a subscription, and unsubscribing. However, this standard also allows Publish/Subscribe services to advertise and describe the supported message delivery protocols such as SOAP messaging, ATOM, and AMQP. Message delivery protocols should be considered to be independent of the Publish/Subscribe 1.0 standard. Therefore OGC Publish/Subscribe only includes metadata relating to message delivery protocols in sufficient detail to allow for different implementations of Publish/Subscribe 1.0 to interoperate.

This specification defines an extension to the OGC Publish/Subscribe (PubSub) 1.0 Core to allow for Publish/Subscribe communications using the SOAP protocol.

ii. Keywords

The following are keywords to be used by search engines and document catalogues.

Ogcdoc pubsub core standard soap soap protocol binding

iii. Preface

The OGC® Abstract Specification does not require any changes to accommodate the technical contents of this document.

No future work currently anticipated.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

iv. Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium Inc.

- National Center for Atmospheric Research (NCAR)

- National Research Council of Italy (CNR)

v. Submitters

All questions regarding this submission should be directed to the editor or the submitters:

Name	Company
Aaron Braeckel	NCAR
Lorenzo Bigagli	CNR

1. Scope

This OGC standard defines core concepts and mechanisms for enabling the Publish/Subscribe messaging pattern with OGC Web Services using the SOAP protocol. This standard utilizes core concepts and mechanisms from the OGC® Publish/Subscribe Interface Standard 1.0 Core.

NOTE This standard does not address access control.

2. Conformance

This OGC interface standard defines conformance classes for Publish/Subscribe behaviour with a SOAP protocol binding, as shown in Table 1. Requirements for the following standardization target types are considered (described further in Clause 4):

- **Publisher** – an entity that can be used to manage subscriptions.

This document establishes several requirements classes shown below in Table 1. Requirements and conformance test URIs defined in this document are relative to http://www.opengis.net/spec/pubsub/1.0/.

According to the target type hierarchy, this specification defines the following conformance levels:

Table 1: Conformance Classes

Conformance Class Name	Conformance Target	Operation or behavior	Conformance Class URI
SOAP Basic Receiver	Receiver	The **Receiver** shall implement the Basic Receiver conformance class from Publish/Subscribe Core. The **Receiver** shall also implement the following	**/conf/soap/basic-receiver**

		operation from the WS-BaseNotification NotificationConsumer interface: □ *Notify*	
SOAP Basic Publisher	Publisher	The **Publisher** shall implement the Basic Publisher conformance class from Publish/Subscribe Core. The **Publisher** shall also implement the following operations from the WS-BaseNotification NotificationProducer interface: □ *Subscribe* The **Publisher** shall also implement the following operations from the WS-BaseNotification Base SubscriptionManager interface: □ *Renew* □ *Unsubscribe*	**/conf/soap/basic-publisher**
SOAP Standalone Publisher	Publisher	The **Publisher** shall implement the SOAP Basic Publisher conformance class. Additionally the **Publisher** shall implement the following operations: □ *GetCapabilities* □ *GetSubscription*	**/conf/soap/standalone-publisher**
SOAP Pausable Publisher	Publisher	The **Publisher** shall implement the SOAP Basic Publisher conformance class, and the following operations from the WS-BaseNotification PausableSubscriptionManager interface: □ PauseSubscription □ ResumeSubscription	**/conf/soap/pausable-publisher**
SOAP Message Batching Publisher	Publisher	The **Publisher** shall implement the SOAP Basic Publisher conformance class. Additionally the **Publisher** shall enable Subscribers to specify	**/conf/soap/message-batching-publisher**

		message-batching capabilities on the *Subscribe* operation. The **Publisher** shall follow message batching directives specified by the Subscriber when delivering messages	
SOAP Heartbeat Publisher	Publisher	The **Publisher** shall implement the SOAP Basic Publisher conformance class. Additionally the **Publisher** shall allow Subscribers to specify heartbeat capabilities on the *Subscribe* operation. The **Publisher** shall follow heartbeat directives specified by the Subscriber and send regular heartbeat messages to allow Receivers to detect a failure or communications problem	**/conf/soap/heartbeat-publisher**
SOAP Brokering Publisher	Publisher	The **Publisher** shall implement the SOAP Standalone Publisher conformance class. Additionally the **Publisher** shall support the management of brokered Publishers: □ *RegisterPublisher* □ *RemovePublisher* □ *GetPublisher* The **Publisher** shall receive messages from the brokered Publishers and republish them	**/conf/soap/brokering-publisher**
SOAP Publication Manager	Publisher	The **Publisher** shall implement the SOAP Basic Publisher conformance class. Additionally the **Publisher** shall support the creation, removal, and subscriptions to user-defined publications: □ *CreatePublication* □ *RemovePublication*	**/conf/soap/publication-manager**
SOAP HTTP Delivery Publisher	Publisher	The **Publisher** shall implement the SOAP Basic Publisher conformance class. The publisher shall implement	**/conf/soap/http-delivery-publisher**

		message delivery (notification) by initiating HTTP POST with SOAP/XML message delivery to SOAP endpoints that implement the WS-BaseNotification NotificationConsumer interface.	
SOAP Reliable Publisher	Publisher	The **Publisher** shall implement the SOAP Basic Publisher conformance class. The **Publisher** shall also implement the following operations from the [OASIS WS-ReliableMessaging] interface: □ *CreateSequence* □ *CloseSequence* □ *TerminateSequence*	**/conf/soap/reliable-publisher**

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site[1].

All requirements-classes and conformance-classes described in this document are owned by the standard(s) identified.

3. References

This *OGC Publish/Subscribe 1.0 SOAP Protocol Binding* standard consists of the present document and an XML Schema.

The complete *OGC Publish/Subscribe 1.0* specification is identified by OGC URI http://www.opengis.net/spec/pubsub/1.0. It is available for download from http://www.opengeospatial.org/standards/pubsub. The XML Schema is posted on-line at http://schemas.opengis.net/pubsub/1.0 as part of the OGC schema repository. In the event of a discrepancy between this document and schema repository versions of the XML Schema files, the schema repository shall be considered authoritative.

The following normative documents contain provisions, which, through reference in this text, constitute provisions of this document. For dated references, subsequent

[1] www.opengeospatial.org/cite

amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

ISO/TS 19103:2005, *Geographic information* — Conceptual schema language

OGC 06-121r3, *OGC Web Services Common Specification*, OGC® Implementation Standard (9 February 2009)

W3C XML Namespaces, *Namespaces in XML*, W3C Recommendation (14 January 1999)

W3C XML Schema Part 1, *XML Schema Part 1: Structures*, W3C Recommendation (2 May 2001)

W3C XML Schema Part 2, *XML Schema Part 2: Datatypes*, W3C Recommendation (2 May 2001).

W3C WS-Addressing Core, Web Services Addressing 1.0 – Core, W3C Recommendation (9 May 2006)

OASIS WS-BaseNotification, *Web Services Base Notification 1.3*, OASIS Standard (1 October 2006)

OASIS WS-BrokeredNotification, *Web Services Brokered Notification 1.3*, OASIS Standard (1 October 2006)

W3C SOAP, *Simple Object Access Protocol (SOAP) 1.2*, W3C Note (27 April 2007)

W3C WSDL, *Web Services Description Language (WSDL) 1.1*, W3C Note (15 Mar 2001)

IETF RFC 2616, *Hypertext Transfer Protocol – HTTP/1.1*, IETF Request for Comments (June 1999)

OGC 13-131, *OGC Publish/Subscribe 1.0 Core*, OGC® Implementation Standard

OASIS WS-BaseFaults, *Web Services Base Faults 1.2*, OASIS Standard (1 April 2006)

OASIS WS-ReliableMessaging, *Web Services Reliable Messaging 1.2* (2 February 2009)

4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of

International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

This document primarily uses terms and definitions from OGC Publish/Subscribe Core. The terms and definitions given in the other normative references also apply.

4.1
Reliable Publisher
A publisher of messages that offers capabilities to detect and recover from message delivery losses, whether caused by network failures, software failures, hardware failures, or other causes.

5. Conventions

5.1 Abbreviations

In this document the following abbreviations and acronyms are used or introduced:

HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
OGC	Open Geospatial Consortium
OMG	Object Management Group
PubSub	Publish/Subscribe
SOAP	Simple Object Access Protocol
UML	Unified Modeling Language (an object modeling language)
WSA	Web Services Addressing
WSN	Web Services Notification
XML	eXtensible Markup Language

5.2 UML Notation

All symbols used in this document are UML 2 (Unified Modeling Language) as defined by OMG and accepted as a publicly available standard by ISO in its earlier 1.3 version.

All classes in this standard are *extensible* and may be extended with application- or domain-specific content via `Extension` blocks.

NOTE The UML shown in this standard is considered conceptual and abstract, and should not be interpreted as an implementation strategy for bindings that extend and implement this standard. For example, TM_Instant from ISO 19108 is used to represent time instants for conceptual clarity, but bindings and implementations of this standard may realize TM_Instant as a GML TimeInstant, an ISO 8601 date string, or any other representation that is consistent with TM_Instant.

5.3 Referencing Conventions

This standard references UML classes from other specifications. When referencing UML classes not defined in this standard, the class name will be qualified with the document of origin. For example, a reference to the ISO 19108 TM_Instant is referenced as:

> TM_Instant [see ISO/TS 19103:2005]

Many referenced UML classes are instantiated as XML schema, such as the GML realization of ISO TC211 standards. This standard only normatively references UML representations.

5.4 XML Namespace Conventions

This standard uses a number of namespace prefixes throughout; they are listed in the following table. Note that the choice of any namespace prefix is not semantically significant.

Table 2: XML Namespace Prefixes

Prefix	Namespace
pubsub	http://www.opengis.net/pubsub/1.0
xsd	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance
soap11	http://schemas.xmlsoap.org/soap/
soap12	http://www.w3.org/2003/05/soap-envelope
wsa	http://www.w3.org/2005/08/addressing
wsn-b	http://docs.oasis-open.org/wsn/b-2
wsn-br	http://docs.oasis-open.org/wsn/br-2

6. SOAP Publish/Subscribe Overview

This document is an extension of the OGC Publish/Subscribe Core 1.0 document using SOAP web services.

This document specifies Publish/Subscribe capabilities using the WS-BaseNotification specification from OASIS. This specification extends WS-BaseNotification operations to include discovery and metadata operations such as *GetCapabilities*.

This document does not mandate a single, specific version of SOAP to be used in message exchanges. This enables established systems that intend to implement OWS Publish/Subscribe functionality to require any version of SOAP for Publisher operations. For maximum interoperability, it is recommended that SOAP Publish/Subscribe services support SOAP version 1.2 (see W3C SOAP) for requests and responses.

NOTE Information on version mismatch can also be communicated per actual message exchange – see http://www.w3.org/TR/soap12-part1/#version for further details.

For maximum interoperability it is suggested that SOAP implementations be compliant with the OASIS WS-Interoperability (WS-I) Basic Profile.

For disambiguation, the following table lists SOAP and [OASIS WS-BaseNotification] terms and their corresponding Publish/Subscribe concepts.

Table 3: SOAP/OASIS WS-BaseNotification Terminology Mapping

SOAP/OASIS WS-BaseNotification Term	Publish/Subscribe Synonym
SOAP Transport Method **WSDL Binding**	**Delivery Method**
Publisher This informal term is referenced, but not formally defined, in [OASIS WS-BrokeredNotification] p. 12	**Sender** Entity that sends messages to Receivers; may (but need not) be the initial creator/producer of the data to be transported in the message payload
SubscriptionManager - *Unsubscribe()* - *Renew()*	**Publisher** - *Unsubscribe()* - *Renew()*
Subscriber	**Subscriber**

NotificationProducer - *Subscribe()* - *GetCurrentMessage()*	**Publisher** - *Subscribe()*
NotificationConsumer - *Notify()*	**Receiver** - *Notify()*
NotificationBroker - *Subscribe()* - *Notify()*	**Broker** - *Subscribe()* - *Receive()*
Topic	No parallel in OGC Publish/Subscribe

Terms as defined by WS-BaseNotification will be explicitly marked as such in this document. If no qualifier (such as "Publish/Subscribe" or "OASIS WS-BaseNotification") is provided for a given term, the Publish/Subscribe terminology is assumed.

7. Requirements Class: SOAP Basic Receiver extends Basic Receiver

Requirements Class	
http://www.opengis.net/spec/pubsub/1.0/req/soap/basic-receiver	
Target type	Receiver
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/core/basic-receiver**
Requirement	/req/soap/basic-receiver/notify

This Requirements Class specifies the basic operation of a SOAP Basic Receiver:

> ***Notify*** – delivery of a message to the Receiver. In the context of Publish/Subscribe this is often the delivery of a message which matches the filter criteria of a given subscription.

7.1 Notify operation

The *Notify* operation from WS-BaseNotification `NotificationConsumer` interface is implemented by a Receiver to allow the delivery of SOAP messages. This interface is specified in [OASIS WS-BaseNotification] Clause 3.

Requirement
/req/soap/basic-receiver/notify
Req 1 A **Receiver** shall offer the *Notify* operation as specified in the WS-BaseNotification `NotificationConsumer` interface

7.1.1 Response

There is no response from the *Notify* operation.

7.1.2 Exceptions

No exceptions are defined for the *Notify* operation.

8. Requirements Class: SOAP Basic Publisher extends Basic Publisher

Requirements Class	
http://www.opengis.net/spec/pubsub/1.0/req/soap/basic-publisher	
Target type	Publisher
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/core/basic-publisher**
Requirement	/req/soap/basic-publisher/envelope-body
Requirement	/req/soap/basic-publisher/getcapabilities-service-metadata
Requirement	/req/soap/basic-publisher/subscribe
Requirement	/req/soap/basic-publisher/unsubscribe
Requirement	/req/soap/basic-publisher/renew
Requirement	/req/soap/basic-publisher/subscribe/dialect-set
Requirement	/req/soap/basic-publisher/subscribe/dialect-valid
Requirement	/req/soap/basic-publisher/subscribe/publication-id
Requirement	/req/soap/basic-publisher/renew-nil-termination-time
Requirement	/req/soap/basic-publisher/unsubscribe/publication-id
Requirement	/req/soap/basic-publisher/fault-issuance
Requirement	/req/soap/basic-publisher/fault-detail

This Requirements Class specifies the basic Publish/Subscribe operations of a Publisher using the SOAP protocol. The SOAP operations in this Requirements Class are implemented as operations from the `NotificationProducer` interface from [OASIS WS-BaseNotification]:

> ***Subscribe*** – allows for the creation of subscriptions against publications offered by a Publisher. This is implemented as the *Subscribe* operation from [OASIS WS-BaseNotification];
>
> ***Renew*** – allows for the renewal of a subscription on a Publisher. This is implemented as the *Renew* operation from [OASIS WS-BaseNotification]; and

Unsubscribe – allows for removal of a subscription on a Publisher. This is implemented as the *Unsubscribe* operation from [OASIS WS-BaseNotification].

This Requirements Class extends the Publish/Subscribe Core Basic Publisher, and as such also returns capabilities metadata that is offered in response to a *GetCapabilities* operation. Therefore this capabilities metadata may either be offered as a Publish/Subscribe *GetCapabilities* operation as described in the Publish/Subscribe Core Standalone Publisher requirements class or through a *GetCapabilities* operation defined by another OGC Web Service such as the OGC Web Feature Service (WFS).

Requirement
/req/soap/basic-publisher/envelope-body
Req 2 A **Publisher** shall produce SOAP messages with one SOAP Envelope containing exactly one SOAP Body

Requirement
/req/soap/basic-publisher/getcapabilities-service-metadata
Req 3 A **Publisher** shall specify the HTTP POST request encodings accepted by including an `ows:Constraint` element, with “PostEncoding” as the value of the `name` attribute and with a value of “SOAP” in *GetCapabilities* responses

Requirement
/req/soap/basic-publisher/subscribe
Req 4 A **Publisher** shall offer the WS-BaseNotification NotificationProducer *Subscribe* operation

Requirement
/req/soap/basic-publisher/unsubscribe
Req 5 A **Publisher** shall offer the WS-BaseNotification SubscriptionManager *Unsubscribe* operation

Requirement
/req/soap/basic-publisher/renew
Req 6 A **Publisher** shall offer the WS-BaseNotification SubscriptionManager *Renew* operation

8.1 Usage of WS-BaseNotification

OGC Publish/Subscribe filtering information is sent in the `/wsn-b:Subscribe/wsn-b:/Filter/wsn-b:MessageContent` section of the request. In this case, the `Dialect` attribute on the `wsn-b:MessageContent` element shall be set appropriately to represent the filtering dialect. The content of the `Dialect` attribute should be one of the `FilterCapabilities` identifiers advertised by the Publisher in the *GetCapabilities* operation.

Subscribe requests must also include the `pubsub:publicationIdentifier` element as a child of the `wsn-b:Subscribe` element.

Requirement
/req/soap/basic-publisher/subscribe/dialect-set
Req 7 A **Publisher** shall raise an Exception if the `/wsn-b:Subscribe/wsn-b:Filter/wsn-b:MessageContent/@Dialect` attribute is not set on a *Subscribe* request

Requirement
/req/soap/basic-publisher/subscribe/dialect-valid
Req 8 A **Publisher** shall raise an Exception if the `/wsn-b:Subscribe/wsn-b:Filter/wsn-b:MessageContent/@Dialect` attribute is not one of the advertised `FilterCapabilities` identifiers

Requirement
/req/soap/basic-publisher/subscribe/publication-id
Req 9 A **Publisher** shall raise an Exception if the `/wsn-b:Subscribe/pubsub:PublicationIdentifier` element is not present on a *Subscribe* request

8.2 Subscribe Operation

8.2.1 Subscription

The following table summarizes how the Publish/Subscribe Subscription is realized in SOAP.

Table 4: Realization of Publish/Subscribe Subscription in SOAP

Publish/Subscribe Core 1.0 – Subscription Property Name	Realization in SOAP binding	Description

identifier	wsn-b:SubscriptionReference	Defined by WS-BaseNotification
publicationIdentifier	pubsub:PublicationIdentifer	Defined in Publish/Subscribe schemas
terminationTime	wsn-b:TerminationTime	Defined by WS-BaseNotification
filter	wsn-b:Filter/ wsn-b:MessageContent	Content depends on filtering functionality supported by the Publisher.
filterLanguageId	wsn-b:Filter/wsn-b:MessageContent/@Dialect	Defined by WS-BaseNotification
deliveryLocation	wsn-b: ConsumerReference	As defined by WS-BaseNotification
deliveryMethod	xs:anyURI	Delivery method URIs are defined by Publish/Subscribe delivery conformance classes
deliveryParameter	xs:anyType	Delivery parameters are defined by Publish/Subscribe delivery conformance classes

8.3 Renew Operation

The semantics for the termination time set in the request are the same in OGC Publish/Subscribe and [OASIS WS-BaseNotification]. [OASIS WS-BaseNotification] also specifies that an exception is thrown if the requested termination time is not in the future.

However, use of the xsi:nil attribute with value “true” on the wsn-b:Renew/wsn-b:TerminationTime to request an infinite duration (see [OASIS WS-BaseNotification] section 6.1.1) is prohibited.

Requirement
/req/soap/basic-publisher/renew-nil-termination-time
Req 10 A **Publisher** shall throw an Exception if it receives a *Renew* request that contains

a `wsn-b:TerminationTime` with `xsi:nil` attribute set to “true”

Renew requests must include the `pubsub:publicationIdentifier` element as a child of the `wsn-b:Renew` element.

The response semantics for the Renew operation are the same in OGC Publish/Subscribe and [OASIS WS-BaseNotification]. In both cases the response indicates that the subscription identified in the request (explicitly in the case of OGC Publish/Subscribe, implicitly in the case of [OASIS WS-BaseNotification]) has been updated to terminate at the requested point in time.

NOTE [OASIS WS-BaseNotification] supports functionality to differentiate and handle different time on the client and server roles (NotificationProducer, SubscriptionManager). OGC Publish/Subscribe does not make this distinction.

8.4 Unsubscribe operation

Unsubscribe requests must include the `pubsub:publicationIdentifier` element as a child of the `wsn-b:Unsubscribe` element.

Requirement
/req/soap/basic-publisher/unsubscribe/publication-id
Req 11 A **Publisher** shall raise an Exception if the `/wsn-b:Unsubscribe/pubsub:PublicationIdentifier` element is not present on a *Unsubscribe* request

8.5 Exception and Fault Usage

In the event that a Publisher encounters an error while processing a request or receives an invalid request, it shall generate a SOAP Fault. Exceptions defined by Publish/Subscribe Core and Publish/Subscribe extensions are encoded as an `ExceptionReport` in the `FaultCause` section of each SOAP Fault. SOAP Faults types and usage are described in [OASIS WS-BaseNotification] and associated specifications. Therefore the Fault type (such as `UnacceptableTerminationTimeFault`) and ExceptionReport type (such as `TerminationUnacceptable`) portions of each issued Fault may be considered independent. Each Exception will be of the appropriate type as described in Publish/Subscribe Core.

Requirement
/req/soap/basic-publisher/fault-issuance
Req 12 A **Publisher** shall issue SOAP Faults as described in [OASIS WS-BaseFaults] in accordance with [OASIS WS-BaseNotification]

Requirement
/req/soap/basic-publisher/fault-detail
Req 13 A **Publisher** shall issue SOAP Faults that incorporate an ExceptionReport valid according to Clause 8 of the OWS Common Specification [OGC 06-121r3] in the `FaultCause` element of the WSRF:BaseFault

For the *Subscribe*, *Unsubscribe*, and *Renew* operations the following Fault and Exception codes are used.

NOTE Topic-related Exceptions are mapped below for completeness as they are defined in [OASIS WS-BaseNotification], but are not utilized in this Standard.

Table 5: BasicPublisher Exception Mappings

Exception Code	**SOAP Fault**
***Subscribe* operation**	
InvalidPublicationIdentifier	ResourceUnknownFault
TerminationUnacceptable	UnacceptableInitialTerminationTimeFault
PastTermination	UnacceptableInitialTerminationTimeFault
InvalidDeliveryMethod	BaseFault
InvalidFilter	InvalidFilterFault
MissingParameterValue	SubscribeCreationFailedFault BaseFault
InvalidParameterValue	SubscribeCreationFailedFault TopicExpressionDialectUnknownFault TopicNotSupportedFault InvalidTopicExpressionFault InvalidProducerPropertiesExpressionFault InvalidMessageContentExpressionFault UnrecognizedPolicyRequestFault UnsupportedPolicyRequestFault NotifyMessageNotSupportedFault BaseFault
NoApplicableCode	SubscribeCreationFailedFault BaseFault
***Unsubscribe* operation**	
InvalidSubscriptionIdentifier	ResourceUnknownFault
NoApplicableCode	BaseFault

***Renew* operation**	
InvalidSubscriptionIdentifier	ResourceUnknownFault
TerminationUnacceptable	UnacceptableTerminationTimeFault
PastTermination	UnacceptableTerminationTimeFault
MissingParameterValue	BaseFault
InvalidParameterValue	BaseFault
NoApplicableCode	BaseFault

8.5.1 SOAP Fault Examples

A sample SOAP 1.1 fault encoding is as follows:

```
<soap11:Envelope xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
    xmlns:wsrf-bf=”http://docs.oasis-open.org/wsrf/bf-2”
    xmlns:soap11=”http://schemas.xmlsoap.org/soap/envelope/”
    xmlns:wsa=”http://www.w3.org/2005/08/addressing”
    xmlns:wsnt=”http://docs.oasis-open.org/wsn/b-2”
    xmlns:pubsub=”http://www.opengis.net/pubsub/1.0”
    xsi:schemaLocation=”http://schemas.xmlsoap.org/soap/envelope/
http://schemas.xmlsoap.org/soap/envelope/
    http://www.opengis.net/pubsub/1.0
http://schemas.opengis.net/pubsub/1.0.0/pubsubAll.xsd
    http://docs.oasis-open.org/wsn/b-2 http://docs.oasis-open.org/wsn/b-2.xsd”>
    <soap11:Header>
        <wsa:Action>http://docs.oasis-open.org/wsn/fault</wsa:Action>
    </soap11:Header>
    <soap11:Body>
        <soap11:Fault>
            <faultcode>soap11:Client</faultcode>
            <faultstring>The publication identifier ‘xyz’ is invalid</faultstring>
            <detail>
                <wsnt:SubscribeCreationFailedFault>
                    <wsrf-bf:Timestamp>
                        2005-05-04T20:18:44.970Z
                    </wsrf-bf:Timestamp>
                    <wsrf-bf:Description>
                        The publication identifier ‘xyz’ is invalid
                    </wsrf-bf:Description>
                    <wsrf-bf:FaultCause>
                        <ows:ExceptionReport version=”1.0.0”
xmlns:ows=”http://www.opengis.net/ows/1.1”>
                            <ows:Exception exceptionCode=”InvalidPublicationIdentifier”
locator=”xyz”>
                                <ows:ExceptionText>
                                    The publication identifier ‘xyz’ is invalid
                                </ows:ExceptionText>
                            </ows:Exception>
                        </ows:ExceptionReport>
                    </wsrf-bf:FaultCause>
                </wsnt:SubscribeCreationFailedFault>
            </detail>
        </soap11:Fault>
    </soap11:Body>
</soap11:Envelope>
```

A sample SOAP 1.2 example is as follows:

```
<soap12:Envelope
    xmlns:soap12=”http://www.w3.org/2003/05/soap-envelope”
    xmlns:wsa=”http://www.w3.org/2005/08/addressing”
    xmlns:wsrf-bf=”http://docs.oasis-open.org/wsrf/bf-2”
    xmlns:wsrf-r=”http://docs.oasis-open.org/wsrf/r-2”
    xmlns:wsnt=”http://docs.oasis-open.org/wsn/b-2”
    xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
    xsi:schemaLocation=”http://schemas.xmlsoap.org/soap/envelope/
http://schemas.xmlsoap.org/soap/envelope/
    http://www.opengis.net/pubsub/1.0
http://schemas.opengis.net/pubsub/1.0.0/pubsubAll.xsd
    http://www.w3.org/2003/05/soap-envelope http://www.w3.org/2003/05/soap-envelope
    http://docs.oasis-open.org/wsrf/r-2 http://docs.oasis-open.org/wsrf/r-2.xsd
    http://docs.oasis-open.org/wsn/b-2 http://docs.oasis-open.org/wsn/b-2.xsd”>
    <soap12:Header>
        <wsa:Action>http://docs.oasis-open.org/wsn/fault</wsa:Action>
    </soap12:Header>
    <soap12:Body>
        <soap12:Fault>
            <soap12:Code>
                <soap12:Value>soap12:Sender</soap12:Value>
            </soap12:Code>
            <soap12:Reason>
                <soap12:Text xml:lang=”en”>The publication identifier ‘xyz’ is
invalid</soap12:Text>
            </soap12:Reason>
            <soap12:Detail>
              <wsnt:SubscribeCreationFailedFault>
                    <wsrf-bf:Timestamp>
                        2005-05-04T20:18:44.970Z
                    </wsrf-bf:Timestamp>
                    <wsrf-bf:Description>
                        The publication identifier ‘xyz’ is invalid
                    </wsrf-bf:Description>
                    <wsrf-bf:FaultCause>
                        <ows:ExceptionReport version=”1.0.0”
xmlns:ows=”http://www.opengis.net/ows/1.1”>
                            <ows:Exception exceptionCode=”InvalidPublicationIdentifier”
locator=”xyz”>
                                <ows:ExceptionText>
                                    The publication identifier ‘xyz’ is invalid
                                </ows:ExceptionText>
                            </ows:Exception>
                        </ows:ExceptionReport>
                    </wsrf-bf:FaultCause>
                </wsnt:SubscribeCreationFailedFault>
            </soap12:Detail>
        </soap12:Fault>
    </soap12:Body>
</soap12:Envelope>
```

9. Requirements Class: SOAP Standalone Publisher extends SOAP Basic Publisher, Standalone Publisher

<table>
<tr><th colspan="2">Requirements Class</th></tr>
<tr><td colspan="2"><b>http://www.opengis.net/spec/pubsub/1.0/req/soap/standalone-publisher</b></td></tr>
<tr><td><b>Target type</b></td><td>Publisher</td></tr>
</table>

Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/soap/basic-publisher**
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/core/standalone-publisher**
Requirement	/req/soap/standalone-publisher/getcapabilities
Requirement	/req/soap/standalone-publisher/getsubscription

This Requirements Class specifies the basic Publish/Subscribe operations of a Publisher using the SOAP protocol:

> ***GetCapabilities*** – allows for the discovery of Publisher metadata, including offered publications, service capabilities, and service provider information; and
>
> ***GetSubscription*** – allows for the retrieval of subscription information.

Standalone SOAP Publishers expose two operations for Publish/Subscribe metadata: *GetCapabilities* and *GetSubscription*. The *GetCapabilities* operation is utilized directly from OWS Common Specification version 1.1 [OGC 06-121r3].

Requirement
/req/soap/standalone-publisher/getcapabilities
Req 14 A **Publisher** shall offer a SOAP pubsub:*GetCapabilities* operation as defined in OGC Publish/Subscribe XSL/XSD schema

Requirement
/req/soap/standalone-publisher/getsubscription
Req 15 A **Publisher** shall offer a SOAP pubsub:*GetSubscription* operation as defined in OGC Publish/Subscribe XSL/XSD schema

9.1 Exception and Fault Usage

Exception behavior for the *GetCapabilities* operation is defined in Table 8 and Clause 8 of the OWS Common Specification [OGC 06-121r3].

Exception behavior for the *GetSubscription* operation is described in Table 6.

Table 6: Standalone Publisher Exception Mapping

Exception Code	SOAP Fault
***GetSubscription* operation**	
InvalidSubscriptionIdentifier	ResourceUnknownFault
NoApplicableCode	BaseFault

10. Requirements Class: SOAP Pausable Publisher extends SOAP Basic Publisher, Pausable Publisher

Requirements Class	
http://www.opengis.net/spec/pubsub/1.0/req/soap/pausable-publisher	
Target type	Publisher
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/soap/basic-publisher**
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/core/pausable-publisher**
Requirement	/req/soap/pausable-publisher/pause
Requirement	/req/soap/pausable-publisher/resume

This Requirements Class specifies the basic Publish/Subscribe operations of a Pausable Publisher using the SOAP protocol:

> ***Pause*** – allows for the pausing of an unpaused subscription, which pauses message delivery; and
>
> ***Resume*** – allows for the resumption of a paused subscription, which resumes message delivery.

These operations are implemented as the *PauseSubscription* and *ResumeSubscription* operations from the `PausableSubscriptionManager` interface described in [OASIS WS-BaseNotification].

Requirement
/req/soap/pausable-publisher/pause
Req 16 A **Publisher** shall offer a SOAP *PauseSubscription* operation from the `PausableSubscriptionManager` interface as described in [OASIS WS-BaseNotification]

Requirement
/req/soap/pausable-publisher/resume
Req 17 A **Publisher** shall offer a SOAP *ResumeSubscription* operation from the `PausableSubscriptionManager` interface as described in [OASIS WS-BaseNotification]

11. Requirements Class: SOAP Message Batching Publisher extends SOAP Basic Publisher, Message Batching Publisher

Requirements Class	
http://www.opengis.net/spec/pubsub/1.0/req/soap/message-batching-publisher	
Target type	Publisher
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/soap/basic-publisher**
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/core/message-batching-publisher**
Requirement	/req/soap/message-batching-publisher/subscribe-message-batching

A SOAP Message Batching Publisher accepts `pubsub:MessageBatchingCriteria` information as part of the *Subscribe* operation as defined in the Publish/Subscribe 1.0 XML schemas.

Requirement
/req/soap/message-batching-publisher/subscribe-message-batching
Req 18 A **Publisher** shall accept `pubsub:MessageBatchingCriteria` as part of `wsn-b:Subscribe` request messages

12. Requirements Class: SOAP Heartbeat Publisher extends SOAP Basic Publisher, Heartbeat Publisher

Requirements Class	
http://www.opengis.net/spec/pubsub/1.0/req/soap/heartbeat-publisher	
Target type	Publisher
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/soap/basic-publisher**
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/core/heartbeat-publisher**
Requirement	/req/soap/heartbeat-publisher/subscribe-heartbeat
Requirement	/req/soap/heartbeat-publisher/publish-heartbeat

A SOAP Heartbeat Publisher accepts `pubsub:HeartbeatCriteria` information as part of the *Subscribe* operation, and publishes heartbeat messages as `pubsub:Heartbeat` messages as defined in the Publish/Subscribe 1.0 XML schemas.

Requirement
/req/soap/heartbeat-publisher/subscribe-heartbeat
Req 19 A **Publisher** shall accept `pubsub:HeartbeatCriteria` as part of `wsn-`

`b:Subscribe` request messages

Requirement
/req/soap/heartbeat-publisher/publish-heartbeat
Req 20 A **Publisher** shall send regular `pubsub:Heartbeat` messages for each subscription as specified by each subscription's `HeartbeatCriteria`

13. Requirements Class – SOAP Brokering Publisher extends SOAP Basic Publisher, Brokering Publisher

Requirements Class	
http://www.opengis.net/spec/pubsub/1.0/req/soap/brokering-publisher	
Target type	Publisher
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/soap/basic-publisher**
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/core/brokering-publisher**
Requirement	/req/soap/brokering-publisher/registerpublisher
Requirement	/req/soap/brokering-publisher/registerpublisher-connect
Requirement	/req/soap/brokering-publisher/removepublisher
Requirement	/req/soap/brokering-publisher/fault-issuance

This Requirements Class specifies the basic Publish/Subscribe operations of a Brokering Publisher using the SOAP protocol. The SOAP operations in this Requirements Class are implemented as operations from interfaces from [OASIS WS-BrokeredNotification]:

> ***RegisterPublisher*** – allows the connection of an external Publisher to the Broker. This is implemented as the *RegisterPublisher* operation from the `NotificationBroker` interface from [OASIS WS-BrokeredNotification]; and
>
> ***RemovePublisher*** – allows the disconnection of a Publisher from the Broker. This is implemented as the *DestroyRegistration* operation from the `PublisherRegistrationManager` interface from [OASIS WS-BrokeredNotification].

Requirement
/req/soap/brokering-publisher/registerpublisher
Req 21 A **Publisher** shall offer the WS-BrokeredNotification NotificationBroker *RegisterPublisher* operation

Requirement
/req/soap/brokering-publisher/registerpublisher-connect
Req 22 When the *RegisterPublisher* operation is executed, A **Publisher** shall retrieve the capabilities document of the registered Publisher from `wsn-br:PublisherReference` and verify that it contains Publications, DeliveryCapabilities, and FilteringCapabilities sections before returning the RegisterPublisherResponse

Requirement
/req/soap/brokering-publisher/removepublisher
Req 23 A **Publisher** shall offer the WS-BrokeredNotification PublisherRegistrationManager *DestroyRegistration* operation

13.1 Exception and Fault Usage

For the *RegisterPublisher*, and *RemovePublisher* operations the following Fault and Exception codes are used.

Table 7: BrokeringPublisher Exception Mappings

Exception Code	SOAP Fault
***RegisterPublisher* operation**	
PublisherRegistrationRejected	PublisherRegistrationRejectedFault
PublisherRegistrationFailed	PublisherRegistrationFailedFault
NoApplicableCode	BaseFault
***RemovePublisher* operation**	
UnknownPublisher	ResourceUnknownFault
NoApplicableCode	BaseFault

Requirement
/req/soap/brokering-publisher/fault-issuance
Req 24 A **Publisher** shall generate SOAP Faults as described in [OASIS WS-BaseFaults] in accordance with [OASIS WS-BaseNotification] and [OASIS WS-BrokeredNotification]

14. Requirements Class – SOAP Publication Manager extends SOAP Basic Publisher, Publication Manager

Requirements Class	
http://www.opengis.net/spec/pubsub/1.0/req/soap/publication-manager	
Target type	Publisher
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/soap/basic-publisher**
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/core/publication-manager**
Requirement	/req/soap/publication-manager/createpublication
Requirement	/req/core/publication-manager/removepublication

This Requirements Class specifies the operations of a Publication Manager using the SOAP protocol:

> ***CreatePublication*** – allows for the creation of a new derived publication based upon an existing publication with an optional filter; and
>
> ***RemovePublication*** – allows for the removal of a derived publication.

Requirement
/req/soap/publication-manager/createpublication
Req 25 A **Publisher** shall offer a SOAP pubsub:*CreatePublication* operation as defined in OGC Publish/Subscribe XSL/XSD schema

Requirement
/req/soap/publication-manager/removepublication
Req 26 A **Publisher** shall offer a SOAP pubsub:*RemovePublication* operation as defined in OGC Publish/Subscribe XSL/XSD schema

14.1 Exception and Fault Usage

Exception behavior for the *CreatePublication* and *RemovePublication* operations is described in Table 8.

Table 8: Publication Manager Exception Mapping

Exception Code	SOAP Fault
***CreatePublication* operation**	
InvalidPublicationIdentifier	ResourceUnknownFault

InvalidFilter	InvalidFilterFault
InvalidParameterValue	BaseFault
NoApplicableCode	BaseFault
***RemovePublication* operation**	
InvalidPublicationIdentifier	ResourceUnknownFault
InvalidParameterValue	BaseFault
NoApplicableCode	BaseFault

15. Requirements Class – SOAP HTTP Delivery Publisher extends SOAP Basic Publisher

Requirements Class	
http://www.opengis.net/spec/pubsub/1.0/req/soap/http-delivery-publisher	
Target type	Publisher
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/soap/basic-publisher**
Requirement	/req/soap/http-delivery-method/notify
Requirement	/req/soap/http-delivery-method/post
Requirement	/req/soap/http-delivery-method/action-header

This Requirements Class specifies a method by which Publishers deliver messages using HTTP POST with SOAP/XML content. Messages are delivered to a SOAP endpoint (SOAP Basic Receiver) using the *Notify* operation from the NotificationConsumer interface described in [OASIS WS-BaseNotification].

Requirement
/req/soap/http-delivery-method/notify
Req 27 A **Publisher** shall deliver matching messages in accordance to [OASIS WS-BaseNotification] Clause 3.2

Requirement
/req/soap/http-delivery-method/post
Req 28 A **Publisher** shall deliver messages to the **Receiver** by initiating an HTTP request using the HTTP POST method

Requirement
/req/soap/http-delivery-method/action-header
Req 29 A **Publisher** shall deliver messages in a HTTP request with the value of the SOAPAction HTTP header field in a quoted string

16. Requirements Class – SOAP Reliable Publisher extends SOAP Basic Publisher

Requirements Class	
http://www.opengis.net/spec/pubsub/1.0/req/soap/reliable-publisher	
Target type	Publisher
Dependency	**http://www.opengis.net/spec/pubsub/1.0/req/soap/basic-publisher**

A SOAP Reliable Publisher is a type of Publisher that sends messages reliably using [OASIS WS-ReliableMessaging]. Basic Publishers only support a single attempted delivery of messages and do not specify behavior where messages are lost in transit.

Annex A. Abstract Test Suite (Normative)

A Publish/Subscribe implementation must satisfy the following system characteristics to be conformant with this specification.

Test, requirement, requirements class, and conformance class identifiers below are relative to http://www.opengis.net/spec/pubsub/1.0/.

The minimum set of conformance classes an implementation needs to pass are either SOAP Basic Receiver (section A.1) or SOAP Basic Publisher (section A.2)

A.1 Conformance class: SOAP Basic Receiver

/conf/soap/basic-receiver	
Dependency	/conf/core/basic-receiver
Requirements Class	/req/soap/basic-receiver

Test: /conf/soap/basic-receiver/notify

Requirement	**/req/soap/basic-receiver/notify**
Test Purpose	A **Receiver** shall offer the *Notify* operation as specified in the WS-BaseNotification `NotificationConsumer` interface
Test Method	Execute the WS-BaseNotification *Notify* operation on the Receiver, with a test SOAP message with the SOAP WS-Addressing Action set to http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify, with test data

A.2 Conformance class: SOAP Basic Publisher

/conf/soap/basic-publisher	
Dependency	/conf/core/basic-publisher
Requirements Class	/req/soap/basic-publisher

Test: /conf/soap/basic-publisher/envelope-body

Requirement	**/req/soap/basic-publisher/envelope-body**
Test Purpose	A **Publisher** shall produce SOAP messages with one SOAP Envelope containing exactly one SOAP Body
Test Method	Execute any operation on the Publisher and verify the condition

Test: /conf/soap/basic-publisher/getcapabilities-service-metadata

Requirement	**/req/soap/basic-publisher/getcapabilities-service-metadata**
Test Purpose	A **Publisher** shall specify the HTTP POST request encodings accepted by including an `ows:Constraint` element, with “PostEncoding” as the value of the `name` attribute and with a value of “SOAP” in *GetCapabilities* responses
Test Method	Execute a *GetCapabilities* operation against the service and verify that the service returns a Capabilities document with a `ows:Constraint` element, with “PostEncoding” as the value of the `name` attribute and with a value of “SOAP”

Test: /conf/soap/basic-publisher/subscribe

Requirement	**/req/soap/basic-publisher/subscribe**
Test Purpose	A **Publisher** shall offer the WS-BaseNotification NotificationProducer *Subscribe* operation
Test Method	Execute the WS-BaseNotification *Subscribe* operation, with a test SOAP message with the SOAP WS-Addressing Action set to “http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/SubscribeRequest”, against a test publication and ensure that the SubscribeResponse includes a valid `Subscription`

Test: /conf/soap/basic-publisher/unsubscribe

Requirement	**/req/soap/basic-publisher/unsubscribe**
Test Purpose	A **Publisher** shall offer the WS-BaseNotification SubscriptionManager *Unsubscribe* operation
Test Method	Execute a *Subscribe* operation against a test publication, record the returned subscription identifier, and execute the WS-BaseNotification *Unsubscribe* operation, with a test SOAP message with the SOAP WS-Addressing Action set to http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeRequest, with the subscription identifier, and ensure that the response is a valid UnsubscribeResponse

Test: /conf/soap/basic-publisher/renew

Requirement	**/req/soap/basic-publisher/renew**
Test Purpose	A **Publisher** shall offer the WS-BaseNotification SubscriptionManager *Renew* operation
Test Method	Execute the WS-BaseNotification *Renew* operation on the Publisher with a test SOAP message with the SOAP WS-Addressing Action set to “http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/RenewRequest”

Test: /conf/soap/basic-publisher/subscribe/dialect-set

Requirement	**/req/soap/basic-publisher/subscribe/dialect-set**
Test Purpose	A **Publisher** shall raise an Exception if the `/wsn-b:Subscribe/wsn-b:Filter/wsn-b:MessageContent/@Dialect` attribute is not set on a *Subscribe* request
Test Method	Execute the WS-BaseNotification *Subscribe* operation without setting the `/wsn-b:Subscribe/wsn-b:Filter/wsn-b:MessageContent/@Dialect` attribute in the request and verify that an Exception is thrown

Test: /conf/soap/basic-publisher/subscribe/dialect-valid

Requirement	**/req/soap/basic-publisher/subscribe/dialect-valid**

Test Purpose	A **Publisher** shall raise an Exception if the `/wsn-b:Subscribe/wsn-b:Filter/wsn-b:MessageContent/@Dialect` attribute is not one of the advertised `FilterCapabilities` identifiers
Test Method	Execute the WS-BaseNotification *Subscribe* operation with the `/wsn-b:Subscribe/wsn-b:Filter/wsn-b:MessageContent/@Dialect` attribute in the request set to a value not advertised in the `FilterCapabilities` and verify that an Exception is thrown

Test: /conf/soap/basic-publisher/subscribe/publication-id

Requirement	**/req/soap/basic-publisher/subscribe/publication-id**
Test Purpose	A **Publisher** shall raise an Exception if the `/wsn-b:Subscribe/pubsub:PublicationIdentifier` element is not present on a *Subscribe* request
Test Method	Execute the WS-BaseNotification *Subscribe* operation without the `/wsn-b:Subscribe/pubsub:PublicationIdentifier` element in the request and verify that an Exception is thrown

Test: /conf/soap/basic-publisher/renew-nil-termination-time

Requirement	**/req/soap/basic-publisher/renew-nil-termination-time**
Test Purpose	A **Publisher** shall throw an Exception if it receives a *Renew* request that contains a `wsn-b:TerminationTime` with `xsi:nil` attribute set to “true”
Test Method	Execute a *Renew* operation with a `wsn-b:TerminationTime` with `xsi:nil` attribute set to “true” and verify that an Exception is thrown

Test: /conf/soap/basic-publisher/unsubscribe/publication-id

Requirement	**/req/soap/basic-publisher/unsubscribe/publication-id**
Test Purpose	A **Publisher** shall raise an Exception if the `/wsn-b:Unsubscribe/pubsub:PublicationIdentifier`

	element is not present on a *Unsubscribe* request
Test Method	Execute the WS-BaseNotification *Unsubscribe* operation without the `/wsn-b:Unsubscribe/pubsub:PublicationIdentifier` element in the request and verify that an Exception is thrown

Test: /conf/soap/basic-publisher/fault-issuance

Requirement	**/req/soap/basic-publisher/fault-issuance**
Test Purpose	A **Publisher** shall issue SOAP Faults as described in [OASIS WS-BaseFaults] in accordance with [OASIS WS-BaseNotification]
Test Method	Execute a request that raises an exception on the Publisher and ensure that the response message contains a SOAP Fault compliant with [OASIS WS-BaseFaults] and [OASIS WS-BaseNotification]

Test: /conf/soap/basic-publisher/fault-detail

Requirement	**/req/soap/basic-publisher/fault-detail**
Test Purpose	A **Publisher** shall issue SOAP Faults that incorporate an ExceptionReport valid according to Clause 8 of the OWS Common Specification [OGC 06-121r3] in the `FaultCause` element of the WSRF:BaseFault
Test Method	Execute a request that raises an exception on the service and ensure that the response message incorporates an ExceptionReport valid according to Clause 8 of the OWS Common Specification [OGC 06-121r3] in the `FaultCause` element of the generated WSRF:BaseFault

A.3 Conformance class: SOAP Standalone Publisher

/conf/soap/standalone-publisher	
Dependency	/conf/soap/basic-publisher
Requirements Class	/req/soap/standalone-publisher

Test: /conf/soap/standalone-publisher/getcapabilities

Requirement	**/req/soap/standalone-publisher/getcapabilities**
Test Purpose	A **Publisher** shall offer a SOAP pubsub:*GetCapabilities* operation as defined in OGC Publish/Subscribe XSL/XSD schema
Test Method	Execute the *GetCapabilities* operation, with a SOAP message with the SOAP WS-Addressing Action set to “http://www.opengis.net/def/serviceOperation/pubsub/1.0/GetCapabilities”, with an `AcceptVersions` section with a single `Version` parameter set to “1.0.0” and the `service` parameter set to “PubSub”, and ensure that the response is a valid PublisherCapabilities document

Test: /conf/soap/standalone-publisher/getsubscription

Requirement	**/req/soap/standalone-publisher/getsubscription**
Test Purpose	A **Publisher** shall offer a SOAP pubsub:*GetSubscription* operation as defined in OGC Publish/Subscribe XSL/XSD schema
Test Method	Execute the *GetSubscription* operation with a SOAP message, with the SOAP WS-Addressing Action set to “http://www.opengis.net/def/serviceOperation/pubsub/1.0/GetSubscription”, without any `subscriptionIdentifier` parameters, and ensure that the response is a valid GetSubscriptionResponse document. For every subscription in the GetSubscriptionResponse, execute the *GetSubscription* operation with the corresponding `subscriptionIdentifier` parameter, and ensure that the response is a valid GetSubscriptionResponse document related to that subscription

A.4 Conformance class: SOAP Pausable Publisher

/conf/soap/pausable-publisher	
Dependency	/conf/soap/basic-publisher
Dependency	/conf/core/pausable-publisher
Requirements	/req/soap/pausable-publisher

Class	

Test: /conf/soap/pausable-publisher/pause

Requirement	**/req/soap/pausable-publisher/pause**
Test Purpose	A **Publisher** shall offer a SOAP *PauseSubscription* operation from the `PausableSubscriptionManager` interface as described in [OASIS WS-BaseNotification]
Test Method	Execute the WS-BaseNotification *PauseSubscription* operation, with a test SOAP message with the SOAP WS-Addressing Action set to "http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/PauseSubscriptionRequest", on a test publication, record the returned subscription identifier, wait for messages to be received for the subscription, then execute the *Pause* operation with the `subscriptionIdentifier` parameter set to the recorded subscription identifier, and ensure that the response is a valid PauseResponse document

Test: /conf/soap/pausable-publisher/resume

Requirement	**/req/soap/pausable-publisher/resume**
Test Purpose	A **Publisher** shall offer a SOAP *ResumeSubscription* operation from the `PausableSubscriptionManager` interface as described in [OASIS WS-BaseNotification]
Test Method	Execute the *Subscribe* operation on a test publication, record the returned subscription identifier, wait for messages to be received for the subscription, execute the *Pause* operation with the `subscriptionIdentifier` parameter set to the recorded subscription identifier, execute the *Resume* operation, with a test SOAP message with the SOAP WS-Addressing Action set to "http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/ResumeSubscriptionRequest", with the `subscriptionIdentifier` parameter set to the recorded subscription identifier, and ensure that the response is a valid ResumeResponse document

A.5 Conformance class: SOAP Message Batching Publisher

/conf/soap/message-batching-publisher	
Dependency	/conf/soap/basic-publisher
Dependency	/conf/core/message-batching-publisher
Requirements Class	/req/soap/message-batching-publisher

Test: /conf/soap/message-batching-publisher/subscribe-message-batching

Requirement	**/req/soap/message-batching-publisher/subscribe-message-batching**
Test Purpose	A **Publisher** shall accept `pubsub:MessageBatchingCriteria` as part of `wsn-b:Subscribe` request messages
Test Method	Execute the WS-BaseNotification *Subscribe* operation to create a test subscription with `pubsub:MessageBatchingCriteria` with the parameter `maxMessageCount` set to "1" and verify that the response is a valid SubscribeResponse

A.6 Conformance class: SOAP Heartbeat Publisher

/conf/soap/heartbeat-publisher	
Dependency	/conf/soap/basic-publisher
Dependency	/conf/core/heartbeat-publisher
Requirements Class	/req/soap/heartbeat-publisher

Test: /conf/soap/heartbeat-publisher/subscribe-heartbeat

Requirement	**/req/soap/heartbeat-publisher/subscribe-heartbeat**
Test Purpose	A **Publisher** shall accept `pubsub:HeartbeatCriteria` as part of `wsn-b:Subscribe` request messages
Test Method	Execute the WS-BaseNotification *Subscribe* operation to create a test subscription with `pubsub:HeartbeatCriteria` with the

	parameter `heartbeatRate` set to "1 minute" and verify that the response is a valid SubscribeResponse

Test: /conf/soap/heartbeat-publisher/publish-heartbeat

Requirement	**/req/soap/heartbeat-publisher/publish-heartbeat**
Test Purpose	A **Publisher** shall send regular `pubsub:Heartbeat` messages for each subscription as specified by each subscription's `HeartbeatCriteria`
Test Method	Execute the WS-BaseNotification *Subscribe* operation to create a test subscription with `pubsub:HeartbeatCriteria` with the parameter `heartbeatRate` set to "10 seconds" and verify that the response is a valid SubscribeResponse. Wait 35 seconds and verify that 3 heartbeat messages were received

A.7 Conformance class: SOAP Brokering Publisher

/conf/soap/brokering-publisher	
Dependency	/conf/soap/basic-publisher
Dependency	/conf/core/brokering-publisher
Requirements Class	/req/soap/brokering-publisher

Test: /conf/soap/brokering-publisher/registerpublisher

Requirement	**/req/soap/brokering-publisher/registerpublisher**
Test Purpose	A **Publisher** shall offer the WS-BrokeredNotification NotificationBroker *RegisterPublisher* operation
Test Method	Execute the WS-BrokeredNotification *RegisterPublisher* operation, with a test SOAP message with the SOAP WS-Addressing Action set to "http://docs.oasis-open.org/wsn/brw-2/RegisterPublisher/RegisterPublisherRequest", and ensure that the response is a valid RegisterPublisherResponse

Test: /conf/soap/brokering-publisher/registerpublisher-connect

Requirement	**/req/soap/brokering-publisher/registerpublisher-connect**
Test Purpose	When the *RegisterPublisher* operation is executed, A **Publisher** shall retrieve the capabilities document of the registered Publisher from `wsn-br:PublisherReference` and verify that it contains Publications, DeliveryCapabilities, and FilteringCapabilities sections before returning the RegisterPublisherResponse
Test Method	Execute the WS-BrokeredNotification *RegisterPublisher* operation, with a test SOAP message with the SOAP WS-Addressing Action set to "http://docs.oasis-open.org/wsn/brw-2/RegisterPublisher/RegisterPublisherRequest", with a `capabilitiesReference` parameter that is resolvable to a valid capabilities document with `FilterCapabilities`, `DeliveryCapabilities`, and `Publications` sections, and ensure that the response is a valid RegisterPublisherResponse

Test: /conf/soap/brokering-publisher/removepublisher

Requirement	**/req/soap/brokering-publisher/removepublisher**
Test Purpose	A **Publisher** shall offer the WS-BrokeredNotification PublisherRegistrationManager *DestroyRegistration* operation
Test Method	Execute the WS-BrokeredNotification *RegisterPublisher* operation, with a test SOAP message with the SOAP WS-Addressing Action set to "http://docs.oasis-open.org/wsn/brw-2/RegisterPublisher/RegisterPublisherRequest", and ensure that the response is a valid RegisterPublisherResponse, then execute the WS-BrokeredNotification PublisherRegistrationManager *DestroyRegistration* operation, with a test SOAP message with the SOAP WS-Addressing Action set to "http://docs.oasis-open.org/wsn/brw-2/PublisherRegistrationManager/DestroyRegistrationRequest", against the same `capabilitiesReference` parameter and ensure that the response is a valid RemovePublisherResponse

Test: /conf/soap/brokering-publisher/fault-issuance

Requirement	**/req/soap/brokering-publisher/fault-issuance**

Test Purpose	A **Publisher** shall generate SOAP Faults as described in [OASIS WS-BaseFaults] in accordance with [OASIS WS-BaseNotification] and [OASIS WS-BrokeredNotification]
Test Method	Execute a request that raises an exception on the Publisher and ensure that the response message contains a SOAP Fault compliant with [OASIS WS-BaseFaults], [OASIS WS-BaseNotification], and [OASIS WS-BrokeredNotification]

A.8 Conformance class: SOAP Publication Manager

/conf/soap/publication-manager	
Dependency	/conf/soap/basic-publisher
Dependency	/conf/core/publication-manager
Requirements Class	/req/soap/basic-publisher

Test: /conf/soap/publication-manager/createpublication

Requirement	**/req/soap/publication-manager/creatpublication**
Test Purpose	A **Publisher** shall offer a SOAP pubsub:*CreatePublication* operation as defined in OGC Publish/Subscribe XSL/XSD schema
Test Method	Create a test publication with a publication identifier of “urn:pubsub:ats:BasePub”. Execute the *CreatePublication* operation, with a test SOAP message, with the SOAP WS-Addressing Action set to “http://www.opengis.net/def/serviceOperation/pubsub/1.0/CreatePublication”, with the `basePublicationIdentifier` parameter set to “urn:pubsub:ats:BasePub” and the `identifier` parameter set to “urn:pubsub:ats:DerivedPub” and the `description` parameter set to “Test description”, ensure that the response is a valid CreatePublicationResponse document

Test: /conf/soap/publication-manager/removepublication

Requirement	**/req/soap/publication-manager/removepublication**

Test Purpose	A **Publisher** shall offer a SOAP pubsub:*RemovePublication* operation as defined in OGC Publish/Subscribe XSL/XSD schema
Test Method	Execute the *CreatePublication* operation with the `basePublicationIdentifier` parameter set to a test publication identifier, ensure that the response is a valid CreatePublicationResponse document, execute the *RemovePublication* operation, , with a test SOAP message, with the SOAP WS-Addressing Action set to "http://www.opengis.net/def/serviceOperation/pubsub/1.0/RemovePublication", against the newly-created Publication, and ensure that the response is a valid RemovePublicationResponse

A.9 Conformance class: SOAP HTTP Delivery Publisher

/conf/soap/http-delivery	
Dependency	/conf/soap/basic-publisher
Requirements Class	/req/soap/http-delivery-publisher

Test: /conf/soap/http-delivery/notify

Requirement	**/req/soap/http-delivery/notify**
Test Purpose	A **Publisher** shall deliver matching messages in accordance to [OASIS WS-BaseNotification] Clause 3.2
Test Method	Execute the *Subscribe* operation on a test publication, wait for messages to be received for the subscription, and verify that the messages are in accordance to [OASIS WS-BaseNotification] Clause 3.2

Test: /conf/soap/http-delivery/post

Requirement	**/req/soap/http-delivery/post**
Test Purpose	A **Publisher** shall deliver messages to the **Receiver** by initiating an HTTP request using the HTTP POST method
Test Method	Execute the *Subscribe* operation on a test publication, wait for messages to be received for the subscription, and verify that the messages are delivered by initiating an HTTP request using the HTTP

	POST method

Test: /conf/soap/http-delivery/action-header

Requirement	**/req/soap/http-delivery/action-header**
Test Purpose	A **Publisher** shall deliver messages in a HTTP request with the value of the SOAPAction HTTP header field in a quoted string
Test Method	Execute the *Subscribe* operation on a test publication, wait for messages to be received for the subscription, and verify that the messages are delivered in a HTTP request with the value of the SOAPAction HTTP header field in a quoted string

A.10 Conformance class: SOAP Reliable Publisher

/conf/soap/reliable-publisher	
Dependency	/conf/soap/basic-publisherget
Requirements Class	/req/soap/reliable-publisher

Annex B. Revision history

Date	Release	Author	Paragraph(s) modified	Description
2015-06-26	1.0-RC0	Aaron Braeckel, Lorenzo Bigagli	All	First draft for internal SWG review
2015-07-31	1.0-RC1	Aaron Braeckel, Lorenzo Bigagli	All	Revised URIs, revised references, edited mentions to Topic
2015-09-08	1.0-RC2	Aaron Braeckel, Lorenzo Bigagli	All	Incorporated comments from OAB review in preparation for public comment
2015-12-07	1.0-RC3	Aaron Braeckel, Lorenzo Bigagli	All	Incorporated changes related to feedback from public comments in preparation for adoption vote
2016-02-12	1.0	Aaron Braeckel, Lorenzo Bigagli	Front page, Abstract	Incorporated comments received during adoption vote Finalisation
2016-03-31	1.0	Scott Simmons	All	Minor edits, prepare for publication