

WS-Eventing Interoperability Scenario

Cetacean Tracking System

Version: 0.0.3

Date: December 6, 2010

Editor: [Gilbert Pilz](#)

Abstract

The following scenario is designed to provide a framework in which to test the interoperability of various WS-Eventing implementations. Because this scenario and the tests defined within it will be used to judge which features of WS-Eventing are implemented and which are not, the feature coverage is intended to be complete.

Timeline

Start	End	Activity

Table of Contents

Abstract.....	1
Timeline.....	1
1 Dependencies.....	2
1.1 Scope.....	2
1.2 Namespaces.....	2
1.3 Preconditions.....	2
2 Scenario Description.....	2
2.1 Event Description.....	3
2.2 Event Timing.....	3
2.3 Tags.....	3
3 Tests.....	3
3.1 Basic Test.....	3
3.2 Wrapped Notifications.....	4
3.3 Duration Expiration Test.....	5
3.4 Specific Time Expiration Test.....	6
3.5 Best Effort Expiration Test.....	7
3.6 Renew Test.....	8
3.7 SubscriptionEnd Test.....	10
3.8 Filter Test – XPath 1.0.....	11
3.9 Filter Test – XPath 2.0.....	12
3.10 Non-Addressable Event Sink Test.....	12
4 WSDLs.....	14
4.1 Event Source WSDL.....	14

4.2 Notification WSDL.....	14
5 EventDescriptions.....	14
6 Schemas.....	14
7 Change Log.....	14

1 Dependencies

1.1 Scope

The following specifications and technologies are in scope for this scenario:

- SOAP 1.1
- WS-Eventing
- WS-EventDescriptions
- WS-MakeConnection
- WS-Policy
- WSDL 1.1

1.2 Namespaces

The following table defines the namespaces used in this document:

Prefix	Namespace	Specification
xsd	http://www.w3.org/2001/XMLSchema	XML Schema
wSDL	http://schemas.xmlsoap.org/wSDL/	WSDL 1.1
soap11	http://schemas.xmlsoap.org/soap/envelope/	SOAP 1.1
wsoap11	http://schemas.xmlsoap.org/wSDL/soap/	WSDL 1.1
wsa	http://www.w3.org/2005/08/addressing	WS-Addressing 1.0
wse	http://www.w3.org/2010/08/ws-evt	WS-Eventing
gpx	http://www.topografix.com/GPX/1/1	GPS eXchange Format

1.3 Preconditions

2 Scenario Description

This scenario presupposes a [cetacean](#) tracking system in which a number of animals have been “tagged” with devices that track their location. These tags periodically communicate via satellite to a central system. External systems can consume this information by using WS-Eventing to subscribe to periodic notifications about the locations of the tags and, presumably, the animals they are attached to.

2.1 Event Description

The location of the tags is expressed in GPS coordinates using the [GPS eXchange Format](#), an XML schema designed as a common GPS data format for software applications. In addition to the basic GPS information (latitude, longitude, elevation, and time), the notifications include an ID that uniquely identifies the tag and, by inference, the animal that the tag is attached to. An EventDescriptions document that describes the structure of the event information within the notifications can be found in Section 5.

2.2 Event Timing

While in the real world the frequency of notifications might be hourly or even daily, for the sake of feasibility we compress time by a scale of 1/120 so that one hour in “scenario time” is thirty seconds in real-world time. The time data contained in the notifications will reflect scenario time.

2.3 Tags

Again for the sake of feasibility, this scenario will only include three tags with the following IDs:

- 13c76450-de3d-11df-85ca-0800200c9a66 (Howard)
- 234b6840-de3d-11df-85ca-0800200c9a66 (Kerry)
- 32675b90-de3d-11df-85ca-0800200c9a66 (Oscar)

3 Tests

The following sub-sections describe tests designed to exercise all the mandatory and optional features of WS-Eventing except for those (such as the use of EventDescriptions or Notification WSDLs) that affect only the process of developing one or more of the components of a WS-Eventing-based system.

Each of these sub-sections is organized into four parts:

- An overview that describes the purpose of the test and the salient features of the messages that are exchanged.
- A sequence diagram that illustrates the sequence of events in the test.
- A list of criteria used to judge the success of the test.
- A conformance section that enumerates the conditions under which conforming implementations are allowed to either not implement the test or fail one or more of the success criteria.

3.1 Basic Test

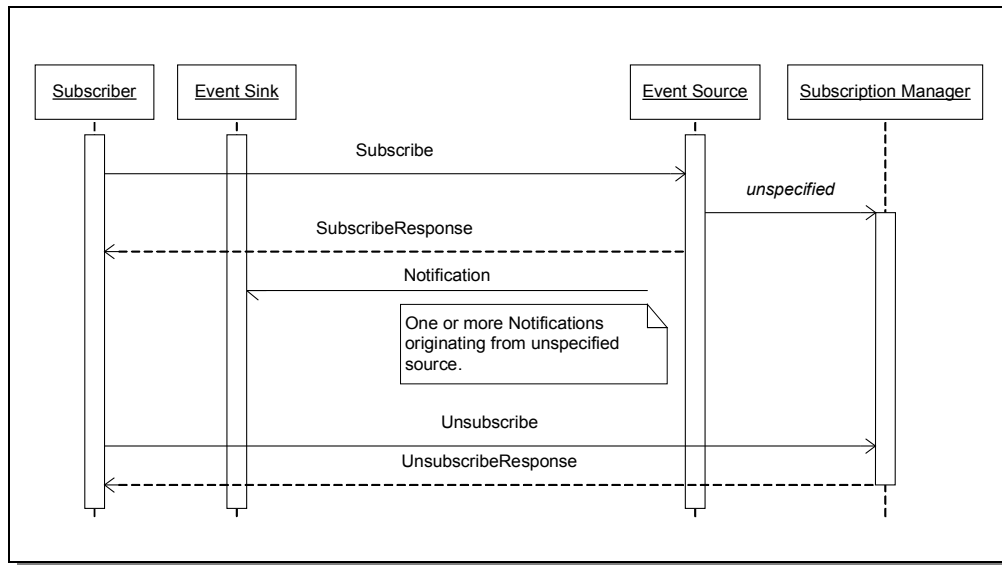
This test verifies the ability to subscribe and receive notifications. The initial Subscribe request has the following features:

- expiration time chose by Event Source/Subscription Manager

- no EndTo EPR
- no Filters
- unwrapped notifications

Sequence

The following diagram illustrates the sequence of messages for the Basic Test.



Success Criteria

1. Receipt of a valid **Subscribe** message by the Event Source.
2. Receipt of a valid **SubscribeResponse** message by Subscriber.
3. Receipt of one or more unwrapped **Notifications** by the Event Sink.
4. Receipt of a valid **Unsubscribe** message by the Subscription Manager.
5. Receipt of a valid **UnsubscribeResponse** message by the Subscriber.

Conformance

Because this test involves only operations and elements that are required, there are no allowable failure cases.

Any failure to meet the above success criteria indicates that either, or both, of the implementations participating in the test do not conform to WS-Eventing.

An implementation that is unable to support this test does not conform to WS-Eventing.

3.2 *Wrapped Notifications*

This test verifies the simple ability to subscribe and receive wrapped notifications. The initial **Subscribe** request has the following features:

- expiration time chosen by Event Source/Subscription Manager
- no EndTo EPR
- no Filters
- wrapped notifications

Sequence

The messaging sequence for this test is identical to that of the Basic Test.

Success Criteria

1. Receipt of a valid Subscribe message by the Event Source.
2. Receipt of a valid SubscribeResponse message by Subscriber.
3. Receipt of one or more wrapped Notifications by the Event Sink.
4. Receipt of a valid Unsubscribe message by the Subscription Manager.
5. Receipt of a valid UnsubscribeResponse message by the Subscriber.

Conformance

Because this test involves the use of the optional wrapped delivery format, there are a number of failure cases that fall within the boundaries of conforming behavior.

A conforming Subscriber/Event Sink MAY NOT be capable of implementing this test due to its inability to support wrapped notifications.

A conforming Event Source MAY respond to the initial Subscribe request with a `wse:DeliveryFormatRequestUnavailable` fault.

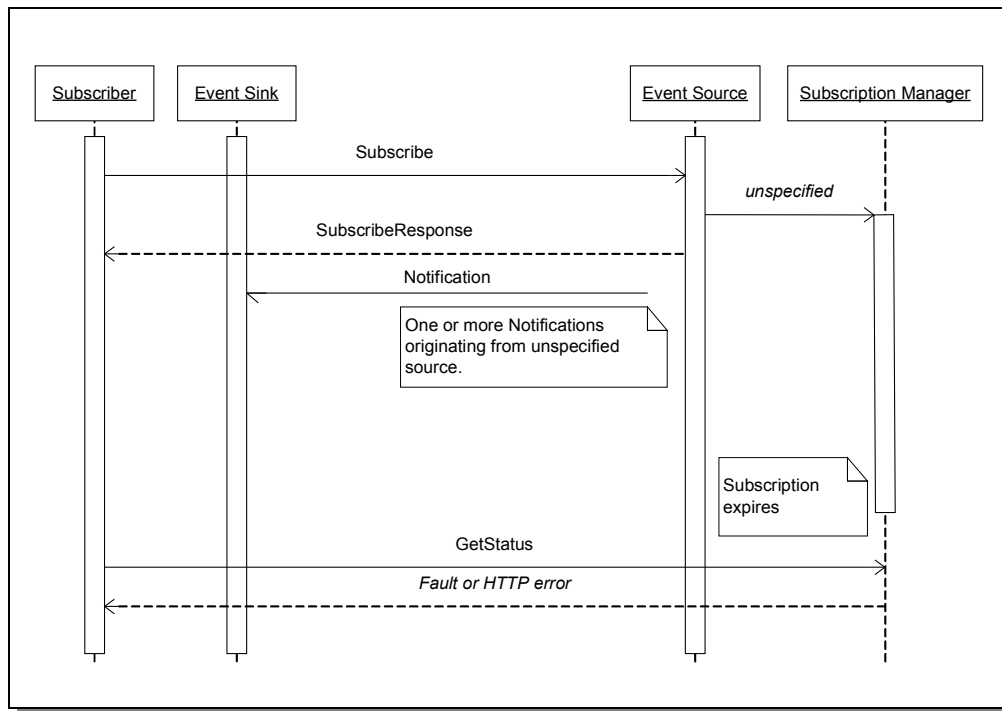
3.3 Duration Expiration Test

This test verifies the correct implementation of the expiration feature on the Event Source/Subscription Manager. The initial Subscribe message has the following features:

- (short) expiration time chosen by Subscriber as `xs:duration`
- no EndTo EPR
- no Filters
- unwrapped notifications

Sequence

The following diagram illustrates the sequence of messages for the Duration Expiration Test. Note that the Subscriber waits until the expiration time has passed before sending the GetStatus request.



Success Criteria

1. Receipt of a valid Subscribe message by the Event Source.
2. Receipt of a valid SubscribeResponse message by Subscriber.
3. Receipt of one or more unwrapped Notifications by the Event Sink.
4. Receipt of a valid GetStatus message by the Subscription Manager.
5. Receipt, by the Subscriber, of either the “UnknownSubscription” fault (defined by Section 6.10 of WS-Eventing), a SOAP fault that indicates that the Subscription Manager no longer exists, or an HTTP error (i.e. “404”) that indicates the Subscription Manager no longer exists

Conformance

Because this test involves the use of the optional `wse:Expires` element, a conforming Subscriber MAY NOT be capable of implementing this test due to its inability to support `wse:Expires`.

Note that, because `wse:Expires` is sender-optional and support for `xs:duration` is required, there are no valid reasons for a conforming Event Source/Subscription Manager implementation to either be unable to implement this test or to fail to meet one of the defined success criteria.

3.4 Specific Time Expiration Test

This test verifies the correct implementation of the expiration feature on the Event Source/Subscription Manager. The initial Subscribe request has the following features:

- (short) expiration time chosen by Subscriber as `xs:dateTime`
- no EndTo EPR
- no Filters
- unwrapped notifications

Sequence

The messaging sequence for this test is identical to that of the Duration Expiration Test.

Success Criteria

1. Receipt of a valid Subscribe message by the Event Source.
2. Receipt of a valid SubscribeResponse message by Subscriber.
3. Receipt of one or more unwrapped Notifications by the Event Sink.
4. Receipt of a valid GetStatus message by the Subscription Manager.
5. Receipt, by the Subscriber, of either the “UnknownSubscription” fault (defined by Section 6.10 of WS-Eventing), a SOAP fault that indicates that the Subscription Manager no longer exists, or an HTTP error (i.e. “404”) that indicates the Subscription Manager no longer exists.

Conformance

Because this test involves the use of both the optional `wse:Expires` element and the optional `xs:dateTime` type, there are a number of failure cases that fall within the boundaries of conforming behavior.

A conforming Subscriber MAY NOT be capable of implementing this test either due to its inability to support the `wse:Expires` element or the `xs:dateTime` type.

A conforming Event Source MAY respond to the initial Subscribe request with a `wse:UnsupportedExpirationType` fault.

3.5 Best Effort Expiration Test

This test verifies the correct implementation of the “best effort” expiration feature on the Event Source/Subscription Manager. The initial subscription has the following features:

- expiration time chosen by Subscriber as `xs:duration` with `@BestEffort='true'`
- no EndTo EPR
- no Filters
- unwrapped notifications

Sequence

The messaging sequence for this test is identical to that of the Duration Expiration Test.

Success Criteria

1. Receipt of a valid Subscribe message by the Event Source.
2. Receipt of a valid SubscribeResponse message by Subscriber.
3. Receipt of one or more unwrapped Notifications by the Event Sink.
4. Receipt of a valid GetStatus message by the Subscription Manager.
5. Receipt, by the Subscriber, of either the “UnknownSubscription” fault (defined by Section 6.10 of WS-Eventing), a SOAP fault that indicates that the Subscription Manager no longer exists, or an HTTP error (i.e. “404”) that indicates the Subscription Manager no longer exists.

Conformance

Because this test involves the use of both the optional `wse:Expires` element and the optional `BestEffort` attribute, there are a number of failure cases that fall within the boundaries of conforming behavior.

A conforming Subscriber MAY NOT be capable of implementing this test either due to its inability to support the `wse:Expires` element or the `BestEffort` attribute.

Note that, because both `wse:Expires` and `BestEffort` are sender-optional, there are no valid reasons for a conforming Event Source/Subscription Manager implementation to either be unable to implement this test or to fail to meet one of the defined success criteria.

3.6 Renew Test

This test verifies the ability of a Subscriber to update the expiration time of a Subscription via a Renew request. The initial Subscribe request has the following features:

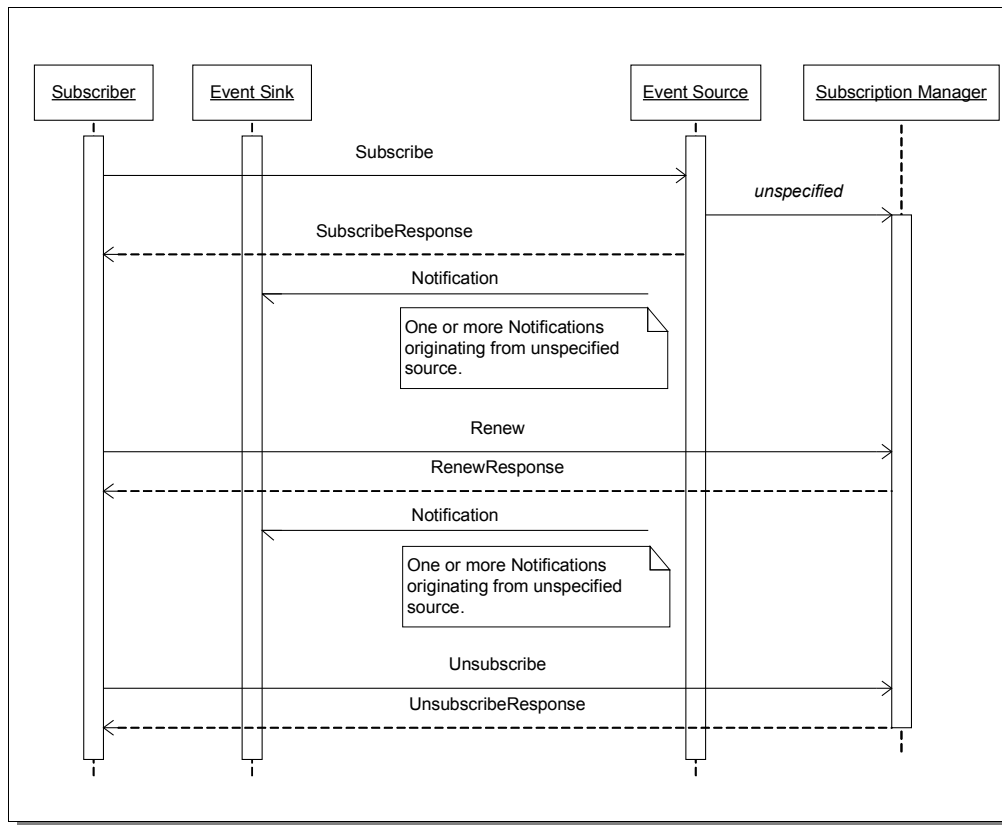
- (short) expiration time chosen by Subscriber as `xs:duration`
- no EndTo EPR
- no Filters
- unwrapped notifications

The Renew request has the following features:

- (short) expiration time chosen by Subscriber as `xs:duration`

Sequence

The following diagram illustrates the sequence of messages for the Renew Test.



Success Criteria

1. Receipt of a valid `Subscribe` message by the Event Source.
2. Receipt of a valid `SubscribeResponse` message by Subscriber.
3. Receipt of one or more wrapped `Notifications` by the Event Sink.
4. Prior to the expiration time elapsing, receipt of a valid `Renew` message by the Subscription Manager.
5. Receipt of a valid `RenewResponse` message by the Subscriber.
6. Subsequent to the `Renew/RenewResponse` exchange, receipt of one or more wrapped `Notifications` by the Event Sink.
7. Receipt of a valid `Unsubscribe` message by the Subscription Manager.
8. Receipt of a valid `UnsubscribeResponse` message by the Subscriber.

Conformance

Because this test involves the use of the optional `wse:Expires` element, a conforming Subscriber MAY NOT be capable of implementing this test due to its inability to support `wse:Expires`.

Note that, because `wse:Expires` is sender-optional and support for `xs:duration` is required, there are no valid reasons for a conforming Event Source/Subscription Manager

implementation to either be unable to implement this test or to fail to meet one of the defined success criteria.

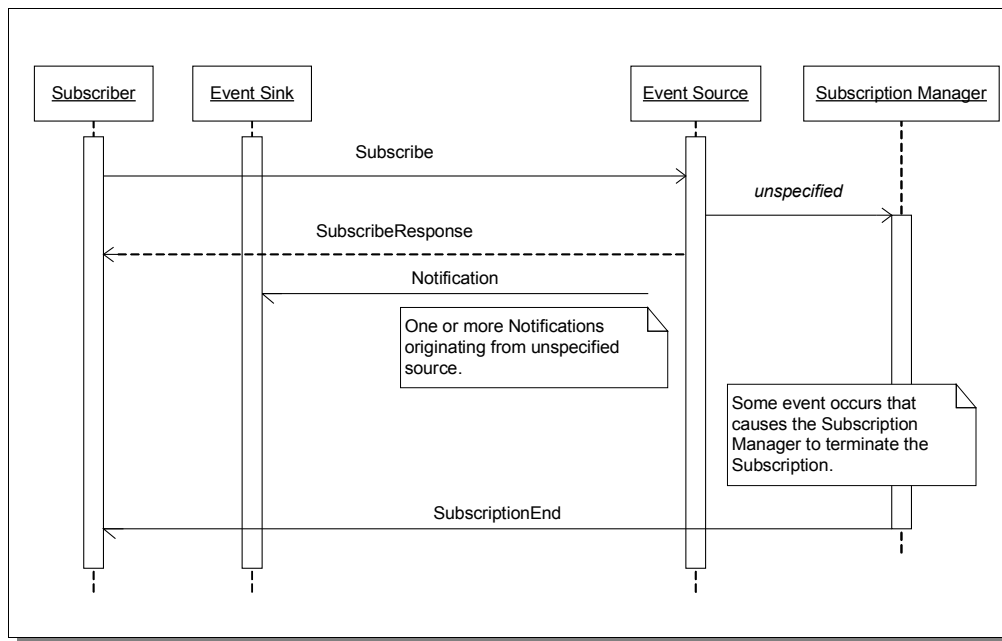
3.7 SubscriptionEnd Test

This test verifies the correct implementation of the SubscriptionEnd feature for both the Subscription Manager and the target of the SubscriptionEnd message. The initial Subscribe request has the following features:

- expiration time chosen by Event Source/Subscription Manager
- EndTo EPR
- no Filters
- unwrapped notifications

Sequence

The following diagram illustrates the sequence of messages for the SubscriptionEnd Test.



Success Criteria

1. Receipt of a valid Subscribe message by the Event Source.
2. Receipt of a valid SubscribeResponse message by the Subscriber.
3. Receipt of one or more wrapped Notifications by the Event Sink.
4. Receipt of a valid SubscriptionEnd message by the Subscriber (or whomever is indicated by the EndTo EPR).

Conformance

Because this test involves the use of the optional `wse:EndTo` element there are a number of failure cases that fall within the boundaries of conforming behavior.

A conforming Subscriber/Event Sink MAY NOT be capable of implementing this test due to its inability to support the `wse:EndTo` element or the `SubscriptionEnd` message.

A conforming Event Source MAY respond to the initial `Subscribe` request with a `wse:EndToNotSupported` fault.

3.8 Filter Test – XPath 1.0

This test verifies the ability of the Event Source/Subscription Manager to correctly implement XPath 1.0 filters. The initial `Subscribe` request has the following features:

- expiration time chosen by Event Source/Subscription Manager
- no `EndTo` EPR
- Filter in dialect “`http://www.w3.org/2010/08/ws-evt/Dialects/XPath10`” that selects those events that apply to tag `234b6840-de3d-11df-85ca-0800200c9a66` (Kerry).
(TBD – add exact filter expression)
- unwrapped notifications

Sequence

The messaging sequence for this test is identical to that of the [Basic Test](#). The difference between this test and the Basic Test is that only Notifications applying to tag `234b6840-de3d-11df-85ca-0800200c9a66` are received by the Event Sink.

Success Criteria

1. Receipt of a valid `Subscribe` message by the Event Source.
2. Receipt of a valid `SubscribeResponse` message by Subscriber.
3. Receipt of one or more unwrapped Notifications for the tag `234b6840-de3d-11df-85ca-0800200c9a66` by the Event Sink.
4. Receipt of a valid `Unsubscribe` message by the Subscription Manager.
5. Receipt of a valid `UnsubscribeResponse` message by the Subscriber.

Conformance

Because this test involves the use of the optional `wse:Filter` element there are a number of failure cases that fall within the boundaries of conforming behavior.

A conforming Subscriber/Event Sink MAY NOT be capable of implementing this test due to its inability to support the `wse:Filter` element or the XPath 1.0 dialect.

A conforming Event Source MAY respond to the initial `Subscribe` request with either a `wse:FilteringNotSupported` fault or a `wse:FilteringRequestedUnavailable` fault.

3.9 Filter Test – XPath 2.0

This test verifies the ability of the Event Source/Subscription Manager to correctly implement XPath 2.0 filters. The initial Subscribe request has the following features:

- expiration time chosen by Event Source/Subscription Manager
- no EndTo EPR
- Filter in dialect “<http://www.w3.org/2010/08/ws-evt/Dialects/XPath20>” that selects those events that apply to tag 32675b90-de3d-11df-85ca-0800200c9a66 (Oscar). (TBD – add exact filter expression)
- unwrapped notifications

Sequence

The messaging sequence for this test is identical to that of the [Basic Test](#). The difference between this test and the Basic Test is that only Notifications applying to tag 32675b90-de3d-11df-85ca-0800200c9a66 are received by the Event Sink.

Success Criteria

1. Receipt of a valid Subscribe message by the Event Source.
2. Receipt of a valid SubscribeResponse message by Subscriber.
3. Receipt of one or more unwrapped Notifications for the tag 32675b90-de3d-11df-85ca-0800200c9a66 by the Event Sink.
4. Receipt of a valid Unsubscribe message by the Subscription Manager.
5. Receipt of a valid UnsubscribeResponse message by the Subscriber.

Conformance

Because this test involves the use of the optional `wse:Filter` element there are a number of failure cases that fall within the boundaries of conforming behavior.

A conforming Subscriber/Event Sink MAY NOT be capable of implementing this test due to its inability to support the `wse:Filter` element or the XPath 2.0 dialect.

A conforming Event Source MAY respond to the initial Subscribe request with either a `wse:FilteringNotSupported` fault or a `wse:FilteringRequestedUnavailable` fault.

3.10 Non-Addressable Event Sink Test

This test verifies the ability to subscribe and receive notifications in an environment in which the Event Sink cannot accept connections from systems outside its network (i.e. the Event Sink is non-addressable). The facilities described by WS-MakeConnection are used by the Event Sink to poll for Notifications from the Event Source.

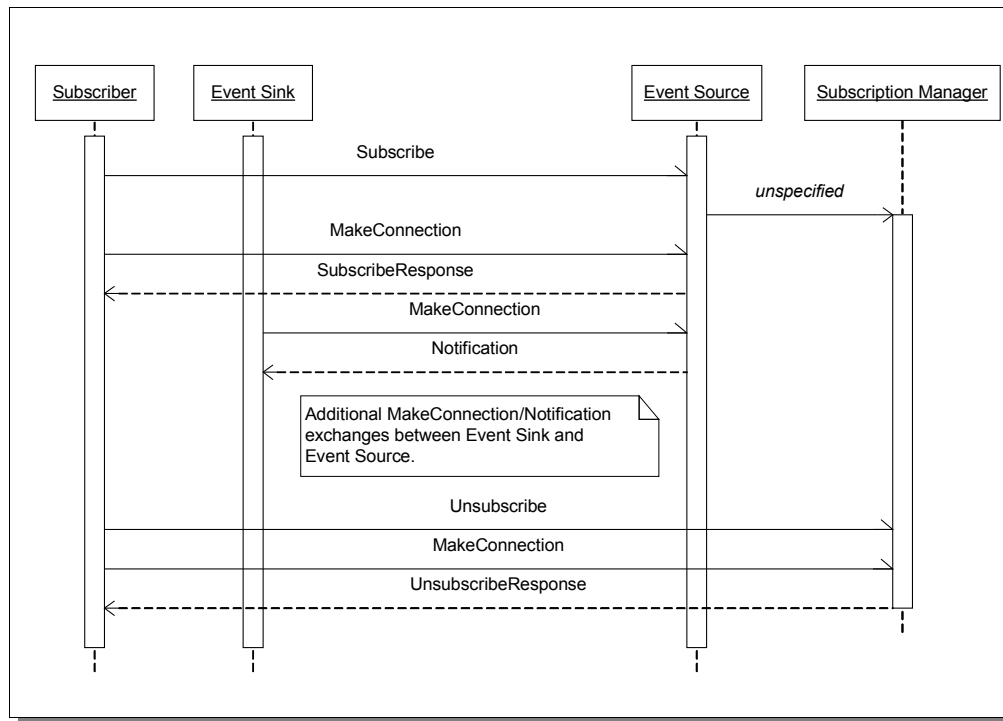
The initial Subscribe request has the following features:

- expiration time chose by Event Source/Subscription Manager

- no EndTo EPR
- no Filters
- unwrapped notifications
- the value of `wse:Delivery/wse:NotifyTo/wsa:Address` is an instance of the **MakeConnection anonymous URI** (e.g. `http://docs.oasis-open.org/ws-rx/wsmc/200702/anonymous?id=550e8400-e29b-11d4-a716-446655440000`).

Sequence

The following diagram illustrates the sequence of messages for the Non-Addressable Event Sink Test.



Note that the `MakeConnection` requests that follow both the `Subscribe` and the `Unsubscribe` requests are optional. It may happen that the `SubscribeResponse` and `UnsubscribeResponse` are both transmitted on the back-channel of their corresponding requests.

Success Criteria

1. Receipt of a valid `Subscribe` message by the Event Source.
2. Receipt of a valid `SubscribeResponse` message by Subscriber.
3. Receipt of one or more unwrapped `Notifications` by the Event Sink.
4. Receipt of a valid `Unsubscribe` message by the Subscription Manager.
5. Receipt of a valid `UnsubscribeResponse` message by the Subscriber.

Conformance

Because this test involves the use of WS-MakeConnection there are a number of failure cases that fall within the boundaries of conforming behavior.

A conforming Subscriber/Event Sink MAY NOT be capable of implementing this test due to its inability to support WS-MakeConnection.

A conforming Event Source MAY respond to the initial Subscribe request with a `wse:UnusableEPR` fault. However, because WS-Eventing does not require Event Sources to validate the NotifyTo EPR at subscribe-time, it MAY be that the Subscribe request succeeds (although the SubscribeResponse is never delivered to the Subscriber) but Notifications are simply not delivered to the Event Sink.

Because Event Sinks and Subscription Managers are not required to implement WS-MakeConnection, the MakeConnection requests MAY elicit a `wsa:ActionNotSupported` fault response or some other, unspecified behavior.

4 WSDLs

4.1 Event Source WSDL

TBD

4.2 Notification WSDL

TBD

5 EventDescriptions

TBD

6 Schemas

TBD

7 Change Log

Version	Date	Author	Changed
Initial	2010-10-26	Gilbert Pilz	Initial revision
0.0.1	2010-10-27	Gilbert Pilz	Fleshed out Basic, Wrapped, and Expiration tests; added sequence diagrams. Added stubs for Renew and Non-Addressable Event Sink tests.
0.0.2	2010-10-28	Gilbert Pilz	Editorial fixes. Changed animal names in honor of the Irish light-bellied Brent Geese tracked by the WWT (http://www.wwt.org.uk/).

Version	Date	Author	Changed
0.0.3	2010-12-03	Gilbert Pilz	Added "Conformance" sections to each test that describe any allowable failures. Added sequence diagrams to Renew Test, SubscriptionEnd Test, and Non-Addressable Event Sink Test.