

Proposal for Issue

http://www.w3.org/Bugs/Public/show_bug.cgi?id=6425

Proposed Changes to WS-Eventing:

1. We propose to add the following paragraph to Section 3.1 of WS-Eventing right after the request message outline box.

The event source is addressed by a WS-Addressing 1.0 EPR and the request message shall be sent to this EPR, as defined in WS-Addressing 1.0 Core Section 3.1-3.3.

2. We may add a sample message (from the above) to illustrate this constraint.

Background Information and Rationale:

WS-Eventing section 2.3 defines Event Source as follows:

Event Source

A Web service that sends notifications and accepts requests to create subscriptions.

In Section 1.4 Table 1, the example message indicates the WS-Addressing EPR <wsa:To> is used to address the event source.

Lines (07-09) in [table 1](#) indicate the message is a request to create a subscription, and Line (16) indicates that it is sent to a hypothetical event source of ocean events.

But it is not clear from this definition and example how a Subscribe request message addresses dynamic event sources exposed by an Event Source.

An example of dynamic event source is the sessions defined in ECMA-366, where a WS-Session service (an Event Source as defined by WS-Eventing) manages multiple sessions created by clients. Each session is an “event source” because it generates an event. The event subscribers are expected to subscribe to individual sessions, instead of the entire service. Since each session is identified by a unique session ID, the session ID needs to be conveyed to the Event Source in the Subscribe request message.

Another example of dynamic event source is the monitors in ECMA-348, where a Switching Function Web service (Event Source) manages multiple monitors created by Computing Function (Event Subscriber), each generating some events. The event subscriber is allowed to subscribe to individual monitor, so as to direct events from different monitors to different sinks. Again, each monitor has a unique ID and this ID needs to be conveyed to the Event Source in the Subscribe request message.

Sometimes it is possible to express this information as a filter. But it depends on existence of a filter implementation. It also does not tie the lifecycle of subscription with that of the dynamic event sources.

Another similar situation is that an Event Source may expose multiple port types, each having some event operations. It would be very useful if an event subscriber can subscribe to a particular port type to receive a subset of events from the Event Source, for which a filter is unnecessary, if not difficult, to define. The identification of port type is necessary because many ports often share the same service location.

The above use cases require an event subscriber to provide some additional information, besides the service location URI defined in the Event Source WSDL, to associate a subscription with the targeted individual event source (a resource or a port type). Current WS-Eventing does not constrain how such additional information should be expressed. Consequently, this information can end up in any extension point in the Subscribe request message, creating unnecessary interoperability issues.

To reduce such ambiguity, we propose to mandate Event Subscriber to express all information pertinent to event source in the [destination] and [reference parameters] properties of the Subscribe request message, as defined in WS-Addressing 1.0 Core Sections 3.1-3.3. Following this approach, given an EPR to address a monitor with ID 12345 in ECMA-348:

```
<wsa:EndpointReference>
  <wsa:Address>http://www.example.org/csta/CallControl</wsa:To>
  <wsa:ReferenceParameters>
    <csta:MonitorCrossRefID>12345</csta:MonitorCrossRefID>
  </wsa:ReferenceParameters>
</wsa:EndpointReference >
```

The corresponding Subscribe request SOAP message sent to this EPR will contain:

```
<s12:Header>
  <wsa:To>http://www.example.org/csta/CallControl</wsa:To>
  <csta:MonitorCrossRefID wsa:IsReferenceParameter='true'>
    12345
  </csta:MonitorCrossRefID>
</s12:Header>
```

Similarly, given an EPR to address a port type dedicated to hurricane alerts:

```
<wsa:EndpointReference>
  <wsa:Address>http://www.example.org/oceanwatch/EventSource</wsa:To>
  <wsa:ReferenceParameters>
    <wsdl:portType name='hurricanePortType' />
  </wsa:ReferenceParameters>
</wsa:EndpointReference >
```

The corresponding Subscribe request SOAP message sent to this EPR will contain:

```
<s12:Header>
  <wsa:To>http://www.example.org/oceanwatch/EventSource</wsa:To>
  <wsdl:portType wsa:IsReferenceParameter='true' name='hurricanePortType' />
</s12:Header>
```

