

Onvif 之 Event Handling（事件处理）

—— 堕落的氏族 整理

ONVIF™

Core Specification

Version 2.2

May , 2012

译者注：被文档一些源文档中的 n 类型的词汇直接拿来代替汉语中的名词，原因是，1、每个人对这些词汇在使用过程理解不同，把原词汇发在这更合适一些。2、有些是 API 词汇不能翻译，有些是在文档中尤其重要指代的词汇，这些词汇可能在一些协议栈中出现，如：gsoap，更有利于一一对应。

声明：此文档仅供参考，不能作为 ONVIF 中该部分的译文使用。

告警是能被设备探测到的事件或行为，而这些告警是可以被客户订阅的。告警由 event server 处理。一个符合 ONVIF 协议的设备应该提供在[ONVIF Event WSDL]里定义的告警服务。设备和客户两者都应该为告警服务支持[WS-Addressing]。

规范里的事件处理是以[WS-BaseNotification] 和 [WS-Topics]（这两个）说明为基础的。规范的事件处理依赖于 basic notification interface 实现（basic notification interface 在 section 9.1 作了阐述），而规范的这点是完全符合[WS-BaseNotification] 协议的。除此之外，设备应该实现 Real-time Pull-Point Notification Interface 和 the Notification Streaming Interface，以上的两者分别在 sections 9.2 and 9.3 作了介绍。

事件处理标准描述了通告信息扩展（notification message extensions），而这个扩展允许客户通过告警追溯（告警）源属性信息（例如：视频移动侦测 Property）。（告警处理）性能（Properties）定义在 Section 9.4。（译者注：之所以把 Properties 译成性能是因为这个扩展力描述了 event servers 可以提供的几种告警，每种告警各自的属性）

事件的载体和订阅（告警事件）过滤的描述在 Section 9.5 讨论。Section 9.6 描述了终端同步的请求怎样可以让客户用 3 个 Notification Interfaces 实现。Section 9.7 对可订阅内容做了整体（概要）描述。Section 9.9 对错误处理作了阐述。

最后的 section 在细节上对 Real-Time Pull-Point Notification Interface 的使用作了展示，包括信息过滤和订阅设置（Message Filtering and Topic Set）。basic notification interface 示例可在相应的[WS-BaseNotification]协议中找到。

9.1 Basic Notification Interface

Section 9.1.1 简短地对[WS-BaseNotification]协议中 Basic Notification Interface 作了介绍。Section 9.1.2 概述了[WS-BaseNotification]协议中的命令（mandatory）和 optional interfaces。

9.1.1 Introduction（介绍）

Notification 部分可以分为下面几个逻辑体：

Client: implements the NotificationConsumer interface. （客户端：实现 NotificationConsumer API）

Event Service: implements the NotificationProducer interface. （告警服务器：实现 NotificationProducer API）

Subscription Manager: implements the BaseSubscriptionManager interface. （订阅管理:实现 BaseSubscriptionManager API）

告警服务和订阅管理应该在设备上实现。

典型的 Messages 在三个逻辑体之间的交互见图示 9。首先，客户端创建一个到 Event Server（Event Server：告警服务器）的连接。然后客户端可以通过发送一个 SubscriptionRequest(到 Event Server)订阅某些 Notifications（Notifications：告警上报）。如果 Event Server 接收到订阅请求，则动态地实例化一个 SubscriptionManager（SubscriptionManager：订阅管理）发送订阅（的告警信息）。Event Server 应该在订阅回应中返回 SubscriptionManager 的 WS-Endpoint-Address（译者注即：订阅管理的服务程序所在的 PC 的 IP，端口等信息）。

为了传递符合订阅的 Notifications，还需要在 Event Server 和 Client 之间建立另外一个连接。通过这个连接，Event Server 单向发送 Notify Message 到 Client

的 NotificationConsumer interface （ NotificationConsumer interface ： 客户端 NotificationConsumer API）。只要是订阅有效，符合订阅的 Notifications 就由 Event Server 发送到 Client。

为了控制订阅，Client 直接与在 SubscriptionResponse 中返回的 SubscriptionManager 通信。在 SubscriptionRequest 中，Client 能指定订阅的过期/结束时间。SubscriptionManager 在订阅过期后自动销毁。RenewRequests 能让 Client 修改过期时间。Client 也可以发送 UnsubscribeRequest 主动结束 SubscriptionManager。UnsubscribeRequest 成功后，

SubscriptionManager 就不再存在。

EventService 和 SubscriptionManager 之间的交互不在由[WS-BaseNotification]指定，而是由设备直接执行实现。

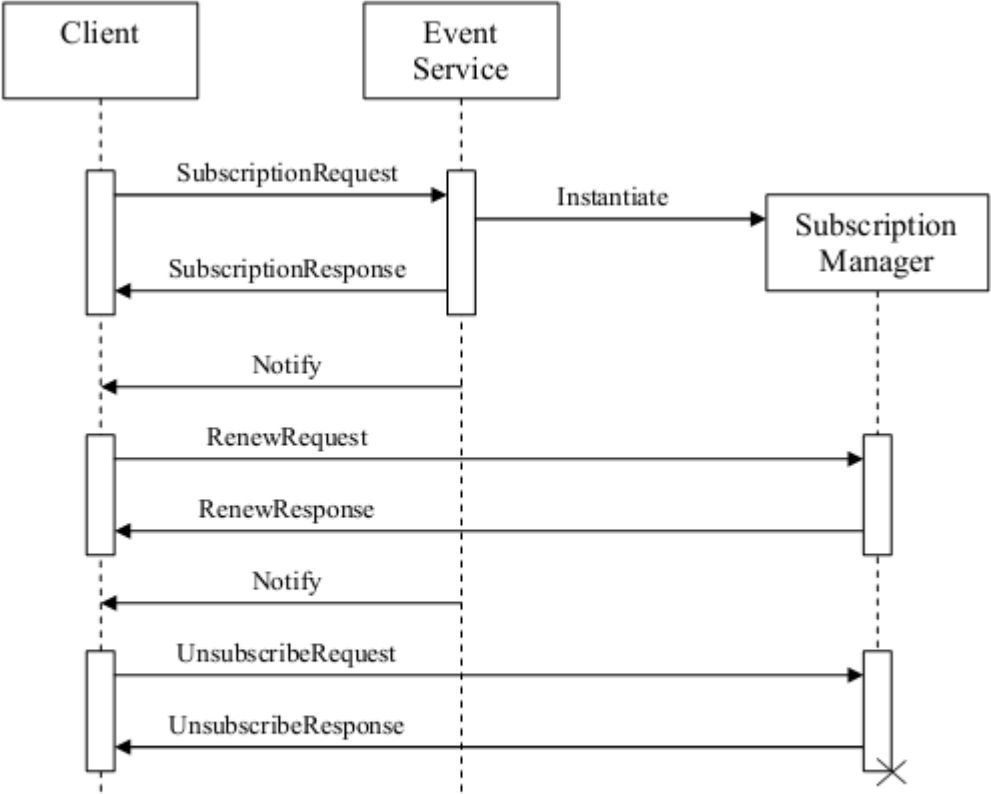


Figure 9: Sequence diagram for the Base Notification Interface

9.1.2 Requirements

这章阐述了设备应该提供的[WS-BaseNotification]中的 Interfaces。

符合 ONVIF 协议的设备应该支持 [WS-BaseNotification] 中的 NotificationProducer Interface。因此，NotificationProducer Resource Properties 是可选（见 Section 9.5）。设备应该至少支持 TopicExpression 和 MessageContent filters 在 Sections 9.5.5 和 9.7.3 中的原语描述。如果设备没有接收到订阅的起始时间，那么它应该用 Fault Message 提供一个有效的起始时间。设备应该用[WS-BaseNotification]文档中的 Notify 格式提供 Notifications。

SubscriptionPolicy wsnt:UseRaw 对设备来说是可选的。尽管[WS-BaseNotification]有 CurrentTime 和 TerminationTime 在 SubscribeResponse 和 RenewResponse 作为可选参数，但是符合 ONVIF 协议的设备还是在 SubscribeResponses 和 RenewResponse 中应该列出它们。

设备也许用一个表示当前没有符合请求类型订阅的 Fault message 回应 GetCurrentMessage 请求。

[WS-BaseNotification]中 Pull-Point Interface 的执行在设备上可选的。

符合 ONVIF 设备应该实现[WS-BaseNotification]描述、符合 Renew and Unsubscribe 操作的 Base Subscription Manager Interface。Pausable Subscription Manager Interface 可选。Subscriptions 中 WS-Resources 的实现是可选的。

符合 ONVIF 设备应该支持请求参数以 Z 标识格式 utc 的 time values, 并且回应所有 time values, 格式为以 Z 标识的 utc。

9.2 Real-time Pull-Point Notification Interface

此章节介绍 Real-time Pull-Point Notification Interface。此 Interface 提供了一个不受防火墙影响、能实时轮训的 Notification Interface, 而且初始化所有的 Client 通信。

Interface 以以下方式使用:

1) The client asks the device for a PullPointSubscription with the

CreatePullPointSubscriptionRequest message. The request contains a detailed description of the Subscription. The ConsumerReference shall be omitted, in contrast to the subscription of the Basic Notification Interface (see Section 9.1).

(译: Client 用 CreatePullPointSubscriptionRequest 信息向设备请求一个 PullPointSubscription。请求信息包含了一个 Subscription 的详细描述。请求中 ConsumerReference 需忽略, 参考 Basic Notification Interface(见 Section 9.1)的 subscription)

2) The device evaluates the Subscription and returns either a

CreatePullPointSubscriptionResponse when the Subscription is accepted or one of the Fault codes.

(译: 设备判断订阅, 而后或者当订阅接收到时返回一个 CreatePullPointSubscriptionResponse, 或者返回一个错误号)

If the Subscription is accepted, the response contains a WS-EndpointReference to a SubscriptionManager. This WS-Endpoint shall provide a PullMessages operation, which is used by the client to retrieve Notifications and by the Base Subscription Manager Interface described in the [WS-BaseNotification] specification. The Base Subscription Manager Interface consists of PullMessages, Renew and Unsubscribe operations. The sequence diagram of the interaction is

shown in Figure 10 . The PullMessagesRequest contains Timeout and MessageLimit parameters.

(译：如果订阅被接收到，回应就要包含一个到 SubscriptionManager 的 WS-EndpointReference 信息。WS-Endpoint 应该提供一个 PullMessages 操作，而其用于 Client 接收 Notifications 信息和[WS-BaseNotification]中描述的 Subscription Manager Interface。Base Subscription Manager Interface 包含 PullMessages，Renew 和 Unsubscribe 操作。该类型的交互操作见 Figure 10。PullMessagesRequest 包括 Timeout 和 MessageLimit 参数)。

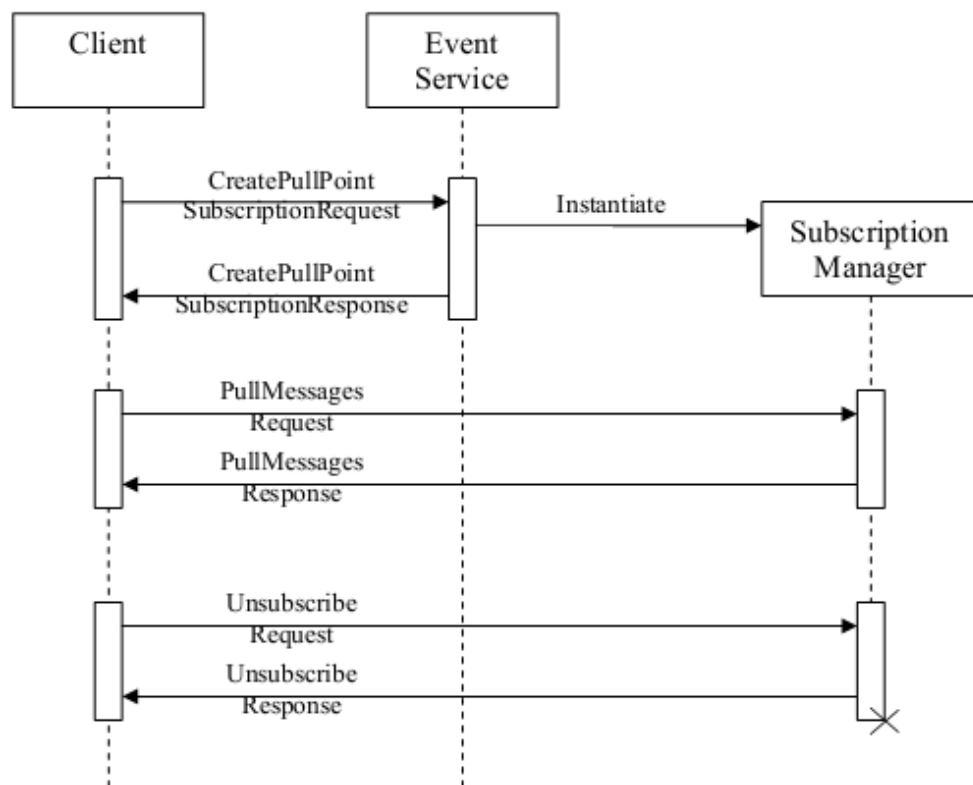


Figure 10: Sequence diagram for the Real-time Pull-Point Notification Interface.

The device shall immediately respond with notifications that have been aggregated on behalf of the client. If there are no aggregated notifications, the device waits with its response until either a notification is produced for the client or the specified Timeout is exceeded. In any case, the response will contain, at most, the number of notifications specified by the MessageLimit parameter. The client can poll the notifications in real-time when it starts a new PullMessagesRequest immediately after each PullMessagesResponse.

(译：设备应该及时回应获取到的符合 Client 订阅的 Notifications。如果没有获取到符合订阅的 Notification，那么设备阻塞回应直到设备获取符合 Client 订阅一个 Notification 或

者指定的 timeout 到达。在任何状况下，回应将最多包含 MessageLimit 参数指明的 Notification 数目。当 Client 在收到 PullMessagesResponse 后立即发送下一个 PullMessagesRequest，以此 Client 能实时轮训获取到 Notifications。）

If neither a termination time nor a relative termination time is set in the CreatePullPointSubscriptionRequest, each PullMessagesRequest shall be interpreted as a keep-alive for the corresponding PullPointSubscription. The termination time is recomputed according to the relative termination time if available or according to a device internal default value. To inform the client about the updated termination time, the PullMessagesResponse shall contain the CurrentTime and TerminationTime elements. When the PullMessagesRequest is used as keep-alive for the corresponding PullPointSubscription, the RenewRequest, defined by the [WS-BaseNotification], need not be called by a client. Nevertheless, the device shall support it for the PullPointSubscription.

（译：如果在 CreatePullPointSubscriptionRequest 中既没有设置绝对超时时间也没有设置相对超时时间，那么每个 PullMessagesRequest 作为一个保持连接的 PullPointSubscription 通信。根据相对时间或者根据设备内部默认值检查过期时间是否有效。为了通知 Client 关于更新结束时间，PullMessagesResponse 应该包含 CurrentTime 和 TerminationTime 项。当 PullMessagesRequest 用于 PullPointSubscription 通信的保持连接，[WS-BaseNotification]定义的 RenewRequest 不需要 Client 调用。虽然如此，设备应该支持 PullPointSubscription。）

9.2. 1 Create pull point subscription

设备应该支持下面的 CreatePullPointSubscription 信令。如果没有指定 Filter 项，Pullpoint 应该通知给 Client 所有出现告警。

设备需支持在 utc 中指定绝对时间值和 InitialTerminationTime 参数的相对时间值。设备要回应 CurrentTime 和 TerminationTime 参数格式为 Z 标识的 utc。

Table 94: CreatePullPointSubscription command

CreatePullPointSubscription		Access Class: READ_MEDIA
Message name	Description	
CreatePullPointSubscriptionRequest	<p><i>This message contains the same elements as the SubscriptionRequest of the [WS-BaseNotification] without the ConsumerReference:</i></p> <p>wsnt:FilterType Filter [0][1] wsnt:AbsoluteOrRelativeTimeType InitialTerminationTime [0][1] xs:any SubscriptionPolicy [0][1]</p>	
CreatePullPointSubscriptionResponse	<p><i>The response contains the same elements as the SubscriptionResponse of the [WS-BaseNotification]:</i></p> <p>wsa:EndpointReferenceType SubscriptionReference [1][1] xs:dateTime CurrentTime [1][1] xs:dateTime TerminationTime [1][1]</p>	
Fault codes	Description	
	<p><i>The same faults as for Subscription Request of the [WS-BaseNotification] are used.</i></p>	

9.2.2 Pull messages（轮训 Message）

设备需为 CreatePullPointSubscription 信令请求返回的所有 SubscriptionManager endpoints 提供下面 PullMessages 信令。

这个信令需至少支持一分钟的 Timeout。没有生成错误信息返回的情况下，设备支持缺失请求类型 Messages 的检索。

设备将回应以 Z 标识 utc 格式的 CurrentTime 和 TerminationTime 参数。

Table 95: PullMessages command

PullMessages		Access Class: READ_MEDIA
Message name	Description	
PullMessagesRequest	<p><i>This message shall be addressed to a SubscriptionManager in order to pull notifications:</i></p> <p>xs:duration Timeout [1][1] xs:int MessageLimit [1][1]</p>	
PullMessagesResponse	<p><i>The response contains a list of notifications together with an updated TerminationTime for the SubscriptionManager.</i></p>	

	xs:dateTime CurrentTime [1][1] xs:dateTime TerminationTime [1][1] wsnt:NotificationMessageHolderType NotificationMessage [0][unbounded]
PullMessagesFaultResponse	<i>The Timeout exceeds the upper limit supported by the device. The Fault Message shall contain the upper limits for both parameters.</i> xs:duration MaxTimeout [1][1] xs:int MaxMessageLimit [1][1]
Fault codes	Description
	<i>No specific fault codes.</i>

9.3 Notification Streaming Interface

ONVIF Media Service 说明中的 Metadata Configuration 章节描述了 metadata configurations 的 creation、deletion 和 modification。某些 metadata configurations 可以包含多重订阅，notification 订阅中 subscription 结构是相同的。当包含订阅的 metadata configuration 赋值到 profile 中时，Client 使用这个 profile 去获取一个 RTP Stream，此 RTP Stream 拥有 Metadata 配置的 Notifications。取于 RTP 的 Notification Stream 应该在符合 ONVIF 协议、支持 ONVIF Media Server 的设备上实现。

[WS-BaseNotification]定义了以 wsnt:NotificationMessage 标签封装 Message Payload，Topic 和 ProducerReference。Message 结构与直接 Notification 请求格式相同（格式在 Section 9.5 有其描述）。wsnt:NotificationMessage 项多重嵌套的形式可以由在 Real-time Viewing 章节中描述的 Metadata Document 代替。

没有明确的 Stream 化 Notifications 的 SubscriptionReference。因此，wsnt:NotificationMessage 不应该包含 SubscriptionReference 项。

9.4 Properties

Properties 是以名字和值成对使用来描述唯一、地址可设置的数据的综合。它们是众多 Topic、Source 和 Key 值共用的唯一识别方式，且像普通 Events 一样封装。一个 Property 也可拥有一个追加 Flag，标识它是否为新创、修改或删除。

当 Client 订阅一个标识某种性能的 Topic 时，设备应该提供符合请求属性的所有告警 Notifications 传递给 Client，只要在订阅期限内 property 存在。Client 也能在任何时间通过访问同步 Point，请求所有 Client 订阅过的当前存在的 property（属性）的值（见 section 9.6）。

Property Interface 定义在这个规范里，为了组织所有关于 Events 的 Property 在一起，并统一提供给 Client。推荐使用 Property Interface，只要适用。Section 9.5 详细说明了 Events

和 Properties 的结构。

9.4. 1 Property Example

下面的 Video analytics 示例展示了动态运行状况 Properties: Video Analytics Detector 的 Rule Engine Interface 能定义 Fields。这种 Detector Field 在 Image Plane 中以多边形描述。对于屏幕中的每个物体, Rule Engine 决定了哪种物体在这个多边形内。Client 能订阅获取关于 Detector Filed 的符合物体进入 Property 的信息。每次一个物体出现在屏幕中, 一个新的物体进入的 Property 就被创建。Client 能被通知, 通过相应”Property Created“ Notification, 内容为物体是否出现在了多边形的内部或外部。每次物体进入或离开这个多边形, 一个“property changed”

Notification 就被生成, 内容为该物体已经变化的内部物体 Property。当一个物体离开了屏幕, 相应内部物体 Property 将被删除, 同时 Client 获得一个 “p roperty deleted” Notification 被告知。

9.5 Notification Structure

下面的 xml 代码是 wsnt:NotificationMessage 在[WS-BaseNotification]的格式:

```
<xs:complexType name="NotificationMessageHolderType" >
  <xs:sequence>
    <xs:element ref="wsnt:SubscriptionReference" minOccurs="0" />
    <xs:element ref="wsnt:Topic" minOccurs="0" />
    <xs:element ref="wsnt:ProducerReference" minOccurs="0" />
    <xs:element name="Message">
      <xs:complexType>
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

```
<xs:element name="NotificationMessage"
            type="wsnt:NotificationMessageHolderType"/>
```

下面是相应的 XML 结构:

```
<wsnt:NotificationMessage>
  <wsnt:SubscriptionReference>
    wsa:EndpointReferenceType
  </wsnt:SubscriptionReference>
  <wsnt:Topic Dialect="xs:anyURI">
    ...
  </wsnt:Topic>?
  <wsnt:ProducerReference>
    wsa:EndpointReferenceType
  </wsnt:ProducerReference>
  <wsnt:Message>
    ...
  </wsnt:Message>
</wsnt:NotificationMessage>
```

wsnt:Message 标签间包含的是 Notification 的真实载体。Message 的 XML 格式可以由 TopicTree 定义中指定（参考 Section 9.7）。

Section 9.5.1 给出了一个 Client 通过 Notifications 获取告警上报的示例。

Section 9.5.2 给出了一个详细的 Message 载体的格式，Section 9.5.4 介绍 Message 载体的描述语言。Section 9.5.5 定义了 Message 内容上用于过滤 Notifications 订阅的语法。

9.5.1 Notification information

Notification 回答至少以下问题:

什么时候发生的?

这 Event 谁产生的?

出现了什么状况?

“什么时候发生的?”问题由增加一个时间属性到 NotificationMessage 的 Message 项回

答。符合 ONVIF 设备应该包括 Message 项的时间属性。

“这 Event 谁产生的？”问题分为两部分。一部分是 WS-Endpoint，它指明设备或设备上的服务器那里已经产生了 Notification。因此，WS-Endpoint 应该在 NotificationMessage 的 ProducerReference 项里指定。第二部分是 WS-Endpoint 中成员的标准识别，以保证 Notification 的产生可查。关于这些成员，嵌套/多重参数或空也许用于成员的唯一性鉴别。这些参数填写在 Message 结构内的 Source 项中 items 中。

“出现了什么状况？”的问题回答份两个步骤。首先，NotificationMessage 的 Topic 项用于 Event 分类。其次，Itmes 增加到 Message 结构的数据项中以描述 Event 的细节。

当 topic 指向这些 Properties 时，Client 用 NotificationProducer、Topic、Source Items 和可选的 Key Items（参考 Section 9.5）来识别 Property。这些值将构成唯一性的鉴别。

9.5.1.1 Event Example

下面的例子展示了 Notification 不同的部分。

```
<wsnt:NotificationMessage>

...

<wsnt:Topic Dialect="...Concrete">

    tns1:PTZController/PTZPreset/Reached

</wsnt:Topic>

<wsnt:Message>

    <tt:Message UtcTime="...">

        <tt:Source>

            <tt:SimpleItem Name="PTZConfigurationToken" Value="PTZConfig1"/>

        </tt:Source>

        <tt:Data>

            <tt:SimpleItem Name="PresetToken" Value="Preset5"/>

            <tt:SimpleItem Name="PresetName" Value="ParkingLot"/>

        </tt:Data>

    </tt:Message>

</wsnt:Message>

</wsnt:NotificationMessage>
```

Item “PTZConfigurationToken”标识了成员的唯一性，其为获取到 Event 的标志。在这个示例中，成员是 PZT Configuration “PTZConfig1”中提到的 PZT Node。

Event tns1:PTZController/PTZPreset/Reached 表明 PTZ 部分已经预先设置。Data 区域包含了预设中它的信息。因此，Preset 定义在用 “PresetName”命名 PresetToken “Preset5”部分。

9.5.2 Message Format

NotificationMessage 的 Message element 定义在[ONVIF Schema]。定义如下：

```
<wsnt:NotificationMessage>
...
<wsnt:Topic Dialect="...Concrete">
    tns1:PTZController/PTZPreset/Reached
</wsnt:Topic>
<wsnt:Message>
    <tt:Message UtcTime="...">
        <tt:Source>
            <tt:SimpleItem Name="PTZConfigurationToken" Value="PTZConfig1"/>
        </tt:Source>
        <tt:Data>
            <tt:SimpleItem Name="PresetToken" Value="Preset5"/>
            <tt:SimpleItem Name="PresetName" Value="ParkingLot"/>
        </tt:Data>
    </tt:Message>
</wsnt:Message>
</wsnt:NotificationMessage>
```

Message element 中的 Items 有 3 部分组成：Source，Key 和 Data。Key Group 和 Properties 无关，不应该用于 Notifications。嵌套的 Sipple 和 Element IItems 在彼此的内部能相互替换。

每个 Item 有一个名字和一个值。一个 ElementItem，它的 value 可以有 ElementItem element 内的一个 XML element 表示。一个 SimpleItem，它的 value 应该有 Value Attribute 指定。所有 Items 的 Name 在这 Message 任何 Group 的众多 Item 中是唯一的。

私有特定扩展要以 QName 形式标明在 SimpleItem 和 ElementItem 的 Name Attribute。但要避免私有特定扩展和后续 Onvif 扩展之间的 Name 冲突问题。

协议推荐在用 SimpleItems element 可以代替 ElementItems element 时候以 SimpleItems 代替，因为 SimpleItems 简化 Message 综合后给一般 Client。Simple 和 ElementItems 两者的确切类型信息可以在 TopicSet 中提取到，而这些信息可以在 Message 负载描述中扩展。

PropertyOperation 应在 Notification 关联到某种 Property 前进行预先设置。使用 操作模式 “Initialized” 通知 Client 关于 Property 的创建状态。当 Synchronization Point 已经请求成功时，操作模式 “Initialized” 也应该被使用。

9.5.3 Property example, continued

section 9.4.1 中示例请求了一个可选 Key Item(本人怀疑文档有缺失,section 9.4.1 没有该示例)。本章节中的示例展示了 Key Item 的应用方法。Rule Engine 可以包含 FieldDetector Rules。Rules 为屏幕中的每一个（影像）个体定义了一个 ObjectsInside Property。当一个新的（影像）个体出现在这个 Field 外部时，就会产生如下面的 Notification：

```
<wsnt:NotificationMessage>
...
<wsnt:Topic Dialect="...Concrete">
    tns1:RuleEngine/FieldDetector/ObjectsInside
</wsnt:Topic>
<wsnt:Message>
    <tt:Message UtcTime="..." PropertyOperation="Initialized">
        <tt:Source>
            <tt:SimpleItem Name="VideoSourceConfigurationToken" Value="1"/>
            <tt:SimpleItem Name="VideoAnalyticsConfigurationToken" Value="1"/>
            <tt:SimpleItem Name="Rule" Value="myImportantField"/>
        </tt:Source>
        <tt:Key>
```

```

        <tt:SimpleItem    Name="ObjectId"    Value="5"/>
    </tt:Key>

    <tt:Data>

        <tt:SimpleItem Name="IsInside"    Value="false"/>

    </tt:Data>

</tt:Message>

</wsnt:Message>

</wsnt:NotificationMessage>

```

Source Items 描述了产生该类 Notification 的 Rule。当（影像）个体重叠出现在屏幕中时，这些个体的每一个个体都有自己的 ObjectsInside Property。因此，Object ID 用作额外 Key Item 以标识 Property 的唯一性。 IsInside Item 是一个 Boolean 值，用于标识个体在 Field 域内还是外。

当（影像）个体进入某 Field 时， Rule 产生一个 “property changed” Message，其格式如下：

```

<wsnt:NotificationMessage>

...

<wsnt:Topic Dialect="...Concrete">

    tns1:RuleEngine/FieldDetector/ObjectsInside

</wsnt:Topic>

<wsnt:Message>

    <tt:Message UtcTime="..." PropertyOperation="Changed">

        <tt:Source>

            <tt:SimpleItem Name="VideoSourceConfigurationToken"    Value="1"/>

            <tt:SimpleItem Name="VideoAnalyticsConfigurationToken" Value="1"/>

            <tt:SimpleItemName="Rule"                                Value="myImportantField"/>

        </tt:Source>

        <tt:Key>

            <tt:SimpleItem Name="ObjectId"                                Value="5"/>

```

```

        </tt:Key>

        <tt:Data>

            <tt:SimpleItem Name="IsInside"                                Value="true"/>

        </tt:Data>

    </tt:Message>

</wsnt:Message>

</wsnt:NotificationMessage>

```

最后，当（影像个体）离开屏幕时，“property deleted” Message 就会产生，格式如下：

```

<wsnt:NotificationMessage>

...

    <wsnt:Topic Dialect="...Concrete">

        tns1:RuleEngine/FieldDetector/ObjectsInside

    </wsnt:Topic>

    <wsnt:Message>

        <tt:Message UtcTime="..." PropertyOperation="Deleted">

            <tt:Source>

                <tt:SimpleItem Name="VideoSourceConfigurationToken"    Value="1"/>

                <tt:SimpleItem Name="VideoAnalyticsConfigurationToken" Value="1"/>

                <tt:SimpleItem      Name="Rule"    Value="myImportantField"/>

            </tt:Source>

            <tt:Key>

                <tt:SimpleItem      Name="ObjectId"    Value="5"/>

            </tt:Key>

        </tt:Message>

    </wsnt:Message>

</wsnt:NotificationMessage>

```

有一种情况是，在（影像）个体和它的对应的 Property 不存在时，Data Item 能省略。

9.5.4 Message Description Language

Message 载体结构已经在前面的章节介绍。结构有 3 部分组成：Source, Key, and Data。每部分包含一系列的 Simple 和 ElementItems。对于每个 Topic，设备必须能用 Message Description Language 描述哪个 Item 是属于该 Topic 产生的 Notification 的部分。下面 Description Language 描述了 Message 必须的 3 个 Item：

```
<xs:complexType name="MessageDescription">
  <xs:sequence>
    <xs:element name="Source" type="tt:ItemListDescription"
      minOccurs="0"/>
    <xs:element name="Key" type="tt:ItemListDescription" minOccurs="0"/>
    <xs:element name="Data" type="tt:ItemListDescription" minOccurs="0"/>
    ...
  </xs:sequence>
  <xs:attribute name="IsProperty" type="xs:boolean"/>
</xs:complexType>

<xs:complexType name="ItemListDescription">
  <xs:sequence>
    <xs:element name="SimpleItemDescription"
      minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="Name" type="xs:string" use="required"/>
        <xs:attribute name="Type" type="xs:QName" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="ElementItemDescription"
      minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="Name" type="xs:string" use="required"/>
```



```

        <xs:attribute name="Type" type="xs:QName" use="required"/>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

一个 Item 的 Name 属性应该在相同部分(Source, Key, Data)所有 Item 中是唯一的。

IsProperty 属性在相关 Property 的被描述 Message 时，应该设置成“true”。然而，Message 没有相关 Property，Key 部分应该省略。SimpleItemDescriptor 的 Type 属性应以简化形式定义在 XML schema（以简化形式创建），ONVIF schemas 或私有 schemas 中。同样，ElementItemDescriptor 的 Type 属性应该与一个 XML 模式的全局 Element 声明相匹配。

所有用于描述 Message 载体的 Schema 文件源址在 Section 9.8 的 GetEventPropertiesResponse 信息中列出。

9.5.4.1 Message Description Example

下面的代码是一个符合 Section 9.5.3 Property 示例的 Message Description 示例。

```

<tt:MessageDescription IsProperty="true">
    <tt:Source>
        <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
                                Type="tt:ReferenceToken"/>
        <tt:SimpleItemDescriptionD Name="VideoAnalyticsConfigurationToken"
                                Type="tt:ReferenceToken"/>
        <tt:SimpleItemDescription Name="Rule"
                                Type="xs:string"/>
    </tt:Source>
    <tt:Key>
        <tt:SimpleItemDescription Name="ObjectId"
                                Type="xs:integer"/>
    </tt:Key>

```

```

<tt:Data>

  <tt:SimpleItemDescription Name="IsInside"

    Type="xs:boolean"/>

</tt:Data>

</tt:MessageDescription>

```

9.5.5 Message Content Filter

在 Subscription 请求中，Client 能通过 TopicExpression 和 MessageContent 过滤 Notifications。对于后者，[WS-BaseNotification] 建议使用 XPath 1.0 原语。由于协议要求的特定的 Message 结构，协议需要 XPath 1.0 语法的一部分。符合 ONVIF 协议设备应该支持 XPath 1.0 的部分语法。类似下面 URI 的语法被引用：
Dialect=http://www.onvif.org/ver10/tev/messageContentFilter/ItemFilter。

优先级与结合性：

‘and’运算比‘or’运算的优先级高。‘and’和‘or’运算都是左结合。

在下面语法定义中的‘and’和‘or’运算的优先级和结合性与 XPath 1.0 协议相同。

表达式形式如下：

- [1] Expression ::= BoolExpr | Expression ‘and’ Expression
 | Expression ‘or’ Expression | ‘(’ Expression ‘)’ | ‘not’ ‘(’ Expression ‘)’
- [2] BoolExpr ::= ‘boolean’ ‘(’ PathExpr ‘)’
- [3] PathExpr ::= [‘//’ Prefix? ‘SimpleItem’ | ‘//’ Prefix? ‘ElementItem’] NodeTest
- [4] Prefix ::= NamespacePrefix ‘:’ | ‘’
- [5] NodeTest ::= ‘[’ AttrExpr ‘]’
- [6] AttrExpr ::= AttrComp | AttrExpr ‘and’ AttrExpr | AttrExpr ‘or’ AttrExpr | ‘(’
 AttrExpr ‘)’
 | ‘not’ ‘(’ AttrExpr ‘)’
- [7] AttrComp ::= Attribute ‘=’ ‘ “ ’ String ‘ ” ’
- [8] Attribute ::= ‘@Name’ | ‘@Value’

语法允许测试属于各部分 (Source, Key or Data) 的 Simple 或 ElementItems 的出现 (即: 使用)。此外, SimpleItems 的值能被检测。SimpleItem 和 ElementItem 的 Prefix namespace (即: 命名空间前缀) 与 “http://www.onvif.org/ver10/schema” 相符。

最后, 测试的属性 boolean 综合表示是合理的。下面的表达被允许:

仅返回包含关于 VideoSourceConfiguration “1” 的 Notifications:

```
boolean(/tt:SimpleItem[@Name=" VideoSourceConfigurationToken " and
@Value=" 1" ])
```

仅返回不包含关于 VideoAnalyticsConfiguration 的 Notification:

```
not( boolean(/tt:SimpleItem[@Name=" VideoAnalyticsConfigurationToken
" ]))
```

仅返回关于 VideoAnalyticsConfiguration “2” 运行和 VideoSourceConfiguration “1” 的 Notification:

```
boolean(/tt:SimpleItem[@Name=" VideoAnalyticsConfigurationToken " and
@Value=" 2" ])
and
boolean(/tt:SimpleItem[@Name=" VideoSourceConfigurationToken " and
@Value=" 1" ])
```

仅返回关于 VideoSourceConfiguration “1” 但不涉及 VideoAnalyticsConfigurations 的 Notifications:

```
boolean(/tt:SimpleItem[@Name=" VideoSourceConfigurationToken " and
@Value=" 1" ])
and
not(boolean(/tt:SimpleItem[@Name=" VideoAnalyticsConfigurationToken "
]))
```

```
仅返回有个体进入或出现在 “myImportantField” 域时的 Notifications:  
  
boolean(//tt:SimpleItem[@Name=” IsInside” and @Value=” true” ] )  
  
and  
  
boolean(//tt:SimpleItem[@Name=” Rule” and @Value=” myImportantField” ] )
```

9.6 Synchronization Point

在 section 9.4 介绍的 Properties 以统一的方式通知 Client 关于 Property 的创建,修改和删除。当 Client 想它的 properties 与设备上的 properties 同步时, 发送 Synchronization Point 请求令其重新生成 Properties 中 Client 订阅过的 Properties 的当前状态。所有已生成的 Notifications 的 PropertyOperation 被设置成“Initialized”(参考 Section 9.5)。Synchronization Point 由 SubscriptionResponse 或 CreatePullPointSubscriptionResponse 返回的 SubscriptionManager 请求。Property 更新通过 Notification Interface 以 Notification 通信方式传递。所有 Subscription Manager Endpoints 必须支持下面的操作:

Table 96: SetSynchronizationPoint command

SetSynchronizationPoint		Access Class: READ_MEDIA
Message name	Description	
SetSynchronizationPoint-Request	This message is empty.	
SetSynchronizationPoint-Response	This message is empty.	
Fault codes	Description	
	No command specific faults!	

当 Client 用 Notification Streaming Interface 时, 那么他应该使用在 ONVIF Media Service Specification 定义的 SetSynchronizationPoint 操作。

9.7 Topic Structure

此说明扩展了在[WS-Topics] 标准里定义的 Topic 框架。Section 9.7.1 阐述 ONVIF Topic Namespace, 这些 ONVIF Topic Namespace 可以作为私有特定 topics 的标准。附录中 “Fehler! Verweisquelle konnte nicht gefunden werden.”为这种扩展做了典型的模板展示。Section 9.7.2 为 topic properties 定义了一个 Interface。这个 Interface 是符合 ONVIF 标准设备要实现的。

Section 9.7.3 把 section 9.5.4 中定义的 Message Description Language 融进了 TopicSet 结构。来自于 ONVIF Topic Namespace 的所有所有 topics 根据 section 9.7.3 描述 topic 的类型。Section 9.7.3 规定了设备必需支持的 Topic Expression Dialects。

9.7.1 ONVIF Topic Namespace

[WS-Topics]标准在归属于某特定 Topic Namespace 的定义和由某特定 Web Service 提供的 Topic Set 间有所区别。这个区别允许供应商定义一个仅用于被定义部分的普通 NameSpace。如果已经存在的 Topic Namespace 的 Topic Tree 包含了某设备仅有的有效 topics 部分,那么 Topic Tree 能通过定义一个新的 Topic Namespace 进行增加。新的 Topic Namespace 以 [WS-Topics]描述的标准追加到已经存在的 Topic Namespace 中去。

下面的根 topic 定义在 ONVIF Namespace 中。所有涉及这些 topics 的 Notifications 应该用在 Section 9.5.2 中 Message Formate。

```
<wstop:TopicNamespace name="ONVIF"
targetNamespace="http://www.onvif.org/ver10/topics" >
  <wstop:Topic name="Device"/>
  <wstop:Topic name="VideoSource"/>
  <wstop:Topic name="VideoEncoder"/>
  <wstop:Topic name="VideoAnalytics"/>
  <wstop:Topic name="RuleEngine"/>
  <wstop:Topic name="PTZController"/>
  <wstop:Topic name="AudioSource"/>
  <wstop:Topic name="AudioEncoder"/>
  <wstop:Topic name="UserAlarm"/>
  <wstop:Topic name="MediaControl"/>
  <wstop:Topic name="RecordingConfig"/>
  <wstop:Topic name="RecordingHistory"/>
  <wstop:Topic name="VideoOutput"/>
  <wstop:Topic name="AudioOutput"/>
  <wstop:Topic name="VideoDecoder"/>
  <wstop:Topic name="AudioDecoder"/>

  <wstop:Topic name="Receiver"/>
</wstop:TopicNamespace>
```

9.7.2 Topic Type Information

规范的 Information 通过 添加一个 Section 9.5.4 定义的 MessageDescriptionType 类型的 MessageDescription 元素的格式添加到 topic 元素里。Topic 元素能被 wstop:topic Attribute 以 value 值为“true”确定。

下面的示例展示了 TopicSet 的 Topics 怎么样以 Message Descriptions 的格式添加:

```

<tnsl:RuleEngine wstop:topic="true">
  <tnsl:LineDetector wstop:topic="true">
    <tnsl:Crossed wstop:topic="true">
      <tt:MessageDescription>
        <tt:Source>
          <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
                                   Type="tt:ReferenceToken"/>
          <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
                                   Type="tt:ReferenceToken"/>
          <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
        </tt:Source>
        <tt:Data>
          <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
        </tt:Data>
      </tt:MessageDescription>
    </tnsl:Crossed>
  </tnsl:LineDetector>
  <tnsl:FieldDetector wstop:topic="true">
    <tnsl:ObjectsInside wstop:topic="true">
      <tt:MessageDescription IsProperty="true">
        <tt:Source>
          <tt:SimpleItemDescription Name="VideoSourceConfigurationToken"
                                   Type="tt:ReferenceToken"/>
          <tt:SimpleItemDescription Name="VideoAnalyticsConfigurationToken"
                                   Type="tt:ReferenceToken"/>
          <tt:SimpleItemDescription Name="Rule" Type="xs:string"/>
        </tt:Source>
        <tt:Key>
          <tt:SimpleItemDescription Name="ObjectId" Type="xs:integer"/>
        </tt:Key>
        <tt:Data>
          <tt:SimpleItemDescription Name="IsInside" Type="xs:boolean"/>
        </tt:Data>
      </tt:MessageDescription>
    </tnsl:ObjectsInside>
  </tnsl:FieldDetector>
</tnsl:RuleEngine>

```

9.7.3 Topic Filter

符合 ONVIF 设备应该支持[WS-Topics]标准中定义的 Concrete Topic Expressions。标准定义了 Topic Trees 中的特定 Topic 的识别方式。下面的 Dialect 原语要被指定，只要 Concrete Topic Expression 用于 Subscription Filter 的 Concrete Topic Expression：
<http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete>。

下面的 Topic Expression 语法需要被设备支持。

此语句通过一个“or”运算和相匹配字符串的 topic 子树扩展 Concrete Topic Expressions。这种扩展语句允许在单独 Subscription 中的 TopicSet 属性选择。这种语法以和 Topic Expressions 相同的方式描述在[WS-Topics 1.3] 标准中：

[3] TopicExpression ::= TopicPath ('|' TopicPath)*

[4] TopicPath ::= RootTopic ChildTopicExpression* ("/./)?

[5] RootTopic ::= QName

如果一个 Namespace 前缀被包含在 RootTopic，那么它将需要符合一个有效 Topic

Namespace 的定义, 并且 Local Name 也需要与前面 Namespace 中定义的 Root Topic 的 Name 一致

[6] ChildTopicExpression ::= '/' ChildTopicName

[7] ChildTopicName ::= QName | NCName

。

NCName 或 QName 的局部部分应在继承于 RootTopic 的层次目录中的 Topic 的 Name 一致, 这些地方上, 每个深度减少预示着此目录中的子 Topic 元素的另一级别。

为了引用 TopicExpression Dialect, 下面的 URI 需要被引用:

Dialect=<http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet>

如果 TopicExpression 以 “//.” 字符结束, 则预示着 TopicExpression 匹配一个 Topic Sub-Tree 。 例如: “ tns1:RuleEngine/FieldDetector//.” 。 这预示着 Sub-Tree 由 tns1:RuleEngine/FieldDetector 和其派生组成。

下面的例子展示了 ConcreteSet topicExpression 的使用:

查找拥有同源 Topic 的 VideoAnalytics Topic 的 Notifications :

```
<wsnt:TopicExpression                               Dialect                               =  
http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet" >  
    tns1:VideoAnalytics//.  
</wsnt:TopicExpression>
```

查找拥有同源 Topic 的 VideoAnalytics topic 或 RuleEngine 的 Notifications:

```
<wsnt:TopicExpression Dialect =  
    "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet" >  
    tns1:VideoAnalytics//. | tns1:RuleEngine//.  
</wsnt:TopicExpression>
```

查找由 LineDetector 或 FieldDetector 产生的 Notifications:

```
<wsnt:TopicExpression Dialect =  
    "http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet">  
    tns1:RuleEngine/FieldDetector//. | tns1:RuleEngine/LineDetector//.  
</wsnt:TopicExpression>
```

9.8 Get event properties

[WS-BaseNotification]标准定义了可选 WS-ResourceProperties 的设置。标准中没要求必须实现 WS-ResourceProperty Interface。取而代之，随后的直接 Interface 将被符合 Onvif 的设备实现，以提供关于 FilterDialects、Schema files 和设备支持的 topics 的 Information（Information 译：信息）。

Table 97: GetEventProperties command

GetEventProperties		Access Class: READ_MEDIA
Message name	Description	
GetEventPropertiesRequest	<i>This is an empty message.</i>	
GetEventPropertiesResponse	xs:anyURI TopicNamespaceLocation [1][unbounded] xs:boolean FixedTopicSet [1][1] wstop:TopicSetType TopicSet [1][1] xs:anyURI TopicExpressionDialect [1][unbounded] xs:anyURI MessageContentFilterDialect [1][unbounded] xs:anyURI ProducerPropertiesFilterDialect [0][unbounded] xs:anyURI MessageContentSchemaLocation [1][unbounded]	
Fault codes	Description	
	<i>No command specific faults!</i>	

符合 ONVIF 的设备应该回应并告知它的 TopicSet 是否为固定，提供哪种 Topics 和哪种 Dialects 被支持。

下面的 TopicExpressionDialects 对于符合 ONVIF 设备是强制要求的（参考 Section 9.7.3）：

<http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete>

<http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet>

下面的 MessageContentFilterDialects 对于符合 ONVIF 设备是强制要求（参考 Section 9.5.5）：

<http://www.onvif.org/ver10/tev/messageContentFilter/ItemFilter>

标准没有要求设备支持 ProducerPropertiesDialect。

Message Content Description Language（Message 内容描述语言），其在介绍于 Section 9.5.4，允许拥有私有特定类型。为了简化提供给 Client 应用这些 types，

GetEventPropertiesResponse 应该罗列出用于描述 Notification 的 Types 来源的 Schema Files 的所有 URI, 其以 MessageContentSchemaLocation 元素给出。这个罗列中至少要包含 ONVIF schema File 的 URI。

9.9 Capabilities

Capabilities 反应了可选的功能和服务器的能力。信息是固定的, 不可设备运行期间更改。

下面的 Capabilities 是有效的:

WSSubscriptionPolicySupport: Indication if the device supports the WS Subscription policy according to Section 9.1.2 (表示设备是否支持符合 Section 9.1.2 的 WS Subscription 方法)

WSPullPointSupport: Indication if the device supports the WS Pull Point according to Section 9.1.2 (表示设备是否支持符合 Section 9.1.2 的 WS Pull Point 方法)

WSPausableSubscriptionManagerInterfaceSupport: Indication if the device supports WS Pausable Subscription (表示如果设备支持 WS Pausable Subscription, 其是否支持 Indication)

Table 98: GetServiceCapabilities command

GetServiceCapabilities		Access Class: PRE_AUTH
Message name	Description	
GetServiceCapabilitiesRequest	This is an empty message.	
GetServiceCapabilitiesResponse	The capability response message contains the requested service capabilities using a hierarchical XML capability structure. tev:Capabilities Capabilities [1][1]	
Fault codes	Description	
	No command specific faults!	

9.10 SOAP Fault Messages

当 Client 或 Subscription Manager 在处理[WS-BaseNotification] 信息时, 如果设备遇到错误, 其将生成 SOAP 1.2 的错误信息。

所有 SOAP 1.2 错误信息将根据[WS-BaseNotification]和[WS-Topics]标准生成。

9.11 Notification example

下面的例子是 Notification 的完整通信方式。它用 Real-time Pull-Point Notification Interface 去接收 Notifications。

9.11.1 GetEventPropertiesRequest

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

<SOAP-ENV:Envelope

  xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"

  xmlns:wsa="http://www.w3.org/2005/08/addressing"

  xmlns:tet="http://www.onvif.org/ver10/events/wsdl">

  <SOAP-ENV:Header>

    <wsa:Action>

      http://www.onvif.org/ver10/events/wsdl/EventPortType/GetEventPropertiesRequest

    </wsa:Action>

  </SOAP-ENV:Header>

  <SOAP-ENV:Body>

    <tet:GetEventProperties>

    </tet:GetEventProperties>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

9.11.2 GetEventPropertiesResponse

在这个示例中，设备回应使用 ONVIF Topic Namespace（其描述能在 <http://www.onvif.org/onvif/ver10/topics/topicsns.xml> 下载到）

Topic Set 不能改变结束时间和单个 Topic tns1:RuleEngine/LineDetector/Crossed 的成员。带有此 Topic 交互的 Message 包含含有 VideoSourceConfigurationToken、VideoAnalyticsConfigurationToken 和在这条线上的个体的 Interface。

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

<SOAP-ENV:Envelope

xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"

xmlns:wsa="http://www.w3.org/2005/08/addressing"

xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"

xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"

xmlns:tet="http://www.onvif.org/ver10/events/wsd1"

xmlns:tns1="http://www.onvif.org/ver10/topics"

xmlns:tt="http://www.onvif.org/ver10/schema">

<SOAP-ENV:Header>

<wsa:Action>

http://www.onvif.org/ver10/events/wsd1/EventPortType/GetEventPropertiesResponse

</wsa:Action>

</SOAP-ENV:Header>

<SOAP-ENV:Body>

<tet:GetEventPropertiesResponse>

<tet:TopicNamespaceLocation>

http://www.onvif.org/onvif/ver10/topics/topicns.xml

</tet:TopicNamespaceLocation>

<wsnt:FixedTopicSet>

true

</wsnt:FixedTopicSet>

<wstop:TopicSet>

<tns1:RuleEngine>

<tns1:LineDetector>

<tns1:Crossed wstop:topic="true">

<tt:MessageDescription>

<tt:Source>

<tt:SimpleItemDescription Name="VideoSourceConfigurationToken"

Type="tt:ReferenceToken"/>

<tt:SimpleItemDescription

```

Name="VideoAnalyticsConfigurationToken"
                                Type="tt:ReferenceToken"/>
    </tt:Source>
    <tt:Data>
        <tt:SimpleItemDescription Name="ObjectId"
                                Type="xs:integer"/>
    </tt:Data>
</tt:MessageDescription>
</tns1:Crossed>
</tns1:LineDetector>
</tns1:RuleEngine>
</wstop:TopicSet>
<wsnt:TopicExpressionDialect>
    http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet
</wsnt:TopicExpressionDialect>
<wsnt:TopicExpressionDialect>
    http://docs.oasis-open.org/wsnt/t-1/TopicExpression/ConcreteSet
</wsnt:TopicExpressionDialect>
<wsnt:MessageContentFilterDialect>
    http://www.onvif.org/ver10/tev/messageContentFilter/ItemFilter
</wsnt:MessageContentFilterDialect>
<tt:MessageContentSchemaLocation>
    http://www.onvif.org/onvif/ver10/schema/onvif.xsd
</tt:MessageContentSchemaLocation>
</tet:GetEventPropertiesResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

9.11.3 CreatePullPointSubscription

Client 能订阅带有来自 TopicProperties 的信息的特定 Notification。下面的 XML 示例展

示了由设备 Rule Engine 产生的 Notification 的订阅。Client 只关心关于 VideoAnalyticsConfiguration “2” 和 VideoSourceConfiguration “1” 的 Notification。

Subscription 有一个一分钟的结束时间。如果订阅不明确地更新或 Message 不定期轮训，那它在这个时间后自动结束。

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  xmlns:tet="http://www.onvif.org/ver10/events/wsdl"
  xmlns:tns1="http://www.onvif.org/ver10/topics">
  <SOAP-ENV:Header>
    <wsa:Action>
      http://www.onvif.org/ver10/events/wsdl/EventPortType/CreatePullPointSubscriptionReq
      uest
    </wsa:Action>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <tet:CreatePullPointSubscription>
      <tet:Filter>
        <wsnt:TopicExpression
          Dialect="http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet">
          tns1:RuleEngine/.
        </wsnt:TopicExpression>
        <wsnt:MessageContent
          Dialect="http://www.onvif.org/ver10/tev/messageContentFilter/ItemFilter">
          boolean(/tt:SimpleItem[@Name="VideoAnalyticsConfigurationToken"
            and @Value="2"] ) and
          boolean(/tt:SimpleItem[@Name="VideoSourceConfigurationToken"
            and @Value="1"] )
        </wsnt:MessageContent>
      </tet:Filter>
    </tet:CreatePullPointSubscription>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

        </wsnt:MessageContent>

    </tet:Filter>

    <tet:InitialTerminationTime>

        PT1M

    </tet:InitialTerminationTime>

</tet:CreatePullPointSubscription>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

```

9.11.4 CreatePullPointSubscriptionResponse

当设备接收到订阅，它将返回 `http://160.10.64.10/Subscription?Idx=0` URI，其代表了 Subscription 的 Endpoint。除此之外，Client 也被告知了设备的当前时间和创建的 Subscription 的结束时间。

```

<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope

    xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"

    xmlns:wsa="http://www.w3.org/2005/08/addressing"

    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"

    xmlns:tet="http://www.onvif.org/ver10/events/wsdl">

    <SOAP-ENV:Header>

        <wsa:Action>

            http://www.onvif.org/ver10/events/wsdl/EventPortType/CreatePullPointSubscription

            Response

        </wsa:Action>

    </SOAP-ENV:Header>

    <SOAP-ENV:Body>

        <tet:CreatePullPointSubscriptionResponse>

            <tet:SubscriptionReference>

                <wsa:Address>

                    http://160.10.64.10/Subscription?Idx=0

```

```

        </wsa:Address>
    </tet:SubscriptionReference>

    <wsnt:CurrentTime>
        2008-10-09T13:52:59
    </wsnt:CurrentTime>

    <wsnt:TerminationTime>
        2008-10-09T13:53:59
    </wsnt:TerminationTime>
</tet:CreatePullPointSubscriptionResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

9.11.5 PullMessagesRequest

Client 发送 PullMessagesRequest 请求到在 CreatePullPointSubscriptionResponse 给出的 Endpoint，以获取符合 Client 订阅的 Notifications。下面的示例请求包括一个 5 秒的过期时间和在回应中 Message 数目上限为 2 的限制。

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://www.w3.org/2005/08/addressing"
    xmlns:tet="http://www.onvif.org/ver10/events/wsd1" >
    <SOAP-ENV:Header>
        <wsa:Action>
            http://www.onvif.org/ver10/events/wsd1/PullPointSubscription/PullMessagesRequest
        </wsa:Action>
        <wsa:To>http://160.10.64.10/Subscription?Idx=0</wsa:To>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <tet:PullMessages>
            <tet:Timeout>

```

```

PT5S

</tet:Timeout>

<tet:MessageLimit>

2

</tet:MessageLimit>

</tet:PullMessages>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

```

9.11.6 PullMessagesResponse

下面的 PullMessageResponse 包含符合订阅的两个 Notifications。此回应告诉 Client，有符合 Rules “MyImportantFence1” and “MyImportantFence2” 两个物体已经在线上。

```

<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  xmlns:tet="http://www.onvif.org/ver10/events/wsdl"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">

  <SOAP-ENV:Header>

    <wsa:Action>

      http://www.onvif.org/ver10/events/wsdl/PullPointSubscription/PullMessagesResponse

    </wsa:Action>

  </SOAP-ENV:Header>

  <SOAP-ENV:Body>

    <tet:PullMessagesResponse>

      <tet:CurrentTime>

        2008-10-10T12:24:58

```



```

</tet:CurrentTime>

<tet:TerminationTime>

    2008-10-10T12:25:58

</tet:TerminationTime>

<wsnt:NotificationMessage>

<wsnt:Topic
Dialect="http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet">

    tns1:RuleEngine/LineDetector/Crossed

</wsnt:Topic>

<wsnt:Message>

    <tt:Message UtcTime="2008-10-10T12:24:57.321Z">

        <tt:Source>

            <tt:SimpleItem Name="VideoSourceConfigurationToken"

                Value="1"/>

            <tt:SimpleItem Name="VideoAnalyticsConfigurationToken"

                Value="2"/>

            <tt:SimpleItem Value="MyImportantFence1" Name="Rule"/>

        </tt:Source>

        <tt:Data>

            <tt:SimpleItem Name="ObjectId" Value="15" />

        </tt:Data>

    </tt:Message>

</wsnt:Message>

</wsnt:NotificationMessage>

<wsnt:NotificationMessage>

    <wsnt:Topic
Dialect="http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet">

        tns1:RuleEngine/LineDetector/Crossed

    </wsnt:Topic>

    <wsnt:Message>

```

```

<tt:Message UtcTime="2008-10-10T12:24:57.789Z">
  <tt:Source>
    <tt:SimpleItem Name="VideoSourceConfigurationToken"
      Value="1"/>
    <tt:SimpleItem Name="VideoAnalyticsConfigurationToken"
      Value="2"/>
    <tt:SimpleItem Value="MyImportantFence2" Name="Rule"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItem Name="ObjectId" Value="19"/>
  </tt:Data>
</tt:Message>
</wsnt:Message>
</wsnt:NotificationMessage>
</tet:PullMessagesResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

9.11.7 UnsubscribeRequest

Client 必须明确的用 UnsubscribeRequest 结束其订阅，以便设备能立即释放资源。请求直接发送到在 CreatePullPointSubscriptionResponse 中返回的 Subscription Endpoint。

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2" >
  <SOAP-ENV:Header>
    <wsa:Action>
      http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeRequest
    </wsa:Action>

```

```

        <wsa:To>http://160.10.64.10/Subscription?Idx=0</wsa:To>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <wsnt:Unsubscribe/>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

9.11.8 UnsubscribeResponse

一旦设备应答了一个 UnsubscribeResponse, 那么 Subscription Endpoint 将不再有效。

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://www.w3.org/2005/08/addressing"
    xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2" >
    <SOAP-ENV:Header>
        <wsa:Action>
            http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeResponse
        </wsa:Action>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <wsnt:UnsubscribeResponse/>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

9.12 Service specific fault codes

Event 服务器不定义[WS-BaseNotification]定义之外的任何服务性特定错误。

