
QCG Notification v. 3.0

Technical Guide and Manual

Bartosz Bosak <bbosak@man.poznan.pl>
Krzysztof Kurowski <krzysztof.kurowski@man.poznan.pl>

3.0.3

Copyright © 2008-2013 Poznan Supercomputing and Networking Center

Table of Contents

Introduction	1
Motivations	2
Purpose of the Document	2
Status	2
Terminology and WS-Notification Concept	2
WS-Notification Concept	2
UML-style Diagrams	8
User Guide for a Publisher, Subscriber and NotificationConsumer	12
Prefixes and Namespaces	12
Interaction with <i>QCG Notification</i>	12
Topics, TopicExpressions and Dialects	16
Extensions	16
Methods Description	23
Administration Guide of <i>QCG Notification</i>	50
Prerequisites	50
Building	50
Creating database for <i>QCG Notification</i> - the PostgreSQL example	51
Configuration of <i>QCG Notification</i>	52
Configuration File: <i>qcg-ntfd.xml</i>	52
Module configuration	57
WS-Topics management	60
Starting <i>QCG Notification</i>	61
Stopping <i>QCG Notification</i>	61

Introduction

QCG Notification is a reference implementation of a set of industry Web Service specifications (WS-Notification OASIS standard) [http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn] for message push/pull-style notification, using a topic-based publish/subscribe pattern. It supports standard message exchanges to be implemented by any Web Service providers (including also services within QCG Service Bus), that wish to participate in notifications, standard message exchanges for a notification-broker service provider (allowing publication of messages from entities that are not themselves service providers), operational requirements expected of service providers and requesters that participate in notifications, and an XML model that describes topics (i.e., items of interest for subscriptions.).

Motivations

Reliable notification and messaging services are key components in service-oriented architectures (SOA) and distributed computing. However, early Web Services, such as those for getting stock or whether daily reports, were seemingly all remote procedure calls (RPC) oriented. In RPC systems, the receiver typically performs the requested Web Service and sends a response back to the consumer over the same connection on which the request arrived. Communication in this case is fully synchronous, and often also called a pull-style message exchange. Today, many existing SOA environments are based on synchronous message exchange, but in the future we expect more and more needs for asynchronous push Web Services. Moreover, existing approaches in a relatively simple and often non standard based way provide notification capabilities for distributed Web Services. Available reference implementations of industry standards, e.g. defined by OASIS standardization body, either implement only partially notification standards or do not meet high performance requirements. Therefore, our main motivation was to design and implement *QCG Notification* to offer more advanced push/pull (synchronous/asynchronous) standard notification routines for QCG Service Bus services as well as any third party Web Services.

Purpose of the Document

The main goal of this document is to define a set of initial use cases, scenarios and specify software requirements for *QCG Notification*, its programming application interface, functional units and the integration with both internal and external modules.

The document is aimed at software architects and programmers who will be developing software components based on this document. Many requirements and *QCG Notification* concept itself might be interesting for developers and end users of any Web Services.

This document assumes that the reader has at least a basic background in Web Service and SOA technologies.

Status

This document is updated periodically on no particular schedule.

Terminology and WS-Notification Concept

The aim of this section is to capture high-level functionality of *QCG Notification* and/or interaction without providing excessive implementation or technical details of the system.

WS-Notification Concept

WS-Notification Fundamentals

Notification-based pattern is commonly used for interaction between distributed objects. WS-Notification is a family of related specifications that standardizes notification concept and topic-based information distribution in SOA and Web Service based systems. There are three normative OASIS specifications: WS-BaseNotification, WS-BrokeredNotification and WS-Topics.

WS-BaseNotification	Standardizes the terminology, concepts, operations, WSDL and XML needed to express the basic roles involved in Web services publish and subscribe for notification message exchange.
---------------------	--

Specification URL:

http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf
[http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-0.2.pdf]

WS-BrokeredNotification Describes specific objectives for expanded version of notification messages exchange with indirect NotificationBroker service.

Specification URL:

http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf [http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-pr-0.2.pdf]

WS-Topics Defines mechanism to organize and categorize items of interest for subscriptions known as "topics".

Specification URL:

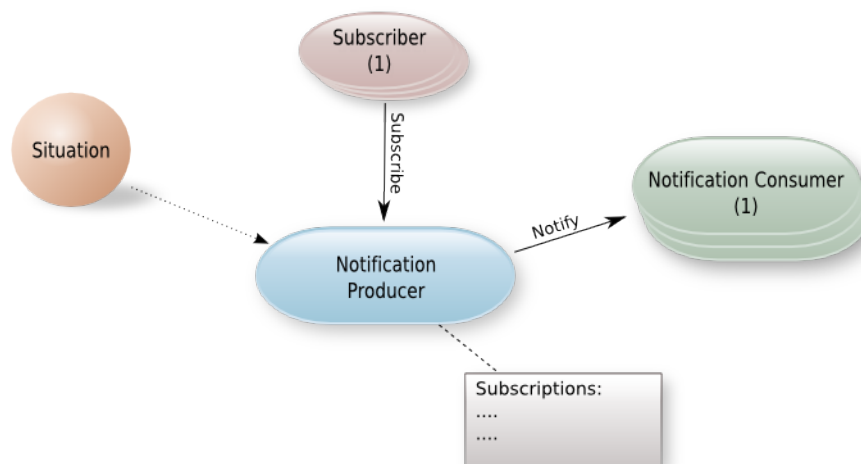
http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf [http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-pr-0.2.pdf]

The next subsections will present main notification concepts defined in the mentioned specifications.

WS-BaseNotification

The WS-BaseNotification document is a basic reference on which all the other specifications in the standard WS-Notification family depend. In fact, it defines precisely two important notification roles/actors called respectively NotificationProducer and NotificationConsumer. An example system architecture based on WS-BaseNotification is presented below.

Figure 1. Web Services BaseNotification entities



The basic terminology related to the WS-BaseNotification is following:

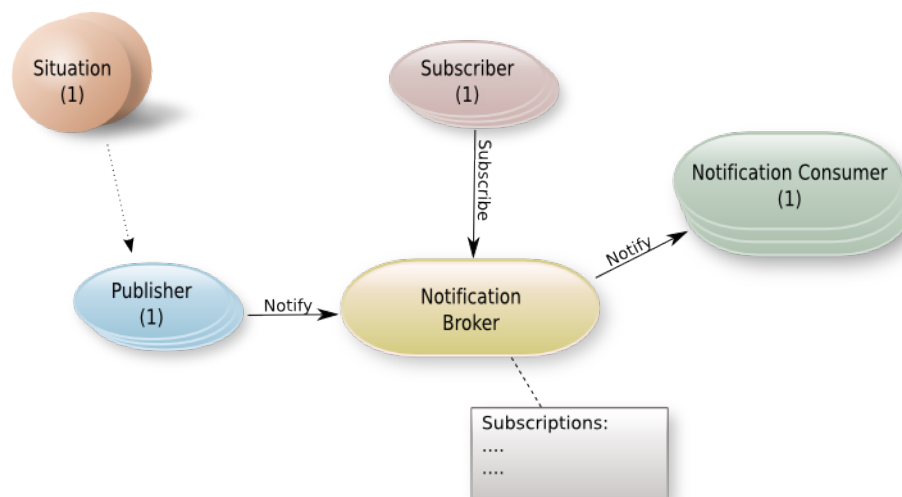
Situation	Some occurrence known by a NotificationProducer. A Situation is a reason of dispatching a Notification.
------------------	---

Notification	An artifact of a Situation containing information about that Situation. A Notification is represented as an XML element and dispatched by a NotificationProducer.
Subscription	Represents relations between a NotificationProducer and a NotificationConsumer. A Subscription is created by a Subscriber and is consisted with filtering parameters such as a Topic, policies or context information.
Subscriber	An unit that creates Subscriptions. It sends the SubscribeRequest message to a NotificationProducer service.
NotificationProducer	Produces Notifications. In other words NotificationProducer detects Situation(s) and translates it to Notification message(s). It is also a Web Service responsible for creating Subscriptions by the SubscribeRequest message.
NotificationConsumer	An endpoint designated to receive Notifications produced by a NotificationProducer as a result of a Subscription.
PullPoint	The purpose of PullPoints is enabling pull-style notifications. In general, PullPoints accumulate notifications and offer an interface for retrieving these notifications by remote consumers. Since a PullPoint implements a NotificationConsumer interface it can be selected in a Subscription as a receiver of notifications.
CurrentMessages	CurrentMessages is a buffer that keeps the latest messages produced on each topic. A new notification replaces the previous related to the same topic and currently buffered in CurrentMessages.

WS-BrokeredNotification

In a nutshell, the WS-BrokeredNotification specification is an extension of described above the WS-BaseNotification concept. However, it introduces a new important notification role called NotificationBroker and defines more advanced notification scenarios. The figure below shows an example scenario and notification message flow where NotificationBroker is present.

Figure 2. Web Services BrokeredNotification entities - QCG Notification entities



Note, that there are some important enlargements and modifications in the WS-BrokeredNotification terminology. Below the most relevant definitions are presented.

Situation	Some occurrence, known by a Publisher. A Situation is a reason of dispatching a Notification.
Notification	An artifact of a Situation containing information about that Situation. A Notification is represented as an XML element, dispatched by a Publisher and redistributed by a NotificationBroker.
Publisher	An unit that creates Notification(s) - detects Situation(s) and translates it to Notification message(s). It doesn't have to be a Web Service.
Subscription	Represents relations between a Publisher and a NotificationConsumer. A Subscription is created by a Subscriber and is consisted with filtering parameters such as a Topic, policies or a context information.
Subscriber	An unit that creates Subscriptions. It sends the SubscribeRequest message to a SubscriptionManager interface.
NotificationBroker	An intermediary Web service that decouples Publishers and NotificationConsumers. It redistributes Notifications received from Publishers to NotificationConsumers on the basis of Subscriptions' parameters.
NotificationConsumer	An endpoint designated to receive Notifications produced by a Publisher and later redistributed by a NotificationBroker as a result of a Subscription.
PullPoint	The purpose of PullPoints is enabling pull-style notifications. In general, PullPoints accumulate notifications and offer an interface for retrieving these notifications by remote consumers. Since a PullPoint implements a NotificationConsumer interface it can be selected in a Subscription as a receiver of notifications.
CurrentMessages	CurrentMessages is a buffer that keeps the latest messages published on each topic. A new notification replaces the previous related to the same topic and currently buffered in CurrentMessages.

The main difference between WS-Base and WS-Brokered Notification specifications is that in the second case NotificationProducers do not have to be implemented as a Web Services. Publishers do not have to implement and be compliant with defined NotificationProducer interfaces. Finally, having a NotificationBroker as a mediator between different actors in a notification flow, NotificationConsumers are fully separated and do not have to know details about NotificationProducers.

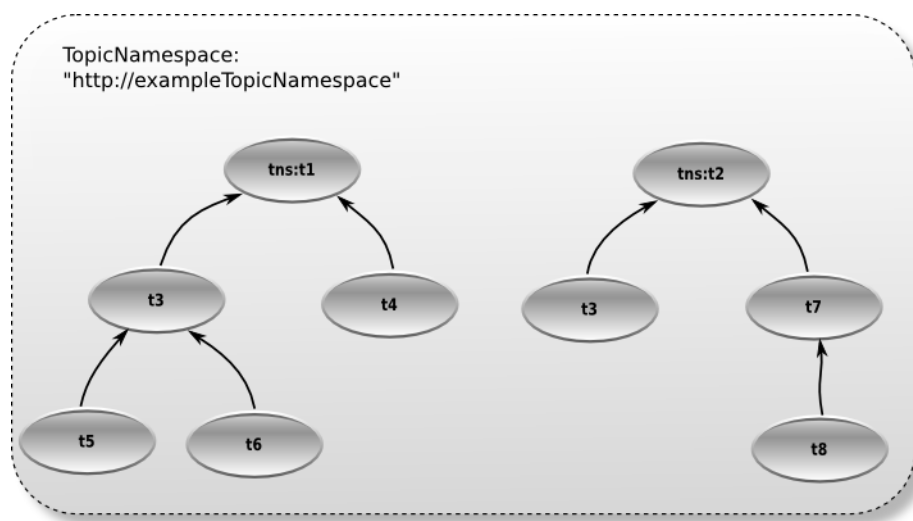
Depending on distributed system requirements and use cases various notification routines, objects, and roles are usually involved. For instance, in the simplest case where an asynchronous communication is required, there is no need to use a notification mechanisms defined by WS-BrokeredNotification and basic notification routines defined by WS-BaseNotification are good enough to satisfy distributed system needs. The example scenario is presented on Figure 1. For more advanced notification scenarios in service oriented systems we recommend to use notification capabilities defined in WS-BrokeredNotification. One of the main scenario defined by WS-BrokeredNotification is presented on Figure 2 .

WS-Topics

The third important notification specification document called WS-Topics is strictly associated with WS-BaseNotification and precisely defines a "topics" mechanism. Important definitions used in the WS-Topics specification are presented below:

Topic	A concept used to categorize Notifications and their related Notification schemes. Topics are used to determine which subscribing NotificationConsumers should receive a specific Notification.
Topic Tree	A hierarchical group of Topics.
Topic Namespace	A forest of Topic Trees grouped into one namespace
TopicExpression	An expression that identifies Topics from a Topic Namespace
Dialect	Identifies the type of a grammar used in a TopicExpression. The specification defines four Dialects: <i>Simple</i> , <i>Concrete</i> , <i>Full</i> and <i>XPath</i> .

Figure 3. Example Topic Namespace



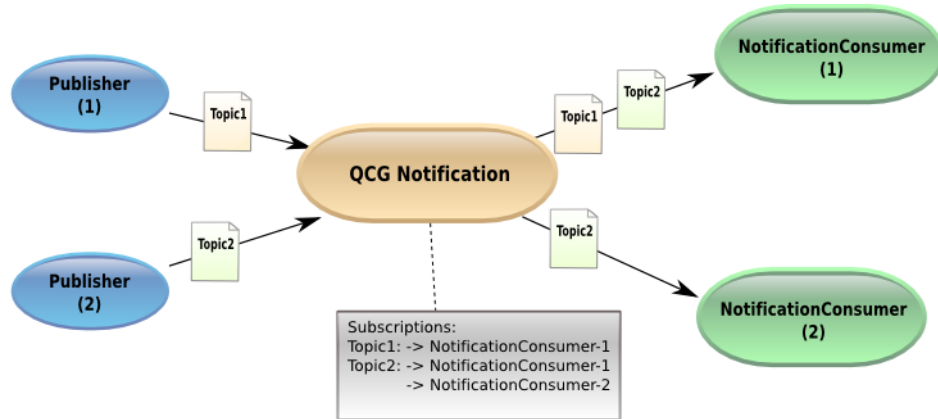
Base of QCG Notification

QCG Notification is a reference implementation of WS-BaseNotification, WS-BrokeredNotification and WS-Topics OASIS industry standards. As a consistent and highly efficient notification system it supports advanced brokered Web Service notification push/pull scenarios described in the Section 2.1.3. It has been designed to satisfy notification requirements and demands of large scale Web Service based systems, SOA environments and it acts as a core component in QCG Service Bus. To meet security requirements, *QCG Notification* (or *QCG Notification Provider*) exposes well defined Web Service and programming interfaces over QCG security mechanisms for dynamic notification, subscription, registration, forwarding, etc. As it was explained in the previous sections it can be used and integrated with various external third party Web Services playing roles of a notification Publisher, Subscriber or Consumer.

A flexible plug-in architecture of *QCG Notification* enables using different communication protocols for messages transmission. Currently messages can be transported between *QCG Notification* and third party entities not only through HTTP protocol, but also through XMPP (jabber) as well as SMTP (e-mail) protocols. It is possible to variously mix channels of communication, for example a notification sent by a Publisher being WS-Client as a HTTP message can be distributed over XMPP to jabber consumers.

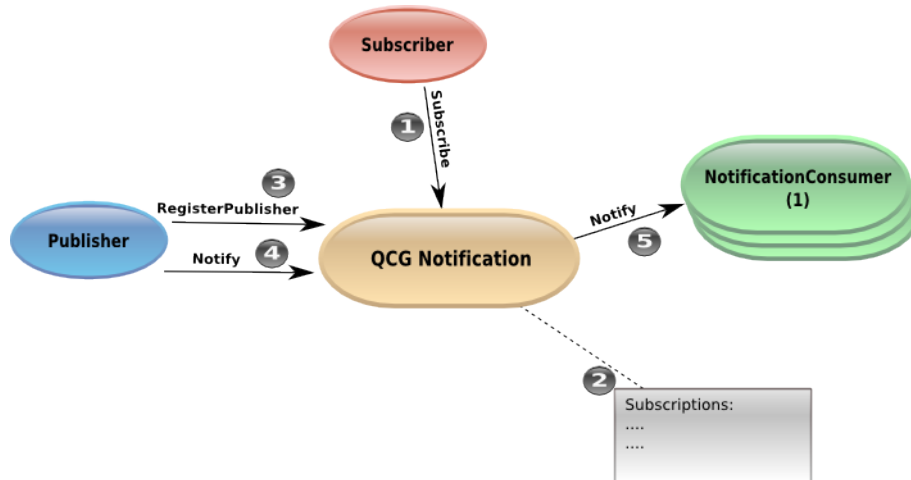
Below we present a set of Web Service notification use cases demonstrating capabilities offered by *QCG Notification*.

Figure 4. Notification with multiple Publishers and multiple NotificationConsumers



The diagram 4 shows one of the basic notification flows supported by *QCG Notification*. Depending on records, Notifications produced by a Publisher and received by the *QCG Notification* may or may not be dispatched to concrete NotificationConsumer. Note, that Subscriptions represent relationships between Topics and NotificationConsumers in this scenario. Before any Notification is delivered to a NotificationConsumer, a Subscriber has to send first a SubscribeRequest message to *QCG Notification* in which all parameters of a Subscription are specified. Main parameters of a SubscribeRequest message are Consumer-Reference and TopicExpression defined according to the specific Dialect type. On the other hand, before a Publisher starts sending Notifications it may be obligated to register himself in *QCG Notification*. The diagram below shows a sequence of events in order to initialize an asynchronous process of notifications between a Publisher and a NotificationConsumer.

Figure 5. Message flows: creating a Subscription, notifying NotificationConsumers



1. A Subscriber sends the SubscribeRequest message with a NotificationConsumer location and Topic-Expression parameters to *QCG Notification*.
2. *QCG Notification* prepares appropriate records for a Subscription.
3. A Publisher register himself in *QCG Notification* by sending the RegisterPublisher message.
4. A registered Publisher sends the Notify message on a specified Topic to a *QCG Notification* service.
5. *QCG Notification* checks if there are NotificationConsumers subscribed on a specified Topic. If so, *QCG Notification* sends asynchronous Notifications to these NotificationConsumers.

UML-style Diagrams

The document uses UML-style use-case diagrams to illustrate high-level use cases. The following list is probably sufficient for this version of the document:

- Stick figures represents actors or roles in a scenario. These can be human beings or software systems.
- Ellipses represent use cases, i.e. actions or units of functionality in a system.
- Lines between actors and use cases indicate a participation of the actor in the use case. At this level, no direction or payload of data flow is expressed by the lines between actors and use cases.

The document uses UML-style sequence diagrams to illustrate detailed use case scenarios and later in the next sections. Meanings of graphics in these diagrams are as follows:

- Boxes at the top of the diagram represent an actor in the scenario.
- Arrows with a solid head represent a message sent from one actor to another. The arrow points from sender to receiver.
- Arrows with a line head represent the return value of a message. The arrow points from the receiver of the earlier message to the sender.
- A dotted line running down the diagram from a box indicates that arrows whose endpoints (tail or head) is on the line apply to that actor.
- Intersections between arrows and dotted lines are meaningless.
- Vertical layout represents time. Messages (arrows) farther down on the page happen after messages higher on the page.
- Horizontal layout has no formal meaning. Since right-pointing arrows look better, actors that initiate a scenario tend to appear leftward of actors they send messages to.

Sequence diagrams are used in this document for more concrete design, and that actors and messages are often objects and object methods. They provide value for this document in that they give a clearly ordered message layout and connections among functional units. The actors and messages in the sequence diagrams are more properly roles in a scenario and actions associated with that scenario.

Main Use Cases

Figure 6. Use case diagram for a Publisher

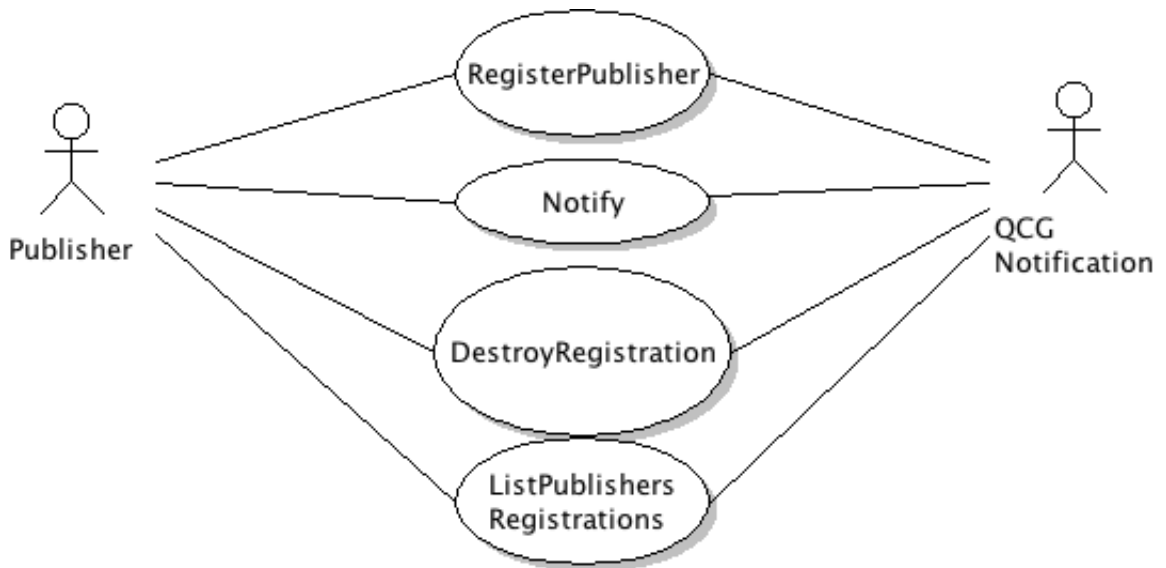


Figure 7. Use case diagram for a Subscriber

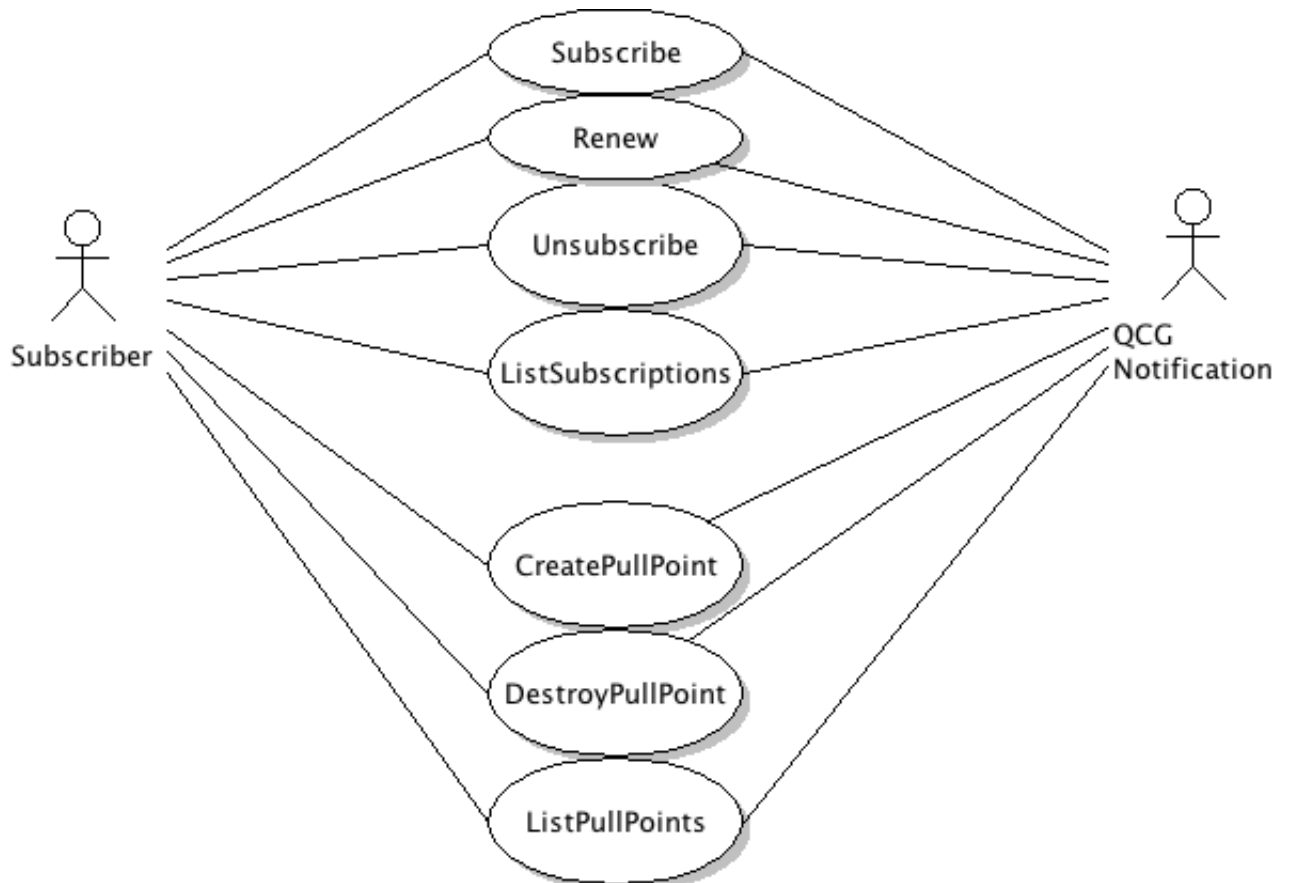


Figure 8. Use case diagram for a NotificationConsumer

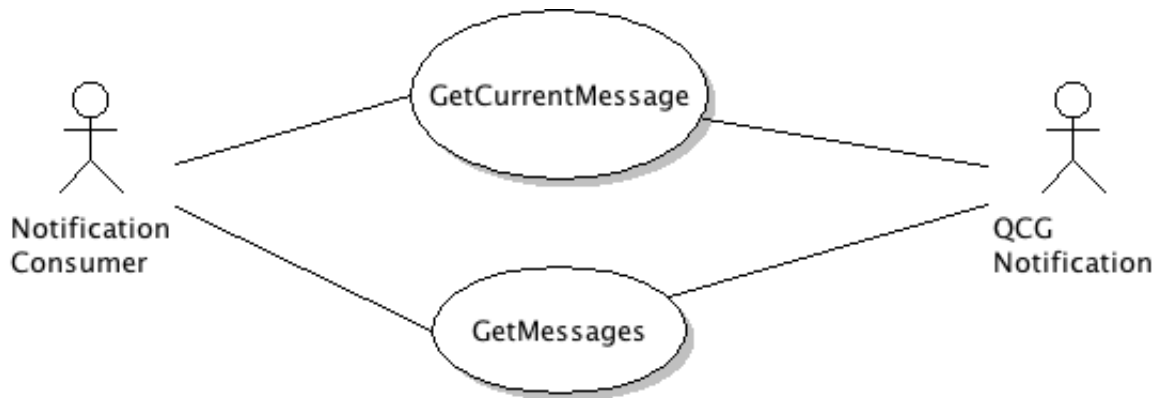
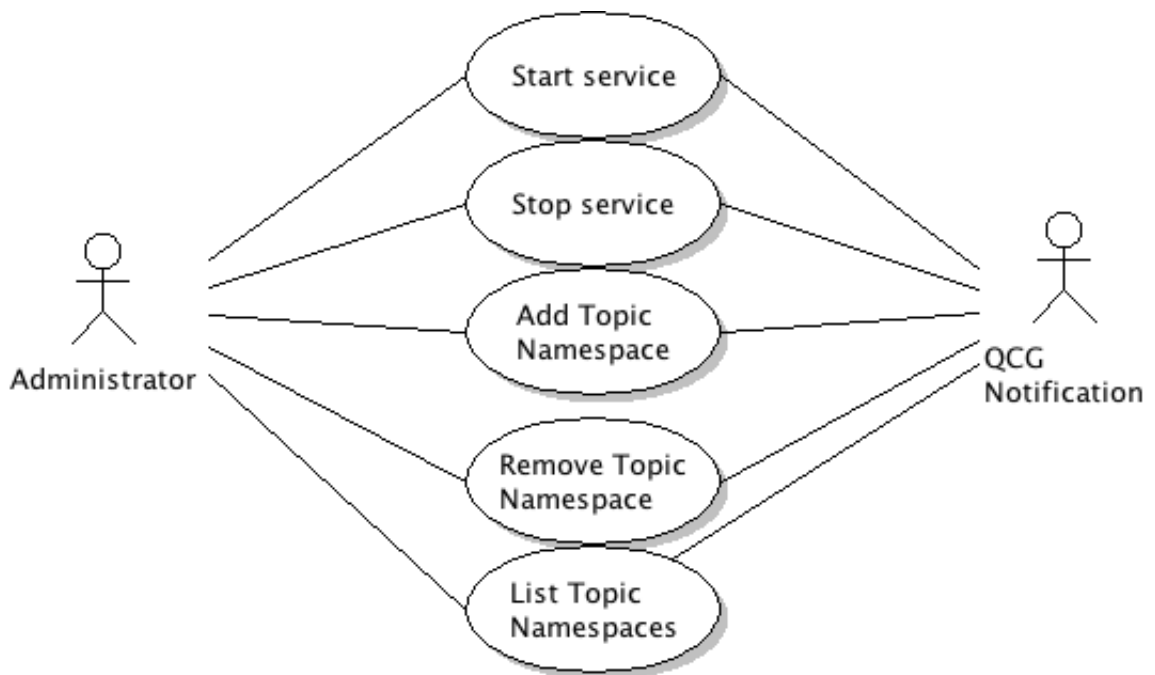


Figure 9. Use case diagram for QCG Notification

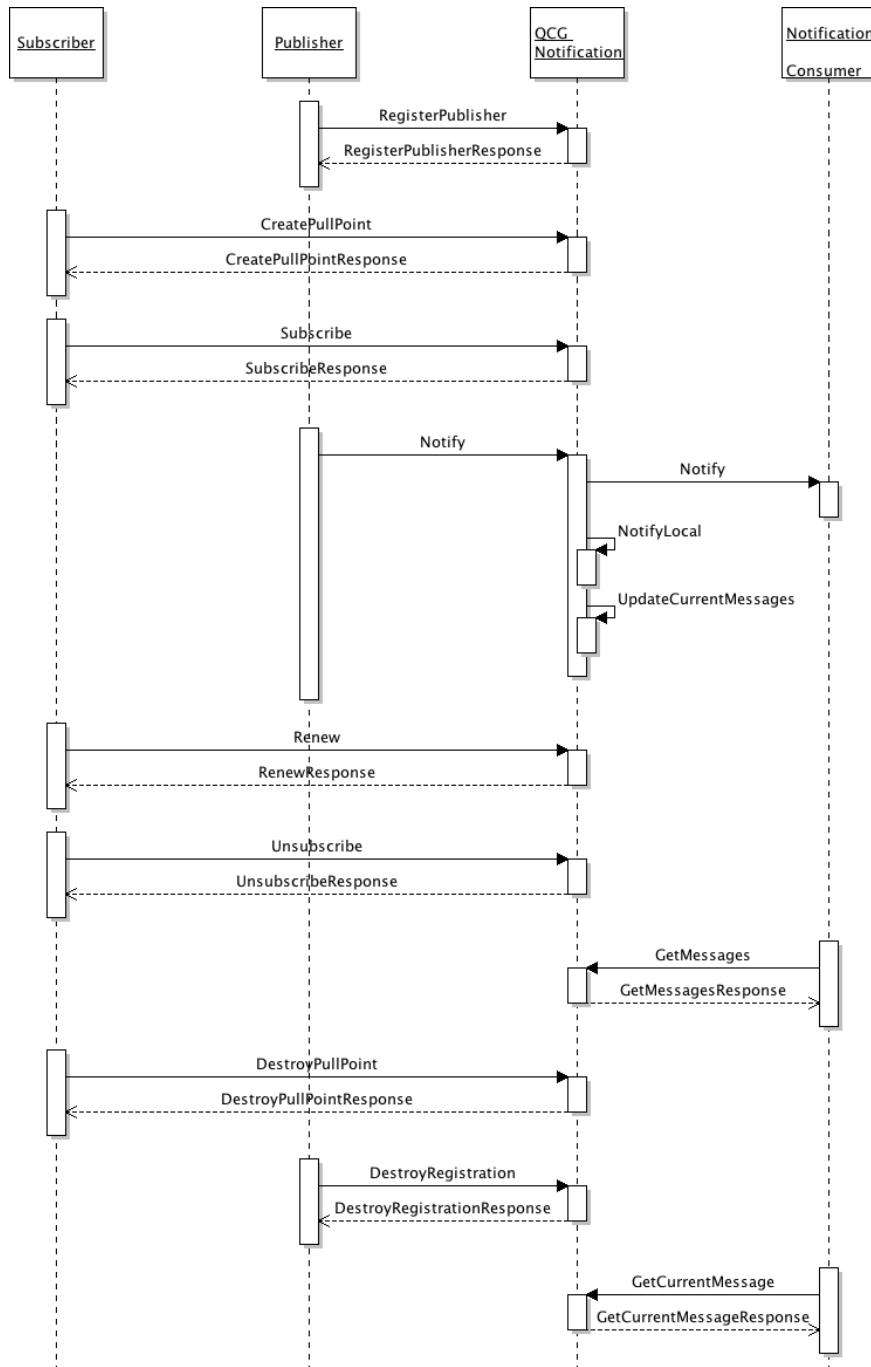


Figure 10. Use case diagram for a QCG Notification Administrator



Main Sequence Diagrams

Figure 11. General sequence diagram



User Guide for a Publisher, Subscriber and NotificationConsumer

Prefixes and Namespaces

The following table presents namespaces tied with prefixes. The prefixes are used in example SOAP messages presented in the further sections.

Table 1. Prefixes and Namespaces

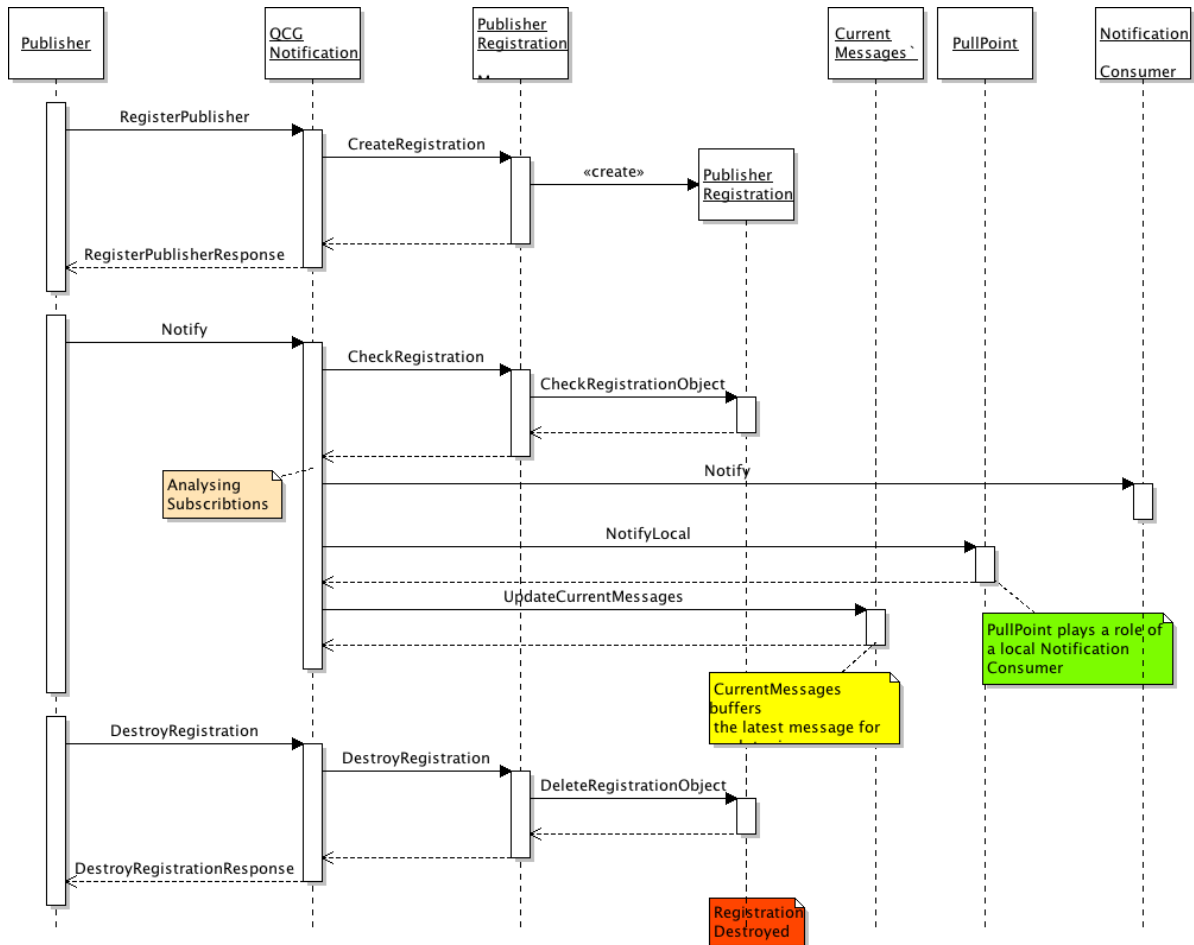
Prefix	Namespace
SOAP-ENV	http://schemas.xmlsoap.org/soap/envelope/
SOAP-ENC	http://schemas.xmlsoap.org/soap/encoding/
xsi	http://www.w3.org/2001/XMLSchema-instance
xsd	http://www.w3.org/2001/XMLSchema
wsa	http://www.w3.org/2005/08/addressing
wsbf	http://docs.oasis-open.org/wsrf/bf-2
wstop	http://docs.oasis-open.org/wsn/t-1
wsr	http://docs.oasis-open.org/wsrf/r-2
wsnt	http://docs.oasis-open.org/wsn/b-2
wsntw	http://docs.oasis-open.org/wsn/bw-2
wsnbr	http://docs.oasis-open.org/wsn/br-2
smn-ext	http://schemas.qoscosgrid.org/ntf/2011/04/extensions

Interaction with *QCG Notification*

In the next subsections an issue of cooperation with *QCG Notification* is presented independently for a Publisher, Subscriber and NotificationConsumer. Although a conceptual style of interaction is described here in details, technical particularities of constructing messages are presented later.

Publisher Interaction

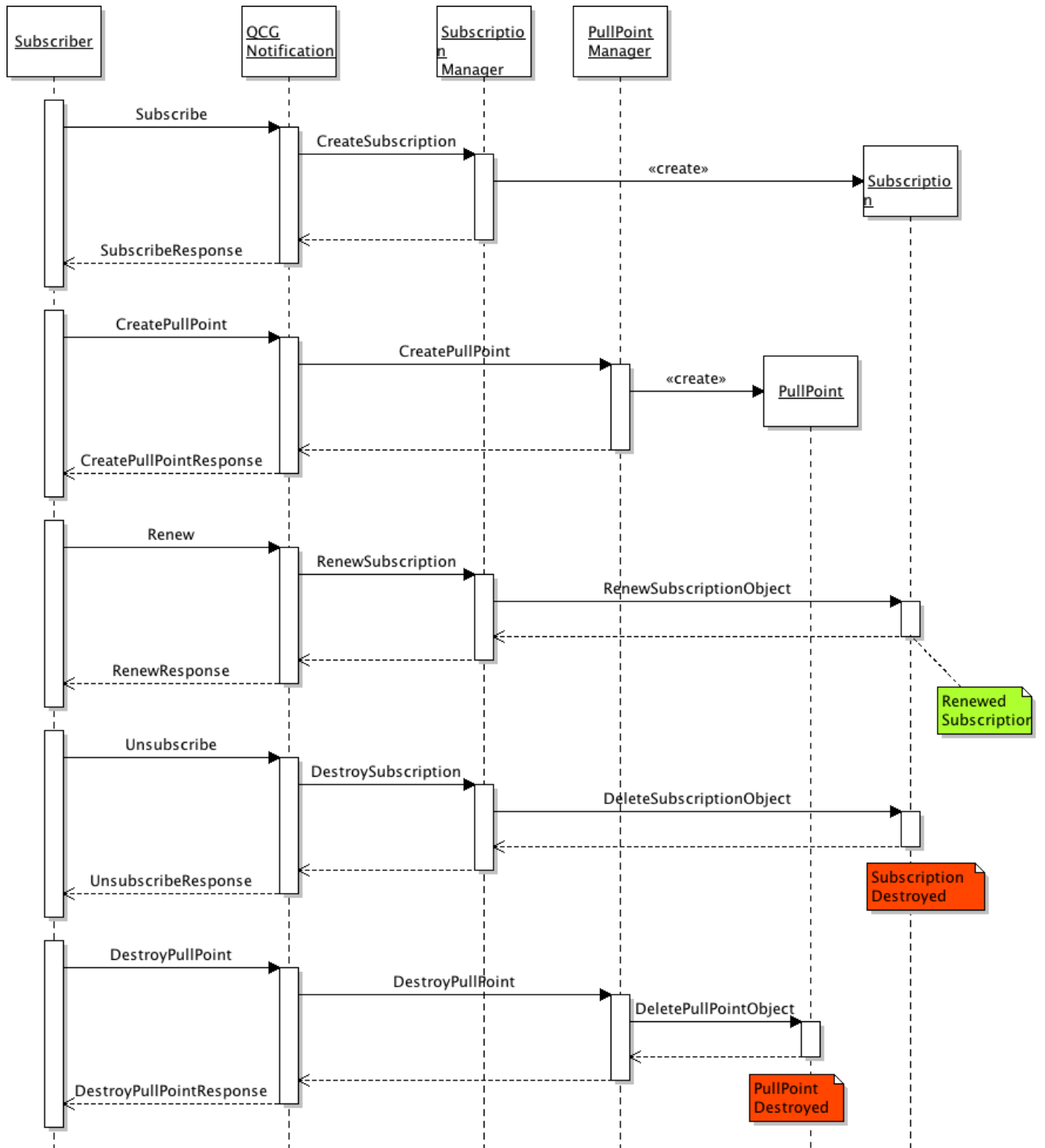
Figure 12. Publisher interaction sequence diagram



In the most general configuration of the service, a Publisher has to be registered before it starts sending Notifications. In this situation, a Publisher sends the **RegisterPublisherRequest** message to *QCG Notification*. If everything goes well it receives the **RegisterPublisherResponse** message with a session ID. From this time a Publisher is entitled for sending **Notify** messages - the **Notify** message has to include received session ID in its content. The notifications can be forwarded to remote Notification Consumers or buffered locally in PullPoints. Moreover, every new notification replaces the old one in **CurrentMessages**. After work, a Publisher should unregister himself by sending **DestroyRegistration**. This message has to include a session ID likewise.

Subscriber interaction

Figure 13. Subscriber interaction sequence diagram



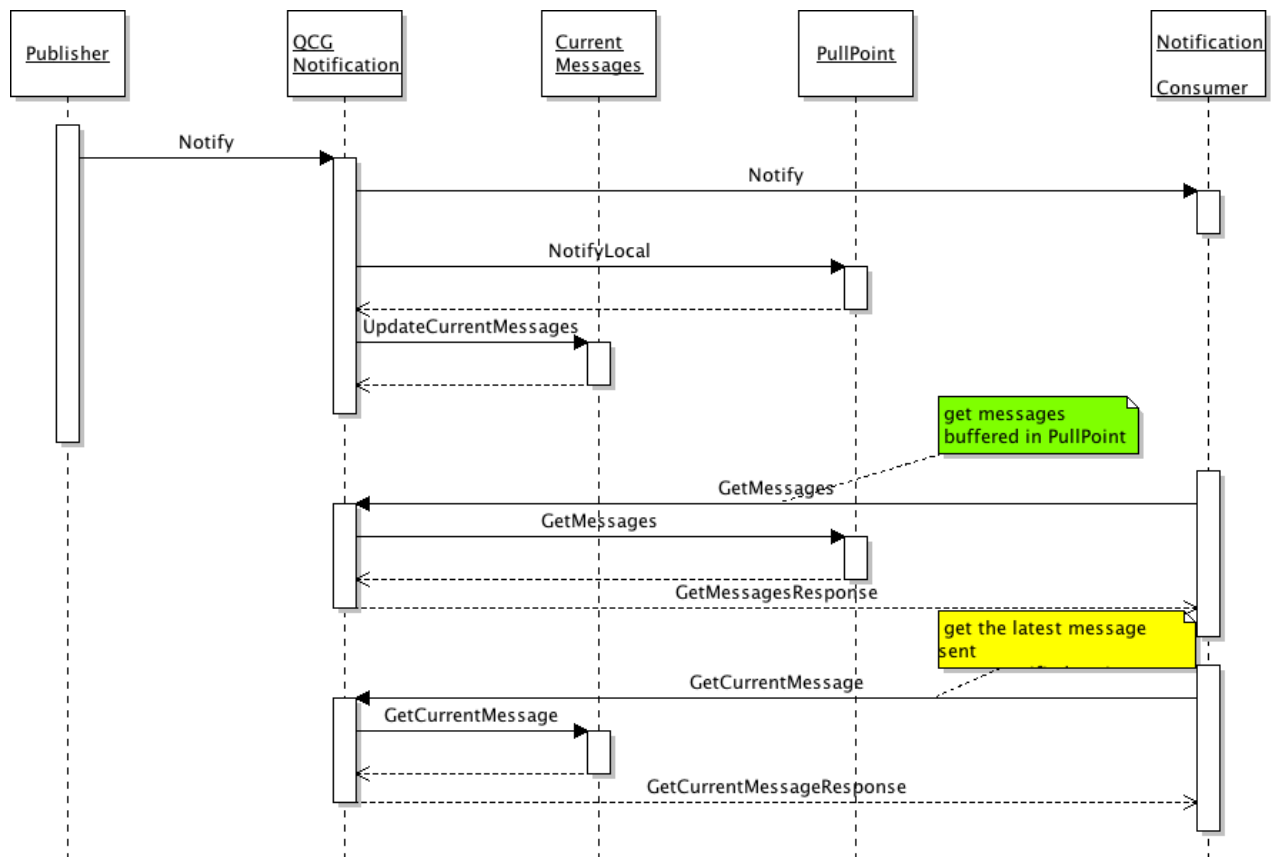
When a Subscriber wants to create a Subscription, it sends the SubscribeRequest message to *QCG Notification*. This message should contain an address of NotificationConsumer and a TopicExpression that specifies a Topic or a set of Topics on which NotificationConsumer should receive Notifications. Additionally a Subscriber can define a boolean XPath expression that will be evaluated over Notification Messages

for more sophisticated filtering. From the version 2.5.0 of the system, the Subscriber may specify XSLT document that will be used to transform message content by certain publishing modules. After creation of a Subscription *QCG Notification* sends the SubscribeResponse message with a created Subscription ID. This ID should be attached to the UnsubscribeRequest message when a Subscription is not wanted and the Renew message when a Subscription should be refreshed. A Subscriber is allowed to create many equal Subscriptions.

If a scenario requires pull-style of notifications, a Subscriber can create a set of PullPoints. In a such situation, it has to invoke the CreatePullPoint method offered by *QCG Notification*. The response parameter of this method will be an address to the newly created *PullPoint* resource (*Subscribers can assign this resources as NotificationConsumers in theirs Subscriptions*). A *PullPoint* resource can be destroyed by the DestroyPullPoint message sent directly to this resource.

NotificationConsumer Interaction

Figure 14. NotificationConsumer interaction sequence diagram



The main capability of a NotificationConsumer is a receiving of Notify messages. A NotificationConsumer can implement an interface for receiving Notify messages sending by *QCG Notification* or can be a client of a PullPoint. In the second case, a NotificationConsumer can invoke GetMessages method offered by a certain PullPoint. Messages buffered in the PullPoint will be removed from the buffer and delivered in GetMessagesResponse to the consumer.

On the other hand, when a NotificationConsumer wants to receive the last Notification message sent on a specific Topic, it can do this by sending GetCurrentMessageRequest to *QCG Notification*. Requested Notification will be supplied in the GetCurrentMessageResponse message.

Topics, TopicExpressions and Dialects

Before start-up of a *QCG Notification* service, there should be defined Topic Namespaces in a system. Topic Namespaces should be created (usually by Publishers) and delivered to a *QCG Notification* administrator in a form of XML documents. A following record presents a sample Topic Namespace identified by <http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics>.

```
<wstop:TopicNamespace
  xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"
  targetNamespace="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics"
  <wstop:Topic name="House">
    <wstop:Topic name="Kitchen">
      <wstop:Topic name="Friedge"/>
      <wstop:Topic name="Cook"/>
      <wstop:Topic name="Light"/>
    </wstop:Topic>
    <wstop:Topic name="Toilet">
      <wstop:Topic name="Light"/>
    </wstop:Topic>
    <wstop:Topic name="Room1">
      <wstop:Topic name="Light"/>
    </wstop:Topic>
    <wstop:Topic name="Room2">
      <wstop:Topic name="Light"/>
    </wstop:Topic>
  </wstop:Topic>
</wstop:TopicNamespace>
```

QCG Notification supports three types of dialects: *Simple*, *Concrete* and *Full*. If a message refer to Topics, it has to define TopicExpression that identifies one or more Topics (note there are situations when TopicExpression has to refer to exactly one Topic). TopicExpression has to be written in a supported Dialect. Further more TopicExpression has to define Topics from supported Topic Namespaces. To select an appropriate Topic Namespace, the first part of TopicExpression should be presented in a form of a prefix that refers to this namespace.

For more details see: http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf [http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-pr-0.2.pdf] and check out examples of methods in the section 3.4.

Extensions

The Oasis standard for notifications doesn't define details for lower Levels of functionality so an architect of system is obligated to introduce some extensions and clarifications. In the *QCG Notification* scenario there are several messages that have to be expanded with additional information written as embedded XML. Some of these extensions are related with other Oasis standards but some are quite new. Technical details are described in the successive subsections.

TopicExpression

QCG Notification remote peers have to use an appropriate TopicExpression XML element. The schema of this element is defined according to Oasis standard in <http://docs.oasis-open.org/wsn/b-2.xsd> [<http://docs.oasis-open.org/wsn/b-2.xsd>] [<http://docs.oasis-open.org/wsn/b-2.xsd>]. Please note here that the Notify and GetCurrentMessageRequest messages contain a Topic element which contains a required @Dialect attribute. Also TopicExpression has to define a @Dialect attribute. For conformity with the WSDL docu-

ment, a @Dialect from the Topic element is significant. It is not necessary but preferred to attach the same @Dialect in the TopicExpression element against.

QCG Notification Extensions

QCG Notification introduces some extensions to WSDL as well as schema documents related to the Oasis standard. The following schema defines these extensions. The first three elements named RegisterPublisherID, SubscriptionID and PullPointID are IDs for Publisher Registrations, created Subscriptions and PullPoints. CurrentMessage is used for supplementation of GetCurrentMessageResponse. The rest of the elements is utilized in the managing methods, the elements are used in order to: listing Subscriptions, listing PublisherRegistrations and listing PullPoints.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  xmlns="http://schemas.qoscosgrid.org/ntf/2011/04/extensions"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
  targetNamespace="http://schemas.qoscosgrid.org/ntf/2011/04/extensions"
  elementFormDefault="qualified">

  <xsd:import namespace="http://docs.oasis-open.org/wsn/b-2"
    schemaLocation="http://docs.oasis-open.org/wsn/b-2.xsd"/>

  <xsd:element name="RegisterPublisherID" type="ID"/>
  <xsd:element name="SubscriptionID" type="ID"/>
  <xsd:element name="PullPointID" type="ID"/>
  <xsd:element name="CurrentMessage" type="CM"/>
  <xsd:element name="XSLTTranslation" type="XSLTTrans"/>

  <xsd:simpleType name="ID">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="32"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="CM">
    <xsd:all>
      <xsd:element name="NotifyTime" type="xsd:dateTime" minOccurs="0"/>
      <xsd:element ref="wsnt:ProducerReference" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="Message" type="MSG" minOccurs="1"/>
    </xsd:all>
  </xsd:complexType>

  <xsd:complexType name="MSG">
    <xsd:sequence>
      <xsd:any minOccurs="0" maxOccurs="1" processContents="lax"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:simpleType name="XSLTTrans">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="4096"/>
    </xsd:restriction>
  </xsd:simpleType>
```

```
<!-- ===== QCG Notification manager : ListPullPoints =====>

<xsd:complexType name="ListSubscriptionsFilter">
  <xsd:sequence>
    <xsd:element name="SubscriptionID" type="ID" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ConsumerReference" type="wsa:EndpointReferenceType"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CreationTimeMin" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CreationTimeMax" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="TerminationTimeMin" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="TerminationTimeMax" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="wsnt:Filter" minOccurs="0" maxOccurs="1"/>
    <xsd:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="ListSubscriptions">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Filter" type="ListSubscriptionsFilter"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="MaxNumber" type="xsd:integer"
        minOccurs="0" maxOccurs="1"/>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="Subscription">
  <xsd:sequence>
    <xsd:element name="SubscriptionID" type="ID"/>
    <xsd:element name="ConsumerReference" type="wsa:EndpointReferenceType"
      minOccurs="0" maxOccurs="1" />
    <xsd:element name="CreationTime" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="TerminationTime" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="wsnt:Filter" minOccurs="0" maxOccurs="1"/>
    <xsd:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="ListSubscriptionsResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Subscription" type="Subscription"

```

```
        minOccurs="0" maxOccurs="unbounded"/>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!-- ===== QCG Notification manager : ListPublishersRegistrations =====>

<xsd:complexType name="ListPublishersRegistrationsFilter">
  <xsd:sequence>
    <xsd:element name="PublisherRegistrationID" type="ID"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PublisherReference" type="wsa:EndpointReferenceType"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CreationTimeMin" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CreationTimeMax" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="TerminationTimeMin" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="TerminationTimeMax" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="ListPublishersRegistrations">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Filter" type="ListPublishersRegistrationsFilter"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="MaxNumber" type="xsd:integer"
        minOccurs="0" maxOccurs="1"/>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="PublisherRegistration">
  <xsd:sequence>
    <xsd:element name="PublisherRegistrationID" type="ID"/>
    <xsd:element name="PublisherReference" type="wsa:EndpointReferenceType"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CreationTime" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="TerminationTime" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:element name="ListPublishersRegistrationsResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="PublisherRegistration" type="PublisherRegistration"
        minOccurs="0" maxOccurs="unbounded"/>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!-- ===== QCG Notification manager : ListPullPoints =====>

<xsd:complexType name="ListPullPointsFilter">
  <xsd:sequence>
    <xsd:element name="PullPointID" type="ID"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PullPointReference" type="wsa:EndpointReferenceType"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CreationTimeMin" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CreationTimeMax" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="TerminationTimeMin" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="TerminationTimeMax" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
    <xsd:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="ListPullPoints">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Filter" type="ListPullPointsFilter"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="MaxNumber" type="xsd:integer"
        minOccurs="0" maxOccurs="1"/>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="PullPoint">
  <xsd:sequence>
    <xsd:element name="PullPointID" type="ID"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PullPointReference" type="wsa:EndpointReferenceType"
      minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CreationTime" type="wsnt:AbsoluteOrRelativeTimeType"
      nillable="true" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:element name="TerminationTime" type="wsnt:AbsoluteOrRelativeTimeType"
  nillable="true" minOccurs="0" maxOccurs="1"/>
<xsd:any namespace="##other" processContents="lax"
  minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>

<xsd:element name="ListPullPointsResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="PullPoint" type="PullPoint"
        minOccurs="0" maxOccurs="unbounded"/>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!-- ===== QCG Notification manager : ListTopicNamespaces =====>

<xsd:complexType name="ListTopicNamespacesFilter">
  <xsd:sequence>
    <xsd:element name="targetNamespace" type="xsd:anyURI"
      minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="ListTopicNamespaces">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Filter" type="ListTopicNamespacesFilter"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="MaxNumber" type="xsd:integer"
        minOccurs="0" maxOccurs="1"/>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="TopicNamespace">
  <xsd:sequence>
    <xsd:element name="targetNamespace" type="xsd:anyURI" use="required"/>
    <xsd:element name="Topics" type="xsd:anyURI" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="ListTopicNamespacesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="TopicNamespace" type="TopicNamespace"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:any namespace="##other" processContents="lax"
  minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<!-- ===== QCG Notification manager : AddTopicNamespace =====>

<xsd:element name="AddTopicNamespace">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="wstop:TopicNamespace" />
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="AddTopicNamespaceResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!-- ===== QCG Notification manager : RemoveTopicNamespace =====>

<xsd:element name="RemoveTopicNamespace">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="targetNamespace" type="xsd:anyURI"
        use="required" />
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="RemoveTopicNamespaceResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

</xsd:schema>
```

Methods Description

The main idea of this chapter is to present the types of messages used in the *QCG Notification* scenario as well as introduce how to construct and interpret these messages. The majority of them is two-way style and initiated by *QCG Notification* clients. Exception is the Notify message that is one-way style. In the first phase it is transported from a Publisher to *QCG Notification* and in the second it is transported from *QCG Notification* to subscribed NotificationConsumers.

All supported by *QCG Notification* message types are presented and explained below.

Note

- The key information are thick;
- A base of the messages structure is a delivered WSDL file, however there are also extensions;
- For all messages it is important to set a correct WS-Addressing Action property which should be the same as SOAP Action.

SubscribeRequest

SubscribeRequest is sent by a Subscriber to *QCG Notification*. The message should contain information about a NotificationConsumer and an expression defining Topic or Topics that the Subscription will affect. If a message processing is successful, a Subscription is created. It is possible to create many equal Subscriptions by sending numerously the same SubscribeRequest message.

```
SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      NotificationProducer/SubscribeRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:Subscribe>
      <wsnt:ConsumerReference>
        <wsa:Address>
          https://localhost:2233
        </wsa:Address>
      </wsnt:ConsumerReference>
      <wsnt:Filter>
        <wsnt:TopicExpression
          xmlns:tns="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics"
          Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full">
            tns://*
          </wsnt:TopicExpression>
          <wsnt:MessageContent
            xmlns:smnex=http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousemessag
            Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
              boolean(//smnex:State[text()='on'])
            </wsnt:MessageContent>
          </wsnt:Filter>
          <wsnt:InitialTerminationTime>
            2007-07-23T10:24:14+02:00
          </wsnt:InitialTerminationTime>
        </wsnt:Filter>
      </wsnt:Subscribe>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

```
</wsnt:InitialTerminationTime>
<smn-ext:XSLTTranslation>
  <![CDATA
    <?xml version="1.0" encoding="ISO-8859-1"?>
    <xsl:stylesheet version="1.0"
      xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
      xmlns:s="http://schemas.qoscosgrid.org/ntf/1011/04/sample/myhousemessage">
      <xsl:output method="text"/>
      <xsl:template match="/">
        ### Sample QCG-Notification message ###
        <xsl:apply-templates/>
      </xsl:template>
      <xsl:template match="s:State">
        Value of State is: <xsl:value-of select="."/>
      </xsl:template>
    </xsl:stylesheet>
  ]]>
</smn-ext:XSLTTranslation>
</wsnt:Subscribe>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://docs.oasis-open.org/wsn/bw-2/ NotificationProducer/SubscribeRequest*. If differ, a fault will be returned.

- *//wsnt:Subscribe/wsnt:ConsumerReference/wsa:Address*

A Consumer address. It is significant to specify a correct prefix (http, https, xmpp). The prefix should be adequate to the expected type of connection between *QCG Notification* and NotificationConsumer(s): http for anonymous, https for ssl-based and xmpp for sending notifications to jabber clients.

- *//wsnt:Subscribe/wsnt:Filter*

In this element, a Subscriber defines filtering parameters to the Subscription being created. In other words, it identifies a type of Notifications that should be forwarded to the selected Notification Consumer.

- *//wsnt:Subscribe/wsnt:Filter/wsnt:TopicExpression*

This element defines Topic or Topics that Subscription should affect. Topics should come from supported Topic Namespaces that are also defined for the element. TopicExpression may be defined in two ways: (1) using prefixed form (e.g. tns:/*) - the topicNamespace is defined in a TopicExpression element and assigned to some prefix (e.g. tns) or (2) using inline form (e.g. {http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics}/*).

- *//wsnt:Subscribe/wsnt:Filter/wsnt:TopicExpression/@Dialect*

URI that specifies a TopicExpression Dialect type. Allowed Dialects types are as follows: *"http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple"*, *"http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete"*, *"http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full"*

- *//wsnt:Subscribe/wsnt:Filter/wsnt:MessageContent*

An optional element. If present, the MessageContent should define an XPath boolean expression that will be evaluated over redistributed Notification Messages to true or false. If

such an evaluation returns true, the message is forwarded, otherwise it is not. The namespaces used in the expression may be defined inside the MessageContent element as additional attributes or using in-line form e.g. `boolean(/{http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousemessage}State[text()='on'])`. The Dialect should always be: `http://www.w3.org/TR/1999/REC-xpath-19991116`

- `//wsnt:Subscribe/wsnt:InitialTerminationTime`

An optional element. If present, must contain a Subscriber suggestion for the termination time of the Subscription being created (expressed as a local time with timezone). If not present, the termination time is determined automatically - *QCG Notification* specify a default time period for the Subscription to be valid.

- `//wsnt:Subscribe/smn-ext:XSLTTranslation`

An optional element. Inside the XSLTTranslation a Subscriber may deliver an XSLT document that will be used by some publishing modules to transform notification messages to a different form (for example user readable). Note that an actual XSLT document should be enclosed in `<![CDATA` and `]]>` markers.

SubscribeResponse

In answer on SubscribeRequest there is sent the SubscribeResponse message. *QCG Notification* deliver in this message a reference to the created Subscription. This ID can be used for manipulation and to remove the Subscription.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      NotificationProducer/SubscribeResponse
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:SubscribeResponse>
      <wsnt:SubscriptionReference>
        <wsa:Address>
          http://some.address
        </wsa:Address>
        <wsa:ReferenceParameters>
          <SubscriptionID
            xmlns="http://schemas.qoscosgrid.org/ntf/2011/04/extensions">
              506435877
            </SubscriptionID>
          </wsa:ReferenceParameters>
        </wsnt:SubscriptionReference>
      </wsnt:SubscribeResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

- `//SOAP-ENV:Header/wsa:Action`

For this message should always be: `http://docs.oasis-open.org/wsn/bw-2/ NotificationProducer/SubscribeResponse`. If differ, a fault will be returned.

- `//wsnt:SubscribeResponse/wsnt:SubscriptionReference/wsa:Address`

An address of a SubscriptionManager interface. In the current solution it is the *QCG Notification* service address.

- *//wsnt:SubscribeResponse/wsnt:SubscriptionReference/*

wsa:ReferenceParameters/SubscriptionID

An extensive element in xsd:any that contains Subscription ID. At this time it is at most 32 char long string.

RenewRequest

When a Subscriber wants to renew a Subscription, it can send the RenewRequest message to *QCG Notification*. The message should contain the Subscription ID element and a new termination time suggestion.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      SubscriptionManager/RenewRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:Renew>
      <wsnt:TerminationTime>
        2007-07-25T12:24:14+02:00
      </wsnt:TerminationTime>
      <smn-ext:SubscriptionID>
        506435877
      </smn-ext:SubscriptionID>
    </wsnt:Renew>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://docs.oasis-open.org/wsn/bw-2/ SubscriptionManager/RenewRequest*. If differ, a fault will be returned.

- *//wsnt:Renew/wsnt:TerminationTime*

A Subscriber suggestion for the new termination time of the Subscription being renewed (expressed as a local time with timezone).

- *//wsnt:Renew/smn-ext:SubscriptionID*

A Subscription ID that was received in the SubscribeResponse message.

RenewResponse

An answer to the RenewRequest message is RenewResponse. The message contains a termination time for the renewed Subscription.

```
<SOAP-ENV:Envelope ... >
```

```
<SOAP-ENV:Header>
  <wsa:Action SOAP-ENV:mustUnderstand="1">
    http://docs.oasis-open.org/wsn/bw-2/
    SubscriptionManager/RenewResponse
  </wsa:Action>
  ...
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <wsnt:RenewResponse>
    <wsnt:TerminationTime>
      2007-07-25T10:24:14Z
    </wsnt:TerminationTime>
    <wsnt:CurrentTime>
      2007-07-24T10:24:14Z
    </wsnt:CurrentTime>
  </wsnt:RenewResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://docs.oasis-open.org/wsn/bw-2/ SubscriptionManager/RenewResponse*. If differ, a fault will be returned.

- *//wsnt:RenewResponse/wsnt:TerminationTime*

A new termination time for the renewed Subscription (expressed as UTC Time).

- *//wsnt:RenewResponse/wsnt:CurrentTime*

A *QCG Notification's* time when the Subscription was renewing (expressed as UTC Time).

UnsubscribeRequest

When a Subscriber wants to terminate a Subscription, it has to send the UnsubscribeRequest message to *QCG Notification*. The parameter attached to this request is a Subscription ID.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      SubscriptionManager/UnsubscribeRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:Unsubscribe>
      <smn-ext:SubscriptionID>
        506435877
      </smn-ext:SubscriptionID>
    </wsnt:Unsubscribe>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeRequest>. If differ, a fault will be returned.

- *//wsnt:Unsubscribe/smn-ext:SubscriptionID*

A Subscription ID that was received in the SubscribeResponse message.

UnsubscribeResponse

UnsubscribeResponse is returned to a Subscriber after successfully processing the UnsubscribeRequest message.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      SubscriptionManager/UnsubscribeResponse
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:UnsubscribeResponse>
    </wsnt:UnsubscribeResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: <http://docs.oasis-open.org/wsn/bw-2/SubscriptionManager/UnsubscribeResponse>. If differ, a fault will be returned.

RegisterPublisherRequest

In order to sending Notifications, a Publisher can be obligated to register himself previously in *QCG Notification*. For this purpose, it sends the RegisterPublisherRequest message to *QCG Notification*.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      RegisterPublisher/RegisterPublisherRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnbr:RegisterPublisher>
      <wsnbr:Demand>
        false
      </wsnbr:Demand>
      <wsnbr:InitialTerminationTime>
        2007-07-23T10:24:14+02:00
      </wsnbr:InitialTerminationTime>
    </wsnbr:RegisterPublisher>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: <http://docs.oasis-open.org/wsn/brw-2/RegisterPublisher/RegisterPublisherRequest>. If differ, a fault will be returned.

- *//wsnbr:RegisterPublisher/wsnbr:Demand*

Determine a demand-based pattern. At this time this pattern isn't supported so default *false* is the sole accepted value.

- *//wsnbr:RegisterPublisher/wsnbr:InitialTerminationTime*

An optional element. If present, must contain the requester suggestion for the termination time of the Registration being created (expressed as a local time with timezone). If not present, the termination time is determined automatically - *QCG Notification* specify the default time period for the Registration to be valid. Every Notify message renews the termination time.

RegisterPublisherResponse

In a response to a successful registration process, *QCG Notification* sends RegisterPublisherResponse. This message contains a reference to the created Registration. The attached ID is necessary to send any Notification message.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/brw-2/
      RegisterPublisher/RegisterPublisherResponse
    </wsa:Action>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnbr:RegisterPublisherResponse>
      <wsnbr:PublisherRegistrationReference>
        <wsa:Address>
          http://some.address
        </wsa:Address>
        <wsa:ReferenceParameters>
          <RegisterPublisherID
            xmlns="http://schemas.qoscosgrid.org/ntf/2011/04/extensions">
              1207382668
            </RegisterPublisherID>
          </wsa:ReferenceParameters>
        </wsnbr:PublisherRegistrationReference>
      </wsnbr:RegisterPublisherResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: <http://docs.oasis-open.org/wsn/brw-2/RegisterPublisher/RegisterPublisherResponse>. If differ, a fault will be returned.

- *//wsnbr:RegisterPublisherResponse/wsnbr:PublisherRegistrationReference/wsa:Address*

An address of a PublisherRegistration Manager interface. In the current solution it is the *QCG Notification* service address.

- *//wsn:RegisterPublisherResponse/wsn:PublisherRegistrationReference/wsa:ReferenceParameters/RegisterPublisherID*

An extensive element in xsd:any that contains a Publisher Registration ID. It defines a session that is created after registration and removed by sending the DestroyRegistration message. At this time it is at most 32 char long string.

Notify

Notify consists of one or many NotificationMessages. Any of them is a separate Notification message and is processed independently. NotificationMessage should define a single Topic it concerns. TopicExpression should be written in the same Dialect that the Subscription was created in. The Publisher Registration ID has to be attached to Notify messages.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify
    </wsa:Action>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:Notify>
      <wsnt:NotificationMessage>
        <wsnt:Topic
          Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full"
          xmlns:tns="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics">
            tns:House/Kitchen/Light
          </wsnt:Topic>
        <wsnt:ProducerReference>
          <wsa:Address>
            http://some.address
          </wsa:Address>
        </wsnt:ProducerReference>
        <wsnt:Message>
          <smnex:State
            xmlns:smnex="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhouseemes
on
          </smnex:State>
        </wsnt:Message>
      </wsnt:NotificationMessage>
      <smn-ext:RegisterPublisherID>
        1207382668
      </smn-ext:RegisterPublisherID>
    </wsnt:Notify>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://docs.oasis-open.org/wsn/bw-2/NotificationConsumer/Notify*. If differ, a fault will be returned.

- *//wsnt:Notify*

A set of Notification messages.

- *//wsnt:Notify/wsnt:NotificationMessage*

A single Notification message.

- *//wsnt:Notify/wsnt:NotificationMessage/wsnt:Topic*

The content of this element is an expression that defines exactly one Topic in a specified Dialect type. The Topic should be supported. The Topic may be defined in two ways: (1) using prefixed form (e.g. `tns:House/Kitchen/Light`) - the `topicNamespace` is defined in a `Topic` element and assigned to some prefix (e.g. `tns`) or (2) using inline form (e.g. `{http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics}House/Kitchen/Light`).

- *//wsnt:Notify/wsnt:NotificationMessage/wsnt:Topic/@Dialect*

URI that specifies TopicExpression Dialect type. Allowed Dialect types are as follows: "`http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple`", "`http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete`", "`http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full`".

- *//wsnt:Notify/wsnt:NotificationMessage/wsnt:ProducerReference/wsa:Address*

A Notification producer (Publisher) address.

- *//wsnt:Notify/wsnt:NotificationMessage/wsnt:Message*

A Notify message content. Currently supported are messages shorter than 4096 bytes.

- *//wsnt:Notify/smn-ext:RegisterPublisherID*

A Publisher Registration ID that was received in the RegisterPublisherResponse message.

DestroyRegistrationRequest

A Publisher can terminate a Registration by sending the DestroyRegistrationRequest message to *QCG Notification*. The parameter attached to this request is a Publisher Registration ID.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/brw-2/
      PublisherRegistrationManager/DestroyRegistrationRequest
    </wsa:Action>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnbr:DestroyRegistration>
      <smn-ext:RegisterPublisherID>
        387837036
      </smn-ext:RegisterPublisherID>
    </wsnbr:DestroyRegistration>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: <http://docs.oasis-open.org/wsn/brw-2/PublisherRegistrationManager/DestroyRegistrationRequest>. If differ, a fault will be returned.

- `//wsn:br:DestroyRegistration/smn-ext:RegisterPublisherID`

A Publisher Registration ID that was received in a RegisterPublisherResponse message.

DestroyRegistrationResponse

DestroyRegistrationResponse is returned to a Publisher after a successful process of destroying a Publisher Registration, in a response to the DestroyRegistration message.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/brw-2/
      PublisherRegistrationManager/DestroyRegistrationResponse
    </wsa:Action>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsn-br:DestroyRegistrationResponse>
    </wsn-br:DestroyRegistrationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- `//SOAP-ENV:Header/wsa:Action`

For this message should always be: <http://docs.oasis-open.org/wsn/brw-2/PublisherRegistrationManager/DestroyRegistrationResponse>. If differ, a fault will be returned.

CreatePullPointRequest

In scenarios where the pull-style of notifications delivery is needed, Subscribers can create PullPoints. To do this, a Subscriber sends the CreatePullPointRequest message to *QCG Notification PullPoint*.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      CreatePullPoint/CreatePullPointRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:CreatePullPoint>
    </wsnt:CreatePullPoint>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- `//SOAP-ENV:Header/wsa:Action`

For this message should always be: <http://docs.oasis-open.org/wsn/bw-2/CreatePullPoint/CreatePullPointRequest>. If differ, a fault will be returned.

CreatePullPointResponse

The CreatePullPointResponse message is returned in a response to the request of PullPoint creation. This message contains a reference to the just created PullPoint.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      CreatePullPoint/CreatePullPointResponse
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:CreatePullPointResponse>
      <wsa:Address xmlns:wsa="http://www.w3.org/2005/08/addressing">
        http://some.address:2211?ppID=1100992448
      </wsa:Address>
    </wsnt:CreatePullPointResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://docs.oasis-open.org/wsn/bw-2/CreatePullPoint/CreatePullPointResponse*. If differ, a fault will be returned.

- *//wsnt:CreatePullPointResponse/wsa:Address*

An logical address of the created PullPoint.

GetMessages

A Pull-style NotificationConsumer can get messages buffered in a PullPoint by invoking the GetMessages method. It sends the GetMessagesRequest message directly to the specified PullPoint. The message can contain a specification of a maximal number of Notification messages that NotificationConsumer wants to receive in the response.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      PullPoint/GetMessagesRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:GetMessages>
      <wsnt:MaximumNumber>10</wsnt:MaximumNumber>
    </wsnt:GetMessages>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: <http://docs.oasis-open.org/wsn/bw-2/PullPoint/GetMessagesRequest>. If differ, a fault will be returned.

- `//wsnt:GetMessages/wsnt:MaximumNumber`

An optional element. It defines a maximal number of messages that will be gotten from the PullPoint.

GetMessagesResponse

GetMessagesResponse, a response message to GetMessages, consists of a set of Notifications received from the PullPoint.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      PullPoint/GetMessagesResponse
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:GetMessagesResponse>
      <wsnt:NotificationMessage>
        <wsnt:Topic>
          Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full">
            xmlns:tns="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics">
              tns:House/Kitchen/Light
          </wsnt:Topic>
          <wsnt:ProducerReference>
            <wsa:Address>
              http://some.address
            </wsa:Address>
          </wsnt:ProducerReference>
          <wsnt:Message>
            <smnex:State
              xmlns:smnex="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousemes
              on
            </smnex:State>
          </wsnt:Message>
        </wsnt:NotificationMessage>
        <wsnt:NotificationMessage>
          ...
        </wsnt:NotificationMessage>
        ...
      </wsnt:GetMessagesResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

- `//SOAP-ENV:Header/wsa:Action`

For this message should always be: <http://docs.oasis-open.org/wsn/bw-2/PullPoint/GetMessagesResponse>. If differ, a fault will be returned.

- `//wsnt:GetMessagesResponse`

A set of Notification messages. The number of messages is restricted by the wsnt:MaximumNumber parameter defined in the GetMessagesRequest message.

- *//wsnt:GetMessagesResponse/wsnt:NotificationMessage*

A single Notification message.

- *//wsnt:GetMessagesResponse/wsnt:NotificationMessage/wsnt:Topic*

The content of this element is an expression that defines exactly one Topic in a specified Dialect type. The Topic should be supported.

- *//wsnt:GetMessagesResponse/wsnt:NotificationMessage/wsnt:Topic/@Dialect*

URI that specifies a TopicExpression Dialect type. Allowed Dialect types are as follows: "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple", "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete", "http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full".

- *//wsnt:GetMessagesResponse/wsnt:NotificationMessage/wsnt:ProducerReference/wsa:Address*

A Notification producer (Publisher) address.

- *//wsnt:GetMessagesResponse/wsnt:NotificationMessage/wsnt:Message*

A Notify message content. Currently supported are messages shorter than 4096 bytes.

DestroyPullPointRequest

A PullPoint can be destroyed by invoking a DestroyPullPoint method. An empty request message should be sent directly to the PullPoint.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      PullPoint/DestroyPullPointRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:DestroyPullPoint>
    </wsnt:DestroyPullPoint>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://docs.oasis-open.org/wsn/bw-2/PullPoint/DestroyPullPointRequest*. If differ, a fault will be returned.

DestroyPullPointResponse

A response to a DestroyPullPointRequest message is an empty DestroyPullPointResponse message.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
```

```
<wsa:Action SOAP-ENV:mustUnderstand="1">
  http://docs.oasis-open.org/wsn/bw-2/
  PullPoint/DestroyPullPointResponse
</wsa:Action>
...
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <wsnt:DestroyPullPointResponse>
  </wsnt:DestroyPullPointResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://docs.oasis-open.org/wsn/bw-2/PullPoint/DestroyPullPointResponse*. If differ, a fault will be returned.

GetCurrentMessageRequest

GetCurrentMessageRequest is useful especially when a new NotificationConsumer is connecting and want to get the last published Notification Message on a specified Topic. Single GetCurrentMessage should refer exactly one Topic and has to be expressed in a supported Dialect type.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      NotificationProducer/GetCurrentMessageRequest
    </wsa:Action>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:GetCurrentMessage>
      <wsnt:Topic
        Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full">
          xmlns:tns="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics"
          tns:House/Kitchen/Light
        </wsnt:Topic>
      </wsnt:GetCurrentMessage>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://docs.oasis-open.org/wsn/bw-2/ NotificationProducer/GetCurrentMessageRequest*. If differ, a fault will be returned.

- *//wsnt:GetCurrentMessage/wsnt:Topic*

The content of this element is an expression that defines exactly one Topic in a specified Dialect type. The topic should be supported. The Topic may be defined in two ways: (1) using prefixed form (e.g. *tns:House/Kitchen/Light*) - the topicNamespace is defined in a Topic element and assigned to some prefix (e.g. *tns*) or (2) using inline form (e.g. *{http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics}House/Kitchen/Light*).

- *//wsnt:GetCurrentMessage/wsnt:Topic/@Dialect*

URI that specifies TopicExpression Dialect's type. Allowed dialect's types are as follows: "<http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple>", "<http://docs.oasis-open.org/wsn/t-1/TopicExpression/Concrete>", "<http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full>".

GetCurrentMessageResponse

In a response to GetCurrentMessageRequest *QCG Notification* prepares GetCurrentMessageResponse. This message includes the content of the last Notification GetCurrentMessageRequest refers to.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      NotificationProducer/GetCurrentMessageResponse
    </wsa:Action>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <wsnt:GetCurrentMessageResponse>
      <smn-ext:CurrentMessage>
        <smn-ext:NotifyTime>
          2007-07-23T10:43:14Z
        </smn-ext:NotifyTime>
        <wsnt:ProducerReference>
          <wsa:Address>
            http://some.address
          </wsa:Address>
        </wsnt:ProducerReference>
        <smn-ext:Message>
          <smnex:State
            xmlns:smnex="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousemeson
          </smnex:State>
        </smn-ext:Message>
      </smn-ext:CurrentMessage>
    </wsnt:GetCurrentMessageResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: [http://docs.oasis-open.org/wsn/bw-2/ NotificationProducer/GetCurrentMessageResponse](http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/GetCurrentMessageResponse). If differ, a fault will be returned.

- *//wsnt:GetCurrentMessageResponse/smn-ext:GetCurrentMessage*

The element that contains current Notification and information related to this notification.

- *//wsnt:GetCurrentMessageResponse/smn-ext:GetCurrentMessage/smn-ext:NotifyTime*

The time when notification was received by *QCG Notification*; expressed as universal time (UTC).

- *//wsnt:GetCurrentMessageResponse/wsnt:ProducerReference/wsa:Address*

A Notification producer (Publisher) address.

- *//wsnt:GetCurrentMessageResponse/smn-ext:GetCurrentMessage/smn-ext:Message*

An extensive element. The content of this element is the last Notification message published to the Topic specified in *GetCurrentMessageRequest*. Currently supported are messages shorter than 4096 bytes.

ListSubscriptions

The first method from the the group of administrative methods. It may be invoked to get an information about available subscriptions in the *QCG Notification* service. The method consists of various filtering parameters.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://schemas.qoscosgrid.org/ntf/2011/04/
      QCGNotificationManager/ListSubscriptionsRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <smn-ext:ListSubscriptions>
      <smn-ext:Filter>
        <smn-ext:SubscriptionID>
          1568334
        </smn-ext:SubscriptionID>
        <smn-ext:ConsumerReference>
          <wsa:Address>
            http://localhost:2233
          </wsa:Address>
        </smn-ext:ConsumerReference>
        <smn-ext:CreationTimeMin>
          2010-07-09T08:21:58Z
        </smn-ext:CreationTimeMin>
        <smn-ext:CreationTimeMax>
          2010-07-09T10:21:58Z
        </smn-ext:CreationTimeMax>
        <smn-ext:TerminationTimeMin>
          2010-07-09T08:21:58Z
        </smn-ext:TerminationTimeMin>
        <smn-ext:TerminationTimeMax>
          2010-07-09T10:21:58Z
        </smn-ext:TerminationTimeMax>
      </smn-ext:Filter>
      <wsnt:Filter>
        <wsnt:MessageContent>
          xmlns:smnex="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousemessa
          Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
          boolean(//smnex:State[text()='on'])
        </wsnt:MessageContent>
        <wsnt:TopicExpression>
          xmlns:tns="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics"
          Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full">
          tns://*
        </wsnt:TopicExpression>
      </wsnt:Filter>
```

```
</smn-ext:Filter>
</smn-ext:ListSubscriptions>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/ListSubscriptionsRequest*. If differ, a fault will be returned.

- *//smn-ext:ListSubscriptions/smn-ext:Filter*

The expanding element from *QCG Notification* schema. It consists of a number of optional filtering parameters.

- *//smn-ext:ListSubscriptions/smn-ext:Filter/SubscriptionID*

A Subscription identifier. Thanks to this parameter Subscriptions may be simply filtered on the basis of Subscription ID. The value is at most 32 char long string.

- */smn-ext:ListSubscriptions/smn-ext:Filter/smn-ext:ConsumerReference/wsa:Address*

The parameter that enables filtering Subscriptions based on addresses of Notification Consumers.

- */smn-ext:ListSubscriptions/smn-ext:Filter/smn-ext:CreationTimeMin*

/smn-ext:ListSubscriptions/smn-ext:Filter/smn-ext:CreationTimeMax

/smn-ext:ListSubscriptions/smn-ext:Filter/smn-ext:TerminationTimeMin

/smn-ext:ListSubscriptions/smn-ext:Filter/smn-ext:TerminationTimeMax

All these elements define filtering parameters related to lifetime of a Subscription, respectively: minimal creation time, maximal creation time, minimal termination time and maximal termination time. The content of these elements is a time expressed as a local time with a timezone.

- */smn-ext:ListSubscriptions/smn-ext:Filter/wsnt:Filter*

The same parameter as the parameter specified in the Subscribe message. In this method it is utilized to filter Subscriptions on the basis of notification filtering parameters.

ListSubscriptionsResponse

The message is sent in the response to ListSubscriptions. It includes a list of information sets about available and probably filtered Subscriptions.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://schemas.qoscosgrid.org/ntf/2011/04/
      QCGNotificationManager/ListSubscriptionsResponse
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <smn-ext:ListSubscriptionsResponse>
```

```
<smn-ext:Subscription>
  <smn-ext:SubscriptionID>
    16807
  </smn-ext:SubscriptionID>
  <smn-ext:ConsumerReference>
    <wsa:Address>
      http://localhost:2233
    </wsa:Address>
  </smn-ext:ConsumerReference>
  <smn-ext:CreationTime>
    2010-07-09T09:21:48Z
  </smn-ext:CreationTime>
  <smn-ext:TerminationTime>
    2010-07-09T09:21:48Z
  </smn-ext:TerminationTime>
  <wsnt:Filter>
    <wsnt:MessageContent
      xmlns:smnex="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousemessag
      Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
        boolean(//smnex:State[text()="on"])
      </wsnt:MessageContent>
    <wsnt:TopicExpression
      xmlns:tns="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics"
      Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Full">
        tns:/*
      </wsnt:TopicExpression>
    </wsnt:Filter>
  </smn-ext:Subscription>
</smn-ext:Subscription>
...
</smn-ext:ListSubscriptionsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/ListSubscriptionsResponse*. If differ, a fault will be returned.

- *//smn-ext:ListSubscriptionsResponse*

This element contains a list of subscriptions received from *QCG Notification* (probably also filtered on the basis of the *smn-ext:Filter* element sent in *ListSubscriptions*).

- *//smn-ext:ListSubscriptionsResponse/smn-ext:Subscription*

The element provides information about a single Subscription.

- *//smn-ext:ListSubscriptionsResponse/smn-ext:Subscription/SubscriptionID*

A Subscription identifier. The value is at most 32 char long string.

- *//smn-ext:ListSubscriptionsResponse/smn-ext:Subscription/smn-ext:ConsumerReference/wsa:Address*

An address of Notification Consumer declared in a Subscription.

- */smn-ext:ListSubscriptionsResponse/smn-ext:Subscription/smn-ext:CreationTime*

/smn-ext:ListSubscriptionsResponse/smn-ext:Subscription/smn-ext:TerminationTime

Creation time and termination time of a Subscription. The content of these elements is a time expressed as UTC time.

- */smn-ext:ListSubscriptionsResponse/smn-ext:Subscription/wsnt:Filter*

The same parameter as the parameter specified in the Subscribe message. It specifies Notification filtering parameters of a Subscription.

ListPublishersRegistrations

The next method from the group of administrative methods. It may be utilized to list Publishers Registrations. The method consists of several filtering parameters.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://schemas.qoscosgrid.org/ntf/2011/04/
      QCGNotificationManager/ListPublishersRegistrationsRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <smn-ext:ListPublishersRegistrations>
      <smn-ext:Filter>
        <smn-ext:PublisherRegistrationID>
          984943658
        </smn-ext:PublisherRegistrationID>
        <smn-ext:PublisherReference>
          <wsa:Address>
            127.0.0.1
          </wsa:Address>
        </smn-ext:PublisherReference>
        <smn-ext:CreationTimeMin>
          2010-07-08T09:37:04Z
        </smn-ext:CreationTimeMin>
        <smn-ext:CreationTimeMax>
          2010-07-08T11:37:04Z
        </smn-ext:CreationTimeMax>
        <smn-ext:TerminationTimeMin>
          2010-07-08T09:37:04Z
        </smn-ext:TerminationTimeMin>
        <smn-ext:TerminationTimeMax>
          2010-07-08T11:37:04Z
        </smn-ext:TerminationTimeMax>
      </smn-ext:Filter>
    </smn-ext:ListPublishersRegistrations>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/ListPublishersRegistrationsRequest*. If differ, a fault will be returned.

- *//smn-ext:ListPublishersRegistrations/smn-ext:Filter*

The element from *QCG Notification* schema. It consists of optional filtering parameters.

- *//smn-ext:ListPublishersRegistrations/smn-ext:Filter/PublisherRegistrationID*

A Publisher Registration identifier. This parameter enables filtering of Publisher Registrations based on Publishers Registrations ID. The value is at most 32 char long string.

- */smn-ext:ListPublishersRegistrations/smn-ext:Filter/smn-ext:PublisherReference/wsa:Address*

This element enables filtering based on an addresses of Publishers. Depending on configuration, this address may be held in the *QCG Notification* service as an address sent in the From element of WSA Header of RegisterPublisher message or an IP address of a Publisher.

- */smn-ext:ListPublishersRegistrations/smn-ext:Filter/smn-ext:CreationTimeMin*

/smn-ext:ListPublishersRegistrations/smn-ext:Filter/smn-ext:CreationTimeMax

/smn-ext:ListPublishersRegistrations/smn-ext:Filter/smn-ext:TerminationTimeMin

/smn-ext:ListPublishersRegistrations/smn-ext:Filter/smn-ext:TerminationTimeMax

The elements define filtering parameters related to lifetime of a Publisher Registration, respectively: minimal creation time, maximal creation time, minimal termination time and maximal termination time. The content of these elements is a time expressed as a local time with a timezone.

ListPublisherRegistrationsResponse

The response message to ListPublishersRegistrations. It includes a list of information sets about Publishers Registrations.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://schemas.qoscosgrid.org/ntf/2011/04/
      QCGNotificationManager/ListPublishersRegistrationsResponse
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <smn-ext:ListPublishersRegistrationsResponse>
      <smn-ext:PublisherRegistration>
        <smn-ext:PublisherRegistrationID>
          16807
        </smn-ext:PublisherRegistrationID>
        <smn-ext:PublisherReference>
          <wsa:Address>
            http://localhost:2233
          </wsa:Address>
```

```
</smn-ext:PublisherReference>
<smn-ext:CreationTime>
  2010-07-09T09:21:48Z
</smn-ext:CreationTime>
<smn-ext:TerminationTime>
  2010-07-09T09:21:48Z
</smn-ext:TerminationTime>
</smn-ext:PublisherRegistration>
<smn-ext:PublisherRegistration>
  ...
</smn-ext:PublisherRegistration>
...
</smn-ext:ListPublishersRegistrationsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/ListPublishersRegistrationsResponse*. If differ, a fault will be returned.

- *//smn-ext:ListPublishersRegistrationsResponse*

The element contains a list of Publishers Registrations received from *QCG Notification* (probably also filtered on the basis of the *smn-ext:Filter* element sent in a *ListPublishersRegistrations* message).

- *//smn-ext:ListPublishersRegistrationsResponse/smn-ext:PublisherRegistration*

The element provides information about a single Publisher Registration.

- *//smn-ext:ListPublishersRegistrationsResponse/smn-ext:PublisherRegistration/PublisherRegistrationID*

A Publisher Registration identifier. The value is at most 32 char long string.

- */smn-ext:ListPublishersRegistrationsResponse/smn-ext:PublisherRegistration/smn-ext:PublisherReference/wsa:Address*

An address of Publisher. It may be an address defined in a From element of WSA Header of a RegisterPublisher message or an IP address of a publisher.

- */smn-ext:ListPublishersRegistrationsResponse/smn-ext:PublisherRegistration/smn-ext:CreationTime*

/smn-ext:ListPublishersRegistrationsResponse/smn-ext:PublisherRegistration/smn-ext:TerminationTime

Creation time and termination time of a Publisher Registration. The content of these elements is a time expressed as UTC time.

ListPullPoints

The next administrative method. It enables listing of available PullPoints. The method consists of several optional filtering parameters.

```
<SOAP-ENV:Envelope ... >
```

```
<SOAP-ENV:Header>
  <wsa:Action SOAP-ENV:mustUnderstand="1">
    http://schemas.qoscosgrid.org/ntf/2011/04/
    QCGNotificationManager/ListPullPointsRequest
  </wsa:Action>
  ...
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <smn-ext:ListPullPoints>
    <smn-ext:Filter>
      <smn-ext:PullPointID>
        9342232
      </smn-ext:PullPointID>
      <smn-ext:PullPointReference>
        <wsa:Address>
          http://localhost:2211?ppID=9342232
        </wsa:Address>
      </smn-ext:PullPointReference>
      <smn-ext:CreationTimeMin>
        2010-07-08T09:38:03Z
      </smn-ext:CreationTimeMin>
      <smn-ext:CreationTimeMax>
        2010-07-08T11:38:03Z
      </smn-ext:CreationTimeMax>
      <smn-ext:TerminationTimeMin>
        2010-07-08T09:38:03Z
      </smn-ext:TerminationTimeMin>
      <smn-ext:TerminationTimeMax>
        2010-07-08T11:38:03Z
      </smn-ext:TerminationTimeMax>
    </smn-ext:Filter>
  </smn-ext:ListPullPoints>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/ListPullPointsRequest*. If differ, a fault will be returned.

- *//smn-ext:ListPullPoints/smn-ext:Filter*

The element from *QCG Notification* schema. It consists of optional filtering parameters.

- *//smn-ext:ListPullPoints/smn-ext:Filter/smn-ext:PullPointID*

A PullPoint identifier. This parameter enables filtering of PullPoints based on PullPoint ID. The value is at most 32 char long string.

- *//smn-ext:ListPullPoints/smn-ext:Filter/smn-ext:PullPointReference/wsa:Address*

This element enables filtering based on an addresses of PullPoint. The PullPoint address is assigned after creation of a PullPoint in *QCG Notification* and returned inside CreatePullPointResponse.

- *//smn-ext:ListPullPoints/smn-ext:Filter/smn-ext:CreationTimeMin*

/smn-ext:ListPullPoints/smn-ext:Filter/smn-ext:CreationTimeMax

/smn-ext:ListPullPoints/smn-ext:Filter/smn-ext:TerminationTimeMin

/smn-ext:ListPullPoints/smn-ext:Filter/smn-ext:TerminationTimeMax

The elements define filtering parameters related to PullPoint lifetime, respectively: minimal creation time, maximal creation time, minimal termination time and maximal termination time. The content of these elements is a time expressed as a local time with a timezone.

ListPullPointsResponse

The response message to ListPullPoints. It includes a list of information sets about filtered Pullpoints.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://schemas.qoscosgrid.org/ntf/2011/04/
QCGNotificationManager/ListPullPointsResponse
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <smn-ext:ListPullPointsResponse>
      <smn-ext:PullPoint>
        <smn-ext:PullPointID>
9342232
        </smn-ext:PullPointID>
        <smn-ext:PullPointReference>
          <wsa:Address>
http://localhost:2211?ppID=9342232
          </wsa:Address>
        </smn-ext:PullPointReference>
        <smn-ext:CreationTime>
2010-07-09T09:31:28Z
        </smn-ext:CreationTime>
        <smn-ext:TerminationTime>
2010-07-09T09:31:28Z
        </smn-ext:TerminationTime>
      </smn-ext:PullPoint>
      <smn-ext:PullPoint>
        ...
      </smn-ext:PullPoint>
      ...
    </smn-ext:ListPullPointsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/ListPullPointsResponse*. If differ, a fault will be returned.

- *//smn-ext:ListPullPointsResponse*

The element contains a list of PullPoints received from *QCG Notification* (probably also previously filtered on the basis of the smn-ext:Filter element sent in a ListPullPoints message).

- *//smn-ext:ListPullPointsResponse/smn-ext:PullPoint*

The element provides particular information about a single PullPoint.

- *//smn-ext:ListPullPointsResponse/smn-ext:PullPoint/PullPointID*

A PullPoint identifier. The value is at most 32 char long string.

- *//smn-ext:ListPullPointsResponse/smn-ext:PullPoint/smn-ext:PullPointReference/wsa:Address*

An address of PullPoint. For particular PullPoint it is the same address as the address from a corresponding CreatePullPointResponse message.

- *//smn-ext:ListPullPointsResponse/smn-ext:PullPoint/smn-ext:CreationTime*

//smn-ext:ListPullPointsResponse/smn-ext:PullPoint/smn-ext:TerminationTime

Creation time and termination time of a PullPoint. The content of these elements is a time expressed as UTC time.

ListTopicNamespaces

An administrative method. It is designed to listing available Topic Namespaces and Topics assigned to this namespaces.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://schemas.qoscosgrid.org/ntf/2011/04/
      QCGNotificationManager/ListTopicNamespacesRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <smn-ext:ListTopicNamespaces>
      <smn-ext:Filter>
        <smn-ext:targetNamespace>
          http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics
        </smn-ext:targetNamespace>
      </smn-ext:Filter>
    </smn-ext:ListTopicNamespaces>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/ListTopicNamespacesRequest*. If differ, a fault will be returned.

- *//smn-ext:ListTopicNamespaces/smn-ext:Filter*

The element from *QCG Notification* schema. It consists of optional filtering parameters.

- *//smn-ext:ListPullPoints/smn-ext:Filter/smn-ext:targetNamespace*

A targetNamespace of a TopicNamespace. This parameter enables filtering of TopicNamespaces based on their targetNamespace names.

ListTopicNamespacesResponse

The response message to ListTopicNamespaces. The message consists of information sets for each filtered TopicNamespace. A single set includes targetNamespaces and a list of topics.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://schemas.qoscosgrid.org/ntf/2011/04/
      QCGNotificationManager/ListPullPointsResponse
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <smn-ext:ListTopicNamespacesResponse>
      <smn-ext:TopicNamespace>
        <smn-ext:targetNamespace>
          http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics
        </smn-ext:targetNamespace>
        <smn-ext:Topic>
          http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics/House
        </smn-ext:Topic>
        <smn-ext:Topic>
          ...
        </smn-ext:Topic>
        ...
      </smn-ext:TopicNamespace>
      <smn-ext:TopicNamespace>
        ...
      </smn-ext:TopicNamespace>
      ...
    </smn-ext:ListTopicNamespacesResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/ListTopicNamespacesResponse*. If differ, a fault will be returned.

- *//smn-ext:ListTopicNamespacesResponse*

The element contains a list of information about TopicNamespaces received from *QCG Notification* (optionally filtered on the basis of the smn-ext:Filter element sent in a ListTopicNamespaces message).

- *//smn-ext:ListTopicNamespacesResponse/smn-ext:TopicNamespace*

The element provides particular information about a single TopicNamespace.

- *//smn-ext:ListTopicNamespacesResponse/smn-ext:TopicNamespace/smn-ext:targetNamespace*

A targetNamespace identifying single TopicNamespace. The value should be compliant with anyURI.

- */smn-ext:ListTopicNamespacesResponse/smn-ext:TopicNamespace/smn-ext:Topic*

A list of topics belonging to a TopicNamespace. Each listed topic is preceded by a targetNamespace so it is Unix.

AddTopicNamespace

An administrative method. The method may be invoked to add a new TopicNamespace to the *QCG Notification*.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://schemas.qoscosgrid.org/ntf/2011/04/
      QCGNotificationManager/AddTopicNamespaceRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <smn-ext:AddTopicNamespace>
      <wstop:TopicNamespace
        targetNamespace="http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetop"
        <wstop:Topic final="false" name="House">
          <wstop:Topic ...>
            ...
          </wstop:Topic>
          ...
        </wstop:Topic>
        ...
      </wstop:TopicNamespace>
    </smn-ext:AddTopicNamespace>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/AddTopicNamespaceRequest*. If differ, a fault will be returned.

- *//smn-ext:AddTopicNamespaces/wstop:TopicNamespace*

The element defining a single Topic Namespace. It comes from WS-Topics schema.

AddTopicNamespaceResponse

A response message to a AddTopicNamespaceRequest. It is empty and only confirms successful addition of a TopicNamespace to the *QCG Notification* service.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      PullPoint/AddTopicNamespaceResponse
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
```

```
<SOAP-ENV:Body>
  <smn-ext:AddTopicNamespaceResponse>
</smn-ext:AddTopicNamespaceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: <http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/AddTopicNamespaceResponse>. If differ, a fault will be returned.

RemoveTopicNamespace

The next administrative method. It is used to remove a TopicNamespace from the *QCG Notification* system. Note that the removing may be only temporary (active to the restart of the system) if the TopicNamespace file is protected by disabling write permission.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://schemas.qoscosgrid.org/ntf/2011/04/
      QCGNotificationManager/RemoveTopicNamespaceRequest
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <smn-ext:RemoveTopicNamespace>
      <smn-ext:targetNamespace>
        http://schemas.qoscosgrid.org/ntf/2011/04/sample/myhousetopics2/House
      </smn-ext:targetNamespace>
    </smn-ext:RemoveTopicNamespace>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: <http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/RemoveTopicNamespaceRequest>. If differ, a fault will be returned.

- *//smn-ext:RemoveTopicNamespaces/smn-ext:targetNamespace*

The targetNamespace name of a TopicNamespace dedicated to remove - anyURI.

RemoveTopicNamespaceResponse

A response message to a RemoveTopicNamespaceRequest. It is empty and confirms the successful deletion of a TopicNamespace.

```
<SOAP-ENV:Envelope ... >
  <SOAP-ENV:Header>
    <wsa:Action SOAP-ENV:mustUnderstand="1">
      http://docs.oasis-open.org/wsn/bw-2/
      PullPoint/RemoveTopicNamespaceResponse
    </wsa:Action>
    ...
  </SOAP-ENV:Header>
```

```
<SOAP-ENV:Body>
  <smn-ext:RemoveTopicNamespaceResponse>
  </smn-ext:RemoveTopicNamespaceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *//SOAP-ENV:Header/wsa:Action*

For this message should always be: *http://schemas.qoscosgrid.org/ntf/2011/04/QCGNotificationManager/RemoveTopicNamespaceResponse*. If differ, a fault will be returned.

Administration Guide of QCG Notification

Prerequisites

In order to build and install *QCG Notification*, an administrator has to appropriately configure operating system. There is some dependencies to external software that should be satisfied.

QCG Core library

QCG Notification is based on the *QCG Core* library developed by the *QCG* team. QCG Core is a common utility library providing set of basic modules used by other components within QCG Project. For more information see QCG Core documentation.

The QCG Core library should be properly configured in prior to preparation of *QCG Notification*.

Building

Before you compile *QCG Notification*, you need to run the configuration script first. It will try to check if all the requirements are met and whether needed third party software is available. Simply type the following command:

```
$ ./configure
```

There are several options that can be passed to the script. If you type the `--help` parameter you will see all available options. These are the most important ones:

- `--prefix=DIR`

specifies directory where *QCG Notification* will be installed.

Default: `/opt/qcg`

- `--with-qcg-core=PFX`

specifies where QCG Core library is installed.

Default: no default value. `qcg-core-config` script is used to determine necessary paths

- `--with-client`

Determine if the test client program should be build

- `--with-consumer`

Determine if the test NotificationConsumer service should be build

- --with-addressing-handler

Determine if the test Web Services Addressing handler service should be build

After successful configuration step, compiling and installing of *QCG Notification* should be a straightforward process:

```
$ make
$ make install
```

Note

The installation step should be done either as `root` or the user that will execute *QCG Notification* service, assuming he has necessary privileges to do this.

Creating database for *QCG Notification* - the PostgreSQL example

QCG Notification can be run in two basic modes related to the database connection: without database or with database. Although the first option is sufficient for simple scenarios, it reduces the overall functionality of the *QCG Notification* (e.g. disable PullPoints). Therefore using the database is recommended and can be often essential.

You will find more detailed description of administrative tasks in PostgreSQL tutorial. However, the following basic steps are needed to set up a database and appropriate tables for *QCG Notification* assuming that you have already setup a PostgreSQL database:

By default, the first database user created will also be named `postgres`. In order to create a user for *QCG Notification* usage, for instance `qcg-ntf`, type the following:

```
$ createuser -U postgres -W qcg-ntf -P
```

If you use a password then remember to supply it in `qcg-ntf.xml` configuration file. Create a database, for example `qcg-ntf`, by typing the following command:

```
$ createdb qcg-ntf
```

and set up a table structure using the `qcg-ntf-psql.sql` file from *QCG Notification* distribution - it can be found in `/opt/qcg/share/tools` directory by default.

```
$ psql -d qcg-ntf -U qcg-ntf < qcg-ntf-psql.sql
```

Eventually you need to add this database as the ODBC Data Source Name. You can do this by editing system-wide configuration file (e.g. `/etc/odbc.ini`) or `.odbc.ini` in home directory of the user that will be used to start *QCG Notification*.

Assuming that ODBC are installed in `/usr/local/lib` and the choosed name for *QCG Notification* is `qcg-ntf` the configuration file should looks similar to the following:

```
[qcg-ntf]
Description      = QCG Notification database
Driver           = /usr/local/lib/libodbcpsql.so
Setup            = /usr/local/lib/libodbcpsqls.so
Database         = qcg-ntf
Servername       = localhost
Port             = 5432
```

```
ReadOnly      = No
Trace         = 0
TraceFile     = odbcc.trace
Debug         = 0
DebugFile     = odbcc.debug
```

You can also try some GUI tools to set up and manage ODBC and databases. Check out if the following tools are available on your machine:

\$ DataManager

\$ ODBCConfig

Configuration of *QCG Notification*

Configuration File: `qcg-ntfd.xml`

```
<sm:QCGCore
  xmlns:sm="http://schemas.qoscosgrid.org/core/2011/04/config"
  xmlns="http://schemas.qoscosgrid.org/ntf/2011/04/config"
  xmlns:smn="http://schemas.qoscosgrid.org/ntf/2011/04/config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Configuration>
    <sm:ModuleManager>
      <sm:Directory>/opt/qcg/lib/qcg-core/modules</sm:Directory>
      <sm:Directory>/opt/qcg/lib/modules</sm:Directory>
    </sm:ModuleManager>

    <sm:Service xsi:type="ntfd" description="QCG Notification">
      <sm:Logger>
        <sm:Filename>/opt/qcg/var/log/qcg-ntf/qcg-ntfd.log</sm:Filename>
        <sm:Level>DEBUG</sm:Level>
      </sm:Logger>

      <sm:Transport>
        <sm:Module xsi:type="sm:ecm_gsoap.service">
          <sm:Host>localhost</sm:Host>
          <sm:Port>2211</sm:Port>
          <sm:UseWSA>true</sm:UseWSA>
        </sm:Module>
        <sm:Module xsi:type="smn:ntf-service-gsoap"/>
      </sm:Transport>

      <sm:Authentication>
        <sm:Module xsi:type="sm:atc_transport_http"/>
      </sm:Authentication>

      <sm:Module xsi:type="smn:ntf-publisher-gsoap-anonymous">
        <UseWSA>true</UseWSA>
        <RecvTimeout>10</RecvTimeout>
        <SendTimeout>10</SendTimeout>
      </sm:Module>
```

```
<sm:Module xsi:type="smn:ntf-publisher-gsoap-ssl">
  <UseWSA>true</UseWSA>
  <RecvTimeout>10</RecvTimeout>
  <SendTimeout>10</SendTimeout>
  <X509CertKeyFile>/opt/qcg/etc/ssl/broker.pem</X509CertKeyFile>
  <X509KeyPass>somepass</X509KeyPass>
  <sm:TrustedCAFile>/opt/qcg/etc/ssl/certs/qcg-ntf/ca.pem</sm:TrustedCAFile>
</sm:Module>

<Core>
  <NotificationsVar>
    <TopicNsDir>/opt/qcg/etc/topicns</TopicNsDir>
    <StrictCorrectnessCheckingEnabled>>false</StrictCorrectnessCheckingEnabled>
    <DisableCurrentMessages>>false</DisableCurrentMessages>
    <SubscriptionManager>
      <ContentFilteringEnabled>true</ContentFilteringEnabled>
      <SubscriptionPolicy>DUPLICATE</SubscriptionPolicy>
      <DialectConsistencyRequired>true</DialectConsistencyRequired>
      <SubscriptionPeriod>10</SubscriptionPeriod>
      <RefreshRate>3600</RefreshRate>
    </SubscriptionManager>
    <PublisherManager>
      <RegistrationRequired>>false</RegistrationRequired>
      <RegistrationPeriod>10</RegistrationPeriod>
      <RefreshRate>3600</RefreshRate>
    </PublisherManager>
    <PullPointManager>
      <PullPointsEnabled>true</PullPointsEnabled>
      <PullPointSize>100</PullPointSize>
      <PullPointPeriod>10</PullPointPeriod>
      <RefreshRate>3600</RefreshRate>
    </PullPointManager>
    <Publishing>
      <NotificationsBufferEnabled>true</NotificationsBufferEnabled>
      <NotificationsBufferSize>1000</NotificationsBufferSize>
      <NotificationsResendingRate>60</NotificationsResendingRate>
      <MaxSendTries>5</MaxSendTries>
    </Publishing>
  </NotificationsVar>

  <Database>
    <DatabaseEnabled>true</DatabaseEnabled>
    <DSN>qcg-ntf</DSN>
    <User>qcg-ntf</User>
    <Password>qcg-ntf</Password>
    <CleanAtStart>>false</CleanAtStart>
  </Database>
</Core>

</sm:Service>

</Configuration>
</sm:QCGCore>
```

- *//sm:ModuleManager/sm:Directory* [string]
Directories with modules that will be used in the service.
- *//sm:Service* [sm:QCGBaseServiceType]
Configuration of the services. It is possible to configure many instances of the service in the one configuration file.
- *//sm:Service/sm:Logger* [sm:LoggerType]
Configuration of the logging.
- *//sm:Service/sm:Logger/sm:FileName* [string]
A log file path.
- *//sm:Service/sm:Logger/sm:Level* [enumeration]
DEBUG, INFO, NOTICE, WARNING, ERROR, CRIT, EMERG - filter less important messages.
- *//sm:Service/sm:Transport* [sm:TransportType]
This element aggregates configuration of transport modules. At least one module of type ecm* should be specified in this place. For full list of supported ecm* modules see the QCG Core documentation
- *//sm:Service/sm:Transport/sm:Module[@type=sm:ecm_gsoap.service]*
Basic configuration of the Web Service interface. The most important configuration covers such aspects as host and port where the service will be listening. The instruction of using other options is located in QCG Core documentation.
- *//sm:Service/sm:Transport/sm:Module[@type=smn:ntf-service-gsoap]*
Define ntf-service-gsoap module as a default module for the service. This option shouldn't be changed.
- *//sm:Service/sm:Authentication* [sm:AuthenticationType]
Configuration of authentication modules for the service. For full list of supported authentication modules see the QCG Core documentation (the most basic are: sm:atc_transport_http for anonymous communication and sm:atc_transport_https for SSL based communication).
- *//sm:Service/sm:Module/[@type=smn:ntf-publisher*]*
Initialization and configuration of the modules for the Publisher side of *QCG Notification*. The following modules are currently supported:
 - smn:ntf-publisher-gsoap-anonymous for anonymous SOAP communication;
 - smn:ntf-publisher-gsoap-ssl for SSL based SOAP communication;
 - smn:ntf-publisher-xmpp for XMPP communication;
 - smn:ntf-publisher-smtp for SMTP communication;For details see the Section 5.2.
- *//smn:Core/smn:NotificationsVar* [smn:NotificationsVarType]
Various notification settings closely related to QCG Notification service implementation.

- *//smn:Core/smn:NotificationsVar/smn:TopicNsDir* [string]
Location of the directory with files containing Topic definitions.
- *//smn:Core/smn:NotificationsVar/smn:StrictCorrectnessCheckingEnabled* [boolean]
true, false - determines a strict validation of incoming messages; if true, it allows receiving detailed information about faults but it causes greatest delays; if false, then faults processing is minimal and delays are smaller. Default: false
- *//smn:Core/smn:NotificationsVar/smn:DisableCurrentMessages* [boolean]
true, false - enable or disable support for CurrentMessages; If set to true, the last notification messages wouldn't be stored in a buffer for the gathering by the GetCurrentMessage method. Default: false
- *//smn:Core/smn:NotificationsVar/smn:SubscriptionManager/* [smn:SubscriptionManagerType]
Settings related to Subscriptions management.
- *//smn:Core/smn:NotificationsVar/smn:SubscriptionManager/smn:ContentFilteringEnabled* [boolean]
true, false - determines if a content filtering of notification messages can be used; if true, Subscribers can specify content filters in their Subscriptions, if false, it is disallowed. Default: true
- *//smn:Core/smn:NotificationsVar/smn:SubscriptionManager/smn:SubscriptionPolicy* [enumeration]
DUPLICATE, DELETE_OLD, RETURN_FAULT - parameter defines an action which should be taken when a Subscriber wants to create a Subscription with the same parameters as the parameters of already existing Subscription. Default: **DELETE_OLD (before the version 2.6.2 it was DUPLICATE)**
- *//smn:Core/smn:NotificationsVar/smn:SubscriptionManager/smn:DialectConsistencyRequired* [boolean]
true, false - determines if a Subscription covers the Notify messages with a different Topics Dialect than the Topic Dialect of the Subscription; if true, the Subscription covers only Notify messages with the same Dialect; if false, the Subscription covers Notify messages with simpler Dialects (i.e. Subscription with Full type of Dialect covers notifications sent with Concrete or Simple type). Default: true
- *//smn:Core/smn:NotificationsVar/smn:SubscriptionManager/smn:SubscriptionPeriod* [integer]
A default time period for a Subscription to be valid; in minutes; if negative the Subscription time checking will be disabled. Default: **43200 (before the version 2.6.2 it was 10)**
- *//smn:Core/smn:NotificationsVar/smn:SubscriptionManager/smn:RefreshRate* [integer]
A time period between successive Subscription Manager refreshing events (events when outdated Subscriptions are deleted); the time period is measured in seconds; if value is set to 0 or negative then the periodical refreshing is disabled. Default: 3600
- *//smn:Core/smn:NotificationsVar/smn:PublisherManager/* [smn:PublisherManagerType]
Settings related to Publishers management.
- *//smn:Core/smn:NotificationsVar/smn:PublisherManager/smn:RegistrationRequired* [boolean]
true, false - determines a requirement of Publishers Registrations; if true, a Publisher has to register himself before sending notifications; if false, it has not. Default: false
- *//smn:Core/smn:NotificationsVar/smn:PublisherManager/smn:RegistrationPeriod* [integer]

A default time period for a Publisher Registration to be valid; in minutes; if negative the Registration time checking will be disabled. **Default: 43200 (before the version 2.6.2 it was 10)**

- `//smn:Core/smn:NotificationsVar/smn:PublisherManager/smn:RefreshRate` [integer]

A time period between successive Publisher Manager refreshing events (events when outdated Publishers are deleted); the time period is measured in seconds; if value is set to 0 or negative then the periodical refreshing is disabled. Default: 3600

- `//smn:Core/smn:NotificationsVar/smn:PullPointManager/` [smn:PullPointManagerType]

Settings related to PullPoints management.

- `//smn:Core/smn:NotificationsVar/smn:PullPointManager/smn:PullPointsEnabled` [boolean]

true, false - determines if PullPoints for pull style of notifications can be used. Default: true

- `//smn:Core/smn:NotificationsVar/smn:PullPointManager/smn:PullPointSize` [integer]

A maximal number of messages that can be buffered in a single PullPoint; if negative then the number of messages is unbounded. Default: **1000 (before the version 2.6.2 it was 100)**

- `//smn:Core/smn:NotificationsVar/smn:PullPointManager/smn:PullPointPeriod` [integer]

A default time period for a PullPoint to be valid; in minutes; if negative the PullPoint time checking will be disabled. Default: **43200 (before the version 2.6.2 it was 10)**

- `//smn:Core/smn:NotificationsVar/smn:PullPointManager/smn:RefreshRate` [integer]

A time period between successive PullPoint Manager refreshing events (events when outdated PullPoints are deleted); the time period is measured in seconds; if value is set to 0 or negative then the periodical refreshing is disabled. Default: 3600

- `//smn:Core/smn:NotificationsVar/smn:Publishing/` [smn:PublishingType]

A group of settings related to Publishing procedure.

- `//smn:Core/smn:NotificationsVar/smn:Publishing/smn:NotificationsBufferEnabled` [boolean]

true, false - determines if Notifications Buffer should be enabled; Notifications Buffer is used for re-sending Notifications if those weren't sent successfully at the first attempt. Default: true

- `//smn:Core/smn:NotificationsVar/smn:Publishing/smn:NotificationsBufferSize` [integer]

A maximal number of Notifications that can be buffered in Notifications Buffer. Default: **10000 (before the version 2.6.2 it was 1000)**

- `//smn:Core/smn:NotificationsVar/smn:Publishing/smn:NotificationsResendingRate` [integer]

A time period between successive notification resending procedures; in seconds; if 0 or negative then the resending will be disabled. Default: 60

- `//smn:Core/smn:NotificationsVar/smn:Publishing/smn:MaxSendTries` [integer]

A maximal number of consecutive unsuccessful tries of sending notification; unsuccessful tries of sending notifications are counted independently for every Subscription; a certain Subscription will be delayed if its counter become equal to *MaxSendTries*; if value is set to 0 or negative then this option is disabled. Default: 5

- *//smn:Core/Database/DatabaseEnabled* [boolean]

true, false - determines if a database should be enabled; in the simplest configuration, *QCG Notification* can be used also without a database connection, however many options will be then unavailable. Default: true
- *//smn:Core/Database/DSN* [string]

Data Source Name of a database. Note, that you should use ODBC DSN **not** the database name.
- *//smn:Core/Database/User* [string]

A database user name.
- *//smn:Core/Database/Password* [string]

A password for a database user.
- *//smn:Core/Database/CleanAtStart* [boolean]

true, false - determines whether all information stored in a database should be cleaned at starting of the *QCG Notification* service.

Module configuration

Modules are located in the PREFIX/*/lib/modules directories and are loaded during the *QCG Notification* run time. Module definitions are specified in appropriate sections of the configuration file.

Publisher modules

As it was discussed in the previous sections, in a typical brokered notification scenario there are two types of communication and notification channels that need to be setup. The first one is a communication channel between Subscriber/Publisher clients and *QCG Notification*. This type of communication is currently completely based on QCG Core library.

The second type is an outgoing communication between *QCG Notification* and NotificationConsumers and it is particular for *QCG Notification*. In this situation *QCG Notification* plays a role of a Notification Publisher. At this time, supported modules for this channel utilize a Web Service interface, XMPP interface and SMTP interface.

- **Publisher Anonymous WS transport**

This type of transport is recommended only for internal, secure network environments or testing.

```
<Module xsi:type="smn:ntf-publisher-gsoap-anonymous">
  <smn:UseWSA>true</smn:UseWSA>
  <smn:RecvTimeout>10</smn:RecvTimeout>
  <smn:SendTimeout>10</smn:SendTimeout>
  <smn:NotifyTopicConversion>WSN</smn:NotifyTopicConversion>
</Module>
```

- **Publisher SSL WS transport and authentication**

```
<Module xsi:type="smn:ntf-publisher-gsoap-ssl">
  <smn:UseWSA>true</smn:UseWSA>
  <smn:RecvTimeout>10</smn:RecvTimeout>
  <smn:SendTimeout>10</smn:SendTimeout>
```

```
<smn:X509CertKeyFile>/opt/qcg/etc/ssl/qcg-ntf/qcg-ntfd.pem</smn:X509CertKeyFile>
<smn:X509KeyPass>qcgpass</smn:X509KeyPass>
<smn:TrustedCAFile>/opt/qcg/etc/ssl/certs/qcg-ntf/ca.pem</smn:TrustedCAFile>
<smn:TrustedCAPath>/opt/qcg/etc/ssl/certs/qcg-ntf/ca.pem</smn:TrustedCAPath>
<smn:NotifyTopicConversion>WSN</smn:NotifyTopicConversion>
</Module>
```

- *//smn:UseWSA* [boolean]

Determine if WS-Addressing should be used.

- *//smn:RecvTimeout* [positiveInteger]

The timeout in seconds for stall on the recv function. If negative, time is measured in microseconds.

- *//smn:SendTimeout* [positiveInteger]

The timeout in seconds for stall on the send function. If negative, time is measured in microseconds.

- *//smn:X509CertKeyFile* [string]

A path to the file with both X509 certificate and private key. Mandatory argument.

- *//smn:X509KeyPass* [string]

An optional password to X509 private key.

- *//smn:TrutedCAFile* [string]

A path to the file with trusted CAs certificates. Either TrustedCAFile or TrustedCAPath must be specified.

- *//smn:TrutedCAPath* [string]

A path to the directory with the files with trusted CAs certificates. Either TrustedCAFile or TrustedCAPath must be specified.

- *//smn:NotifyTopicConversion* [enumeration]

WSN, WSN_WITH_ADDITIONAL_NS, INLINE, LACK - defines a type of conversion that will be applied (if needed) to the Topic element of Notify messages.

- **Publisher XMPP transport**

The XMPP publisher module can send two versions of notifications via XMPP. The first version are notifications expressed as SOAP messages, the second are notifications sent as a typical "chat" messages. The second version of notifications is selected automatically when the XSLT document was provided in Subscribe message: the XSLT document will be used to transform the message to the appropriate form.

```
<sm:Module xsi:type="smn:ntf-publisher-xmpp">
  <smn:Server>xmppserver.qcg.com</smn:Server>
  <smn:Port>5222</smn:Port>
  <smn:Username>qcgaccount</smn:Username>
  <smn>Password>qcgpass</smn>Password>
  <smn:Jid>qcgaccount@xmppserver.qcg.com</smn:Jid>
  <smn:ResourceName>QCGntfXmppPublisher</smn:ResourceName>
```

```
<smn:Encryption>TLS</smn:Encryption>
<smn:CertFingerprint>00112233445566778899AABBCCDDEE</smn:CertFingerprint>
<smn:NotifyTopicConversion>WSN</smn:NotifyTopicConversion>
</sm:Module>
```

- *//smn:Server* [string]

Hostname of the XMPP server on which the XMPP account is set up.

- *//smn:Port* [positiveInteger]

Port number of the XMPP server. Default is 5222.

- *//smn:Username* [string]

Username used for the XMPP account.

- *//smn:Password* [string]

The secret key for the XMPP account.

- *//smn:Jid* [string]

JID of the XMPP account

- *//smn:ResourceName* [string]

Resource name used by publisher's client connection.

- *//smn:Encryption* [string]

Encryption to use. By default it's "none". Use "TLS" for STARTTLS connection encryption. Use "SSL" for old-style SSL encryption (typically port 5223).

- *//smn:CertFingerprint* [string]

MD5 fingerprint of the encryption certificate used by the XMPP server. Upon mismatch, the connection will fail. Use empty string to accept any certificate.

- *//smn:NotifyTopicConversion* [enumeration]

WSN, WSN_WITH_ADDITIONAL_NS, INLINE, LACK - defines a type of conversion that will be applied (only if needed) to the Topic element of Notify messages.

- **Publisher SMTP transport**

Similarly to the XMPP publisher module, the SMTP module utilizes the XSLT documents to transform the content of the notification to the human-friendly form.

```
<sm:Module xsi:type="smn:ntf-publisher-smtp">
  <smn:ServerURL>smtp://localhost:25</smn:ServerURL>
  <smn:User>someuser</smn:User>
  <smn:Password>somepassword</smn:Password>
  <smn:Encryption>TLS</smn:Encryption>
  <smn:TrustedCAFile>somefile</smn:TrustedCAFile>
  <smn:TrustedCAPath>somedir</smn:TrustedCAPath>
  <smn:From>some@from</smn:From>
```

```
<smn:Subject>Some subject</smn:Subject>
<smn:Trace>false</smn:Trace>
<smn:SendTimeout>10</smn:SendTimeout>
</sm:Module>
```

- *//smn:Server* [string]

URL of the SMTP server. Mandatory argument.

- *//smn:User* [string]

Name of the account on SMTP server. Optional argument.

- *//smn:Password* [string]

The secret key for the account. Optional argument.

- *//smn:Encryption* [string]

Used to enable encryption. The only supported type of encryption is "TLS". If not present, encryption will be switched off. Optional argument.

- *//smn:TrustedCAFile* [string]

A path to the file with trusted CAs certificates. Optional argument.

- *//smn:TrustedCAPath* [string]

A path to the directory with the files with trusted CAs certificates. Optional argument.

- *//smn:From* [string]

Defines a name that will be placed in From field of the e-mail message. Optional argument.

- *//smn:Subject* [string]

Defines a Subject of the e-mail message. Optional argument.

- *//smn:Trace* [boolean]

Switches trace logging. Optional argument, default is false.

- *//smn:SendTimeout* [int]

The timeout in seconds for stall on the send function. Optional argument, default is 10.

WS-Topics management

In the current release, a topics management in *QCG Notification* is relatively simple. The first step is a preparation of an XML document with a Topic Namespace definition. This step can be done by one of the peers cooperating with *QCG Notification*. Then the document should be delivered to the computer where *QCG Notification* is installed. It can be done in two ways: the document may be copied manually by QCG Notification administrator to TopicNamespaces directory in prior to a *QCG Notification* startup or it can be send through the *QCG Notification* interface when *QCG Notification* is already running. In the last case, the TopicNamespaces are stored also in a files in TopicNamespaces directory so they can be automatically loaded after restart of the service. The TopicNamespaces loaded from files may be protected from unwanted removal by unsetting write permissions for a certain file.

An XML document describing a sample topic namespace is presented in the Section 3.3.

Starting *QCG Notification*

To start *QCG Notification* as a daemon type:

```
$ ./qcg-ntfd
```

If you start *QCG Notification* with parameter `-h` you will get an optional parameters list:

```
$ ./qcg-ntfd -h
usage: ./qcg-ntfd [-d] [-z config_file]

-d          do not detach and log to stderr
-D          do not detach and force logging at debug level to stderr
-z config_file configuration file, default is /opt/qcg/etc/qcg-ntfd.xml
-v          print version information
-h          print this help
```

Once you run *QCG Notification* as a daemon, a log file is created. Using various log levels, you can log less or more details.

Stopping *QCG Notification*

At the moment, shutdown is performed simply by sending SIGINT signal to the service, e.g.:

```
$ kill -SIGINT `pidof qcg-ntfd`
```