UPnP QosPolicyHolder:3
Service Template Version 1.01

For UPnP Version 1.0
Status: Standardized DCP
Date: November 30, 2008

Authors	Member
Ally Yu-kyoung Song	LGE
Amol Bhagwat (editor)	CableLabs
Bruce Fairman	Sony
Daryl Hlasny	Sharp Labs of America
Dieter Verslype	Ghent University
Fred Tuck (co-chair)	EchoStar
Jelle Nelis	Ghent University
Michael van Hartskamp (co-chair)	Philips
Narm Gadiraju	Intel Corporation
Puneet Sharma	HP
Richard Chen	Philips
Sherman Gavette	Sharp Labs of America
Steve Wade	Sharp Labs of America
Suman Sharma	Intel Corporation

Authors	Member
Zong Wu	Entropic

The UPnP Forum in no way guarantees the accuracy or completeness of this author list and in no way implies any rights for or support from those members listed. This list is not the specifications' contributor list that is kept on the UPnP Forum's website.

Contents

List of Tables

1. Overview and Scope

This service definition is compliant with the UPnP Device Architecture version 1.0.

This service type enables modeling of the '*QosPolicyHolder*' function capabilities. The functionality for the *QosPolicyHolder* Service can be implemented by any device on the network. The *QosPolicyHolder* Service is responsible for providing the traffic policy values for any given traffic stream as requested by an entity that manages the network traffic, typically a *QoS Manager*. The traffic policy values are determined by applying the policy rules, which are configured for the network, to the requested traffic information.

A *QosPolicyHolder* is a dual-role entity that exposes a *QosPolicyHolder* Service to the Control Point (mainly the *QoS Manager*) while acting as a Control Point for the *QosDevice* Services running on the network. This document describes the components of the *QosPolicyHolder* Service and the *QoS Policy Holder*. The *QoS Policy Holder* provides the Control Point functionality that discovers and controls *QosDevice* Services, mainly for the propagation and synchronization of the preferred *QosPolicyHolder* Service information. Additional information concerning the *QoS Policy Holder* may be found in:

- UPnP-QoS Architecture document
- UPnP QosDevice Service Definition Document

1.1. Referenced Specifications

Unless explicitly stated otherwise herein, implementation of the mandatory provisions of any standard referenced by this specification shall be mandatory for compliance with this specification.

1.1.1. Normative References

This section lists the normative references used in this document and includes the tag inside square brackets that is used for each sub reference:

[XML] – *Extensible Markup Language (XML) 1.0 (Second Edition)*, T. Bray, J.Paoli, C. M. Sperberg-McQueen, E Maler, eds. W3C Recommendations, 6 October 2000.

[DEVICE] - UPnP Device Architecture, version 1.0, UPnP Forum, July 20, 2006.
Available at: http://upnp.org/specs/arch/UPnP-DeviceArchitecture-v1.0-20060720.pdf
Latest version available at: http://upnp.org/specs/arch/UPnP-DeviceArchitecture-v1.0.pdf

[QM] – UPnP QosManager:3 Service Document: This reference is informative except for the definitions of the following state variables, which are normative: *A_ARG_TYPE_TrafficDescriptor*, and *A_ARG_TYPE_ListOfTrafficDescriptors*.
Available at: http://www.upnp.org/specs/qos/UPnP-qos-QosManager-v3-Service-20081130.pdf
Latest version available at: http://www.upnp.org/specs/qos/UPnP-qos-QosManager-v3-Service.pdf

[RFC3339] – *Date and Time on the Internet: Timestamps*, G. Klyne, July 2002.
http://www.ietf.org/rfc/rfc3339.txt

1.1.2. Informative References

This section lists the informative references used in this document and includes the tag inside square brackets that is used for each sub reference:

[QoS Architecture] – *UPnP QosDevice:3* Service Document
Available at: http://www.upnp.org/specs/qos/UPnP-qos-QosDevice-v3-Service-20081130.pdf
Latest version available at: http://www.upnp.org/specs/qos/UPnP-qos-QosDevice-v3-Service.pdf

[QoS DEV] – *UPnP QosDevice:3* Service Document
Available at: http://www.upnp.org/specs/qos/UPnP-qos-QosDevice-v3-Service-20081130.pdf
Latest version available at: http://www.upnp.org/specs/qos/UPnP-qos-QosDevice-v3-Service.pdf

[IEEE 802.1D] – IEEE 802.1D-2004, Annex G, *IEEE Standard for Information technology - Telecommunications and information exchange between systems - IEEE standard for local and metropolitan area networks - Common specifications - Media access control (MAC) Bridges*, 2004.

2. Service Modeling Definitions

2.1. ServiceType

The following service type identifies a service that is compliant with this template:

urn:schemas-upnp-org:service:*QosPolicyHolder:3*

The shorthand '*QosPolicyHolder* Service' is used herein to refer to this service type.

2.2. Derived Data Types

This section defines some derived data types that are represented as UPnP **string** data types with special syntax.

2.2.1. XML Fragments as UPnP Arguments

UPnP-QoS often uses XML Fragments as arguments in UPnP actions. The containing UPnP data type is a **string**. This places restrictions on a string's content; it has to represent a well-formed XML fragment (this includes a complete XML document).

An XML fragment, in adherence to the UPnP Device Architecture 1.0 [DEVICE], MUST be escaped by using the normal XML rules, [XML]Section 2.4 Character Data and Markup, before embedding it in a SOAP request / response message or an event notification message. The XML escaping rules are summarized:

- The (<) character is encoded as (<)
- The (>) character is encoded as (>)
- The (&) character is encoded as (&)
- The (") character is encoded as (")
- The (') character is encoded as (')

In their XML fragments, implementations MAY use an explicit reference to appropriate namespaces.

2.2.2. Extensibility of XML

The names of UPnP-QoS namespaces come in two flavors. The ones in HTTP-form are existing UPnP-QoS v1 and v2 namespace names. The ones in URN-form are introduced in UPnP-QoS v3 (or later).

In order to maintain the extensibility of a namespace, all future modifications of the schema definition will be proper supersets. The namespace name will not change even when the service version number changes.

The v2, v3, v4 tags within a schema allow for the UPnP Forum to add newly standardized elements to the schema definitions without impacting implementations based on previous version(s) of the schema. UPnP-QoS v3 introduces the v4 tags in a similar way as UPnP-QoS v2 defined the v3 tags. The contents of the v4 tags MAY be (re)defined in UPnP-QoS v4 depending on needs.

At several places in the XML schemas there is also room for vendor differentiation or future revisions through the use of the “any”-tag. This tag is placed both in the original schema as well as within the v2 and v3 tags to allow extensions related to those versions of the specification.

When extending UPnP-QoS with their own XML tags, vendors SHOULD use a namespace to prevent collisions of their tags with those of other vendors. It is RECOMMENDED that implementations are not required to retrieve the corresponding schemas from the Internet. For example, a vendor MAY provide its own enhancements within the schema.

Below is an example using extensions to TrafficPolicy v2

```
<TrafficPolicy
  xmlns="http://www.upnp.org/schemas/TrafficPolicy.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:prv="http://myPrivate.com"
  xmlns:prv2="http://myPrivate2.com"
  xlmns:prv3="urn:schemas-myprivate-com:v3Extensions"
  xlmns:prv4="urn:schemas-myprivate-com:v4Extensions"
  xsi:schemaLocation="http://www.upnp.org/schemas/TrafficPolicy.xsd
    http://www.upnp.org/schemas/qos/TrafficPolicy-v2.xsd">
    <AdmissionPolicy>Enabled</AdmissionPolicy>
    <TrafficImportanceNumber>3</TrafficImportanceNumber>
    <UserImportanceNumber>128</UserImportanceNumber>
    <v2>
       <PolicyHolderId>uuid:2fac1234-31f8-11b4-a222-
08002b34c003:serviceId:qph</PolicyHolderId>
       <PolicyLastModified>2004-11-26T15:03:23-08:00</PolicyLastModified>
       <PolicyModifyingUserName>Jimmy</PolicyModifyingUserName>
      <PolicyHolderConfigUrl>http://10.0.0.5/ConfPolicy.html</PolicyHolderConfigUrl>
          <v3>
              <!-- UPnP Forum v3 extensions go here -->
                    <v4>
                   <!-- UPnP Forum v4 extensions go here -->
                   <prv4:MyPrivate4>whatever</prv4:MyPrivate4>
                       </v4>
               <prv3:MyPrivate3>whatever</prv3:MyPrivate3>
              </v3>
       <prv2:MyPrivate2>whatever</prv2:MyPrivate2>
    </v2>
    <prv:MyPrivate1>whatever</prv:MyPrivate1>
</TrafficPolicy>
```

2.3. State Variables

The *QosPolicyHolder* Service is 'action' based. This service's state variables exist primarily to support argument passing of the service's actions. A client retrieves *QosPolicyHolder* Service information via the return parameters of the actions defined in section 2.5.

Reader Note: For first-time reader, it may be more insightful to read the theory of operations first and then the action definitions before reading the state variable definitions.

Table 2-1: State Variables

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value	Default Value[2]	Eng. Units
A_ARG_TYPE_TrafficDescriptor	R	**string** (XML fragment)	See §2.3.1	n/a	n/a
A_ARG_TYPE_TrafficPolicy	R	**string** (XML fragment)	See §2.3.2	n/a	n/a
A_ARG_TYPE_ListOfTrafficDescriptors	R	**string** (XML fragment)	See §2.3.3	n/a	n/a
A_ARG_TYPE_ListOfTrafficPolicies	R	**string** (XML fragment)	See §2.3.4	n/a	n/a
A_ARG_TYPE_IsPreferred	O	**boolean**	See §2.3.5	False	n/a
A_ARG_TYPE_QphPolicyRule	O	**string** (XML fragment)	See §2.3.6	n/a	n/a
A_ARG_TYPE_ListOfQphPolicyRule	O	**string** (XML fragment)	See §2.3.7	n/a	n/a
A_ARG_TYPE_Position	O	ui4	See §2.3.8	n/a	n/a
A_ARG_TYPE_TIN	O	ui4	See §2.3.9	n/a	n/a
A_ARG_TYPE_IN	O	ui4	See §2.3.10	n/a	n/a
A_ARG_TYPE_ReasonCode	O	ui4	See §2.3.11	n/a	n/a
A_ARG_TYPE_PolicyHandle	O	ui4	See §2.3.12	n/a	n/a

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value	Default Value[2]	Eng. Units
A_ARG_TYPE_ListPolicyHandle	O	String (XML fragment)	See §2.3.13	n/a	n/a
PolicyVersion	O	ui4	See §2.3.14	n/a	n/a

[1] R = Required, O = Optional, X = Non-standard.

[2] Values listed in this column are required. To specify standard optional values or to delegate assignment of values to the vendor, you must reference a specific instance of an appropriate table below.

2.3.1. A_ARG_TYPE_TrafficDescriptor

This is a **string** containing an XML fragment. It contains information describing a traffic descriptor. Refer to the UPnP *QosManager:3* [QM] for details of this XML document using the namespace.

2.3.2. A_ARG_TYPE_TrafficPolicy

This is a **string** containing an XML fragment. It contains information describing TrafficPolicy information. The XML fragment in this argument MUST validate against the XML schema for `TrafficPolicy` in the XML namespace "`http://www.upnp.org/schemas/TrafficPolicy.xsd`" which is located at `http://www.upnp.org/schemas/qos/TrafficPolicy-v2.xsd`.

2.3.2.1. Description of fields in the TrafficPolicy structure

TrafficPolicy structure consists of the following seven elements:

- *AdmissionPolicy* is a required field and is set to "Enabled".
- *TrafficImportanceNumber* is a required field of type integer with values in the range of 0 through 7. This value conforms to the numbering scheme for traffic classes as described in IEEE 802.1D Annex G [IEEE 802.1D] and with additional traffic classes described in the *QosManager:3* [QM]. This value is used by *QosDevice* service(s) in the traffic's path to indicate what priority level to utilize when priority tagging the traffic's network packets.
- *UserImportanceNumber* is a required field of type integer with values in the range of 0 through 255. This is used by a *QoS Manager* for Preemption. This value is applicable only when the *AdmissionPolicy* is enabled. Note that a value of 255 is the highest user importance and 0 is the lowest.
- *PolicyHolderId* is an optional field. Refer to the *PolicyHolderId* field in the *TrafficDescriptor* structure in the *QosManager:3* [QM] for the definition and more details.
- *PolicyLastModified* is an optional field. Refer to the *PolicyLastModified* field in the *TrafficDescriptor* structure in the *QosManager:3* [QM] for the definition and more details.
- *PolicyModifyingUserName* is an optional field. Refer to the *PolicyModifyingUserName* field in the *TrafficDescriptor* structure in the *QosManager:3* [QM] for the definition and more details.
- *PolicyHolderConfigUrl* is an optional field. Refer to the *PolicyHolderConfigUrl* field in the *TrafficDescriptor* structure in the *QosManager:3* [QM] for the definition and more details.

2.3.2.2. Sample Argument XML String

Illustrated below are two separate examples of TrafficPolicy structure.

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<TrafficPolicy
  xmlns="http://www.upnp.org/schemas/TrafficPolicy.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/TrafficPolicy.xsd
    http://www.upnp.org/schemas/qos/TrafficPolicy-v2.xsd">
    <AdmissionPolicy>Enabled</AdmissionPolicy>
    <TrafficImportanceNumber>3</TrafficImportanceNumber>
    <UserImportanceNumber>128</UserImportanceNumber>
    <v2>
      <PolicyHolderId>uuid:2fac1234-31f8-11b4-a222-08002b34c003:urn:upnp-
org:serviceId:QosPolicyHolder-3a</PolicyHolderId>
      <PolicyLastModified>2004-11-26T15:03:23-08:00</PolicyLastModified>
      <PolicyModifyingUserName>Jimmy</PolicyModifyingUserName>
      <PolicyHolderConfigUrl>http://10.0.0.50/ConfigPolicy.html</PolicyHolderConfigUrl>
    </v2>
</TrafficPolicy>
```

Example 2:

```
<?xml version="1.0" encoding="UTF-8"?>
<TrafficPolicy
  xmlns="http://www.upnp.org/schemas/TrafficPolicy.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/TrafficPolicy.xsd
    http://www.upnp.org/schemas/qos/TrafficPolicy-v2.xsd">
    <AdmissionPolicy>Enabled</AdmissionPolicy>
    <TrafficImportanceNumber>5</TrafficImportanceNumber>
    <v2>
      <PolicyHolderId>uuid:2fac1234-31f8-11b4-a222-08002b34c003:urn:upnp-
org:serviceId:QosPolicyHolder-3b</PolicyHolderId>
      <PolicyHolderConfigUrl>http://10.0.0.50/ConfigPolicy.html</PolicyHolderConfigUrl>
    </v2>
</TrafficPolicy>
```

2.3.3. A_ARG_TYPE_ListOfTrafficDescriptors

This is a **string** containing an XML fragment. It contains information describing *ListOfTrafficDescriptors* structure. This structure contains a list of traffic descriptor each with the information for a traffic stream. Refer to the UPnP *QosManager:3* [QM] for details of this XML document using the namespace.

2.3.4. A_ARG_TYPE_ListOfTrafficPolicies

This is a **string** containing an XML fragment. It contains information describing the *ListOfTrafficPolicies* structure. This structure contains traffic policies for a list of traffic streams. The XML fragment in this argument MUST validate against the XML schema for *ListOfTrafficPolicies* in the XML namespace "`urn:schemas-upnp-org:qos:ListOfTrafficPolicies`" which is located at `http://www.upnp.org/schemas/qos/ListOfTrafficPolicies-v3.xsd`.

2.3.4.1. Description of fields in the ListOfTrafficPolicies structure

The *ListOfTrafficPolicies* structure consists of the following elements.

AdmissionPolicy: This is a required field. Refer to section 2.3.2.1 for details.

PolicyHolderId: This is a required field. Refer to the *PolicyHolderId* field in the *TrafficDescriptor* structure in the *QoS Manager:3* [QM] for the definition and more details.

PolicyHolderConfigUrl: This is a required field. Refer to the *PolicyHolderConfigUrl* field in the *TrafficDescriptor* structure in the *QoS Manager:3* [QM] for the definition and more details.

PolicyLastModified: This is a required field. Refer to the *PolicyLastModified* field in the *TrafficDescriptor* structure in the *QoS Manager:3* [QM] for the definition and more details.

PolicyModifyingUserName: This is a required field. Refer to the *PolicyModifyingUserName* field in the *TrafficDescriptor* structure in the *QoS Manager:3* [QM] for the definition and more details.

TdPolicy: This is a required structure. This contains traffic policies per TSPEC for different traffic descriptors identified by a *TrafficHandle*.

2.3.4.2. Description of fields in the TdPolicy structure

TrafficHandle: This is a required field. It identifies a traffic descriptor in the list. Refer to the *TrafficHandle* field in the *TrafficDescriptor* structure in the *QoS Manager:3* [QM] for the definition and more details.

TdPolicyPerTspec: This is a required structure. This contains traffic policies for different TSPECs of a traffic descriptor identified by the *TrafficHandle*

2.3.4.3. Description of fields in the TdPolicyPerTspec structure

TspecIndex: This is a required field. It identifies a TSPEC in the list of TSPECs. Refer to the *TspecIndex* field in the *TrafficDescriptor* structure in the *QoS Manager:3* [QM] for the definition and more details.

TrafficImportanceNumber: This is a required field. Refer to section 2.3.2.1 for details.

UserImportanceNumber: This is a required field. Refer to section 2.3.2.1 for details.

2.3.4.4. Sample Argument XML String

Illustrated below is an example of ListOfTrafficPolicies structure.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2008 sp1 (http://www.altova.com)-->
<ListOfTrafficPolicies xsi:schemaLocation="urn:schemas-upnp-org:qos:ListOfTrafficPolicies
ListOfTrafficPolicies-v3.xsd" xmlns="urn:schemas-upnp-org:qos:ListOfTrafficPolicies"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <TdPolicy>
                <TrafficHandle>TH1b4-a222-08002b34c0037f921234-723c-11b4</TrafficHandle>
                <TdPolicyPerTspec>
                        <TspecIndex>1</TspecIndex>
                        <TrafficImportanceNumber>7</TrafficImportanceNumber>
                        <UserImportanceNumber>0</UserImportanceNumber>
                </TdPolicyPerTspec>
        </TdPolicy>
        <TdPolicy>
                <TrafficHandle>TH712-a213-0807823742c0037f921234-723c-11b4</TrafficHandle>
                <TdPolicyPerTspec>
                        <TspecIndex>1</TspecIndex>
                        <TrafficImportanceNumber>7</TrafficImportanceNumber>
                        <UserImportanceNumber>1</UserImportanceNumber>
                </TdPolicyPerTspec>
        </TdPolicy>
        <AdmissionPolicy>Enabled</AdmissionPolicy>
        <PolicyHolderId>fac1234-31f8-11b4-a222-08002b34c003:urn:upnp-
org:serviceId:QosPolicyHolder-3a</PolicyHolderId>
        <PolicyHolderConfigUrl>http://192.168.1.2/QPH.html</PolicyHolderConfigUrl>
        <PolicyLastModified>2006-12-19T16:39:57-08:00</PolicyLastModified>
        <PolicyModifyingUserName>jpaine</PolicyModifyingUserName>
</ListOfTrafficPolicies>
```

2.3.5. A_ARG_TYPE_IsPreferred

This is a **boolean** variable. The value of **"1"** indicates that a *QosPolicyHolder* Service is selected as preferred. The value of **"0"** indicates that no *QosPolicyHolder* Service is set as preferred.

2.3.6. A_ARG_TYPE_QphPolicyRule

This is a **string** containing an XML fragment. It contains information describing the *QosPolicyHolder* Service policy details including a policy classifier and corresponding parameters. The XML fragment in this argument MUST validate against the XML schema for *QphPolicyRule* in the XML namespace "http://www.upnp.org/schemas/QphPolicyRule.xsd" which is located at http://www.upnp.org/schemas/qos/QphPolicyRule-v3.xsd

2.3.6.1. Description of A_ARG_TYPE_QphPolicyRule

The *QphPolicyRule* structure consists of two structures: the *QphPolicyClassifier* and the *QphPolicyParameter*, whose descriptions are provided below.

QphPolicyClassifier: This structure contains fields/information that will be used for classifying a traffic stream to which QoS policy will be applied.

QphPolicyParameter: This structure contains the actual policy parameters (*TIN* and *IN*) applied for a traffic stream matching a classifier.

2.3.6.1.1. Classifying a traffic stream

QphPolicyClassifier consists of a number of optional fields to test whether a traffic stream gets classified by this rule. A field that is absent satisfies all tests for the corresponding field in the TrafficDescriptor of the traffic stream. A traffic stream matches a *QphPolicyRule* if and only if all present fields of *QphPolicyClassifier* satisfy all tests for the corresponding fields in the TrafficDescriptor of the traffic stream.

In this section we provide information on how to interpret classification by certain types.

2.3.6.1.1.1.Integer value

An upper and lower boundary can be provided for any field containing an integer value. A field containing an integer value satisfies the provided boundaries if and only if the integer value is lower or equal to the upper boundary and the integer value is higher or equal to the lower boundary. If the upper boundary isn't provided, the upper boundary test always succeeds. If the lower boundary isn't provided, the lower boundary test always succeeds.

2.3.6.1.1.2.String value

A field that contains a **string** value satisfies the test when the string value matches the corresponding field in the TrafficDescriptor under test or when the field is absent.

2.3.6.1.1.3.IP address

An upper and lower boundary can be provided for any field containing an IP address. An IP address satisfies the provided boundaries if and only if every octet is within the bounds provided for that octet (see section 2.3.6.1.1.1). If the upper boundary isn't provided, the upper boundary defaults to the IP address of all ones (255.255.255.255 for IPv4, ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff for IPv6). If the lower boundary isn't provided, the lower boundary defaults to the IP address of all zeroes (0.0.0.0 for IPv4, :: for IPv6).

2.3.6.1.1.4.TrafficId

Refer to section 2.2.2.2.2 of [QM] for the definition of *TrafficId*.

An upper and lower boundary can be provided for any field containing a *TrafficId*. A *TrafficId* satisfies the provided boundaries if and only if all present fields are within the bounds for the individual fields. If the upper boundary isn't provided, the upper boundary defaults to a *TrafficId* where default upper boundary values apply for all fields (refer to section 2.3.6.1.1.1 and section 2.3.6.1.1.3). If the lower boundary isn't provided, the lower boundary defaults to a TrafficId where default lower boundary values apply for all fields (refer to section 2.3.6.1.1.1 and section 2.3.6.1.1.3).

2.3.6.1.1.5.Tspec

Refer to section 2.2.2.2.3.1 of [QM] for the definition of Tspec.

An upper and lower boundary can be provided for any field containing a Tspec. A Tspec satisfies the provided boundaries if and only if all present fields are within the bounds for the individual fields. If the upper boundary isn't provided, the upper boundary defaults to a Tspec where default upper boundary values apply for all fields (refer to section 2.3.6.1.1.1 and section 2.3.6.1.1.3). If the lower boundary isn't provided, the lower boundary defaults to a Tspec where default lower boundary values apply for all fields (refer to section 2.3.6.1.1.1 and section 2.3.6.1.1.3)

2.3.6.1.2. Description of fields in the QphPolicyClassifier structure

TrafficIdUpLimit: Refer to section 2.3.6.1.1.4 for more information on this variable. This optional field provides an upper boundary for the TrafficId of the stream under test.

TrafficIdLowLimit: Refer to section 2.3.6.1.1.4 for more information on this variable. This optional field provides a lower boundary for the TrafficId of the stream under test.

TspecUpLimit: Refer to section 2.3.6.1.1.5 for more information on this variable. This optional field provides an upper boundary for the TSPEC indicated by the ActiveTspecIndex of the stream under test.

TspecLowLimit: Refer to section 2.3.6.1.1.5 for more information on this variable. This optional field provides a lower boundary for the TSPEC indicated by the ActiveTspecIndex of the stream under test.

QosBoundarySourceAddressUpLimit: Refer to *QosBoundarySourceAddress* definition in section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.3 for more information on this variable. This opiotnal field provides an upper boundary for *QosBoundarySourceAddress* of the stream under test.

QosBoundarySourceAddressLowLimit: Refer to *QosBoundarySourceAddress* definition in section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.3 for more information on this variable. This OPTIONAL field provides a lower boundary for *QosBoundarySourceAddress* of the stream under test.

QosBoundaryDestinationAddressUpLimit: Refer to *QosBoundaryDestinationAddress* definition in section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.3 for more information on this variable. This optional field provides an upper boundary for *QosBoundaryDestinationAddress* of the stream under test.

QosBoundaryDestinationAddressLowLimit: Refer to *QosBoundaryDestinationAddress* definition in section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.3 for more information on this variable. This optional field provides a lower boundary for *QosBoundaryDestinationAddress* of the stream under test.

UserName: Refer to section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.2 for more information on this variable. This is an optional field.

CpName: Refer to section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.2 for more information on this variable. This is an optional field.

VendorApplicationName: Refer to section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.2 for more information on this variable. This is an optional field.

PortName: Refer to section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.2 for more information on this variable. This is an optional field.

ServiceProviderServiceName: Refer to section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.2 for more information on this variable. This is an optional field.

TrafficLeaseTimeUpLimit: Refer to *TrafficLeaseTime* definition in section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.1 for more information on this variable. This optional field provides an upper boundary for the *TrafficLeaseTime* of the stream under test.

TrafficLeaseTimeLowLimit: Refer to *TrafficLeaseTime* definition in section 2.2.2.2 of [QM] for definition of this variable. Refer to section 2.3.6.1.1.1 for more information on this variable. This optional field provides a lower boundary for the *TrafficLeaseTime* of the stream under test.

Critical: Refer to section 2.2.2.2 of [QM] for definition of this variable. This optional field matches if it equals the value of *Critical* in the TrafficDescriptor of the traffic stream under test or if this field is absent.

StartTime: This optional variable is of type unsigned integer and contains the scheduled start time of the validity interval of this *QphPolicyRule* which is repeated every week. Time is specified in seconds and it ranges from 0 – 604800 (to cover 7 days of week). Time 0 is Sunday 12 midnight (start of Sunday). Refer to section 2.3.6.1.1.1 for more information on this variable.

EndTime: This optional variable is of type unsigned integer and contains the scheduled end time of the validity interval of this *QphPolicyRule* which is repeated every week. Time is specified in seconds and it ranges from 0 – 604800 (to cover 7 days of week). Time 0 is Sunday 12 midnight (start of Sunday). *StartTime* MUST be less than *EndTime*. Refer to section 2.3.6.1.1.1 for more information on this variable.

2.3.6.1.3. Description of fields in the QphPolicyParameter structure

TrafficImportanceNumber (*TIN*): Refer to section 1.4.2.1

ImportanceNumber (*IN*): This is an unsigned integer value in the range of 0 through 255. This is used by a Control Point to indicate the static priority of a stream matching the *QphPolicyClassifier*. During the preemption process, this information is used by *QosPolicyHolder* Service to derive the relative *UserImportanceNumber(UIN)* of streams.

2.3.6.2. Sample argument XML string

Illustrated below is an example of *QphPolicyRule* structure.

```
<?xml version="1.0" encoding="UTF-8"?>
<QphPolicyRule xmlns="urn:schemas-upnp-org:qos:QphPolicyRule.xsd"
xmlns:td="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd">
        <QphPolicyClassifier>
                <TrafficIdUpLimit>
                        <td:SourceAddress>
                                <td:Ipv4>1.2.3.4</td:Ipv4>
                        </td:SourceAddress>
                        <td:SourcePort>47420</td:SourcePort>
                        <td:DestinationAddress>
                                <td:Ipv4>1.2.3.4</td:Ipv4>
                        </td:DestinationAddress>
                        <td:DestinationPort>46807</td:DestinationPort>
                        <td:IpProtocol>161</td:IpProtocol>
                </TrafficIdUpLimit>
```

```
                <TrafficIdLowLimit>
                        <td:SourceAddress>
                                <td:Ipv4>2.199.252.164</td:Ipv4>
                        </td:SourceAddress>
                        <td:SourcePort>45521</td:SourcePort>
                        <td:DestinationAddress>
                                <td:Ipv4>1.2.3.4</td:Ipv4>
                        </td:DestinationAddress>
                        <td:DestinationPort>9932</td:DestinationPort>
                        <td:IpProtocol>205</td:IpProtocol>
                </TrafficIdLowLimit>
                <UserName>SiZnk3</UserName>
                <CpName>fhnnUjy70vPT3CdEotrbuMVVqjKlw6u2oug6WM</CpName>
                <VendorApplicationName>poLkLGv</VendorApplicationName>
                <PortName>F</PortName>
                <ServiceProviderServiceName>PyJNLvgq7AfoP_w</ServiceProviderServiceName>
                <TrafficLeaseTimeUpLimit>8615</TrafficLeaseTimeUpLimit>
                <TrafficLeaseTimeLowLimit>1738</TrafficLeaseTimeLowLimit>
                <Critical>false</Critical>
                <StartTime>238</StartTime>
                <EndTime>474652</EndTime>
        </QphPolicyClassifier>
        <QphPolicyParameter>
                <TrafficImportanceNumber>0</TrafficImportanceNumber>
                <ImportanceNumber>0</ImportanceNumber>
        </QphPolicyParameter>
</QphPolicyRule>
```

2.3.7. A_ARG_TYPE_ListOfQphPolicyRule

This is a **string** containing an XML fragment. It contains information listing one or more *QosPolicyHolder* Service policy rules. The XML fragment in this argument MUST validate against the XML schema for *ListOfQphPolicyRule* in the XML namespace.
"http://www.upnp.org/schemas/ListOfQphPolicyRule.xsd" which is located at http://www.upnp.org/schemas/qos/ListOfQphPolicyRule-v3.xsd

2.3.7.1. Description of A_ARG_TYPE_ListOfQphPolicyRule

NumberQphPolicyRule: This is an unsigned integer value containing the number of *QphPolicyRule* instances included in *ListOfQphPolicyRule*.

QphPolicyList: This is an XML structure that contains information on a single policy rule. *A_ARG_TYPE_ListOfQphRule* MUST contain exactly *NumberQphPolicyRule* instances of *QphPolicyList*.

2.3.7.1.1. QphPolicyList

This XML structure identifies a rule and its position. It contains the following fields

QphPolicyRule: This required filed of type *A_ARG_TYPE_QphPolicyRule*. It contains the policy rule.

PolicyHandle: This required field is of type *A_ARG_TYPE_PolicyHandle*. It contains the *PolicyHandle* of the *QphPolicyRule* included in *ListOfQphPolicyRule*.

Position: This required field is of type *A_ARG_TYPE_Position*. It represents the position of the *QphPolicyRule* in the *QosPolicyHolder* Service policy database.

2.3.7.2. Sample argument XML string

Illustrated below is an example of *ListOfQphPolicyRule* structure.

```
<?xml version="1.0" encoding="UTF-8"?>
<ListOfQphPolicyRule xmlns="urn:schemas-upnp-org:qos:ListOfQphPolicyRule.xsd"
xmlns:td="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd" xmlns:pr="urn:schemas-upnp-
org:qos:QphPolicyRule.xsd">
```

```
        <NumberQphPolicyRule>1</NumberQphPolicyRule>
        <QphPolicyList>
                <QphPolicyRule>
                        <pr:QphPolicyClassifier>
                                <pr:TrafficIdUpLimit>
                                        <td:SourceAddress>
                                                <td:Ipv4>1.2.3.4</td:Ipv4>
                                        </td:SourceAddress>
                                        <td:SourcePort>47420</td:SourcePort>
                                        <td:DestinationAddress>
                                                <td:Ipv4>1.2.3.4</td:Ipv4>
                                        </td:DestinationAddress>
                                        <td:DestinationPort>46807</td:DestinationPort>
                                        <td:IpProtocol>161</td:IpProtocol>
                                </pr:TrafficIdUpLimit>
                                <pr:TrafficIdLowLimit>
                                        <td:SourceAddress>
                                                <td:Ipv4>2.199.252.164</td:Ipv4>
                                        </td:SourceAddress>
                                        <td:SourcePort>45521</td:SourcePort>
                                        <td:DestinationAddress>
                                                <td:Ipv4>1.2.3.4</td:Ipv4>
                                        </td:DestinationAddress>
                                        <td:DestinationPort>9932</td:DestinationPort>
                                        <td:IpProtocol>205</td:IpProtocol>
                                </pr:TrafficIdLowLimit>
                                <pr:UserName>SiZnk3</pr:UserName>
                                <pr:CpName>fhnnUjy70vPT3CdEotrbuMVVqjKlw6u2oug6WM</pr:CpName>
                                <pr:VendorApplicationName>poLkLGv</pr:VendorApplicationName>
                                <pr:PortName>F</pr:PortName>
                        <pr:ServiceProviderServiceName>PyJNLvgq7AfoP_w</pr:ServiceProviderServiceName>
                                <pr:TrafficLeaseTimeUpLimit>8615</pr:TrafficLeaseTimeUpLimit>
                                <pr:TrafficLeaseTimeLowLimit>1738</pr:TrafficLeaseTimeLowLimit>
                                <pr:Critical>false</pr:Critical>
                                <pr:StartTime>555238</pr:StartTime>
                                <pr:EndTime>474652</pr:EndTime>
                        </pr:QphPolicyClassifier>
                        <pr:QphPolicyParameter>
                                <pr:TrafficImportanceNumber>0</pr:TrafficImportanceNumber>
                                <pr:ImportanceNumber>0</pr:ImportanceNumber>
                        </pr:QphPolicyParameter>
                </QphPolicyRule>
                <PolicyHandle>0</PolicyHandle>
                <Position>0</Position>
        </QphPolicyList>
</ListOfQphPolicyRule>
```

2.3.8. A_ARG_TYPE_Position

This is an unsigned integer (ui4). It is used by the Control Point to indicate the position where the new policy rule MUST be added in *QosPolicyHolder* Service database. Position is used to determine the order in which traffic descriptors are compared against classifiers. The comparison is done in ascending order of position. The first match found is used. The first position has the value 0.

2.3.9. A_ARG_TYPE_TIN

This is an unsigned integer (ui4). Refer to the description of *TrafficImportanceNumber* (*TIN*) in section 2.3.2.1 for details.

2.3.10.A_ARG_TYPE_IN

This is an unsigned integer (ui4). Refer to the description of *ImportanceNumber* (*IN*) in section 2.3.6.1 for details

2.3.11.A_ARG_TYPE_ReasonCode

This is an unsigned integer (ui4). This variable is used by *QosPolicyHolder* Service to indicate the reason for not assigning the requested *IN* to the Control Point as part of *AddQphPolicy()* action. When the *QosPolicyHolder* Service successfully accepts the requested *IN*, **"0"** is returned as part of this variable. Different values of reason code are defined as part of action definition.

2.3.12.A_ARG_TYPE_PolicyHandle

This is an unsigned integer (ui4). This variable is used as identifier of the successfully added policy in *QosPolicyHolder* Service database. The value of this variable is defined by the *QosPolicyHolder* Service. The purpose of this variable is to uniquely identify a rule in the *QosPolicyHolder* database (*QphPolicyRule*).

2.3.13.A_ARG_TYPE_ListPolicyHandle

This is a **string** containing an XML fragment. It contains information listing zero or more *PolicyHandle* structures. The XML fragment in this argument MUST validate against the XML schema for *ListPolicyHandle* in the XML namespace "http://www.upnp.org/schemas/ListPolicyHandle.xsd" which is located at http://www.upnp.org/schemas/qos/ListPolicyHandle-v3.xsd

2.3.13.1. Description of A_ARG_TYPE_ListPolicyHandle

NumberOfPolicyHandle: This is an unsigned integer value containing the number of *PolicyHandle* instances included as part inside *ListPolicyHandle*. Value of 0 is used to indicate all policies stored in *QosPolicyHolder* Service.

PolicyHandle: There can be zero or more instances of *PolicyHandle*. Number of instances MUST be equal to *NumberOfPolicyHandle*.

2.3.13.2. Sample argument XML string

Illustrated below is an example of *ListPolicyHandle* structure.

```
<?xml version="1.0" encoding="UTF-8"?>
<ListPolicyHandle>
 <NumberPolicyHandle>1</NumberPolicyHandle>

  <PolicyHandle>100</PolicyHandle>
</ListPolicyHandle>
```

2.3.14.PolicyVersion

PolicyVersion is variable of type unsigned integer (ui4) and the *QosPolicyHolder* Service stores the value of this variable. This variable is incremented each time there is a successful change to *QosPolicyHolder* Service *QphPolicyRule* database.

2.3.15.Relationships Between State Variables

There are no relationships between any of the state variables for this service.

2.4. Eventing and Moderation

Table 2-2: Event Moderation

Variable Name	Evented	Moderated Event	Max Event Rate[1]	Logical Combination	Min Delta per Event[2]
A_ARG_TYPE_TrafficDescriptor	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_TrafficPolicy	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_ListOfTrafficDescriptors	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_ListOfTrafficPolicies	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_IsPreferred	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_QphPolicyRule	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_ListOfQphPolicyRule	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_Position	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_TIN	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_IN	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_ReasonCode	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_PolicyHandle	NO	n/a	n/a	n/a	n/a
A_ARG_TYPE_ListPolicyHandle	NO	n/a	n/a	n/a	n/a
PolicyVersion	YES	YES	2	n/a	n/a

[1] Determined by N, where Rate = (Event)/(N secs).
[2] (N) * (allowedValueRange Step)

2.4.1. Event Model

PolicyVersion **:** The state variable is optional and MUST be evented, when implemented. If the *SetAsPreferred()* action is implemented, this event MUST be implemented. This is a *QosPolicyHolder* Service variable that is evented to allow interested parties to monitor the *QosPolicyHolder* Service state. This variable is incremented each time there is a successful policy change in *QosPolicyHolder* Service database.

2.5. Actions

The *QosPolicyHolder* Service is added to a UPnP device that will manage the QoS policy for the entire network..

Immediately following Table 2-3: Actions is detailed information about the actions listed in this table, including short descriptions of the actions, the effects of the actions on state variables, and error codes defined by the actions.

Table 2-3: Actions

Name	Req. or Opt. [1]
GetTrafficPolicy ()	R

Name	Req. or Opt. [1]
GetListOfTrafficPolicies ()	R
SetAsPreferred ()	O
AddQphPolicy( )	O
RemoveQphPolicy( )	O
RetrieveQphPolicy( )	O
GetPolicyVersion( )	O

[1] R = Required, O = Optional, X = Non-standard.

All four actions *AddQphPolicy()*, *RemoveQphPolicy()*, *RetrieveQphPolicy()* *and* *GetPolicyVersion()* *and the event* *PolicyVersion* MUST be implemented together.

If *SetAsPreferred()* action is implemented, *AddQphPolicy()*, *RemoveQphPolicy()*, *RetrieveQphPolicy()* *and* *GetPolicyVersion()* actions MUST be implemented.

2.5.1. GetTrafficPolicy

This action is invoked to determine the traffic policy for a requested traffic stream. The *QosPolicyHolder* Service returns the traffic policy for the supplied TrafficDescriptor in the *OutputTrafficPolicy* output argument.

2.5.1.1. Arguments

Table 2-4: Arguments for GetTrafficPolicy

Argument	Direction	relatedStateVariable
RequestedTrafficDescriptor	IN	A_ARG_TYPE_TrafficDescriptor
OutputTrafficPolicy	OUT	A_ARG_TYPE_TrafficPolicy

The *RequestedTrafficDescriptor* input argument contains information for the traffic stream requiring some level of QoS. This action will determine the traffic policy for this requested traffic stream.

The *OutputTrafficPolicy* output argument contains the traffic policy for the requested traffic stream.

2.5.1.2. Service requirements

If the input parameter is not a TrafficDescriptor (including non-XML input), *QosPolicyHolder* Service MUST return error code 799.

If a *QoS Manager* does not supply *ActiveTspecIndex* in *RequestedTrafficDescriptor* to *QosPolicyHolder* Service, the *QosPolicyHolder* Service MUST return error code 723.

If a *QoS Manager* does not supply a *TrafficHandle*, or if *TrafficHandle* has a NULL value, in the *RequestedTrafficDescriptor* to *QosPolicyHolder* Service, the *QosPolicyHolder* Service MUST return error code 700.

The *QosPolicyHolder* Service MUST only return the traffic policy for the Tspec indicated by the ActiveTspecIndex.

In the *RequestedTrafficDescriptor* to the *QosPolicyHolder* Service, the *ActiveTspecIndex* indicates the Tspec for which *TrafficPolicy* is needed. *ActiveTspecIndex* MUST be one of the TspecIndex values in the AvailableOrderedTspecList. If not, the *QosPolicyHolder* Service MUST return error code 720.

If the *QoS Manager* provides any elements of TrafficPolicy in the input *RequestedTrafficDescriptor*, then the *QosPolicyHolder* Service MUST ignore those elements and their values.

If a *PolicyHolderId* is specified by the requesting Control Point in the *RequestedTrafficDescriptor* supplied to the *QosManager*, and if it is not the *PolicyHolderId* of this *QosPolicyHolder* service, the *QosPolicyHolder* MUST return error code 781.

The *QosPolicyHolder* Service MUST set the *AdmissionPolicy* element to "Enabled" in the TrafficPolicy returned for a traffic descriptor.

2.5.1.3. CP requirements when calling the action

A Control Point (e.g. *QoS Manager*) MUST supply the *TrafficHandle* in the *RequestedTrafficDescriptor* input argument to *QosPolicyHolder* Service when calling the *GetTrafficPolicy()* acion.

A Control Point (e.g. QoS Manager) MUST supply an *ActiveTspecIndex* that is one of the *TspecIndex* values in the *AvailableOrderedTspecList* in the *RequestedTrafficDescriptor* input argument to *QosPolicyHolder* Service when calling the *QPH:GetTrafficPolicy()* action.

2.5.1.4. Dependency on State (if any)

There is no dependency on the current state of this service when this action gets executed

2.5.1.5. Effect on State (if any)

There is no effect on the state of this service when this action gets executed.

2.5.1.6. Errors

Table 2-5: Error Codes for GetTrafficPolicy

errorCode	errorDescription	Description
700	Traffic Handle missing or empty	Traffic Handle must be filled in as input to this action.
720	ActiveTspecIndex is not a TspecIndex	
723	ActiveTspecIndex missing	Valid ActiveTspecIndex must be filled in as input to this action.
781	PolicyHolderId does not match	This QosPolicyHolder Service is not the same as the PolicyHolderId specified in A_ARG_TYPE_TrafficPolicy.
799	Invalid Input Parameter	The input parameter supplied to Action is invalid.

2.5.2. GetListOfTrafficPolicies

This action is used to determine traffic policies for multiple traffic descriptors using a single action. The *QoS Manager* typically invokes this action to obtain QoS policies for a set of Blocking Streams during the

preemption process while trying to admit a new stream [QM]. If a TrafficDescriptor lists multiple Tspecs, the *QosPolicyHolder* Service returns policies for the individual Tspecs of the TrafficDescriptor.

2.5.2.1. Arguments

Table 2-6: Arguments for GetListOfTrafficPolicies

Argument	Direction	relatedStateVariable
ListOfTrafficDescriptors	IN	A_ARG_TYPE_ListOfTrafficDescriptor
ListOfTrafficPolicies	OUT	A_ARG_TYPE_ListOfTrafficPolicies

The *ListOfTrafficDescriptors* input argument contains information for the list of traffic streams requiring some level of QoS. This action determines traffic policies for the requested list of traffic streams.

The *ListOfTrafficPolicies* output argument contains traffic policies for the requested list of traffic streams. The *ListOfTrafficPolicies* argument lists traffic policy for each individual TrafficDescriptor in the *ListOfTrafficDescriptors* input argument. If an individual TrafficDescriptor in the *ListOfTrafficDescriptors* input argument contains multiple Tspecs, the *ListOfTrafficPolicies* argument will have multiple traffic policies for the TrafficDescriptor, each corresponding to an individual Tspec in the TrafficDescriptor. The higher the value of the UIN, the more important the associated traffic stream. The scope of UINs returned as a part of this action is constrained to this instance of the invocation of this action. The UINs returned as a part of this action MUST be unique within this list.

2.5.2.2. Service requirements

If the input parameter is not a *ListOfTrafficDescriptors* (including non-XML input), *QosPolicyHolder* Service MUST return error code 799.

If a *QoS Manager* does not supply a *TrafficHandle*, or if *TrafficHandle* has a NULL value, in at least one of the TrafficDescriptors supplied to the *QosPolicyHolder* Service, the *QosPolicyHolder* Service MUST return error code 700.

The *QosPolicyHolder* Service MUST ignore all elements of the TrafficPolicy in the input TrafficDescriptor.

If a TrafficDescriptor in the *ListOfTrafficDescriptors* contains multiple Tspecs, the *QosPolicyHolder* Service MUST return traffic policy for each Tspec in the *ListOfTrafficPolicies* output argument.

The *QosPolicyHolder* Service MUST set the *AdmissionPolicy* element to “Enabled” in the TrafficPolicy returned for a list of traffic descriptors.

2.5.2.3. CP requirements when calling the action

A Control Point (e.g. *QoS Manager*) MUST supply the TrafficHandle in each of TrafficDescriptors in the *ListOfTrafficDescriptors* input argument to the *QosPolicyHolder* Service when calling the *GetListOfTrafficPolicies()* action.

2.5.2.4. Dependency on State (if any)

There is no dependency on the current state of this service when this action gets executed.

2.5.2.5. Effect on State (if any)

There is no effect on the state of this service when this action gets executed.

2.5.2.6. Errors

Table 2-7: Error Codes for GetListOfTrafficPolicies

errorCode	errorDescription	Description
700	Traffic Handle missing or empty	Every Traffic Handle must be filled in as input to this action.
799	Invalid Input Parameter	The input parameter supplied to Action is invalid.

2.5.3. SetAsPreferred

This is an optional action. This action is invoked on a *QosPolicyHolder* Service to either set the *QosPolicyHolder* Service as the preferred *QosPolicyHolder* Service on the network or to indicate that no *QosPolicyHolder* Service is preferred on the network.

2.5.3.1. Arguments

Table 2-8: Arguments for SetAsPreferred

Argument	Direction	relatedStateVariable
SelectAsPreferred	IN	A_ARG_Type_IsPreferred

The input argument *SelectAsPreferred* is a **boolean** variable. The value of **"1"** indicates that the *QosPolicyHolder* Service is selected as the preferred *QosPolicyHolder* on the network or and value of **"0"** indicates that no *QosPolicyHolder* Service is preferred on the network.

In order to change the preferred *QosPolicyHolder* selection, it is recommended not to invoke the *SetAsPreferred()* action with the value of **"0"** for the *SelectAsPreferred* input argument on the preferred *QosPolicyHolder* Service prior to selecting a new *QosPolicyHolder* to be preferred. It is sufficient to invoke the *SetAsPreferred()* action on the new *QosPolicyHolder* with the value of **"1"** for the *SelectAsPreferred* input argument.

2.5.3.2. Service requirements

If the *QosPolicyHolder* Service determines that some other *QosPolicyHolder* Service has been selected as the preferred one on the network (as described in step 4 of Section 2.5.3.4 below), the *QosPolicyHolder* Service MUST return an error code of 730.

If the *QosPolicyHolder* Service determines that there is a collision on the network (as described in step 5 of Section 2.5.3.4 below) because multiple *QosPolicyHolder* Services are being set as preferred on the network at the same time, the *QosPolicyHolder* Service MUST return an error code of 731.

2.5.3.3. CP requirements when calling the action

None.

2.5.3.4. QoS Policy Holder Requirements

After the *SetAsPreferred()* action is invoked with *SelectAsPreferred* input argument equal to **"1"**, the *QoS Policy Holder* MUST perform the Preferred *QosPolicyHolder* Synchronization/Propagation Process as described below:

1. The *QoS Policy Holder* MUST determine the fields in the *PreferredQph* input argument for the *QD:SetPreferredQph()* action as follows:

    a. *PreferredQphId* = Its own *QosPolicyHolder* ServiceId;

    b. *QphPreferenceCount* = The highest value of *QphPreferenceCount* returned from all of the available v3 *QosDevice* Services, incremented by one.

    Note: The highest value of *QphPreferenceCount* is determined by comparing the *QphPreferenceCount* variable returned from all the v3 *QosDevice* Services as an output argument of *QD:SetPreferredQph().*

2. The *QoS Policy Holder* MUST invoke the *QD:SetPreferredQph()* action with the above values on each of the available *QosDevice:3* Services that implement this action.

3. If the *QD:SetPreferredQph()* action on all of the available *QosDevice* Services returns the *SetPreferredQphResults* with the value 0 indicating success, the *QoS Policy Holder* concludes that it has successfully established itself as the preferred *QosPolicyHolder* Service on the network. To maintain itself as the preferred *QosPolicyHolder* Service on the network, the *QoS Policy Holder* performs the following:

    a. The *QoS Policy Holder* MUST invoke *QD:SetPreferredQph()* action on every *QosDevice* Service that advertises itself on the network [DEVICE] and implements this action.

    b. Every time the *QosPolicyHolder* Service is rebooted, the *QoS Policy Holder* MUST invoke *QD:SetPreferredQph()* action on all the available *QosDevice* Services that implement this action.

    c. Anytime, if *QD:SetPreferredQph()* action does not return the *SetPreferredQphResults* with the value 0 indicating success, the *QoS Policy Holder* MUST follow the steps 4 or 5 identified below based on the value returned for *SetPreferredQphResults*.

4. If the *QD:SetPreferredQph()* action on any of the *QosDevice* Services returns the *SetPreferredQphResults* of 770 (QphPreferrenceCount mismatch) as defined in *QosDevice* Service [QoS DEV], the *QoS Policy Holder* concludes that the Control Point has selected some other *QosPolicyHolder* Service as the preferred one on the network and the *QoS Policy Holder* MUST not invoke *QD:SetPreferredQph()* actions on the remaining *QosDevice* Services.

5. If the *QD:SetPreferredQph()* action on any of the *QosDevice* Service returns the *SetPreferredQphResults* of 771 (Synchronization error) as defined in [QoS DEV], the *QoS Policy Holder* concludes that there is a collision on the network because multiple *QosPolicyHolder* Services are being set as preferred on the network at the same time and the *QoS Policy Holder* MUST not invoke *QD:SetPreferredQph()* actions on the remaining *QosDevice* Services.

After the *SetAsPreferred()* action is invoked with *SelectAsPreferred* input argument equal to **"0"**, the *QoS Policy Holder* MUST perform the following steps:

1. The *QoS Policy Holder* MUST determine the fields in the *PreferredQph* input argument as follows:

    a. *PreferredQphId*= null;

    b. *QphPreferenceCount*= Highest value of *QphPreferenceCount* state variable on the network incremented by one.

2. The *QoS Policy Holder* MUST invoke the *QD:SetPreferredQph()* action with the above values on each of the *QosDevice* Services on the network that implement this action.

3. If the *QD:SetPreferredQph()* action on any of the QosDevice Services returns the *SetPreferredQphResults* of 770 (QphPreferrenceCount mismatch) as defined in [QoS DEV], the *QoS Policy Holder* concludes that the user has selected some other *QosPolicyHolder* Service as the preferred one on the network and the *QoS Policy Holder* MUST not invoke *QD:SetPreferredQph()* actions on the remaining *QosDevice* Services.

4. If the *QD:SetPreferredQph()* action on any of the *QosDevice* Service returns the *SetPreferredQphResults* of 771 (Synchronization error) as defined in [QoS DEV], the *QoS Policy Holder* concludes that there is a collision on the network because multiple *QosPolicyHolder* Services are being set as preferred on the network at the same time and the *QoS Policy Holder* MUST not invoke *QD:SetPreferredQph()* actions on the remaining *QosDevice* Services.

2.5.3.5. Dependency on State (if any)

There is no dependency on the current state of this service when this action gets executed.

2.5.3.6. Effect on State (if any)

There is no effect on the state of this service when this action gets executed..

2.5.3.7. Errors

Table 2-9: Error Codes for SetAsPreferred

errorCode	errorDescription	Description
730	Preferred QPH Failure	Some other Policy Holder is selected as the preferred one on the network
731	Preferred QPH Sync Error	Multiple Policy Holders are being set as preferred at the same time

2.5.4. AddQphPolicy

This is an optional action. This action is invoked on a *QosPolicyHolder* Service to add a new policy rule in the *QosPolicyHolder* Service database. The *Position* input argument is used by the *QosPolicyHolder* Service to insert the policy rule at the given position and shifts the existing policy rules past the given position.

2.5.4.1. Arguments

Table 2-10: Arguments for *AddQphPolicy*

Argument	Direction	relatedStateVariable
QphPolicyRule	IN	A_ARG_TYPE_QphPolicyRule
Position	IN	A_ARG_TYPE_Position
PolicyVersion	IN	PolicyVersion
TrafficImportanceNumber	OUT	A_ARG_TYPE_ TIN

ImportanceNumber	OUT	A_ARG_TYPE_ IN
ReasonCode	OUT	A_ARG_TYPE_ReasonCode
PolicyHandle	OUT	A_ARG_TYPE_PolicyHandle

2.5.4.2. Service Requirements

QosPolicyHolder Service MUST verify the *PolicyVersion* provided as part of input matches with the current *PolicyVersion*. Otherwise, the *QosPolicyHolder* Service MUST return an error code of 783.

The *QosPolicyHolder* Service MUST check if the *Position* provided as part of input is within the range of policies in the database. The *Position* input argument MUST be used by the *QosPolicyHolder* Service to insert the policy rule at the given position. If the *QosPolicyHolder* Service determines that the value of *Position* is out of range, it MUST append the new rule to the *QosPolicyHolder* Service database.

2.5.4.3. CP requirement when calling this action

Control Point MUST provide *QphPolicyRule*, *Position* and current *PolicyVersion* as input while calling this action.

2.5.4.4. QoS Policy Holder Requirements

None

2.5.4.5. Dependency on State (if any)

There is no dependency on the current state of this service when this action gets executed.

2.5.4.6. Effect on State (if any)

QosPolicyHolder Service *QphPolicyRule* database is affected by this action.

2.5.4.7. Reason Code

Table 2-11: Reason code for AddQphPolicy

ReasonCode	ReasonDescription	Description
0	Success	Requested IN is successfully assigned
1	IN not assignable	QosPolicyHolder Service could not assign the requested IN.
100-200	Reserved	Reserved for vendor-specific reason codes

2.5.4.8. Errors

Table 2-12: Error code for AddQphPolicy

errorCode	errorDescription	Description
783	Incorrect PolicyVersion	If QosPolicyHolder Service finds that *PolicyVersion* variable is not valid

2.5.5. RemoveQphPolicy

This is an optional action. This action is invoked on a *QosPolicyHolder* Service to remove a policy rule from *QosPolicyHolder* Service database.

2.5.5.1. Arguments

Table 2-13: Arguments for RemoveQphPolicy

Argument	Direction	relatedStateVariable
PolicyHandle	IN	A_ARG_TYPE_ PolicyHandle

2.5.5.2. Service Requirements

QosPolicyHolder Service MUST check the existence of the *PolicyHandle* in the database provided as part of input. If the *QosPolicyHolder* Service determines that *PolicyHandle* provided as part of input doesn't exist, the *QosPolicyHolder* Service MUST return an error code of 784. If it is valid, the *QosPolicyHolder* Service MUST delete the policy rule.

2.5.5.3. CP requirement when calling this action

Control Point MUST provide *PolicyHandle* as input while calling this action.

2.5.5.4. QoS Policy Holder Requirements

None

2.5.5.5. Dependency on State (if any)

There is no dependency on the current state of this service when this action gets executed.

2.5.5.6. Effect on State (if any)

QosPolicyHolder Service *QosPolicyRule* database is affected by this action.

2.5.5.7. Errors

Table 2-14: Error code for RemoveQphPolicy

errorCode	errorDescription	Description
784	Invalid PolicyHandle	During the validation, if QosPolicyHolder Service finds that *PolicyHandle* is invalid

2.5.6. RetrieveQphPolicy

This is an optional action. This action is invoked on a *QosPolicyHolder* Service to retrieve one or more policy rules from *QosPolicyHolder* Service database.

2.5.6.1. Arguments

Table 2-15: Arguments for RetrieveQphPolicy

Argument	Direction	relatedStateVariable
ListPolicyHandle	IN	A_ARG_TYPE_ListPolicyHandle
PolicyVersion	OUT	PolicyVersion
ListOfQphPolicyRule	OUT	A_ARG_TYPE_ListOfQphPolicyRule

2.5.6.2. Service Requirements

Upon receiving this action,

a) If *ListPolicyHandle* field indicates that all policies are requested, the *QosPolicyHolder* Service MUST return all the policies stored in the *QosPolicyHolder* Service. Refer to definition of *NumberOfPolicyHandle* in section 2.3.13.1 for details on how to request all the policies.
b) If *ListPolicyHandle* has zero or more occurrences of *PolicyHandle*, the the *QosPolicyHolder* Service MUST return the corresponding policies. If a *PolicyHandle* doesn't exist in the *QosPolicyHolder* Service, it MUST NOT return *QphPolicyList* for that *PolicyHandle* in the *ListOfQphPolicyRule* ouput argument.

2.5.6.3. CP requirement when calling this action

Control Point MUST provide *ListPolicyHandle* as input while calling this action.

2.5.6.4. QoS Policy Holder Requirements

None

2.5.6.5. Dependency on State (if any)

There is no dependency on the current state of this service when this action gets executed.

2.5.6.6. Effect on State (if any)

There is no effect on the current state of this service when this action gets executed.

2.5.6.7. Errors

None

2.5.7. GetPolicyVersion

This is an optional action. This action is invoked on a *QosPolicyHolder* Service to get the *PolicyVersion* state variable.

2.5.7.1. Arguments

Table 2-16: Arguments for *GetPolicyVersion*

Argument	Direction	relatedStateVariable
PolicyVersion	OUT	PolicyVersion

2.5.7.2. Service Requirements

QosPolicyHolder Service MUST return the current *PolicyVersion* as output of this action.

2.5.7.3. CP requirement when calling this action

None

2.5.7.4. QoS Policy Holder Requirements

None

2.5.7.5. Dependency on State (if any)

There is no dependency on the current state of this service when this action gets executed.

2.5.7.6. Effect on State (if any)

There is no dependency on the current state of this service when this action gets executed.

2.5.7.7. Errors

None

2.5.8. Non-Standard Actions Implemented by a UPnP Vendor

To facilitate certification, non-standard actions implemented by UPnP vendors should be included in this service template. The UPnP Device Architecture [DEVICE] specifies naming requirements for non-standard actions (see the section on Description).

2.5.9. Error Code Summary

The following table lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error MUST be returned. These common error codes are defined in the UPnP Device Architecture [DEVICE] and other Technical Committee documents.

Table 2-17: Common Error Codes

errorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control
600-699	TBD	See UPnP Device Architecture section on Control

errorCode	errorDescription	Description
700	Traffic Handle missing or empty	Traffic Handle must be filled in as input to this action.
720	ActiveTspecIndex is not a TspecIndex	
723	ActiveTspecIndex missing	Valid ActiveTspecIndex must be filled in as input to this action.
730	Preferred QPH Failure	Some other Policy Holder is selected as the preferred one on the network
731	Preferred QPH Sync Error	Multiple Policy Holders are being set as preferred at the same time
781	PolicyHolderId does not match	This QosPolicyHolder Service is not the same as the PolicyHolderId specified in A_ARG_TYPE_TrafficPolicy.
783	Incorrect PolicyVersion	If QosPolicyHolder Service finds that *PolicyVersion* variable is not valid
784	Invalid PolicyHandle	During the validation, if QosPolicyHolder Service finds that *PolicyHandle* is invalid
799	Invalid Input Parameter	The input parameter supplied to Action is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

3. Theory of Operation (Informative)

The sole purpose of this service is to provide an interface to a network entity that will host QoS policies. It is necessary for such an entity within the UPnP network to provide some traffic policy values for any network traffic stream that wants to have QoS that is better than default. This is accomplished by this service.

3.1. Retrieving Policies

QoS Policies for a traffic stream or a set of traffic steams are obtained by invoking either the *GetTrafficPolicy()* or the *GetListOfTrafficPolicies()* action, respectively, on the *QosPolicyHolder* Service.

The *GetTrafficPolicy()* action will accept as input a traffic descriptor, defined as an XML string in [QM], which contains all the information needed to generate QoS traffic policy values for this traffic stream. The *QoS Manager* is required to invoke the *GetTrafficPolicy()* action when it is attempting to establish QoS for a Prioritized Stream and for a Hybrid Stream. The *QoS Manager* optionally invokes the *GetTrafficPolicy()* action when it is attempting to establish QoS for a Parameterized Stream. If the input traffic descriptor contains multiple Tspecs, the *GetTrafficPolicy()* action returns traffic policy only for the TSPEC identified by the *ActiveTspecIndex* provided by the Control Point. The QoS traffic policy values, as an XML string, that are returned by the *GetTrafficPolicy()* action for a requested traffic descriptor are summarized in Section 2.3.2.1.

The *GetListOfTrafficPolicies()* action will accept as input a list of traffic descriptors, defined as an XML string in [QM], which contains all the information needed to generate QoS traffic policy values for a list of traffic streams. The Control Point (i.e. *QoS Manager*) typically invokes *GetListOfTrafficPolicies()* action to obtain the relative importance of a set of streams that it intends to preempt while attempting to admit a new stream. The *QosPolicyHolder* Service returns traffic policy for each traffic descriptor. If a traffic descriptor contains multiple Tspecs, the *QosPolicyHolder* Service returns a traffic policy for each Tspec in the traffic descriptor. The QoS traffic policy values returned by the *GetListOfTrafficPolicies()* action for a requested list of traffic descriptor are summarized in Section 2.3.4.1.

From version 3.0 onwards, *QosPolicyHolder* Service is required to enable *AdmissionPolicy*. For admitted Parameterized Streams, the stream will be allocated requested network resources by the UPnP-QoS framework. Network preemption decisions, when *AdmissionPolicy* is "Enabled", are made by using the *UserImportanceNumber* (UIN).

The *QosPolicyHolder* Service takes into consideration various fields in the traffic descriptor of a particular traffic stream to determine its traffic policy values (TIN and UIN).

3.2. Preferred QosPolicyHolder Service Selection

In a scenario of multiple *QosPolicyHolder* Services on the network, a user may desire to have all the policies stored in a single centralized *QosPolicyHolder* Service for consistent application of QoS policies. This is achieved by invoking the *SetAsPreferred()* action with the value of input argument *SelectAsPreferred* set to **"1"**. When a *QosPolicyHolder* Service is selected as preferred, the *QoS Manager* uses this preferred *QosPolicyHolder* Service for all the Hybrid and Parameterized streams. The *QoS Manager* uses the preferred *QosPolicyHolder* Service for prioritized streams only if the Control Point does not specify *PolicyHolderId* parameter in the traffic descriptor. If a user no longer wishes to have a particular *QosPolicyHolder* Service to be preferred, then a user can either select another *QosPolicyHolder* Service as preferred by invoking a *SetAsPreferred()* action on a different *QosPolicyHolder* Service with *SelectAsPreferred* input argument set to **"1"** or by invoking the *SetAsPreferred()* action on the same *QosPolicyHolder* Service with *SelectAsPreferred* input argument set to **"0"**.

SetAsPreferred() is an optional action. Thus, a *QosPolicyHolder* Service that does not implement this action cannot be selected as preferred by the user.

See *QosManager:3* Service [QM] for details on how the *QoS Manager* selects a *QosPolicyHolder* Service for establishing QoS under different *QosPolicyHolder* Service configurations.

3.3. QosPolicyHolder Service Configuration

Policy Add/Delete/Update:

A Control Point interested in adding any *QphPolicyRule* to the *QosPolicyHolder* Service will perform following steps:

1. The Control Point calls the *RetrieveQphPolicy()* action to retrieve all the policies in the *QosPolicyHolder* Service. Output of this action helps the Control Point decide the location where it wants to add the new policy rule. It also provides the current *PolicyVersion* to the Control Point. Refer to the definition of *NumberOfPolicyHandle* for details on specifying all the policies.
2. As a next step, the Control Point populates the *QphPolicyRule* parameter and calls the *AddQphPolicy()* action. Once a *QphPolicyRule* is successfully added, the *QosPolicyHolder* Service returns success. If during the addition process the *QosPolicyHolder* Service makes some changes to *QphPolicyRule*, it provides the appropriate *ReasonCode* as part of the output.

A Control Point interested in deleting any *QphPolicyRule* from the *QosPolicyHolder* Service will perform following steps:

1. The Control Point calls the *RemoveQphPolicy()* action with *PolicyHandle* as input to delete any *QphPolicyRule* from the *QosPolicyHolder* Service. The Control Point gets *PolicyHandle* as a result of calling the *AddQphPolicy()* or the *RetrieveQphPolicy()* action before calling this action.
2. As a next step, the *QosPolicyHolder* Service verifies the *PolicyHandle* and if it's valid, the *QosPolicyHolder* Service removes the *QphPolicyRule* from its database.

Policy Lookup:

1. Upon receiving the *GetTrafficPolicy()* or *GetListOfTrafficPolicies()* action, the *QosPolicyHolder* Service performs policy lookup process for each traffic descriptor and returns the TIN and *UIN*. The policy lookup process, at the least, involves the following:
    a) The *QosPolicyHolder* Service uses the traffic descriptors parameters to perform policy rule lookup in its policy database.
    b) If no matching policy rule is found during the Policy Lookup, the default policy rule is applied.
    c) Policy precedence is defined by the order of occurrence of policy elements in the *QosPolicyHolder* Service policy database, thus a lookup always results in exactly one result.

4. XML Service Description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion> <!-- UPnP version 1.0 -->
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
      <name>GetTrafficPolicy</name>
      <argumentList>
        <argument>
          <name>RequestedTrafficDescriptor</name>
          <relatedStateVariable>A_ARG_TYPE_TrafficDescriptor</relatedStateVariable>
          <direction>in</direction>
        </argument>
        <argument>
          <name>OutputTrafficPolicy</name>
          <relatedStateVariable>A_ARG_TYPE_TrafficPolicy</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetListOfTrafficPolicies</name>
      <argumentList>
        <argument>
          <name>ListOfTrafficDescriptors</name>
       <relatedStateVariable>A_ARG_TYPE_ListOfTrafficDescriptors</relatedStateVaria
       ble>
          <direction>in</direction>
        </argument>
        <argument>
          <name>ListOfTrafficPolicies</name>
      <relatedStateVariable>A_ARG_TYPE_ListOfTrafficPolicies</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>SetAsPreferred</name>
      <argumentList>
        <argument>
          <name>SelectAsPreferred</name>
          <relatedStateVariable>A_ARG_TYPE_IsPreferred</relatedStateVariable>
          <direction>in</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>AddQphPolicy</name>
      <argumentList>
        <argument>
          <name>QphPolicyRule</name>
          <relatedStateVariable>A_ARG_TYPE_QPHPolicyRule</relatedStateVariable>
          <direction>in</direction>
        </argument>
       <argument>
          <name>Position</name>
          <relatedStateVariable>A_ARG_TYPE_Position</relatedStateVariable>
```

```
          <direction>in</direction>
        </argument>
        <argument>
          <name>PolicyVersion</name>
          <relatedStateVariable>PolicyVersion</relatedStateVariable>
          <direction>in</direction>
        </argument>
        <argument>
          <name>TrafficImportanceNumber</name>
          <relatedStateVariable>A_ARG_TYPE_TIN</relatedStateVariable>
          <direction>in</direction>
        </argument>
        <argument>
          <name>ImportanceNumber</name>
          <relatedStateVariable>A_ARG_TYPE_IN</relatedStateVariable>
          <direction>out</direction>
        </argument>
        <argument>
          <name>ReasonCode</name>
          <relatedStateVariable>A_ARG_TYPE_ReasonCode</relatedStateVariable>
          <direction>out</direction>
        </argument>
        <argument>
          <name>PolicyHandle</name>
          <relatedStateVariable>A_ARG_TYPE_PolicyHandle</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>RemoveQphPolicy</name>
      <argumentList>
        <argument>
          <name>PolicyHandle</name>
          <relatedStateVariable>A_ARG_TYPE_PolicyHandle</relatedStateVariable>
          <direction>in</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>RetrieveQphPolicy</name>
      <argumentList>
        <argument>
          <name>ListPolicyHandle</name>
          <relatedStateVariable>A_ARG_TYPE_ListPolicyHandle</relatedStateVariable>
          <direction>in</direction>
        </argument>
        <argument>
          <name>PolicyVersion</name>
          <relatedStateVariable>PolicyVersion</relatedStateVariable>
          <direction>out</direction>
        </argument>
        <argument>
          <name>ListOfQphPolicyRule</name>

       <relatedStateVariable>A_ARG_TYPE_ListOfQphPolicyRule</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetPolicyVersion</name>
      <argumentList>
```

```
        <argument>
          <name>PolicyVersion</name>
          <relatedStateVariable>PolicyVersion</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
    Declarations for other actions added by UPnP vendor (if any) go here
  </actionList>
  <serviceStateTable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_TrafficDescriptor</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_TrafficPolicy</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_ListOfTrafficDescriptors</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_ListOfTrafficPolicies</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_IsPreferred</name>
      <dataType>boolean</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_QphPolicyRule</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_ListOfQphPolicyRule</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_Position</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_TIN</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_IN</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_ReasonCode</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_PolicyHandle</name>
      <dataType>ui4</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>A_ARG_TYPE_ListPolicyHandles </name>
      <dataType>string</dataType>
    </stateVariable>
```

```
    <stateVariable sendEvents="yes">
      <name>PolicyVersion</name>
      <dataType>ui4</dataType>
    </stateVariable>
    Declarations for other state variables added by UPnP vendor (if any) go here
  </serviceStateTable>
</scpd>
```

5. Test

Content requirements for this section to be specified in revision 1a of this standard template.