

Policy Framework
Internet Draft
Expires September 2000

Y. Snir
Y. Ramberg
J. Strassner
R. Cohen
Cisco Systems
February, 2000

[draft-ietf-policy-qos-schema-01.txt](#)

QoS Policy Schema

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

The purpose of this document is to provide a mapping of QoS policy information to a form that can be implemented in a directory that uses (L)DAP as its access protocol. Its derivation is as follows.

The structure of generic policy information is defined in the Policy Core Information Model ([[PCIM](#)]). The mapping of this information to a form that can be implemented in a directory is defined in the Policy Core Schema ([[PFSHEMA](#)]). The [[PCIM](#)] is extended to represent specific information needed to represent QoS policies with the QoS Information Model ([[QOSIM](#)]). This draft, then, maps the data defined in the [[QOSIM](#)] to a form that can be implemented in a directory. Specifically, this draft refines the concept of generic policy rules, conditions and actions to cover extensions necessary for representing policies to control the configuration and management of RSVP and Differentiated Services. This document also discusses LDAP related issues regarding the implementation of such a schema.

Snir, Ramberg, Strassner, Cohen expires September 2000 1

Draft-ietf-policy-qos-schema-01.txt April 2000

Table of Contents

1. Introduction	5
1.1 Related Work	5
1.2 Objective	6
1.3 Mappings	6
1.4 Approach	7
2. The QoS Policy Information Model	8
3. Inheritance Hierarchy for the LDAP QoS Policy Schema	9
3.1 Containment Hierarchy	11
3.2 Reusable vs. Rule-Specific Objects	13
3.3 Policy and DIT Containment	13
4. General Discussion of Mapping the Information Model to LDAP	15
4.1. Use of Distinguished Name in the Schema	15
4.2. QoS Policy Auxiliary Classes	15
4.2.1. Using Attachment of Auxiliary Classes vs. DNs	15
4.2.2. Multiple Attachment	15
4.2.3. Auxiliary Classes - When and How They Should Be Used	15
4.2.3.1. Attach to policyInstance, policyConditionInstance and policyActionInstance Classes	16
4.2.3.2. Attach Specific Containers to Root Objects	16
4.2.3.3. Attach to an Object for Efficient LDAP Retrieval	16
4.2.3.3.1. Attaching qosPolicySimpleCondition to policyRuleConditionAssociation	16
4.2.3.3.2. Attaching QoS Policy Action Classes to policyRuleActionAssociation	17
4.2.3.3.3. Attaching qosPolicyVariable and qosPolicyValue Extensions to qosPolicySimpleCondition	17
4.2.3.3.4 Extensions for Complex Policy Rules	17
5. LDAP Search Efficiency	18
5.1. Reusable Objects	18
5.2. NamedGroupContainer Location	18
5.3. QoS Policy Rules Location	18
5.4. QoS Policy Sub-Rules Location	18
5.5. Condition and Action Object Location	19
5.6. Searching for QoS Policy Objects	19
6. Data Integrity	20
6.1. Order of Insertion of Objects into the Directory Service	20
6.2. Distinguishing between Reusable Objects in the Repository and	

Rule-Specific Objects	21
6.3. Versioning of Objects	21
6.4. Transaction Support	21
6.5 Referential Integrity Support	22
6.6. Data Integrity in Replicated Directories	22
7. Summary of QoS Policy Class Relationships	23

Snir, Ramberg, Strassner, Cohen expires September 2000 2
Draft-ietf-policy-qos-schema-01.txt April 2000

8. Class Definitions	25
8.1. Class policyGroup	26
8.2 Class policyRepository	26
8.3 Class qosRepositoryContainmentAuxClass	26
8.4 Class qosPolicyDomain	26
8.4.1. The Attribute qpDomainName	27
8.4.2. The Attribute qpPHBSet	28
8.4.3. The Attribute qpPolicyRuleMatchMethod	28
8.5. Class qosNamedPolicyContainer	28
8.5.1. The Attribute qpPriority	29
8.5.2. The Attribute qpPolicyRuleMatchMethod	30
8.6. Class qosPolicyPRAction	30
8.6.1. The Attribute qpDirection	31
8.6.2. The Attribute qpSetDSCPvalue	32
8.6.3. The Attribute qpMeter	32
8.6.4. The Attribute qpMeterScope	33
8.6.5. The Attribute qpTrfcProf	33
8.6.6. The Attribute qpOutOfProfileAction	34
8.6.7. The Attribute qpOutOfProfileRemarkValue	34
8.7. Class qosPolicyRSVPAction	35
8.7.1. The Attribute qpRSVPDirection	35
8.7.2. The Attribute qpRSVPMessageType	36
8.7.3. The Attribute qpRSVPStyle	36
8.7.4. The Attribute qpRSVPServiceType	37
8.7.5. The Attribute qpRSVPInstallAction	37
8.7.6. The Attribute qpRSVPCtrlAction	38
8.7.7. The Attribute qpRSVPMeter	38
8.7.8. The Attribute qpRSVPMeterScope	39
8.7.9. The Attribute qpRSVPTrfcProf	39
8.8. Class qosPolicyPRTrfcProf	40
8.8.1. The Attribute qpPRRate	40
8.8.2. The Attribute qpPRNormalBurst	41
8.8.3. The Attribute qpPRExcessBurst	41
8.9. Class qosPolicyRSVPTrfcProf	42
8.9.1. The Attribute qpRSVPTokenRate	42
8.9.2. The Attribute qpRSVPPeakRate	43
8.9.3. The Attribute qpRSVPBucketSize	43
8.9.4. The Attribute qpRSVPResvRate	43
8.9.5. The Attribute qpRSVPResvSlack	44

8.9.6.	The Attribute qpRSVPSessionNum	44
8.9.7.	The Attribute qpMinPolicedUnit	45
8.9.8.	The Attribute qpMaxPktSize	45
8.10.	Class qosPolicyRSVPSignalCtrlAction	46
8.10.1.	The Attribute qpForwardingMode	46
8.10.2.	The Attribute qpSendError	47
8.10.3.	The Attribute qpReplaceDSCP	47
8.10.4.	The Attribute qpReplacePreemptionPriority	48
8.10.5.	The Attribute qpReplaceDefendingPriority	48
8.11.	Class qosPolicyRSVPInstallAction	49
8.11.1.	The Attribute qpSetDSCPValue	50
8.11.2.	The Attribute qpSetDefendingPriority	50
8.11.3.	The Attribute qpSetPreemptionPriority	51

Snir, Ramberg, Strassner, Cohen	expires September 2000	3
Draft-ietf-policy-qos-schema-01.txt		April 2000

8.12.	Class qosPolicySimpleCondition (Aux)	51
8.12.1	The Attribute qpOperator	52
8.12.2.	The Attribute qpVariableAtom	53
8.12.3.	The Attribute qpValueAtom	53
8.13.	Class qosPolicyVariable	54
8.13.1.	The Attribute qpVariableName	55
8.13.2	The Attribute qpValueTypes	57
8.13.3.	The Attribute qpVariableDescription	58
8.13.4.	The Attribute qpValueConstraints	59
8.14.	Class qosPolicyValue	59
8.15.	Class qosPolicyIPv4AddrValue	60
8.15.1.	The Attribute qpIPv4AddrList	60
8.16.	Class qosPolicyIPv6AddrValue	61
8.16.1.	The Attribute qpIPv6AddrList	62
8.17.	Class qosPolicyMACAddrValue	63
8.17.1.	The Attribute qpMACAddrList	64
8.18.	Class qosPolicyStringValue	64
8.18.1.	The Attribute qpStringList	65
8.19	Class qosPolicyBitStringValue	66
8.19.1.	The Attribute qpBitStringList	66
8.20.	Class qosPolicyDNValue	67
8.20.1.	The Attribute qpDNList	68
8.21.	Class qosPolicyAttributeValue	68
8.21.1.	The Attribute qpAttributeName	69
8.21.2.	The Attribute qpAttributeValueList	70
8.22.	Class qosPolicyIntegerValue	70
8.22.1.	The Attribute qpIntegerList	71
8.23.	Class qosPolicyPHBSet	72
8.24.	Class qosPolicyPHB	72
8.24.1.	The attribute qpDSCP	73
8.25.	Class qosPolicyElementAuxClass	73
8.26.	Class qosPolicyMeter	74

9. Extending the QoS Policy Schema	75
9.1. Extending qosPolicyValue	75
9.2. Extending qosPolicySimpleCondition	75
9.3. Extending qosPolicyAction	76
10. Security Considerations	76
11. Acknowledgments	76
12. References	76
13. Author's Addresses	78
14. Full Copyright Statement	78

Snir, Ramberg, Strassner, Cohen expires September 2000 4
Draft-ietf-policy-qos-schema-01.txt April 2000

[1. Introduction](#)

The purpose of this document is to provide a mapping of QoS policy information to a form that can be implemented in a directory that uses (L)DAP as its access protocol. QoS policy information includes not just the definition of policy rules, but also policy conditions, policy actions, and general policy data that is used to make policy decisions. Its derivation is as follows.

The structure of generic policy information is defined in the Policy Core Information Model ([[PCIM](#)]). Note that an information model is NOT bound to any one specific repository. Rather, the information model represents objects and relationships between objects. Therefore, a set of mappings must be defined that translate the data specified in the information model to a specific type of repository. Each mapping will be necessarily different, which reflects the different access protocol(s) and structure of data used by different repositories.

The Policy Core Schema ([[PFSHEMA](#)]) defines a mapping that implements the information specified in the [[PCIM](#)] to a form that can be implemented in a directory. The [[PCIM](#)] is extended to represent specific information needed to represent QoS policies with the QoS Information Model ([[QOSIM](#)]). This draft, then, maps the data defined in the [[QOSIM](#)] to a form that can be implemented in a directory. This mapping is also influenced by the mapping of generic policy information (as specified in [[PCIM](#)]) to a directory (via [[PFSHEMA](#)]). This draft relies on and uses concepts from [[PFSHEMA](#)].

This draft refines the concept of generic policy rules, conditions and actions to cover extensions necessary for representing policies to control the configuration and management of RSVP and Differentiated Services. This document also discusses LDAP related issues regarding the implementation of such a schema.

1.1 Related Work

This document takes as its starting point the object-oriented information model for representing QoS policy information currently under development in the IETF's Policy Framework working group. The IETF document defining this information model is the "QoS Policy Information Model" [[QOSIM](#)]. This model defines the structural and auxiliary object classes needed to represent QoS policy information. In general, these object classes extend the Core Policy object classes as defined in the Policy Core Schema document [[PFSCHEMA](#)]. In addition, the QoS policy schema uses the association and aggregation mechanisms as defined in the [[PFSCHEMA](#)].

Snir, Ramberg, Strassner, Cohen	expires September 2000	5
Draft-ietf-policy-qos-schema-01.txt		April 2000

Specifically, the Policy Core Information Model [[PCIM](#)] defines the generic structure of a policy, and provides a framework for describing specific conditions and actions that are used to construct application and domain-specific policies. The QoS Policy Information model [[QOSIM](#)] then refines this information to describe policy rules, conditions and actions, as well as other data, that are needed to represent network QoS policies.

Related to this is the specification of the underlying QoS mechanisms provided by a device. This is specified in two drafts. The information model for representing device QoS mechanisms is specified in [[QOSDEVIM](#)].

1.2 Objective

The object of the QoS modeling work is to derive two sets of drafts. The first set defines the representation of QoS policies, while the second set defines the QoS mechanisms in a device that provide QoS services. QoS policies, then, are written in order to manage and control the implementation of QoS services by provisioning the QoS mechanisms of devices participating in that service.

1.3 Mappings

Information models are by definition repository-independent. That is, one needs to build a mapping of the data contained in the information model to a form that can be implemented in the target repository. To ensure that this can be done, parallel drafts are being defined for each type of policy that is being developed in the Policy Framework working group.

The first, the Policy Core Schema ([\[PFSCHEMA\]](#)), is a mapping of the information in the [\[PCIM\]](#) to a form suitable for implementation in a directory. The second, this document, is a mapping of the information in the [\[QOSIM\]](#) to a form suitable for implementation in a directory. This document therefore derives from both the [\[PFSCHEMA\]](#) as well as the [\[QOSIM\]](#).

Similarly, [\[QOSDEVIM\]](#) defines a set of extensions that model (in a generic way) QoS mechanisms inherent in devices. A future draft will be written that maps the information specified in [\[QOSDEVIM\]](#) to a form suitable for implementation in the directory.

Snir, Ramberg, Strassner, Cohen	expires September 2000	6
Draft-ietf-policy-qos-schema-01.txt		April 2000

1.4 Approach

This draft defines the mapping of these information model classes to a directory that uses LDAPv3 as its access protocol. In particular, this draft refines the concept of generic policy rules, conditions and actions to cover extensions necessary for representing policies to control the configuration and management of RSVP and Differentiated Services. This information is typically used by QoS Policy Servers to configure network devices according to prescribed QoS Policies. In general, this class hierarchy will need to be mapped to a particular vendor's implementation. This is due to the differences in LDAP directory server implementations.

For the classes in the information model, the mapping is basically one-for-one: information model classes map to LDAP classes, and information model properties map to LDAP attributes.

Implementations that want to implement QoS policies in a directory

SHALL use the LDAP policy schema defined in [[PFSCHEMA](#)] and the QoS extensions defined in this document.

The use of the QoS Policy information model defined in reference [[QOSIM](#)] as the starting point enables the schema and the relationship class hierarchy to be extended and used for different types of repositories. For example, relational databases as well as directories can also use this information. The difference is the mapping that is defined from the information model to the repository.

This document fits into the overall framework for representing, deploying, and managing policies being developed by the Policy Framework Working Group. It also draws on the work done for the Directory-enabled Networks (DEN) specification, reference [4].

Finally, this draft is also meant to interoperate with a companion draft that defines the QoS capabilities of network devices. Again, two versions of the QoS capabilities draft are planned. The first defines the information model that represents QoS capabilities of network devices. This draft is specified in [[QOSDEVIM](#)]. A second draft will be published soon that defines the mapping of the data in [[QOSDEVIM](#)] to a form that can be stored in a directory.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#), reference [5].

Snir, Ramberg, Strassner, Cohen	expires September 2000	7
Draft-ietf-policy-qos-schema-01.txt		April 2000

[2. The QoS Policy Information Model](#)

This document contains an LDAP schema representing the QoS Policy Information Model, which is defined in the companion document "QoS Policy Information Model" [[QOSIM](#)]. Other documents may subsequently be produced, with mappings of this same QoS Policy Information Model to other storage technologies. Since the detailed semantics of the QoS Policy classes appear only in reference [[QOSIM](#)], that document is a prerequisite for reading and understanding this document.

The QoS Policy schema by itself may be insufficient to model a particular set of QoS services and systems. This is because the purpose of this draft is to define QoS policies in as generic a way as

possible. In general, this might be insufficient in three ways. The first is that application-specific policies are not represented. In this case, such policies SHOULD be implemented as extensions to this draft. Extensions can take two forms:

- new functions can be represented as subclasses of classes defined in this document
- new attributes can be defined for existing classes specified in this document

Both of the above methods are aimed at deriving implementation-specific classes from this schema in order to model a specific QoS system in sufficient detail. In fact, the QoS Policy schema is a middle layer in a three-level hierarchy of schemata:

```
Core Policy Schema  is extended by
QoS Policy Schema   is extended by
Implementation-specific schemata that also use the QoS capabilities
draft and extensions of that draft
```

The second way that this schema may prove to be insufficient is in providing an efficient mapping to a given vendor's directory implementation. The guiding principle in this draft is to provide a set of classes and attributes that can be easily implemented in the majority of LDAP implementations. However, certain LDAP functions are implemented in very different ways by different vendors. Thus, it may be necessary to take the basic design presented in this document and modify it to make it fit a particular vendor's implementation better.

The third way that this schema may provide to be insufficient is in accommodating an implementation that is not compliant with the LDAP specifications. This is really a variation of the second case. Certain vendors are not completely compliant with LDAP. However, it is still desirable to define a schema that accommodates them as well as other compliant implementations. The schema presented in this document has made every effort to do so, but edge cases may exist that require the schema presented in this document to be modified to accommodate a particular vendor's needs.

Snir, Ramberg, Strassner, Cohen expires September 2000 8
Draft-ietf-policy-qos-schema-01.txt April 2000

3. Inheritance Hierarchy for the LDAP QoS Policy Schema

The following diagram illustrates the class hierarchy for the LDAP QoS Policy schema classes. These classes will be defined in [section 8](#) of this document, except for the core classes, which are defined in [[PFSCHEMA](#)]. Note that only the [[PFSCHEMA](#)] classes required by this draft are shown.

```

top (the root of the directory)
|
+--policy (abstract) ([PFSHEMA])
|
|   +--policyGroup (structural) ([PFSHEMA])
|   |
|   |   +--qosPolicyDomain (structural)
|   |   |
|   |   |   +--qosNamedPolicyContainer (structural)
|   |   |
|   |   +--policyRule (structural) ([PFSHEMA])
|   |   |
|   |   +--policyRuleConditionAssociation (structural) ([PFSHEMA])
|   |   |
|   |   +--policyRuleActionAssociation (structural) ([PFSHEMA])
|   |   |
|   |   +--policyInstance (structural) ([PFSHEMA])
|   |   |
|   |   +--policyConditionInstance (structural) ([PFSHEMA])
|   |   |
|   |   +--policyActionInstance (structural) ([PFSHEMA])
|   |   |
|   |   +--policyElementAuxClass (auxiliary) ([PFSHEMA])
|   |   |
|   |   +--policyConditionAuxClass (auxiliary) ([PFSHEMA])
|   |   |
|   |   |   +--qosPolicySimpleCondition (auxiliary)
|   |   |
|   |   +-- qosPolicyMeter (auxiliary)
|   |   |
|   |   +-- qosPolicyPRTrfcProf (auxiliary)
|   |   |
|   |   +-- qosPolicyRSVPTrfcProf (auxiliary)
|   |   |
|   |   +-- qosPolicyPHBSet (abstract)
|   |   |
|   |   +-- qosPHB (abstract)
|   |   |
|   |   +--qosPolicyVariable(auxiliary)
|   |
|

```

(Figure 1 is continued on next page)

(Figure 1 is continued from the previous page)

```

top (root of the directory)
|
+--policy (abstract) ([PFSCHEMA])
|   |
|   +--qosPolicyValue(abstract)
|       |
|       +--qosPolicyIPv4AddrValue(auxiliary)
|       |
|       +--qosPolicyIPv6AddrValue(auxiliary)
|       |
|       +--qosPolicyMACAddrValue(auxiliary)
|       |
|       +--qosPolicyStringValue(auxiliary)
|       |
|       +--qosPolicyBitStringValue(auxiliary)
|       |
|       +--qosPolicyDNValue(auxiliary)
|       |
|       +--qosPolicyAttributeValue(auxiliary)
|       |
|       +--qosPolicyIntegerValue(auxiliary)
|       |
|   +--policyActionAuxClass (auxiliary) ([PFSCHEMA])
|       |
|       +-- qosPolicyPRAction (auxiliary)
|       |
|       +-- qosPolicyRSVPAction (auxiliary)
|       |
|       +-- qosPolicyRSVPSignalCtrlAction (auxiliary)
|       |
|       +-- qosPolicyRSVPInstallAction (auxiliary)
|       |
|   +--policyRepository (structural) ([PFSCHEMA])
|   |
|   +--policyGroupContainmentAuxClass (auxiliary) ([PFSCHEMA])
|   |
|   +--policyRuleContainmentAuxClass (auxiliary) ([PFSCHEMA])

```

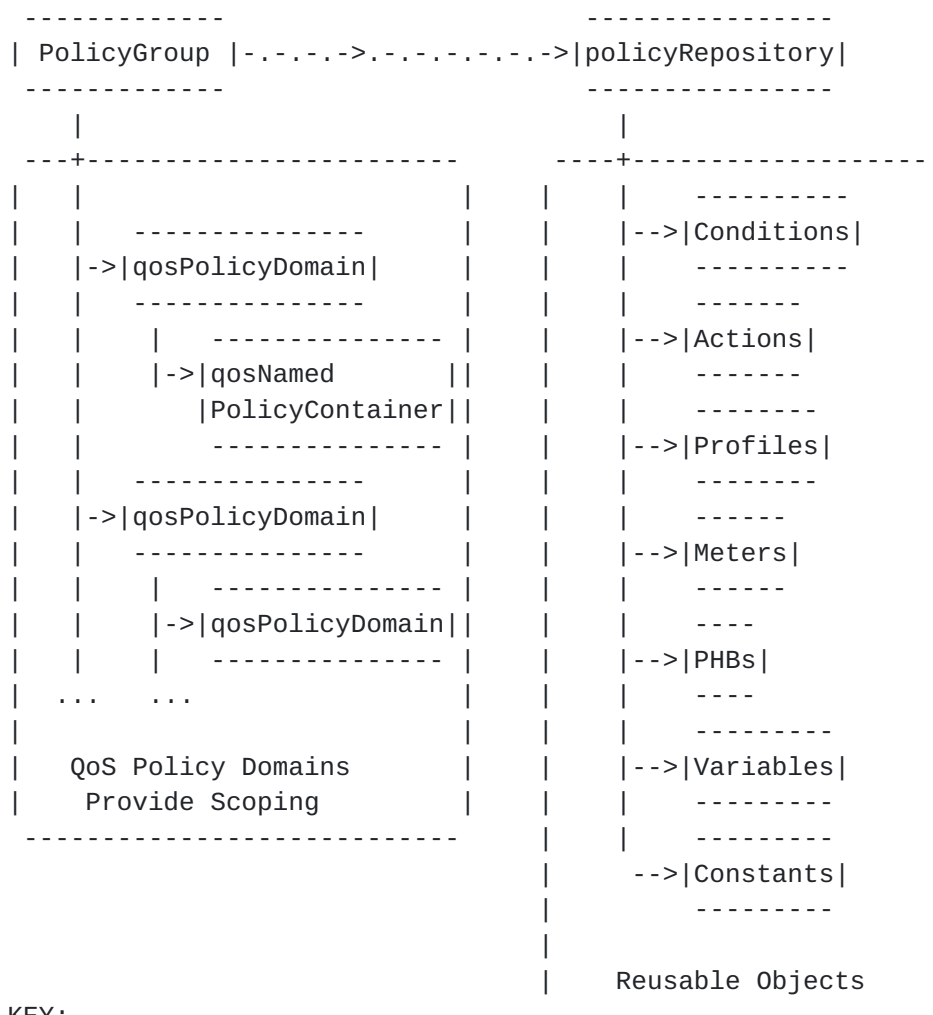
Figure 1. QoS Policy Schema Inheritance Hierarchy

Note: classes with a "qos" prefix are QoS Policy Schema classes.

3.1. Containment Hierarchy

The fundamental data model of the QoS Policy schema is defined by the mapping of the inheritance and aggregation hierarchies defined in the QoS Policy Information Model [[QOSIM](#)]. This mapping, for a directory, forms a strict tree hierarchy (note that other mappings for other types of repositories may be different in their resulting structure, but they will still use the information in [[QOSIM](#)]). This mapping is defined by a combination of this draft and the QoS Policy Core Schema.

Containment is a critical feature of directories. Therefore, Figure 2 shows a summary view of the class containment hierarchy.



KEY :

- ```

-----> Containment.
-...-> Implied containment. That is, the PolicyGroup class
would not contain an instance of the Repository class,
but would rather contain instances of classes that
are contained in the Repository class.

```

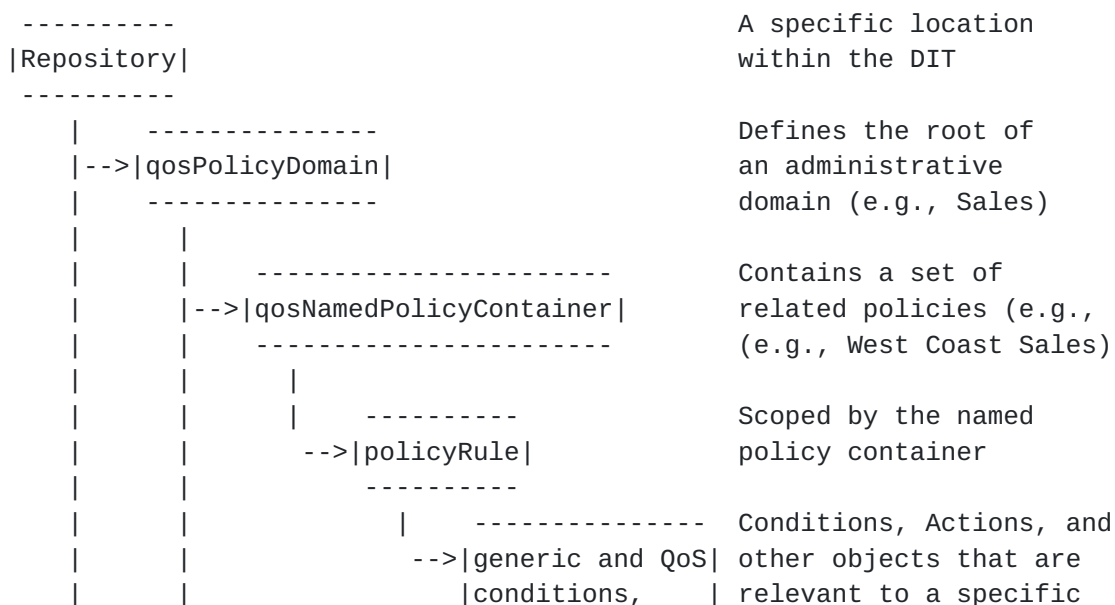
Figure 2: QoS Policy class containment - major classes

Snir, Ramberg, Strassner, Cohen      expires September 2000      11  
 Draft-ietf-policy-qos-schema-01.txt      April 2000

The QoS policy hierarchy is structured as a set of containment relationships. These consist of container objects (on the left) that each have sets of DNs that point to other objects on the right. These contained objects (the ones on the right) model containment by placement. That is, containment is achieved by placing the contained objects as leaf nodes of the container. The container objects (the ones on the left) model containment both by placement as well as by DN reference. For example, a qosNamedPolicyContainer is either placed as a leaf node of a qosPolicyDomain container, or referenced by a DN from the aux class policyGroupContainmentAuxClass (which is attached to a qosPolicyDomain container). This flexibility enables containers to be used to represent various administrative, political, geographical, or other organizational constructs. In addition, separating containment from functionality enables the design of each to avoid affecting the other.

With respect to the Policy Framework core schema specifications [[PFSCHEMA](#)], containers are usually based on auxiliary classes. A container, then, may be attached to a branch-level class so that leaf-node classes that are specific to sub-domains, applications, or other units of scoping may be added underneath the branch-level class.

Figure 3 below illustrates how these classes can be used to scope policies.



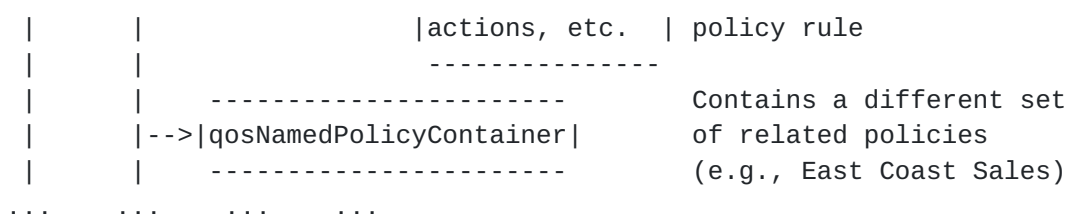


Figure 3. Containment Hierarchy and Policy Scoping

Snir, Ramberg, Strassner, Cohen expires September 2000 12  
 Draft-ietf-policy-qos-schema-01.txt April 2000

### 3.2 Reusable vs. Rule-Specific Objects

This schema provides for two types of objects. Reusable objects are those objects that can be used by multiple policy rules. They live in their own section of the repository (conceptually, a repository in a repository). In a directory implementation, they are pointed to by DNS. Reusable object references cross the containment hierarchy and are not considered part of the policy tree structure. [Section 5.1](#) describes the reusable object repositories.

The advantage of reusable objects is that the same object may be used by multiple policy rules. This has many benefits, the primary one being that it enables a common set of conditions and/or actions to be defined once and used many times. However, it also has disadvantages. The main disadvantage is that in a directory implementation, this means that the object could incur multiple accesses, one for the base object and one (or more) access for each reusable object. See [\[DEREF\]](#) for a possible way of reducing the cost of multiple accesses.

Rule-specific objects are those objects that can only be used by a single rule. In a directory, they are attached directly to the object. Rule-specific objects are explicitly part of the containment hierarchy. The advantage of using rule-specific objects is speed of access. One could also argue that the design is simplified, since the meaning is more direct. The disadvantages are that there is no possibility of reusability, and that using rule-specific objects reduce the extensibility of the schema. Furthermore, there is a greater chance of inconsistency, since the same entity may be defined in different ways if it is used by multiple rule-specific objects.

### 3.3 Policy and DIT Containment

Policy applications do not own the DIT; rather, they must work within an existing DIT. Consequently, this means that there is no standard organization of objects to be controlled via policy. This makes it hard to associate policy objects with other objects in the DIT efficiently.

This schema advocates one particular solution to this problem. This solution extends the DIT structure and builds a special portion in the DIT (conceptually, a sub-repository) that is reserved for policy objects. This avoids needless replication of policy objects, and promotes reusability of policy objects.

The root of the policy portion of the DIT is a single-instance object of type policyGroup, defined in the Policy Core Schema [[PCIM](#)]. This class serves as the root of the policy schema, and provides scoping for two main branches of the policy schema: reusable-objects repositories and a policy tree that contains policy rules and their building blocks.

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 13         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

Figures 2 and 3 show that multiple qosPolicyDomain containers can be used to provide scoping for a set of policyGroup and/or policyRule classes (these are both defined in [[PFSCHEMA](#)]). Each qosPolicyDomain can contain its own set of policyRules and groups of rules. However, each qosPolicyDomain contains policies that are specific to a particular administrative domain.

The reusable-objects repository section contains information that every container can use, and is divided into different categories of information (conditions, actions, etc.). This lets multiple policy servers in different policy domains share and reuse common information.

These concepts are explained in more detail in the QoS Policy Information Model [[QOSIM](#)].

## **4. General Discussion of Mapping the Information Model to LDAP**

### **4.1. Use of Distinguished Name in the Schema**

Distinguished names are object primary keys in LDAP. The QoS Policy schema makes ample use of DNS in various places according to the concepts defined in the Core Schema. Here are the major uses of DNS:

- \* Object containers - throughout the schema, the relationships of a container to the set of objects that it contains are prevalent. Containers may hold lists of DNS of contained objects. A container may be attached to a node on the tree, thus adding another level of scoping to the hierarchy.
- \* Static branches - leaves of the tree can be pointed to by DNS to incorporate information specific to a particular branch of the tree
- \* Cross hierarchy reference - references from a given entity to another entity (e.g., a repository object ) can be referenced by means of a DN

### **4.2. QoS Policy Auxiliary Classes**

#### **4.2.1. Using Attachment of Auxiliary Classes vs. DNS**

For a general discussion of attachment of auxiliary classes and the pros and cons of doing so, see [[PCIM](#)]. QoS policy reusable objects should be stored in the appropriate repository. These objects will be



referred to by DNS. Objects that are not reusable should, if possible, be attached to other classes for efficiency.

Attachment allows a more efficient LDAP data retrieval operation. See [[PCIM](#)].

#### **[4.2.2. Multiple Attachment](#)**

Attaching more than one auxiliary class to a single structural object is allowed. This type of usage is recommended when defining efficient conditions and actions as part of the policy rule itself. For example, a condition that includes a simple condition, variable and one or more values can be attached to the policyRuleConditionAssociation entry. In this example, this method enables the various components that make up the condition to be retrieved in a single operation.

#### **[4.2.3. Auxiliary Classes - When and How They Should Be Used](#)**

Auxiliary objects must be attached to a structural class to be instantiated. There are 3 ways of using these objects.

Snir, Ramberg, Strassner, Cohen      expires September 2000      15  
Draft-ietf-policy-qos-schema-01.txt      April 2000

##### **[4.2.3.1. Attach to policyInstance, policyConditionInstance and policyActionInstance Classes](#)**

Whenever an auxiliary class should be instantiated so that it can be reused, it must be attached to one of three structural objects. These objects are the policyInstance, the policyConditionInstance, and the policyActionInstance objects. These classes not only allow instantiation of an auxiliary class, but also make it a named instance that could be placed under a policy repository namespace and reused. For example, a reusable qosPolicySimpleCondition can be attached to an instance of the policyConditionInstance, which can then be placed in the repository.

##### **[4.2.3.2. Attach Specific Containers to Root Objects](#)**

Some auxiliary classes are attached to the appropriate structural classes defined in the Core Policy Schema. Among such classes are the policyGroupContainmentAuxClass. This is used to associate, for example, qosPolicyDomain objects to, along with other objects that extend the semantics provided by the policyGroup class. This type of association is to be used when general aggregation by DIT location can't be used. For example, a policyGroup object that serves as the root class of policy information could contain two qosPolicyDomain objects as direct

children, and one additional qosPolicyDomain object that is not located as a (direct) child of the policyGroup object. This third object would be referenced using the policyGroupContainmentAuxClass. This structure has defined three different policy domains under the same policy root. This structure simplifies their management.

#### **4.2.3.3. Attach to an Object for Efficient LDAP Retrieval**

##### **4.2.3.3.1. Attaching qosPolicySimpleCondition to policyRuleConditionAssociation**

A policyRuleConditionAssociation includes a single condition, either by attachment or by DN reference. Using attachment allows the retrieval of the association class and the condition itself in a single LDAP search. This creates a rule-specific condition. That is, a condition constructed in this way can not be reused by other policy rules.

Alternatively, a DN reference can be used. This provides reusability by keeping the condition in the policy repository, and using a DN to reference the condition. Conditions formed in this way can be shared by many policy rules.

Snir, Ramberg, Strassner, Cohen      expires September 2000      16  
Draft-ietf-policy-qos-schema-01.txt      April 2000

##### **4.2.3.3.2. Attaching QoS Policy Action Classes to policyRuleActionAssociation**

A policyRuleActionAssociation includes a single action, either by attachment or by DN reference. Using attachment allows the retrieval of the association class and the action itself in a single LDAP search, while using a DN reference provides reusability. The implications are exactly as described in [section 4.2.3.3.1](#) above, except that they apply to actions, not conditions.

##### **4.2.3.3.3. Attaching qosPolicyVariable and qosPolicyValue Extensions to qosPolicySimpleCondition**

A qosPolicySimpleCondition includes a single qosPolicyValue and a single qosPolicyVariable. Each can either be attached or referenced by a DN. Using attachment allows the retrieval of the association class and the condition itself in a single LDAP search, while using a DN promotes reusability. The implications are exactly as described in

[section 4.2.3.3.1](#) above, except that they apply to terms that are used as part of a condition.

#### **[4.2.3.3.4](#) Extensions for Complex Policy Rules**

A complex policy rule is one that contains multiple conditions and/or actions. Construction of a complex policy rule is done by building on the techniques used to assemble simple policy rules. Each of the condition and action terms that make up a complex policy rule can be built using the techniques described in the previous sections. For example, it is recommended that a `qosPolicySimpleCondition` object be constructed through the attachment of `qosPolicyVariable` and `qosPolicyValue` auxiliary classes. This can then be used to build multiple conditions that are combined to form the complex policy.

The exception to this rule is when one or more of these objects are reusable (e.g., not specific to a single policy, and therefore are resident in the reusable-objects repository). In this case, the object should not be attached, but instead, a DN reference to the object should be used.

Snir, Ramberg, Strassner, Cohen      expires September 2000      17  
Draft-ietf-policy-qos-schema-01.txt      April 2000

## **[5. LDAP Search Efficiency](#)**

The ability to efficiently search for policy rules is important. Writers of policy elements should follow a few basic rules in order to allow readers of policy to do this efficiently, with a minimum of LDAP queries.

### **[5.1. Reusable Objects](#)**

Reusable objects are located in the repository sub-trees. Each reusable object is a child of the parent folder it belongs to. The parent folder defines a namespace that the objects that it contains are bound to. Keeping reusable objects in a single repository enhances their management, simplifies their maintenance, and enables rooted searches

(which are more efficient) to be implemented.

## **5.2. QoS Named Group Container Location**

NamedGroupContainers are defined as direct children of their domain Entry (e.g., they are intended to be instantiated under a qosPolicyDomain container). The purpose of the qosNamedPolicyContainer is to enable more specific behavior to be applied to a set of policies that are of a particular type, or related in a particular way. For example, a policy domain can define general traffic conditioning policy rules, which can then be specialized (e.g., subclassed) to suit the needs of particular users or applications.

## **5.3. QoS Policy Rules Location**

The philosophy of this draft is that QoS policy rules will exist as objects that are children of a particular qosNamedPolicyContainer entry. QoS policy rules may be constructed using conditions and actions that are rule-specific, reusable, or a combination of both. The important thing is that the QoS policy rules are grouped together using sets of qosNamedPolicyContainer objects.

## **5.4. QoS Policy Sub-Rules Location**

A QoS policy sub-rule is a rule that is contained in another rule. This concept is not defined in the core schema, and is specific to this QoS schema. Sub-rule entries serve two purposes. The first is to represent rules of a particular policy rule that is more general in usage and/or scope. This usage aggregates a set of sub-rules under a higher-level rule for scoping purposes.

The second use is to represent finer details of a policy. In other words, the set of sub-rules defines how a higher-level rule is structurally defined.

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 18         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

## **5.5. Condition and Action Object Location**

Condition and action objects are either located in the relevant repository (if they are reusable objects) or are defined as children of the specific policy rule that uses them.

## **5.6. Searching for QoS Policy Objects**

Readers of policies will assume that the above rules of entry location

are implemented by applications that write these results. Readers will most likely perform LDAP sub-tree searches. The readers are responsible for validating the completeness and consistency of the policy retrieved by checking that every entry exists, as specified by the relevant container values.

The Policy Core Schema [[PFSHEMA](#)] has been constructed in such a way as to enable the efficient location of policy information. This is done in two ways. First, a special section of the DIT can be designated as the repository for policy information. This in turn provides two important benefits:

- efficient search and retrieval of policy information is enabled by searching in a specific subtree
- reusable policy elements (e.g., conditions and actions) can be stored in a single location in the DIT.

This second benefit is very important. Instead of having to find instances of the same policy class throughout the DIT without any knowledge of where those instances can be located, one can instead use the policy repository to define a common location. This enhances usability and maintainability. Furthermore, if a reusable object needs to be updated, placing it in the repository enables the updating application to look in just one place to find and update the object. All other objects that refer to this object will then have their references updated automatically.

The second method of organizing policy information is through the use of auxiliary classes to "tag" an object as being related to policy. This decouples the design of the policy schema from the design of other schemata that reference it (e.g., the definition of users in an organization). For example, the policy schema can now be associated with other schemata that need policy information by tagging appropriate classes in the non-policy schemata as dependent on policy. This has the effect of defining how two disparate subtrees of the DIT can share information.

Both of these methods are described in more detail in [[PFSHEMA](#)].

Snir, Ramberg, Strassner, Cohen      expires September 2000      19  
Draft-ietf-policy-qos-schema-01.txt      April 2000

## **[6. Data Integrity](#)**

LDAP provides little if any support for data integrity protection. The only guarantee provided by LDAP-based systems is that operations on a single object instance are "atomic". This means that complex schemata,

such as the QoS Policy schema, can't guarantee atomicity of multi-step operations. Note that even reading is not safe: no read consistency is guaranteed whatsoever.

While there are various tactical solutions, a general schema SHOULD NOT rely on the guarantees of any particular directory product that are beyond the LDAP protocol standard specification, as such guarantees are currently proprietary and not supported by all products.

This section discuss the problems associated with data integrity, consistency, concurrency control and transaction handling involved in using the QoS Policy Schema classes, and suggests several approaches to tactical solutions. However, no attempt is made to provide a general strategy to the inherent weaknesses in LDAP.

### **6.1. Order of Insertion of Objects into the Directory Service**

Objects should be placed in the directory server in a particular order to minimize risks of lost updates due to the abnormal termination of clients or servers. In general, referred objects should be placed in the DIT prior to the placement of its DN in the referring object. For example, a policy action object (e.g., an instance of the qosPolicyAction class) should be fully initialized and placed in the DIT before its DN is added to the policyActionDN attribute of the instance of the policyRuleActionAssociation class.

Doing it in the opposite order (i.e., inserting a DN of the qosPolicyAction instance in the policyRuleActionList attribute before placing the action object in the DIT) may result in a "dangling" DN (i.e., a DN that points to nothing). This is because a failure in the modify process can occur which will cause a dangling reference. For example, if the client machine fails to complete its modify operations because it crashes before the second operation completes successfully, the result is that the DN will not point at a real instance.

These insertion ordering tactics comes at a price. For example, the semantics necessary for an object that refers to another object require that the referring and referred objects be placed in the directory such that the referring object is designated as the parent of the referred object. Obviously, no child DN exists before the parent is placed in the DIT. In such a case, one is tempted to write the parent object, thus creating the node in the DIT, and then write the child object. However, an abnormal termination of either the client or the LDAP server before the operation of placing the child in the DIT results in a dangling child DN reference in the parent.

To prevent this, one must pay the price of an extra write operation, as follows:

- First, write the parent with no reference to the child.
- Next, write the child to the correct DIT placement.
- Finally, modify the parent to point to the child.

Note that it is the responsibility of the writing client to eliminate cases of dangling references.

## **6.2. Benefits of Using Reusable Objects**

Reusable objects SHOULD be instantiated in the policy repository part of the DIT. This is because such objects, by definition, are intended to be shared among multiple policy rules. If such objects are not stored in the policy repository, then each change made to that object requires a complete scan of the DIT to make the change to each copy.

Alternatively, if all reusable objects are stored in the policy repository, then they are more easily reused because all policy rules that want to reuse them need only reference them. When a change is made to a reusable object that is located in the repository, no other action is required to insure that the modification is reflected in all referring objects (policies), since such referring objects use DNs to refer to the reusable object. Note also that storing objects in the repository enhances their search and retrieval, since directed sub-tree searches can be used.

## **6.3. Versioning of Objects**

Adding meta information to objects, such as creation / modification time, version and other application-specific information, will allow implementation of application-specific validation, data integrity checking and enforcement. However, discussion of these techniques is beyond the scope of this document. In general, implementation of the QoS Policy Schema SHOULD NOT assume that any versioning support is available.

## **6.4. Transaction Support**

No transaction support is defined in LDAPv3. Implementation of the QoS Policy Schema SHOULD NOT assume that any transaction support is available, and define their use of the DIT by relying solely on the single entry atomic operation LDAP supplies.

## **[6.5](#) Referential Integrity Support**

No referential support is defined in LDAPv3. Implementation of the QoS Policy Schema SHOULD NOT assume that any referential integrity support is available, and define their use of the DIT by relying solely on the single entry atomic operation LDAP supplies.

## **[6.6](#). Data Integrity in Replicated Directories**

Replication of information brings up data integrity, referential integrity, and concurrency control issues. These issues are not related specifically to the QoS Policy Schema (e.g., the QoS Policy Schema does not make things better or worse) and are beyond the scope of this document.

When updating a DN to a referred object, that object version should be checked to make sure that it exists and the object is of the right version. It is also recommend that schema checking be turned on in the server.



## 7. Summary of QoS Policy Class Relationships

All of the classes in the LDAP QoS Policy Schema map directly to corresponding classes in the QoS Policy Information Model [[QOSIM](#)]. The following table summarizes these relationships:

| Information Model Relationship                                  | LDAP Attribute / Class                                                                             |
|-----------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| qosPolicyDomain to<br>policyRepository                          | DIT containment                                                                                    |
| qosPolicyDomain to<br>qosNamedPolicyContainer to<br>policyGroup | DIT containment or<br>policyGroupsAuxContainedSet<br>property of<br>policyGroupContainmentAuxClass |
| qosNamedPolicyContainer to<br>policyRule                        | DIT containment or<br>policyRulesAuxContainedSet<br>property of<br>PolicyRuleContainmentAuxClass   |
| policyRule to<br>policyRuleConditionAssociation                 | DIT containment                                                                                    |
| policyRuleConditionAssociation<br>to qosPolicySimpleCondition   | Attachment or<br>policyConditionDN property of<br>policyRuleConditionAssociation                   |
| qosPolicySimpleCondition to<br>qosPolicyIPv4AddrValue           | Attachment or<br>qpValueAtom property of<br>qosPolicySimpleCondition                               |
| qosPolicySimpleCondition to<br>qosPolicyIPv6AddrValue           | Attachment or<br>qpValueAtom property of<br>qosPolicySimpleCondition                               |
| qosPolicySimpleCondition to<br>qosPolicyMACAddrValue            | Attachment or<br>qpValueAtom property of<br>qosPolicySimpleCondition                               |
| qosPolicySimpleCondition to<br>qosPolicyStringValue             | Attachment or<br>qpValueAtom property of                                                           |

|                             |                          |         |
|-----------------------------|--------------------------|---------|
|                             | qosPolicySimpleCondition |         |
| +-----+                     | +-----+                  | +-----+ |
| qosPolicySimpleCondition to | Attachment or            |         |
| qosPolicyBitStringValue     | qpValueAtom property of  |         |
|                             | qosPolicySimpleCondition |         |
| +-----+                     | +-----+                  | +-----+ |

(table is continued on the next page)

Snir, Ramberg, Strassner, Cohen expires September 2000 23  
Draft-ietf-policy-qos-schema-01.txt April 2000

(table is continued from the previous page)

|                             |                             |         |
|-----------------------------|-----------------------------|---------|
| +-----+                     | +-----+                     | +-----+ |
| qosPolicySimpleCondition to | Attachment or               |         |
| qosPolicyDNValue            | qpValueAtom property of     |         |
|                             | qosPolicySimpleCondition    |         |
| +-----+                     | +-----+                     | +-----+ |
| qosPolicySimpleCondition to | Attachment or               |         |
| qosPolicyAttributeValue     | qpValueAtom property of     |         |
|                             | qosPolicySimpleCondition    |         |
| +-----+                     | +-----+                     | +-----+ |
| qosPolicySimpleCondition to | Attachment or               |         |
| qosPolicyIntegerValue       | qpValueAtom property of     |         |
|                             | qosPolicySimpleCondition    |         |
| +-----+                     | +-----+                     | +-----+ |
| policyRule to               |                             |         |
| policyRuleActionAssociation | DIT containment             |         |
| +-----+                     | +-----+                     | +-----+ |
| policyRuleActionAssociation | Attachment or               |         |
| to qosPolicyPRAction        | policyActionDN property of  |         |
|                             | policyRuleActionAssociation |         |
| +-----+                     | +-----+                     | +-----+ |
| policyRuleActionAssociation | Attachment or               |         |
| to qosPolicyRSVPAction      | policyActionDN property of  |         |
|                             | policyRuleActionAssociation |         |
| +-----+                     | +-----+                     | +-----+ |
| qosPolicyPRAction to        | Attachment or               |         |
| qosPolicyPRTrfcProf         | qpTrfcProf property of      |         |
|                             | qosPolicyPRAction           |         |
| +-----+                     | +-----+                     | +-----+ |
| qosPolicyPRAction to        | Attachment or               |         |
| qosPolicyMeter              | qpMeter property of         |         |
|                             | qosPolicyPRAction           |         |
| +-----+                     | +-----+                     | +-----+ |
| qosPolicyRSVPAction to      | Attachment or               |         |
| qosPolicyRSVPTTrfcProf      | qpTrfcProf property of      |         |

|                               |                                |  |
|-------------------------------|--------------------------------|--|
|                               | qosPolicyRSVPAction            |  |
| qosPolicyRSVPAction to        | Attachment or                  |  |
| qosPolicyRSVPInstallAction    | qpInstallAction property of    |  |
|                               | qosPolicyRSVPAction            |  |
| qosPolicyRSVPAction to        | Attachment or                  |  |
| qosPolicyRSVPSignalCtrlAction | qpSignalCtrlAction property of |  |
|                               | qosPolicyRSVPAction            |  |

(table is continued on the next page)

Snir, Ramberg, Strassner, Cohen expires September 2000 24  
Draft-ietf-policy-qos-schema-01.txt April 2000

(table is continued from the previous page)

|                        |                     |  |
|------------------------|---------------------|--|
| qosPolicyRSVPAction to | Attachment or       |  |
| qosPolicyMeter         | qpMeter property of |  |
|                        | qosPolicyRSVPAction |  |
| policyInstance to      | Attachment          |  |
| qosPolicyPRTrfcProf    |                     |  |
| policyInstance to      | Attachment          |  |
| qosPolicyRSVPTrfcProf  |                     |  |

Table 1. Relationship between classes defined in this draft and [\[QOSIM\]](#)

## 8. Class Definitions

This section contains the class and attribute definitions for this schema. All class and attribute definitions for classes that are defined in either the Policy Core Information Model ([\[PCIM\]](#)) or the QoS Policy Information Model ([\[QOSIM\]](#)) are noted here; however, the definitions of these classes and attributes remain in their respective original documents.

There are two general notes that apply to this section. First, object and attribute OIDs have not been assigned yet. Until their assignment, these will be represented by the following nomenclature:

<qos-oc-#> for object class OIDs and <qos-at-#> for attribute OIDs.

The second note is that some vendor implementations do not allow for one auxiliary class to be derived from another auxiliary class, even though the LDAP and X.500 specs do provide this. Let's call the auxiliary class that is to be derived from an auxiliary class aux-b, and the auxiliary class that is the superclass of aux-b aux-a. There are two possible solutions. The first is to derive aux-b from Top. The problem with this approach is that now, aux-a and aux-b must both be attached to the same class. This is potentially dangerous, as the developer must be explicitly aware of the attributes defined in each of the auxiliary classes so as to avoid attribute name conflicts. One possible solution to this problem is to use different prefixes for the attribute names of aux-a and aux-b.

The second approach is to define a third auxiliary class, aux-c, that contains all of the attributes defined in aux-a and aux-b, and to define a set of unique prefixes for all of the attributes of aux-c.

Snir, Ramberg, Strassner, Cohen      expires September 2000      25  
Draft-ietf-policy-qos-schema-01.txt      April 2000

### **8.1. Class policyGroup**

This class represents the root of the subtree that contains QoS policy information. The policyGroup object contains the references to the repositories that it uses and to the policy definition information that it needs to represent policies. This class is defined in [[PCIM](#)].

### **8.2. Class policyRepository**

This class represents the root (i.e., the top of the subtree) of the QoS policy repository. The policyRepository object contains the DNS of the specific repositories that contain reusable policy information. This class is defined in [[PCIM](#)].

### **8.3. Class policyRule**

This class represents the "If Condition then Action" semantics associated with a policy. This class is defined in [[PCIM](#)].

### **8.4. Class qosPolicyDomain**

This class defines a single administrative QoS policy domain, and contains the domain's policy rules and definitions. This enables the administrator to partition the set of QoS information into different

domains, where each domain may have a potentially different set of policies, access rules, decision strategy or other application of the policy information organized in some fashion (which is represented by the domain) that reflects distinct administrative control (compared to the rest of the DIT). The policyGroup object points to a subtree in the DIT that contains policy information, and each qosPolicyDomain object points to a specific subsection of that subtree that contains specialized policy information. The class definition is as follows:

```

NAME qosPolicyDomain
DERIVED FROM policyGroup (defined in [PCIM])
TYPE Structural
AUXILIARY CLASSES policyGroupContainmentAuxClass,
 policyRuleContainmentAuxClass, policyElementAuxClass,
 (all of these are defined in [PCIM]), and
 qosPolicyElementAuxClass (defined in this document)
OID <tbid>
MUST
MAY qpDomainName, qpPHBSet, qpPolicyRuleMatchMethod

```

Snir, Ramberg, Strassner, Cohen      expires September 2000      26  
Draft-ietf-policy-qos-schema-01.txt      April 2000

The corresponding ABNF is:

```

(<qos-oc-1> NAME 'qosPolicyDomain'
DESC 'A class that is the root of an administrative QoS policy domain,
which resides in the policyGroup container. It contains a group of
named policy containers.'
SUP policyGroup
MAY (qpDomainName $ qpPHBSet $ qpPolicyRuleMatchMethod)
)

```

The following DIT content rule indicates that an instance of the qosPolicyDomain class may have attached to it any of the following four auxiliary classes (or one of their subclasses):  
policyGroupContainmentAuxClass, policyRuleContainmentAuxClass,  
policyElementAuxClass, or qosPolicyElementAuxClass.

```

(<qos-oc-1> NAME 'qosPolicyDomain'
DESC 'Auxiliary classes that may be attached to qosPolicyDomain'
AUX (policyGroupContainmentAuxClass $ policyRuleContainmentAuxClass $
policyElementAuxClass $ qosPolicyElementAuxClass)
)

```

#### **8.4.1. The Attribute qpDomainName**

The purpose of this attribute is to define a user-friendly name of the QoS policy domain. This is the name that users can use to refer to this domain, and not necessarily the fully qualified distinguished name of this attribute. The attribute is defined as follows:

|               |                   |
|---------------|-------------------|
| NAME          | qpDomainName      |
| SYNTAX        | IA5String         |
| OID           | <tdb>             |
| EQUALITY      | CaseExactIA5Match |
| MULTI-VALUED  | No                |
| DEFAULT VALUE | NULL              |

The corresponding ABNF is:

```
(<qos-at-1> NAME 'qpDomainName'
DESC 'A user-friendly name of the QoS policy domain. Default value is
NULL.'
SYNTAX IA5String SINGLE-VALUE
EQUALITY CaseExactIA5Match
)
```

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 27         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

#### **8.4.2. The Attribute qpPHBSet**

This attribute is a distinguished name, and enables the PHB set defined for this domain to be referenced from this QoS domain. The attribute is defined as follows:

|               |                        |
|---------------|------------------------|
| NAME          | qpPHBSet               |
| SYNTAX        | DistinguishedName      |
| OID           | <tdb>                  |
| EQUALITY      | DistinguishedNameMatch |
| MULTI-VALUED  | YES                    |
| DEFAULT VALUE | NULL                   |

The corresponding ABNF is:

```
(<qos-at-2> NAME 'qpPHBSet'
DESC 'DN reference to the PHB set defined for the domain. Default value
```

```

is NULL.'
SYNTAX DistinguishedName
EQUALITY DistinguishedNameMatch
)

```

#### 8.4.3. The Attribute qpPolicyRuleMatchMethod

This attribute is an enumerated integer that defines the matching strategy to be applied on the set of QoS policy rules within the domain. The attribute definition is specified in [section 8.5.2](#).

#### 8.5. Class qosNamedPolicyContainer

This class represents an administrative policy rule container. All policies serving a certain goal, servicing a certain type of application, handling a certain type of flow or devices are administrated in a particular qosNamedPolicyContainer. This enables multiple levels of scoping to be applied: high-level policy aggregation through the policyGroup or qosPolicyDomain classes, and finer-level refinement of policies through instances of the qosNamedPolicyContainer classes. The class definition is as follows:

```

NAME qosNamedPolicyContainer
DERIVED FROM policyGroup (defined in [PCIM])
TYPE Structural
AUXILIARY CLASSES policyRuleContainmentAuxClass, policyElementAuxClass,
 (these are both defined in [PCIM]), and
 qosPolicyElementAuxClass (defined in this document)
OID <tbid>
MUST qpPriority, qpPolicyRuleMatchMethod
MAY

```

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 28         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF is:

```

(<qos-oc-2> NAME 'qosNamedPolicyContainer'
DESC 'A class that is a logical and physical container of policies.'
SUP policyGroup
MUST (qpPriority $ qpPolicyRuleMatchMethod)
)

```

The following DIT content rule indicates that an instance of the qosPolicyDomain class may have attached to it any of the following three auxiliary classes (or one of their subclasses):  
policyRuleContainmentAuxClass, policyElementAuxClass, or

qosPolicyElementAuxClass.

```
(<qos-oc-2> NAME 'qosNamedPolicyContainer'
DESC 'Auxiliary classes that may be attached'
AUX (policyRuleContainmentAuxClass $ policyElementAuxClass $
qosPolicyElementAuxClass)
)
```

#### **8.5.1. The Attribute qpPriority**

This attribute defines the priority of a named group of rules that are resident in one qosPolicyNamedContainer compared to other qosPolicyNamedContainer instances. If two or more qosPolicyNamedContainer objects have the same priority, this means that the order between these containers is of no importance, but that they must each be evaluated before other objects that have a numerically lower priority.

The attribute is defined as follows:

|               |              |
|---------------|--------------|
| NAME          | qpPriority   |
| SYNTAX        | Integer      |
| OID           | <tbd>        |
| EQUALITY      | IntegerMatch |
| MULTI-VALUED  | No           |
| DEFAULT VALUE | NULL         |

The corresponding ABNF is:

Snir, Ramberg, Strassner, Cohen      expires September 2000      29  
Draft-ietf-policy-qos-schema-01.txt      April 2000

```
(<qos-at-3> NAME 'qpPriority'
DESC 'The priority of a named group of rules in one
qosPolicyNamedContainer instance compared to other
qosPolicyNamedContainer instances. If two or more
qosPolicyNamedContainer objects have the same priority, this means that
the order between these containers is of no importance, but that they
must each be evaluated before other objects that have a numerically
lower priority. Default value is NULL.'
```



```
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### 8.5.2. The Attribute qpPolicyRuleMatchMethod

This attribute is an enumerated integer that defines the matching strategy to be applied on this set of QoS policy rules. The attribute is defined as follows:

```
NAME qpPolicyRuleMatchMethod
SYNTAX Integer (ENUM)
 { "FIRST MATCH " = 0; "MATCH ALL " = 1 }
OID <tbid>
EQUALITY IntegerMatch
MULTI-VALUED No
DEFAULT VALUE 0
```

The corresponding ABNF is:

```
(<qos-at-4> NAME 'qpPolicyRuleMatchMethod'
DESC 'The decision strategy to be applied on this set of qos policy
rules by policy servers. Values are: 0="First Match", 1="Match All".
Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### 8.6. Class qosPolicyPRAction

This class defines DiffServ-specific actions to be applied on a flow, including the (re)marking of a DSCP value, along with policing and shaping actions that need to be performed. The semantics of this class require that instances of the auxiliary classes qosPolicyPRTTrfcProf and qosPolicyMeter SHOULD be attached to whatever structural class that the instance of this class (qosPolicyPRAction) is attached to.

The class definition is as follows:

Snir, Ramberg, Strassner, Cohen      expires September 2000      30  
Draft-ietf-policy-qos-schema-01.txt      April 2000

```
NAME qosPolicyPRAction
DESCRIPTION A class that defines provisioning DiffServ
 Traffic actions to be applied on a specific
```

flow or group of flows, if a certain rule's  
 condition is met.  
 DERIVED FROM      policyActionAuxClass (defined in [[PCIM](#)])  
 TYPE                      Auxiliary  
 AUXILIARY CLASSES  
 OID                      <tbid>  
 MUST  
 MAY                      qpDirection, qpSetDSCPvalue, qpMeter,  
                          qpMeterScope, qpPRTTrfcProf, qpOutOfProfileAction,  
                          qpOutOfProfileRemarkValue

The corresponding ABNF is:

```

(<qos-oc-3> NAME 'qosPolicyPRAction'
DESC 'A class that defines provisioning DiffServ Traffic actions to be
applied on a specific flow or group of flows, if the condition of a
certain rule is met.'
SUP policyActionAuxClass AUXILIARY
MAY (qpDirection $ qpSetDSCPvalue $ qpMeter $ qpMeterScope
$ qpPRTTrfcProf $ qpOutOfProfileRemarkValue)
)

```

### **8.6.1. The Attribute qpDirection**

This attribute is an enumerated integer that defines whether this  
 action should be applied on the incoming and/or outgoing interface.  
 Note that incoming and outgoing interface is handled by keeping the  
 enumerated values simple (e.g., IN or OUT) and simply enabling multiple  
 values to be defined.

NAME                      qpDirection  
 SYNTAX                  Integer    (ENUM) {IN=0,OUT=1}  
 OID                      <tbid>  
 EQUALITY                  IntegerMatch  
 MULTI-VALUED          Yes  
 DEFAULT VALUE          0

The corresponding ABNF is:

```

(<qos-at-5> NAME 'qpDirection'
DESC 'this attribute defines the direction of the action (e.g., the
incoming and/or outgoing interfaces). Values are 0=In, 1=Out. Default
value is 0.'
SYNTAX Integer
EQUALITY IntegerMatch
)

```

### **8.6.2. The Attribute qpSetDSCPvalue**

This attribute defines the DSCP value of the marking action. Its definition is as follows:

|               |                |
|---------------|----------------|
| NAME          | qpSetDSCPvalue |
| SYNTAX        | Integer        |
| OID           | <tdb>          |
| EQUALITY      | IntegerMatch   |
| MULTI-VALUED  | No             |
| DEFAULT VALUE | 0              |

The corresponding ABNF is:

```
(<qos-at-6> NAME 'qpSetDSCPvalue'
DESC 'This attribute defines the DSCP value of the mark action. Default
value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### **8.6.3. The Attribute qpMeter**

This attribute is a DN that references a qosPolicyMeter object to be used in this provisioning action.

|               |                        |
|---------------|------------------------|
| NAME          | qpMeter                |
| SYNTAX        | DistinguishedName      |
| OID           | <tdb>                  |
| EQUALITY      | DistinguishedNameMatch |
| MULTI-VALUED  | No                     |
| DEFAULT VALUE | NULL                   |

The corresponding ABNF is:

```
(<qos-at-7> NAME 'qpMeter'
DESC 'A DN reference to a qosPolicyMeter object used in this
provisioning action. Default value is 0.'
SYNTAX DistinguishedName SINGLE-VALUE
EQUALITY DistinguishedNameMatch
)
```

#### **8.6.4. The Attribute qpMeterScope**

This attribute is an enumerated integer that defines what the scope of the metering action is (i.e., to what does the meter apply to). The attribute definition is as follows:

|               |                                              |
|---------------|----------------------------------------------|
| NAME          | qpMeterScope                                 |
| SYNTAX        | Integer ENUM (flow=0, interface=1, device=2) |
| OID           | <tbd>                                        |
| EQUALITY      | IntegerMatch                                 |
| MULTI-VALUED  | No                                           |
| DEFAULT VALUE | 0                                            |

The corresponding ABNF is:

```
(<qos-at-8> NAME 'qpMeterScope'
DESC 'An integer that defines the scope of the metering action. Values
are 0=flow, 1=interface, 2=device. Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.6.5. The Attribute qpPRTrfcProf**

This attribute is a DN that references another object that contains the DiffServ provisioning and policing values. The attribute is defined as follows:

|               |                        |
|---------------|------------------------|
| NAME          | qpPRTrfcProf           |
| SYNTAX        | DistinguishedName      |
| OID           | <tbd>                  |
| EQUALITY      | DistinguishedNameMatch |
| MULTI-VALUED  | No                     |
| DEFAULT VALUE | NULL                   |

The corresponding ABNF is:

```
(<qos-at-9> NAME 'qpPRTrfcProf'
DESC 'This attribute contains the DiffServ / provisioning policing
instruction value, defined as a DN reference to a qosPolicyTrfcProf
entry. Default value is NULL.'
SYNTAX DistinguishedName SINGLE-VALUE
EQUALITY DistinguishedNameMatch
)
```

#### **8.6.6. The Attribute qpOutOfProfileAction**

This attribute is an enumerated integer that defines the action to be applied to out of profile packets. The attribute definition is as follows:

|               |                                             |
|---------------|---------------------------------------------|
| NAME          | qpOutOfProfileAction                        |
| SYNTAX        | Integer [ENUM] {SHAPE=0,DISCARD=1,REMARK=2} |
| OID           | <tdb>                                       |
| EQUALITY      | IntegerMatch                                |
| MULTI-VALUED  | No                                          |
| DEFAULT VALUE | 0                                           |

The corresponding ABNF is:

```
(<qos-at-10> NAME 'qpOutOfProfileAction'
DESC 'The action to be applied to out of profile packets, as defined in
the DiffServPolicer entry. Values are 0=shape, 1=discard, 2=remark.
Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.6.7. The Attribute qpOutOfProfileRemarkValue**

This attribute is an integer that contains the DSCP value to be applied to out of profile packets, if the OutOfProfile attribute action is defined as remark. The attribute definition is as follows:

|               |                           |
|---------------|---------------------------|
| NAME          | qpOutOfProfileRemarkValue |
| SYNTAX        | Integer                   |
| OID           | <tdb>                     |
| EQUALITY      | IntegerMatch              |
| MULTI-VALUED  | No                        |
| DEFAULT VALUE | 0                         |

The corresponding ABNF is:

```
(<qos-at-12> NAME 'qpOutOfProfileRemarkValue'
```

```

DESC 'The DSCP value to be applied to out of profile packets if the
OutOfProfile attribute action is defined as REMARK. Default value is
0.'
```

|                             |  |
|-----------------------------|--|
| SYNTAX Integer SINGLE-VALUE |  |
| EQUALITY IntegerMatch       |  |
| )                           |  |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 34         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

### **8.7. Class qosPolicyRSVPAction**

This class defines a policy action to be applied on RSVP signaling messages that match the rule condition. The semantics of this class require that instances of the auxiliary classes qosPolicyRSVPTrfcProf, qosPolicyRSVPSignalCtrlAction and qosPolicyRSVPInstallAction SHOULD be attached to whatever structural class that the instance of this class (qosPolicyRSVPAction) is attached to.

The class definition is as follows:

|                   |                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NAME              | qosPolicyRSVPAction                                                                                                                                               |
| DERIVED FROM      | policyActionAuxClass (defined in [ <a href="#">PCIM</a> ])                                                                                                        |
| TYPE              | Auxiliary                                                                                                                                                         |
| AUXILIARY CLASSES |                                                                                                                                                                   |
| OID               | <tbd>                                                                                                                                                             |
| MUST              |                                                                                                                                                                   |
| MAY               | qpRSVPDirection, qpRSVPMessageType, qpRSVPStyle,<br>qpRSVPServiceType, qpRSVPInstallAction,<br>qpRSVPCtrlAction, qpRSVPMeter, qpRSVPMeterScope,<br>qpRSVPTrfcProf |

The corresponding ABNF is:

```

(<qos-oc-4> NAME 'qosPolicyRSVPAction'
DESC 'A class that defines an RSVP action to be performed if a certain
rule's condition is met.'
SUP policyActionAuxClass AUXILIARY
MAY (qpRSVPDirection $ qpRSVPMessageType $ qpRSVPStyle $
qpRSVPServiceType $ qpRSVPInstallAction $ qpRSVPCtrlAction
$ qpRSVPMeter $ qpRSVPMeterScope $ qpRSVPTrfcProf)
)
```

#### **8.7.1. The Attribute qpRSVPDirection**

This attribute is an enumerated integer that defines whether this action is applied to the incoming and/or outgoing interface. Note that the syntax is kept simple (IN or OUT) and instead the attribute is allowed to have multiple values. The definition of the attribute is as follows:

|               |                             |
|---------------|-----------------------------|
| NAME          | qpRSVPDirection             |
| SYNTAX        | Integer (ENUM) {IN=0,OUT=1} |
| OID           | <tdb>                       |
| EQUALITY      | IntegerMatch                |
| MULTI-VALUED  | Yes                         |
| DEFAULT VALUE | 0                           |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 35         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF is:

```
(<qos-at-54> NAME 'qpRSVPDirection'
DESC 'this attribute defines the direction of the action (e.g., the
incoming and/or outgoing interfaces). Values are 0=In, 1=Out. Default
value is 0.'
SYNTAX Integer
EQUALITY IntegerMatch
)
```

#### **8.7.2. The Attribute qpRSVPMessageType**

This attribute is an enumerated integer that defines the type of RSVP message to be handled. The attribute definition is as follows:

|               |                                                      |
|---------------|------------------------------------------------------|
| NAME          | qpRSVPMessageType                                    |
| SYNTAX        | Integer (ENUM) { Path=0,Resv=1,ResvErr=2,PathErr=3 } |
| OID           | <tdb>                                                |
| EQUALITY      | IntegerMatch                                         |
| MULTI-VALUED  | NO                                                   |
| DEFAULT VALUE | 0                                                    |

The corresponding ABNF is:

```
(<qos-at-13> NAME 'qpRSVPMessageType'
DESC 'This attribute defines the type of RSVP message to be handled.
Values are 0=Path, 1=Resv, 2=ResvErr, 3=PathErr. Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
```

)

### **8.7.3. The Attribute qpRSVPStyle**

This attribute is an enumerated integer that limits the scope of the action to be enforced only on RSVP Requests with the specified reservation style. The definition of this attribute is as follows:

|               |                                   |
|---------------|-----------------------------------|
| NAME          | qpRSVPStyle                       |
| SYNTAX        | Integer (ENUM) {SE=0, FF=1, WF=2} |
| OID           | <tdb>                             |
| EQUALITY      | IntegerMatch                      |
| MULTI-VALUED  | YES                               |
| DEFAULT VALUE | 0                                 |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 36         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF is:

```
(<qos-at-14> NAME 'qpRSVPStyle'
DESC 'This Property limits the scope of the action to be enforced only
on RSVP Requests with the specified reservation style. The allowed
styles are Shared Explicit (SE), Fixed Filter (FF) and Wildcard Filter
(WF) as defined in RSVP. Values are 0=SE, 1=FF, 2=WF. Default value is
0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### **8.7.4. The Attribute qpRSVPServiceType**

This attribute is an enumerated integer that limits the scope of the action to be enforced only on RSVP Requests asking for a specified integrated service type. The attribute definition is as follows:

|              |                                                                      |
|--------------|----------------------------------------------------------------------|
| NAME         | qpRSVPServiceType                                                    |
| SYNTAX       | Integer (ENUM)<br>{ControlledLoad =0 , GuaranteedService =1, NULL=2} |
| OID          | <tdb>                                                                |
| EQUALITY     | IntegerMatch                                                         |
| MULTI-VALUED | YES                                                                  |



DEFAULT VALUE 0

The ABNF is as follows:

```
(<qos-at-15> NAME 'qpRSVPServiceType'
DESC 'this Property limits the scope of the action to be enforced only
on RSVP Requests asking for specified integrated service type. Values
are 0=ControlledLoad, 1=GuaranteedService, 2=NULL. Default value is 0.'
SYNTAX Integer
EQUALITY IntegerMatch
)
```

#### **8.7.5. The Attribute qpRSVPInstallAction**

This attribute is a DN that is used to reference a QosPolicyRSVPInstallAction object that SHOULD be used in conjunction with the RSVP reservation. The attribute definition is as follows:

|               |                        |
|---------------|------------------------|
| NAME          | qpRSVPInstallAction    |
| SYNTAX        | DistinguishedName      |
| OID           | <tdb>                  |
| EQUALITY      | DistinguishedNameMatch |
| MULTI-VALUED  | No                     |
| DEFAULT VALUE | NULL                   |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 37         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF description is as follows:

```
(<qos-at-16> NAME 'qpRSVPInstallAction'
DESC 'A DN reference to a QosPolicyRSVPInstallAction object used in
conjunction with the RSVP reservation. Default value is NULL.'
SYNTAX DistinguishedName SINGLE-VALUE
EQUALITY DistinguishedNameMatch
)
```

#### **8.7.6. The Attribute qpRSVPCtrlAction**

This attribute is a DN that references a separate object, of type qpRSVPCtrlAction. This object is used in conjunction with the RSVP reservation. The attribute definition is as follows:

|          |                        |
|----------|------------------------|
| NAME     | qpRSVPCtrlAction       |
| SYNTAX   | DistinguishedName      |
| OID      | <tdb>                  |
| EQUALITY | DistinguishedNameMatch |

MULTI-VALUED      No  
DEFAULT VALUE    NULL

The corresponding ABNF is as follows:

```
(<qos-at-17> NAME 'qpRSVPCtrlAction'
DESC 'A DN reference to a qpRSVPCtrlAction object used in conjunction
with the RSVP reservation. Default value is NULL.'
SYNTAX DistinguishedName SINGLE-VALUE
EQUALITY DistinguishedNameMatch
)
```

#### **8.7.7. The Attribute qpRSVPMeter**

This attribute is a DN reference to a separate object, of type qosPolicyMeter. This object is used to provide metering for this RSVP action.

NAME                      qpRSVPMeter  
SYNTAX                  DistinguishedName  
OID                      <tbd>  
EQUALITY                  DistinguishedNameMatch  
MULTI-VALUED      No  
DEFAULT VALUE    NULL

Snir, Ramberg, Strassner, Cohen      expires September 2000      38  
Draft-ietf-policy-qos-schema-01.txt      April 2000

The corresponding ABNF is as follows:

```
(<qos-at-55> NAME 'qpRSVPMeter'
DESC 'A DN reference to a qosPolicyMeter object used in this
provisioning action. Default value is NULL.'
SYNTAX DistinguishedName SINGLE-VALUE
EQUALITY DistinguishedNameMatch
)
```

#### **8.7.8. The Attribute qpRSVPMeterScope**

This attribute is an enumerated integer that defines the scope of the metering action (e.g., to what it should be applied to). The definition of this attribute is as follows:

|               |                                            |
|---------------|--------------------------------------------|
| NAME          | qpRSVPMeterScope                           |
| SYNTAX        | Integer ENUM {flow=0,interface=1 device=2} |
| OID           | <tdb>                                      |
| EQUALITY      | IntegerMatch                               |
| MULTI-VALUED  | No                                         |
| DEFAULT VALUE | 0                                          |

The corresponding ABNF is as follows:

```
(<qos-at-56> NAME 'qpRSVPMeterScope'
DESC 'An integer that defines the scope of the metering action. Values
are 0=flow, 1=interface, 2=device. Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.7.9. The Attribute qpRSVPTrfcProf**

This attribute is a DN that references a qosPolicyTrfcProf object that defines the desired RSVP provisioning action.

|               |                        |
|---------------|------------------------|
| NAME          | qpRSVPTrfcProf         |
| SYNTAX        | DistinguishedName      |
| OID           | <tdb>                  |
| EQUALITY      | DistinguishedNameMatch |
| MULTI-VALUED  | No                     |
| DEFAULT VALUE | NULL                   |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 39         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF is:

```
(<qos-at-57> NAME 'qpRSVPTrfcProf'
DESC 'This attribute contains the DiffServ / provisioning policing
instruction value, defined as a DN reference to a qosPolicyTrfcProf
entry. Default value is NULL.'
SYNTAX DistinguishedName SINGLE-VALUE
EQUALITY DistinguishedNameMatch
)
```

## 8.8. Class qosPolicyPRTTrfcProf

This class represents a provisioning traffic profile, which is used to define the policer or shaper rate values to be enforced on a flow or a set of flows. QosPolicyPRTTrfcProfs may be implemented as reusable or rule-specific objects; see [[QOSIM](#)] for more information. The class definition is as follows:

|                   |                                              |
|-------------------|----------------------------------------------|
| NAME              | qosPolicyPRTTrfcProf                         |
| DERIVED FROM      | Policy (defined in [ <a href="#">PCIM</a> ]) |
| TYPE              | Auxiliary                                    |
| AUXILIARY CLASSES |                                              |
| OID               | <tdb>                                        |
| MUST              |                                              |
| MAY               | qpPRRate, qpPRNormalBurst, qpPRExcessBurst   |

The corresponding ABNF definition is as follows:

```
(<qos-oc-5> NAME 'qosPolicyPRTTrfcProf'
DESC 'A class that defines the policer or shaper rate values to be
enforced on a flow or a set of flows.'
SUP Policy AUXILIARY
MAY (qpPRRate $ qpPRNormalBurst $ qpPRExcessBurst)
)
```

### 8.8.1. The Attribute qpPRRate

This attribute is an integer that specifies the token rate used for policing this flow or set of flows. The definition of this attribute is as follows:

|               |              |
|---------------|--------------|
| NAME          | qpPRRate     |
| SYNTAX        | Integer      |
| OID           | <tdb>        |
| EQUALITY      | IntegerMatch |
| MULTI-VALUED  | No           |
| DEFAULT VALUE | 0            |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 40         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF is as follows:

```
(<qos-at-18> NAME 'qpPRRate'
DESC 'The token rate used for policing this flow or set of flows. It is
specified in units of bits/second. A rate of zero means that all
packets will be out of profile. Default value is 0.'
```

```
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### 8.8.2. The Attribute qpPRNormalBurst

This attribute is an integer that specifies the normal size of a burst. It is defined as follows:

```
NAME qpPRNormalBurst
SYNTAX Integer
OID <tbid>
EQUALITY IntegerMatch
MULTI-VALUED No
DEFAULT VALUE 0
```

The corresponding ABNF is as follows:

```
(<qos-at-19> NAME 'qpPRNormalBurst'
DESC 'The normal size of a burst measured in bytes. Default value is
0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### 8.8.3. The Attribute qpPRExcessBurst

This attribute is an integer that defines the excess size of a burst. Its definition is as follows:

```
NAME qpPRExcessBurst
SYNTAX Integer
OID <tbid>
EQUALITY IntegerMatch
MULTI-VALUED No
DEFAULT VALUE 0
```

The corresponding ABNF is as follows:

```
(<qos-at-20> NAME 'qpPRExcessBurst'
DESC 'The excess size of a burst measured in bytes. Default value is
0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

## 8.9. Class qosPolicyRSVPTrfcProf

This class represents an IntServ RSVP traffic profile. Values of RSVP policers are compared against the Traffic specification (TSPEC) and QoS Reservation requests (RSPEC) carried in RSVP requests. The qosPolicyRSVPTrfcProf class may be implemented as a reusable or as a rule-specific object; see [QOSIM] for more information. The class definition is as follows:

|                   |                                                                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| NAME              | qosPolicyRSVPTrfcProf                                                                                                                      |
| DERIVED FROM      | Policy (defined in [PCIM])                                                                                                                 |
| TYPE              | Auxiliary                                                                                                                                  |
| AUXILIARY CLASSES |                                                                                                                                            |
| OID               | <tdb>                                                                                                                                      |
| MUST              |                                                                                                                                            |
| MAY               | qpRSVPTokenRate, qpRSVPPeakRate, qpRSVPBucketSize,<br>qpRSVPResvRate, qpRSVPResvSlack, qpRSVPSessionNum,<br>qpMinPolicedUnit, qpMaxPktSize |

The corresponding ABNF is as follows:

```
(<qos-oc-6> NAME 'qosPolicyRSVPTrfcProf'
DESC 'A class that defines rate limiting values for QoS requests for a
flow or a set of flow via RSVP'
SUP Policy AUXILIARY
MAY (qpRSVPTokenRate $ qpRSVPPeakRate $ qpRSVPBucketSize
$ qpRSVPResvRate $ qpRSVPResvSlack $ qpRSVPSessionNum
$ qpMinPolicedUnit $ qpMaxPktSize)
)
```

#### **8.9.1. The Attribute qpRSVPTokenRate**

This attribute is an integer that defines the token rate parameter for RSVP flows. The attribute definition is as follows:

|               |                 |
|---------------|-----------------|
| NAME          | qpRSVPTokenRate |
| SYNTAX        | Integer         |
| OID           | <tdb>           |
| EQUALITY      | IntegerMatch    |
| MULTI-VALUED  | No              |
| DEFAULT VALUE | 0               |

The corresponding ABNF is:

```
(<qos-at-21> NAME 'qpRSVPTokenRate'
DESC 'Token Rate parameter, measured in bits/sec. Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### **8.9.2. The Attribute qpRSVPPeakRate**

This attribute is an integer that defines the peak rate parameter for RSVP flows. The attribute definition is as follows:

|               |                |
|---------------|----------------|
| NAME          | qpRSVPPeakRate |
| SYNTAX        | Integer        |
| OID           | <tbid>         |
| EQUALITY      | IntegerMatch   |
| MULTI-VALUED  | No             |
| DEFAULT VALUE | 0              |

The corresponding ABNF is:

```
(<qos-at-22> NAME 'qpRSVPPeakRate. Default value is 0.'
DESC 'Peak rate parameter, measured in bits/sec'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### **8.9.3. The Attribute qpRSVPBucketSize**

This attribute is an integer that defines the maximal allowed bucket size. The attribute definition is as follows:

|               |                  |
|---------------|------------------|
| NAME          | qpRSVPBucketSize |
| SYNTAX        | Integer          |
| OID           | <tbid>           |
| EQUALITY      | IntegerMatch     |
| MULTI-VALUED  | No               |
| DEFAULT VALUE | 0                |

The corresponding ABNF is:

```
(<qos-at-23> NAME 'qpRSVPBucketSize. Default value is 0.'
DESC 'Bucket Size, measured in bytes'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### **8.9.4. The Attribute qpRSVPResvRate**

This attribute is an integer that defines the rate for this conversation. This is the R-Spec parameter in the RSVP Guaranteed service reservation.

The attribute is defined as follows:

|               |                |
|---------------|----------------|
| NAME          | qpRSVPResvRate |
| SYNTAX        | Integer        |
| OID           | <tdb>          |
| EQUALITY      | IntegerMatch   |
| MULTI-VALUED  | NO             |
| DEFAULT VALUE | 0              |

The corresponding ABNF is:

```
(<qos-at-24> NAME 'qpRSVPResvRate'
DESC 'Defines the RSVP Rate. This is the R-Spec parameter in the RSVP
Guaranteed service reservation. Measured in bits/sec. Default value is
0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.9.5. The Attribute qpRSVPResvSlack**

This attribute is an integer that defines the RSVP Slack Term parameter in the RSVP      Guaranteed service reservation. The attribute is defined as follows:

|               |                 |
|---------------|-----------------|
| NAME          | qpRSVPResvSlack |
| SYNTAX        | Integer         |
| OID           | <tdb>           |
| EQUALITY      | IntegerMatch    |
| MULTI-VALUED  | NO              |
| DEFAULT VALUE | 0               |

The corresponding ABNF is:

```
(<qos-at-25> NAME 'qpRSVPResvSlack'
DESC 'Defines the RSVP Slack Term parameter in the RSVP Guaranteed
service reservation. Measured in microseconds. Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.9.6. The Attribute qpRSVPSessionNum**



This attribute is an integer that defines the total number of allowed active RSVP sessions. It is defined as follows:

|               |                  |
|---------------|------------------|
| NAME          | qpRSVPSessionNum |
| SYNTAX        | Integer          |
| OID           | <tdb>            |
| EQUALITY      | IntegerMatch     |
| MULTI-VALUED  | No               |
| DEFAULT VALUE | 0                |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 44         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF is:

```
(<qos-at-26> NAME 'qpRSVPSessionNum'
DESC 'The total number of allowed active RSVP sessions. Default value
is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.9.7. The Attribute qpMinPolicedUnit**

This attribute is an integer that defines the RSVP minimum policed unit, measured in bytes. Its definition is:

|               |                  |
|---------------|------------------|
| NAME          | qpMinPolicedUnit |
| SYNTAX        | Integer          |
| OID           | <tdb>            |
| EQUALITY      | IntegerMatch     |
| MULTI-VALUED  | NO               |
| DEFAULT VALUE | 0                |

The corresponding ABNF is:

```
(<qos-at-27> NAME 'qpMinPolicedUnit'
DESC 'Defines the RSVP minimum policed unit, measured in bytes. Default
value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.9.8. The Attribute qpMaxPktSize**

This attribute is an integer that defines the RSVP maximum allowed packet size. It is defined as follows:

|      |              |
|------|--------------|
| NAME | qpMaxPktSize |
|------|--------------|

```

SYNTAX Integer
OID <td>
EQUALITY IntegerMatch
MULTI-VALUED NO
DEFAULT VALUE 0

```

The corresponding ABNF is:

```

(<qos-at-28> NAME 'qpMaxPktSize'
DESC 'Defines the RSVP maximum allowed packet size, measured in bytes.
Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)

```

Snir, Ramberg, Strassner, Cohen      expires September 2000      45  
Draft-ietf-policy-qos-schema-01.txt      April 2000

### **8.10. Class qosPolicyRSVPSignalCtrlAction**

This class extends the functionality of the qosPolicyRSVPAction class by adding detailed control on the signaling protocol behavior itself. The information carried in RSVP messages can be modified using this action, as well as the RSVP forwarding behavior. This class can be extended to support replacement of additional objects in RSVP messages, beyond the replacement of the DCLASS and PREEMPTION objects that are defined below.

An instance of this class SHOULD be attached to an object together with an instance of the qosPolicyRSVPAction class.

```

NAME qosPolicyRSVPSignalCtrlAction
DESCRIPTION Actions modifying the behavior and content of RSVP
signaling flows.
DERIVED FROM policyActionAuxClass (defined in [PCIM])
TYPE Auxiliary
AUXILIARY CLASSES
OID <td>
MUST
MAY qpForwardingMode, qpSendError, qpReplaceDSCP,
qpReplacePreemptionPriority,
qpReplaceDefendingPriority

```

The corresponding ABNF is:

```

(<qos-oc-7> NAME 'qosPolicyRSVPSignalCtrlAction'
DESC 'Actions modifying the behavior and content of RSVP
Signaling flows.'
SUP 'policyActionAuxClass' AUXILIARY
MAY (qpForwardingMode $ qpSendError $ qpReplaceDSCP
$ qpReplacePreemptionPriority $ qpReplaceDefendingPriority)

```

)

#### **8.10.1. The Attribute qpForwardingMode**

This attribute controls forwarding of RSVP messages. If the mode is set to proxy, an RSVP Path messages is not forwarded and a Resv message is returned as if the Resv was returned by the receiver. The attribute definition is as follows:

|               |                                                      |
|---------------|------------------------------------------------------|
| NAME          | qpForwardingMode                                     |
| DESCRIPTION   | Defines whether to forward or return RSVP signaling. |
| SYNTAX        | Integer (ENUM) {Forward=0 , Proxy=1}                 |
| OID           | <tdb>                                                |
| EQUALITY      | IntegerMatch                                         |
| MULTI-VALUED  | No                                                   |
| DEFAULT VALUE | 0                                                    |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 46         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF is:

```
(<qos-at-29> NAME 'qpForwardingMode'
DESC 'Defines whether to forward or return RSVP signaling.
Values are 0=Forward, 1=Proxy. Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.10.2. The Attribute qpSendError**

This attribute controls generation of Resv-Err and Path-Err messages as defined in [[COPSRSVP](#)]. The attribute definition is as follows:

|               |                                                            |
|---------------|------------------------------------------------------------|
| NAME          | qpSendError                                                |
| DESCRIPTION   | Defines whether to send an RSVP error and warning message. |
| SYNTAX        | Integer {No=0, Yes=1}                                      |
| OID           | <tdb>                                                      |
| EQUALITY      | IntegerMatch                                               |
| MULTI-VALUED  | No                                                         |
| DEFAULT VALUE | 0                                                          |

The corresponding ABNF is:

```
(<qos-at-30> NAME 'qpSendError'
DESC 'Defines whether to send an RSVP error and warning
```

message. Values are 0=No, 1=Yes. Default value is 0.'

SYNTAX Integer SINGLE-VALUE

EQUALITY IntegerMatch

)

### **8.10.3. The Attribute qpReplaceDSCP**

This attribute controls the replacement of a DCLASS object that carries a DSCP value which is carried in an RSVP message. Its attribute definition is as follows:

|               |                                                                                                                                                                                                   |  |  |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
| NAME          | qpReplaceDSCP                                                                                                                                                                                     |  |  |
| DESCRIPTION   | This attribute controls the replacement or addition of a DCLASS object holding a DSCP value carried in an RSVP message. The attribute specifies the DSCP value to be carried in the RSVP message. |  |  |
| SYNTAX        | Integer                                                                                                                                                                                           |  |  |
| OID           | <tdb>                                                                                                                                                                                             |  |  |
| EQUALITY      | IntegerMatch                                                                                                                                                                                      |  |  |
| MULTI-VALUED  | No                                                                                                                                                                                                |  |  |
| DEFAULT VALUE | 0                                                                                                                                                                                                 |  |  |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 47         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF is:

```
(<qos-at-31> NAME 'qpReplaceDSCP'
DESC This attribute controls the replacement or addition of a DCLASS
object holding a DSCP value which is carried in an RSVP message. The
Attribute specifies the DSCP value to be carried in the RSVP message.
Default value is 0.
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### **8.10.4. The Attribute qpReplacePreemptionPriority**

This attribute allows replacing or adding of preemption priority [[RSVP\\_PREEMP](#)] objects to RSVP messages.

|             |                                                                                                      |  |  |
|-------------|------------------------------------------------------------------------------------------------------|--|--|
| NAME        | qpReplacePreemptionPriority                                                                          |  |  |
| DESCRIPTION | A positive integer value specifying the preemption priority that should be carried by RSVP messages. |  |  |
| SYNTAX      | Integer                                                                                              |  |  |
| OID         | <tdb>                                                                                                |  |  |
| EQUALITY    | IntegerMatch                                                                                         |  |  |

MULTI-VALUED      No  
DEFAULT VALUE    0

The corresponding ABNF is:

```
(<qos-at-32> NAME 'qpReplacePreemptionPriority'
DESC 'A positive integer value specifying the preemption
priority that should be carried by RSVP messages.
Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.10.5. The Attribute qpReplaceDefendingPriority**

This attribute allows replacing or adding of preemption priority  
[[RSVP\\_PREEMP](#)] objects to RSVP messages.

|               |                                                                                                                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NAME          | qpReplaceDefendingPriority                                                                                                                                                                            |
| DESCRIPTION   | This attribute allows replacing or adding of<br>preemption priority [ <a href="#">RSVP_PREEMP</a> ] objects to RSVP<br>messages. It specifies the defending priority within<br>the preemption object. |
| SYNTAX        | Integer                                                                                                                                                                                               |
| OID           | <tbid>                                                                                                                                                                                                |
| EQUALITY      | IntegerMatch                                                                                                                                                                                          |
| MULTI-VALUED  | No                                                                                                                                                                                                    |
| DEFAULT VALUE | 0                                                                                                                                                                                                     |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 48         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The ABNF is:

```
(<qos-at-33> NAME 'qpReplaceDefendingPriority'
DESC 'This attribute allows replacing or adding of preemption priority
objects to RSVP messages. It specifies the defending priority within
the preemption object. Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.11. Class qosPolicyRSVPInstallAction**

This class extends the functionality of the qosPolicyRSVPAction class  
by adding detailed control on COPS Install decisions [[COPS](#)]. This  
action allows assigning a preemption priority with an RSVP request, to  
provide a device with information which RSVP requests to accept in case  
of admission failures. This action specifies a DSCP value to set on the

flow RSVP is requesting QoS for.

This class should be extended when additional install decisions need to be controlled.

An instance of this class SHOULD be attached to an object together with an instance of the qosPolicyRSVPAction class.

The class definition is as follows:

|                   |                                                                 |
|-------------------|-----------------------------------------------------------------|
| NAME              | qosPolicyRSVPInstallAction                                      |
| DESCRIPTION       | A class that defines actions to be administered on a PEP.       |
| DERIVED FROM      | policyActionAuxClass (defined in [ <a href="#">PCIM</a> ])      |
| TYPE              | Auxiliary                                                       |
| AUXILIARY CLASSES |                                                                 |
| OID               | <tdb>                                                           |
| MUST              |                                                                 |
| MAY               | qpSetDSCPValue, qpSetDefendingPriority, qpSetPreemptionPriority |

The corresponding ABNF is:

```
(<qos-oc-8> NAME 'qosPolicyRSVPInstallAction'
DESC 'A class that defines actions to be administered on a PEP.'
SUP policyActionAuxClass AUXILIARY
MAY (qpSetDSCPValue $ qpSetDefendingPriority $
qpSetPreemptionPriority)
)
```

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 49         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

#### **8.11.1. The Attribute qpSetDSCPValue**

This attribute is used in the RSVP install action to set the DSCP for the flow in response to an RSVP request. The attribute definition is as follows:

|              |                                                                                     |
|--------------|-------------------------------------------------------------------------------------|
| NAME         | qpSetDSCPValue                                                                      |
| DESCRIPTION  | Defines the value the PEP must use to remark the flow signaled by the RSVP request. |
| SYNTAX       | Integer                                                                             |
| OID          | <tdb>                                                                               |
| EQUALITY     | IntegerMatch                                                                        |
| MULTI-VALUED | No                                                                                  |

DEFAULT VALUE 0

The corresponding ABNF is:

```
(<qos-at-34> NAME 'qpSetDSCPValue'
DESC 'Defines the value the PEP must use to remark the flow signaled by
the RSVP request. Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### **8.11.2. The Attribute qpSetDefendingPriority**

This attribute allows setting the preemption priority [[RSVP\\_PREEMP](#)] of RSVP flows.

|               |                                                                                                                                                                        |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NAME          | qpSetDefendingPriority                                                                                                                                                 |
| DESCRIPTION   | This attribute allows setting the preemption priority [ <a href="#">RSVP_PREEMP</a> ] of RSVP flows. It specifies the defending priority within the preemption object. |
| SYNTAX        | Integer                                                                                                                                                                |
| OID           | <tdb>                                                                                                                                                                  |
| EQUALITY      | IntegerMatch                                                                                                                                                           |
| MULTI-VALUED  | No                                                                                                                                                                     |
| DEFAULT VALUE | 0                                                                                                                                                                      |

```
(<qos-at-35> NAME 'qpSetDefendingPriority'
DESC 'This attribute allows setting the preemption priority of RSVP
flows. It specifies the defending priority within the preemption
object. Default value is 0'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 50         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

### **8.11.3. The Attribute qpSetPreemptionPriority**

This attribute allows setting the preemption priority [[RSVP\\_PREEMP](#)] of RSVP flows.

|             |                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------|
| NAME        | qpSetPreemptionPriority                                                                              |
| DESCRIPTION | This attribute allows setting the preemption priority [ <a href="#">RSVP_PREEMP</a> ] of RSVP flows. |

|               |              |
|---------------|--------------|
| SYNTAX        | Integer      |
| OID           | <tbid>       |
| EQUALITY      | IntegerMatch |
| MULTI-VALUED  | No           |
| DEFAULT VALUE | 0            |

The corresponding ABNF is:

```
(<qos-at-36> NAME 'qpSetPreemptionPriority'
DESC 'This attribute allows setting the preemption priority of RSVP
flows. Default value is 0.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

### **8.12. Class qosPolicySimpleCondition (Aux)**

A simple condition is composed of a <Variable>, an <Operator>, and a <Value> triplet. The operator used in all definitions in this draft is the 'match' operator. Such simple conditions are evaluated by answering the question: Does <variable> match <value>? The operator attribute can be extended to support other relations between variable and values; however, this is beyond the scope of this draft.

Simple conditions are building blocks for more complex Boolean conditions. The qosPolicySimpleCondition is derived from the policyConditionAuxClass class of the Core schema [[PFSHEMA](#)]. QosPolicySimpleCondition is an auxiliary class. This enables the condition to support both rule-specific as well as reusable policy rules. To implement a reusable simple condition, the qosPolicySimpleCondition is stored in the policy repository. It is then attached to an instance of the policyConditionInstance class. To implement a rule-specific simple condition, the instance of the qosPolicySimpleCondition class is attached directly to an instance of the policyRuleConditionAssociation structural class. For a complete explanation of the use of simple conditions, see [[QOSIM](#)].

Variables and/or values can themselves be made reusable or rule-specific. If it is desired to reuse variables and/or values, then they can be referenced by this class using the qpVariableAtom and qpValueAtom attributes, which are attributes are DNs that point to variable and value attributes, respectively. Note that all classes derived from qosPolicyVariable and qosPolicyValue classes can be referenced as well.

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 51         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

If instead rule-specific variables and values are to be defined, then they must be attached to a structural class. This in turn means that the qosPolicySimpleCondition, as well as the variables and/or values



that are rule-specific, MUST be attached to a structural class. This is done by attaching the qosPolicySimpleCondition class and appropriate subclass of qosPolicyVariable and qosPolicyValue to an instance of the policyConditionInstance class. In this case, we need to identify the variables and/or values that are rule-specific. Since the qpVariableAtom and qpValueAtom attributes are DNSs, if we set them to NULL, then that can be used to signal to the system that the variable and/or value are attached directly to the policyConditionInstance entry. Thus, the qpVariableAtom and qpValueAtom attributes (as appropriate) MUST be NULL in order to signify that they are directly attached to the policyConditionInstance entry. The class definition is as follows:

|                   |                                                                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NAME              | qosPolicySimpleCondition                                                                                                                                                              |
| DESCRIPTION       | A class that represents a single Boolean condition. A group of conditions make up a Boolean expression.<br>A simple condition is made of the triplet<br><Variable - relation - Value> |
| DERIVED FROM      | policyConditionAuxClass (defined in [ <a href="#">PCIM</a> ])                                                                                                                         |
| TYPE              | Auxiliary                                                                                                                                                                             |
| AUXILIARY CLASSES |                                                                                                                                                                                       |
| OID               | <tdb>                                                                                                                                                                                 |
| MUST              |                                                                                                                                                                                       |
| MAY               | qpOperator, qpVariableAtom, qpValueAtom                                                                                                                                               |

The corresponding ABNF is:

```
(<qos-oc-9> NAME 'qosPolicySimpleCondition'
DESC 'A class that represents a single Boolean condition. A group of
conditions make up a Boolean expression. A simple condition is made of
the triple <Variable - relation - Value>.'
SUP policyConditionAuxClass AUXILIARY
MAY (qpOperator $ qpVariableAtom $ qpValueAtom)
)
```

#### **8.12.1. The Attribute qpOperator**

This attribute is used to define the relation between a variable and a value. Its definition is:

|               |                                                                           |
|---------------|---------------------------------------------------------------------------|
| NAME          | qpOperator                                                                |
| DESCRIPTION   | The relation between a variable and a value, stored in a directory entry. |
| SYNTAX        | DirectoryString                                                           |
| OID           | <tdb>                                                                     |
| EQUALITY      | CaseIgnoreString                                                          |
| MULTI-VALUED  | No                                                                        |
| DEFAULT VALUE | "match"                                                                   |

```
(<qos-at-37> NAME 'qpOperator'
DESC 'The relation between a variable and a value, stored in a
directory entry. Default value is "match".'
SYNTAX DirectoryString SINGLE-VALUE
EQUALITY CaseIgnoreString
)
```

### **8.12.2. The Attribute qpVariableAtom**

This attribute is used to store a reference to a variable. It therefore implements reusable variables. To implement a rule-specific variable, this attribute MUST be set to NULL. The attribute definition is:

|               |                                                         |
|---------------|---------------------------------------------------------|
| NAME          | qpVariableAtom                                          |
| DESCRIPTION   | A reference to a variable, stored in a directory entry. |
| SYNTAX        | DistinguishedName                                       |
| OID           | <tdb>                                                   |
| EQUALITY      | DistinguishedNameMatch                                  |
| MULTI-VALUED  | No                                                      |
| DEFAULT VALUE | NULL                                                    |

The corresponding ABNF is:

```
(<qos-at-38> NAME 'qpVariableAtom'
DESC 'A reference to a variable, stored in a directory entry. If the
value of this attribute is NULL, then this is a rule-specific variable.
Otherwise, this DN references a variable attribute. Default value is
NULL.'
SYNTAX DistinguishedName SINGLE-VALUE
EQUALITY DistinguishedNameMatch
)
```

### **8.12.3. The Attribute qpValueAtom**

This attribute is used to store a reference to a value. It therefore implements reusable values. To implement a rule-specific value, this attribute MUST be set to NULL. The attribute definition is:

|               |                                                     |
|---------------|-----------------------------------------------------|
| NAME          | qpValueAtom                                         |
| DESCRIPTION   | A reference to a value, stored in a directory entry |
| SYNTAX        | DistinguishedName                                   |
| OID           | <tdb>                                               |
| EQUALITY      | DistinguishedNameMatch                              |
| MULTI-VALUED  | No                                                  |
| DEFAULT VALUE | NULL                                                |

The corresponding ABNF is:

```
(<qos-at-39> NAME 'qpValueAtom'
DESC 'A reference to a value, stored in a directory entry. If the value
of this attribute is NULL, then this is a rule-specific value.
Otherwise, this DN references a value attribute. Default value is
NULL.'
SYNTAX DistinguishedName SINGLE-VALUE
EQUALITY DistinguishedNameMatch
)
```

### 8.13. Class qosPolicyVariable

Variables are used for building individual conditions. The variable specifies the attribute of a flow that should be matched when evaluating the condition.

Not every combination of a variable and a value creates a meaningful condition. For example, a source IP address variable can not be matched against a value that specifies a port number. All variables have particular syntaxes that select the set of values that can be matched.

A variable may also limit the set of values within a particular value type that can be matched against it in a condition. For example, a source-port variable limits the set of values to represent integers in the range of 0-65535. Integers outside this range can not be matched to the 16 bits port entity. However, such checking is implementation-dependent.

The qosPolicyVariable class is an auxiliary class so that it can be used to represent either rule-specific or reusable variables. Variables can either be attached directly to the policyConditionInstance entry (to implement a rule-specific variable) or referenced from a qosPolicySimpleCondition entry (to implement a reusable variable). The class definition is as follows:

|                   |                                                                  |
|-------------------|------------------------------------------------------------------|
| NAME              | qosPolicyVariable                                                |
| DESCRIPTION       | A class that represents a single variable in a Boolean condition |
| DERIVED FROM      | Policy (defined in [ <a href="#">PCIM</a> ])                     |
| TYPE              | Auxiliary                                                        |
| AUXILIARY CLASSES |                                                                  |
| OID               | <td>                                                             |

MUST  
MAY qpVariableName, qpValueTypes, qpVariableDescription,  
qpValueConstraints

Snir, Ramberg, Strassner, Cohen expires September 2000 54  
Draft-ietf-policy-qos-schema-01.txt April 2000

The corresponding ABNF is:

```
(<qos-oc-10> NAME 'qosPolicyVariable'
DESC 'A class that represents a single variable in a Boolean condition'
SUP Policy AUXILIARY
MAY (qpVariableName $ qpValueTypes $ qpVariableDescription
$ qpValueConstraints)
)
```

8.13.1. The Attribute qpVariableName

This attribute defines a unique name for the variable. The attribute is defined as follows:

|               |                                 |
|---------------|---------------------------------|
| NAME          | qpVariableName                  |
| DESCRIPTION   | A unique name for the variable. |
| SYNTAX        | IA5String                       |
| OID           | <tdb>                           |
| EQUALITY      | CaseExactIA5Match               |
| MULTI-VALUED  | No                              |
| DEFAULT VALUE | NULL                            |

Following is a table that defines the predefined Variable names and their bindings. The table indicates which fields are checked in actual filters used in provisioning policies as well as in RSVP signaling messages.

| Variable name | Logical binding                                                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| SourceIP      | The source IP address of the flow. Compared to the source IP header field, or the sender address in the RSVP Filter spec object [RSVP].               |
| SourcePort    | The source Port of a UDP/TCP flow. Compared to the source port field in the TCP/UDP header, or the sender port in the RSVP Filter spec object [RSVP]. |

|                 |                                                              |         |
|-----------------|--------------------------------------------------------------|---------|
| DestinationIP   | The destination IP address of the flow. Compared             |         |
|                 | to the destination IP header field, or the session           |         |
|                 | address in the RSVP SESSION object [ <a href="#">RSVP</a> ]. |         |
| +-----+         | +-----+                                                      | +-----+ |
| DestinationPort | The destination Port of a UDP/TCP flow. Compared             |         |
|                 | to the destination port field in the TCP/UDP                 |         |
|                 | header, or the session port in the RSVP SESSION              |         |
|                 | object [ <a href="#">RSVP</a> ].                             |         |
| +-----+         | +-----+                                                      | +-----+ |

(table continued on next page)

Snir, Ramberg, Strassner, Cohen      expires September 2000      55  
Draft-ietf-policy-qos-schema-01.txt      April 2000

(table continued from previous page)

|                |                                                               |         |
|----------------|---------------------------------------------------------------|---------|
| +-----+        | +-----+                                                       | +-----+ |
| Variable name  | Logical binding                                               |         |
| +-----+        | +-----+                                                       | +-----+ |
| IPProtocol     | The IP protocol number. Compared to the protocol              |         |
|                | number in the IP header field or to the IP                    |         |
|                | protocol in the RSVP SESSION object [ <a href="#">RSVP</a> ]. |         |
| +-----+        | +-----+                                                       | +-----+ |
| ToS            | The ToS variable is bound to the IP header ToS                |         |
|                | byte.                                                         |         |
| +-----+        | +-----+                                                       | +-----+ |
| DSCP           | The DSCP variable is bound to the IP header DSCP              |         |
|                | byte or to DCLASS RSVP object.                                |         |
| +-----+        | +-----+                                                       | +-----+ |
| DestinationMAC | The destination MAC address variable is bound the             |         |
|                | frame destination MAC address.                                |         |
| +-----+        | +-----+                                                       | +-----+ |
| SourceMAC      | The source MAC address variable is bound the frame            |         |
|                | source MAC address.                                           |         |
| +-----+        | +-----+                                                       | +-----+ |
| 8021QID        | The VLAN ID as represented in the 802.1Q field of             |         |
|                | the header.                                                   |         |
| +-----+        | +-----+                                                       | +-----+ |
| Snap           | The snap protocol variable is bound to protocol               |         |
|                | type carried over SNAP encapsulation.                         |         |
| +-----+        | +-----+                                                       | +-----+ |
| Ethertype      | The ethertype variable is bound to the frame                  |         |
|                | header ethertype value.                                       |         |
| +-----+        | +-----+                                                       | +-----+ |
| Ssap           | The source sap variable is bound the frame header             |         |
|                | field containing the source SAP.                              |         |
| +-----+        | +-----+                                                       | +-----+ |

|             |                                                    |         |
|-------------|----------------------------------------------------|---------|
| Dsap        | The destination sap variable is bound the frame    |         |
|             | header field containing the destination SAP.       |         |
| +-----+     | +-----+                                            | +-----+ |
| Application | The ID of the application that generated the flow. |         |
| +-----+     | +-----+                                            | +-----+ |
| User        | The ID of the user that initiated the flow, or is  |         |
|             | designated as the flow owner.                      |         |
| +-----+     | +-----+                                            | +-----+ |

Table 2. Pre-defined Variable Names and Their Bindings

The corresponding ABNF is:

```
(<qos-at-40> NAME 'qpVariableName'
DESC 'A unique name for the variable. Default value is NULL'
SYNTAX IA5String SINGLE-VALUE
EQUALITY CaseExactIA5Match
)
```

Snir, Ramberg, Strassner, Cohen      expires September 2000      56  
Draft-ietf-policy-qos-schema-01.txt      April 2000

### [8.13.2](#) The Attribute qpValueTypes

This attribute specifies an unordered list of possible value types that can be used in a simple condition together with this variable. The value types are specified by their class names. The list of class names allows efficient retrieval of the possible set of relevant values from a repository. The attribute is defined as follows:

```
NAME qpValueTypes
DESCRIPTION A list of class names of possible value types that
can be associated with this variable in a
condition
SYNTAX IA5String
OID <td>
EQUALITY caseIgnoreIA5Match
MULTI-VALUED Yes
DEFAULT VALUE NULL
```

Following is a table of variable names and their default allowed class types.

|               |         |                                                |
|---------------|---------|------------------------------------------------|
| +-----+       | +-----+ | +-----+                                        |
| Variable name |         | Allowed class types                            |
| +-----+       | +-----+ | +-----+                                        |
| SourceIP      |         | qosPolicyIPv4AddrValue, qosPolicyIPv6AddrValue |
| +-----+       | +-----+ | +-----+                                        |
| SourcePort    |         | qosPolicyIntegerValue                          |
| +-----+       | +-----+ | +-----+                                        |

|                 |                                                |         |
|-----------------|------------------------------------------------|---------|
| DestinationIP   | qosPolicyIPv4AddrValue, qosPolicyIPv6AddrValue |         |
| +-----+         | +-----+                                        | +-----+ |
| DestinationPort | qosPolicyIntegerValue                          |         |
| +-----+         | +-----+                                        | +-----+ |
| IPProtocol      | qosPolicyIntegerValue                          |         |
| +-----+         | +-----+                                        | +-----+ |
| ToS             | qosPolicyIntegerValue, qosPolicyBitStringValue |         |
| +-----+         | +-----+                                        | +-----+ |
| DSCP            | qosPolicyIntegerValue, qosPolicyBitStringValue |         |
| +-----+         | +-----+                                        | +-----+ |
| DestinationMAC  | qosPolicyMACAddrValue                          |         |
| +-----+         | +-----+                                        | +-----+ |
| SourceMAC       | qosPolicyMACAddrValue                          |         |
| +-----+         | +-----+                                        | +-----+ |
| 8021QID         | qosPolicyIntegerValue, qosPolicyBitStringValue |         |
| +-----+         | +-----+                                        | +-----+ |
| Snap            | qosPolicyIntegerValue                          |         |
| +-----+         | +-----+                                        | +-----+ |
| Ethertype       | qosPolicyIntegerValue                          |         |
| +-----+         | +-----+                                        | +-----+ |

(table continued on following page)

Snir, Ramberg, Strassner, Cohen      expires September 2000      57  
Draft-ietf-policy-qos-schema-01.txt      April 2000

(table continued from previous page)

|               |                                         |         |
|---------------|-----------------------------------------|---------|
| +-----+       | +-----+                                 | +-----+ |
| Variable name | Allowed class types                     |         |
| +-----+       | +-----+                                 | +-----+ |
| Ssap          | qosPolicyIntegerValue                   |         |
| +-----+       | +-----+                                 | +-----+ |
| Dsap          | qosPolicyIntegerValue                   |         |
| +-----+       | +-----+                                 | +-----+ |
| Application   | qosPolicyDNValue, qosPolicyStringValue, |         |
|               | qosPolicyAttributeValue                 |         |
| +-----+       | +-----+                                 | +-----+ |
| User          | qosPolicyDNValue, qosPolicyStringValue, |         |
|               | qosPolicyAttributeValue                 |         |
| +-----+       | +-----+                                 | +-----+ |

Table 3. Allowed Variable Names and Their Default Class Types

The corresponding ABNF is:

```
(<qos-at-41> NAME 'qpValueTypes'
DESC 'A list of class names of possible value types that can be
associated with this variable in a condition. Default value is NULL.'
```

```
SYNTAX IA5String
EQUALITY caseIgnoreIA5StringMatch
)
```

### **8.13.3. The Attribute qpVariableDescription**

This attribute provides a simple textual description of the variable.  
The attribute is defined as follows:

```
NAME qpVariableDescription
DESCRIPTION A textual description of the variable
SYNTAX DirectoryString
OID <tbid>
EQUALITY CaseIgnoreMatch
MULTI-VALUED No
DEFAULT VALUE NULL
```

The corresponding ABNF is:

```
(<qos-at-42> NAME 'qpVariableDescription'
DESC 'A textual description of the variable'
SYNTAX DirectoryString SINGLE-VALUE
EQUALITY CaseIgnoreMatch
)
```

Snir, Ramberg, Strassner, Cohen      expires September 2000      58  
Draft-ietf-policy-qos-schema-01.txt      April 2000

### **8.13.4. The Attribute qpValueConstraints**

This attribute provides a list of DNS that reference objects that  
provide constraints on this variable. The attribute is defined as  
follows:

```
NAME qpValueConstraints
DESCRIPTION A list of DNS of the objects serving
 as constraints for this variable.
SYNTAX DistinguishedName
OID <tbid>
EQUALITY DistinguishedNameMatch
MULTI-VALUED Yes
DEFAULT VALUE NULL
```

The corresponding ABNF is:

```
(<qos-at-43> NAME 'qpValueConstraints'
```



```

DESC 'A list of DNS of the objects serving as constraints for this
variable. Default value is NULL.'
SYNTAX DistinguishedName
EQUALITY DistinguishedNameMatch
)

```

#### 8.14. Class qosPolicyValue

This is an abstract class, and is used for defining values and constants used in policy conditions. This class provides a common base class for defining application-specific values. The following sections describe some pre-defined subclasses of this class used to describe commonly occurring values and constants used in DiffServ and RSVP policies. The class definition is as follows:

```

NAME qosPolicyValue
DESCRIPTION This class is used as an abstract class for
 defining values and constants used in policy
 conditions
DERIVED FROM Policy (defined in [PCIM])
TYPE Abstract
AUXILIARY CLASSES
OID <td>
MUST
MAY

```

The corresponding ABNF is:

```

(<qos-oc-11> NAME 'qosPolicyValue'
DESC 'This class is used as an abstract class for defining values and
constants used in policy conditions'
SUP Policy ABSTRACT
)

```

Snir, Ramberg, Strassner, Cohen      expires September 2000      59  
Draft-ietf-policy-qos-schema-01.txt      April 2000

#### 8.15. Class qosPolicyIPv4AddrValue

This class is used to define the value(s) for one or more of the following: a single IPv4 address, a hostname, a list of IPv4 addresses, a list of masked IPv4 addresses, or an address range in prefix notation. Each address is contained as a value in the qpIPv4AddrList attribute. The class definition is as follows:

```

NAME qosPolicyIPv4AddrValue
DESCRIPTION This class is used to define the value for one or
 more types of IPv4 addresses.
DERIVED FROM qosPolicyValue
TYPE Auxiliary

```

|                   |                |
|-------------------|----------------|
| AUXILIARY CLASSES |                |
| OID               | <tdb>          |
| MUST              |                |
| MAY               | qpIPv4AddrList |

The corresponding ABNF is:

```
(<qos-oc-12> NAME 'qosPolicyIPv4AddrValue'
DESC 'This class is used to define the value(s) of one or more of the
following: a single IPv4 address, a hostname, a list of IPv4 addresses,
a list of masked IPv4 addresses, or an address range in prefix
notation.'
SUP qosPolicyValue AUXILIARY
MAY (qpIPv4AddrList)
)
```

### **8.15.1. The Attribute qpIPv4AddrList**

This attribute provides an unordered list of strings, each specifying a single IPv4 address or a range of IPv4 addresses. The ABNF definition [\[ABNF\]](#) of IPv4 address is:

```
IPv4address = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
IPv4prefix = IPv4address "/" 1*2DIGIT
IPv4range = IPv4address "-" IPv4address
IPv4maskedaddress = IPv4address", "IPv4address
```

Each string entry is either:

1. A single IPv4address in dot notation as defined above.  
Example: 121.1.1.2
2. A single Hostname. Hostname format MUST follow guidelines and restrictions specified in [\[NAMES\]](#).  
Example: www.bigcompany.com

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 60         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

3. An IPv4range address range defined above, specified by a start address in dot notation and an end address in dot notation, separated by "-". The range includes all addresses between the range's start and end addresses, including the start and end addresses.  
Example: 1.1.22.1-1.1.22.5
4. An IPv4maskedaddress address range defined above, specified by an address and mask. The address and mask are represented in dot

notation separated by a comma ",".

Example: 2.3.128.0,255.255.248.0.

5. An IPv4prefix address range defined above specified by an address and a prefix length separated by "/".

Example: 2.3.128.0/15

The class definition is:

|               |                                                                               |
|---------------|-------------------------------------------------------------------------------|
| NAME          | qpIPv4AddrList                                                                |
| DESCRIPTION   | A list of IP addresses and IP address ranges.                                 |
| SYNTAX        | IA5String                                                                     |
| OID           | <tbid>                                                                        |
| EQUALITY      | caseIgnoreIA5Match                                                            |
| MULTI-VALUED  | Yes                                                                           |
| FORMAT        | IPv4address   hostname   IPv4addressrange  <br>IPv4maskedaddress   IPv4prefix |
| DEFAULT VALUE | NULL                                                                          |

The corresponding ABNF is:

```
(<qos-at-44> NAME 'qpIPv4AddrList'
DESC 'A list of one or more of the following types of addresses: a
single IPv4 address, a hostname, a list of IPv4 addresses, a list of
masked IPv4 addresses, or an address range in prefix notation. Default
value is NULL.'
SYNTAX IA5String
EQUALITY caseIgnoreIA5Match
)
```

#### **8.16. Class qosPolicyIPv6AddrValue**

This class is used to define the value of one or more of the following:  
types of addresses: a single IPv6 address, a hostname, a list of IPv6  
addresses, a list of masked IPv6 addresses, or an address range in  
prefix notation. Each address is contained as a value in the  
qpIPv6AddrList attribute. The class definition is as follows:

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 61         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

|             |                                                                                                          |
|-------------|----------------------------------------------------------------------------------------------------------|
| NAME        | qosPolicyIPv6AddrValue                                                                                   |
| DESCRIPTION | This class is used to define a list of one or more of<br>the following types of addresses: a single IPv6 |

|                   |                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------|
|                   | address, a hostname, a list of IPv6 addresses, a list of masked IPv6 addresses, or an address range in prefix notation. |
| DERIVED FROM      | qosPolicyValue                                                                                                          |
| TYPE              | Auxiliary                                                                                                               |
| AUXILIARY CLASSES |                                                                                                                         |
| OID               | <tbid>                                                                                                                  |
| MUST              |                                                                                                                         |
| MAY               | qpIPv6AddrList                                                                                                          |

The corresponding ABNF is:

```
(<qos-oc-13> NAME 'qosPolicyIPv6AddrValue'
DESC ' This class is used to define a list of one or more of the
following types of addresses: a single IPv6 address, a hostname, a list
of IPv6 addresses, a list of masked IPv6 addresses, or an address range
in prefix notation.'
SUP qosPolicyValue AUXILIARY
MAY (qpIPv6AddrList)
)
```

#### **8.16.1. The Attribute qpIPv6AddrList**

This attribute provides an unordered list of strings, each specifying an IPv6 address or a range of IPv6 addresses. IPv6 address format definition uses the standard address format defined in [\[IPv6\]](#).

The ABNF definition [\[ABNF\]](#) as specified in [\[IPv6\]](#) is:

```
IPv6address = hexpart [":" IPv4address]
IPv4address = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
IPv6prefix = hexpart "/" 1*2DIGIT
hexpart = hexseq | hexseq ":" [hexseq] | "::" [hexseq]
hexseq = hex4 *(":" hex4)
hex4 = 1*4HEXDIG
IPv6range = IPv6address "-" IPv6address
IPv6maskedaddress = IPv6address "/", IPv6address
```

Each string entry is either:

1. A single IPv6address as defined above.
2. A single Hostname. Hostname format MUST follow guidelines and restrictions specified in [\[NAMES\]](#).  
Example: `www.bigcompany.com`
3. An IPv6range address range, specified by a start address in dot notation and an end address in dot notation, separated by "-".  
The range includes all addresses between the range's start and end addresses, including the start and end addresses.

4. An IPv4maskedaddress address range defined above specified by an address and mask. The address and mask are represented in dot notation separated by a comma ",".
5. A single IPv6prefix as defined above.

|               |                                                                               |
|---------------|-------------------------------------------------------------------------------|
| NAME          | qpIPv6AddrList                                                                |
| DESCRIPTION   | A list of IPv6 addresses and IPv6 address ranges.                             |
| SYNTAX        | IA5String                                                                     |
| OID           | <tdb>                                                                         |
| EQUALITY      | caseIgnoreIA5Match                                                            |
| MULTI-VALUED  | Yes                                                                           |
| FORMAT        | IPv6address   hostname   IPv6addressrange  <br>IPv6maskedaddress   IPv6prefix |
| DEFAULT VALUE | NULL                                                                          |

The corresponding ABNF is:

```
(<qos-at-45> NAME 'qpIPv6AddrList'
DESC 'A list of one or more of the following types of addresses: a
single IPv6 address, a hostname, a list of IPv6 addresses, a list of
masked IPv6 addresses, or an address range in prefix notation. Default
value is NULL.'
SYNTAX IA5String
EQUALITY caseIgnoreIA5Match
)
```

### **8.17. Class qosPolicyMACAddrValue**

This class is used to define a list of MAC addresses and MAC address range values. The actual addresses are stored as values in the qpMACAddrList attribute. The class definition is as follows:

|                   |                                                                                       |
|-------------------|---------------------------------------------------------------------------------------|
| NAME              | qosPolicyMACAddrValue                                                                 |
| DESCRIPTION       | This class is used to define a list of MAC<br>addresses and MAC address range values. |
| DERIVED FROM      | qosPolicyValue                                                                        |
| TYPE              | Auxiliary                                                                             |
| AUXILIARY CLASSES |                                                                                       |
| OID               | <tdb>                                                                                 |
| MUST              |                                                                                       |
| MAY               | qpMACAddrList                                                                         |
| DEFAULT VALUE     | NULL                                                                                  |

The corresponding ABNF is:

```
(qos-oc-14> NAME 'qosPolicyMACAddrValue'
DESC 'This class is used to define a list of MAC addresses and MAC
address range values. Default value is NULL.'
SUP qosPolicyValue AUXILIARY
```

```
MAY (qpMACAddrList)
)
```

Snir, Ramberg, Strassner, Cohen      expires September 2000      63  
Draft-ietf-policy-qos-schema-01.txt      April 2000

#### **8.17.1. The Attribute qpMACAddrList**

This attribute provides an unordered list of strings each specifying a MAC address or a range of MAC addresses. 802 MAC address canonical format is used. The ABNF definition [[ABNF](#)] is:

```
MACaddress = 1*4HEXDIG ":" 1*4HEXDIG ":" 1*4HEXDIG
MACmaskedaddress = MACaddress", "MACaddress
```

Each string entry is either:

1. A single MAC address.  
Example: 0000:00A5:0000
2. A MACmaskedaddress address range, which is specified by an address and mask. The mask specifies the relevant bits in the address.  
Example: 0000:00A5:0000, FFFF:FFFF:0000 defines a range of MAC addresses in which the first 4 8-bit bytes are equal to 0000:00A5.

The attribute is defined as follows:

|               |                                                 |
|---------------|-------------------------------------------------|
| NAME          | qpMACAddrList                                   |
| DESCRIPTION   | A list of MAC addresses and MAC address ranges. |
| SYNTAX        | IA5String                                       |
| OID           | <tdb>                                           |
| EQUALITY      | caseIgnoreIA5Match                              |
| MULTI-VALUED  | Yes                                             |
| FORMAT        | MACaddress   MACmaskedaddress                   |
| DEFAULT VALUE | NULL                                            |

The corresponding ABNF is:

```
(<qos-at-46> NAME 'qpMACAddrList'
DESC 'A list of MAC addresses and MAC address ranges. Each entry is
either a single MAC address or a MACmaskedaddress, which is specified
by an address and a mask. The mask specifies the relevant bits in the
address range. Default value is NULL.'
SYNTAX IA5String
EQUALITY caseIgnoreIA5Match
)
```

#### **8.18. Class qosPolicyStringValue**

This class is used to represent a single or set of string values. The

strings are contained as multiple values in the qpStringList attribute.  
The class definition is as follows:

Snir, Ramberg, Strassner, Cohen      expires September 2000      64  
Draft-ietf-policy-qos-schema-01.txt      April 2000

|                   |                                                                     |
|-------------------|---------------------------------------------------------------------|
| NAME              | qosPolicyStringValue                                                |
| DESCRIPTION       | This class is used to define a list of string values with wildcards |
| DERIVED FROM      | qosPolicyValue                                                      |
| TYPE              | Auxiliary                                                           |
| AUXILIARY CLASSES |                                                                     |
| OID               | <tdb>                                                               |
| MUST              |                                                                     |
| MAY               | qpStringList                                                        |

The corresponding ABNF is:

```
(<qos-oc-15> NAME 'qosPolicyStringValue'
DESC 'This class is used to define a list of string
values with wildcards'
SUP qosPolicyValue AUXILIARY
MAY (qpStringList)
)
```

#### **8.18.1. The Attribute qpStringList**

This attribute provides an unordered list of strings, each representing a single string with wildcards. The asterisk character ("\*") is used as a wildcard, and represents an arbitrary sub-string replacement (i.e., zero or more characters). For example, the value "abc\*def" matches "abcxyzdef", and the value "abc\*def\*" match "abcxxxdefyyyzzz". The syntax definition is identical to the sub-string syntax defined in [[LDAP\\_ATTR](#)]. If the asterisk character is required as part of the string value itself, it must be quoted as described in section 4.3 of [[LDAP\\_ATTR](#)]. The attribute is defined as follows:

|               |                                        |
|---------------|----------------------------------------|
| NAME          | qpStringList                           |
| DESCRIPTION   | A list of string values with wildcards |
| SYNTAX        | IA5String                              |
| OID           | <tdb>                                  |
| EQUALITY      | CaseIgnoreIA5Match                     |
| MULTI-VALUED  | Yes                                    |
| DEFAULT VALUE | NULL                                   |

The corresponding ABNF is:

```
(<qos-at-47> NAME 'qpStringList'
DESC 'A list of string values with wildcards. The asterisk character is
used as a wildcard, and represents an arbitrary sub-string replacement
(i.e., zero or more characters). Default value is NULL.'
SYNTAX IA5String
EQUALITY CaseIgnoreIA5Match
)
```

Snir, Ramberg, Strassner, Cohen      expires September 2000      65  
Draft-ietf-policy-qos-schema-01.txt      April 2000

### **8.19 Class qosPolicyBitStringValue**

This class is used to represent a single or set of bit string values.  
The bit strings are stored as values of the qpBitStringList attribute.  
The class definition is as follows:

|                   |                                                              |
|-------------------|--------------------------------------------------------------|
| NAME              | qosPolicyBitStringValue                                      |
| DESCRIPTION       | This class is used to define a list of bit<br>string values. |
| DERIVED FROM      | qosPolicyValue                                               |
| TYPE              | Auxiliary                                                    |
| AUXILIARY CLASSES |                                                              |
| OID               | <tdb>                                                        |
| MUST              |                                                              |
| MAY               | qpBitStringList                                              |

The corresponding ABNF is:

```
(<qos-oc-16> NAME 'qosPolicyBitStringValue'
DESC 'This class is used to define a list of bit string values.'
SUP qosPolicyValue AUXILIARY
MAY (qpBitStringList)
)
```

#### **8.19.1. The Attribute qpBitStringList**

This attribute provides an unordered list of strings, each representing a single bit string or a set of bit strings. The number of bits specified should equal the number of bits of the expected variable. For example, for an 8-bit byte variable, 8 bits should be specified. If the variable does not have a fixed length, the bit string should be matched against the variable's most significant bit. The formal [\[ABNF\]](#) definitions are:



```

binary-digit = "0" / "1"
bitstring = 1*binary-digit
maskedBitString = bitstring", "bitstring

```

Each string entry is either:

1. A single bit string.  
Example: 00111010
2. A range of bit strings specifies using a bit string and a bit mask. The bit string and mask must have the same number of bits specified. The mask bit string specifies the significant bits in the bit string value.  
For example, 110110, 100110 and 110111 would match the maskedBitString 100110,101110 but 100100 would not.

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 66         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The attribute is defined as follows:

|               |                             |
|---------------|-----------------------------|
| NAME          | qpBitStringList             |
| DESCRIPTION   | A list of bit string values |
| SYNTAX        | IA5String                   |
| OID           | <tbd>                       |
| EQUALITY      | CaseIgnoreIA5Match          |
| MULTI-VALUED  | Yes                         |
| FORMAT        | BitString   maskedBitString |
| DEFAULT VALUE | NULL                        |

The corresponding ABNF is:

```

(<qos-at-48> NAME 'qpBitStringList'
DESC 'A list of bit string values. Each string entry is either a single
bit string or a range of bit strings specified using a bit string and a
bit mask. The bit string and mask must have the same number of bits
specified. The mask bit string specifies the significant bits in
the bit string value. Default value is NULL.'
SYNTAX IA5String
EQUALITY CaseIgnoreIA5Match
)

```

## 8.20. Class qosPolicyDNValue

This class is used to represent a single or set of DN values, including wildcards. This value can be used in comparison to DN values carried in RSVP policy objects [IDENT]. The list of DNs are stored as values in the qpDNList attribute. The class definition is as follows:

|                   |                                                                  |
|-------------------|------------------------------------------------------------------|
| NAME              | qosPolicyDNValue                                                 |
| DESCRIPTION       | This class is used to define a list of DN values with wildcards. |
| DERIVED FROM      | qosPolicyValue                                                   |
| TYPE              | Auxiliary                                                        |
| AUXILIARY CLASSES |                                                                  |
| OID               | <tdb>                                                            |
| MUST              |                                                                  |
| MAY               | qpDNList                                                         |

The corresponding ABNF is:

```
(<qos-oc-17> NAME 'qosPolicyDNValue'
DESC 'This class is used to define a list of DN
values with wildcards.'
SUP qosPolicyValue AUXILIARY
MAY (qpDNList)
)
```

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 67         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

### **8.20.1. The Attribute qpDNList**

This attribute provides an unordered list of strings, each representing a Distinguished Name (DN) with wildcards. The format of a DN is defined in [DNDEF]. The asterisk character ("\*") is used as wildcard for either a single attribute value or a wildcard for an RDN. The order of RDNs is significant.

For example: A qpDNList attribute carrying the following value:

"OU=Sales, CN=\*, O=Widget Inc., \*, C=US"

matches:

"OU=Sales, CN=J. Smith, O=Widget Inc, C=US"

and also matches:

"OU=Sales, CN=J. Smith, O=Widget Inc, C=US, CN=CA".

The attribute is defined as follows:

|               |                                           |
|---------------|-------------------------------------------|
| NAME          | qpDNList                                  |
| DESCRIPTION   | A list of DN string values with wildcards |
| SYNTAX        | IA5String                                 |
| OID           | <tdb>                                     |
| EQUALITY      | CaseIgnoreIA5Match                        |
| MULTI-VALUED  | Yes                                       |
| DEFAULT VALUE | NULL                                      |

The corresponding ABNF is:

```
(<qos-at-49> NAME 'qpDNList'
DESC 'A list of DN string values with wildcards. The asterisk character
is used as wildcard for either a single attribute value or a wildcard
for an RDN. The order of RDNs is significant. Default value is NULL.'
SYNTAX IA5String
EQUALITY CaseIgnoreIA5Match
)
```

### **8.21. Class qosPolicyAttributeValue**

This class is used to represent a single attribute or a set of attribute values. The particular match operation used is dependent on the type of attribute, which is specified in the qpAttributeName attribute. (The qpAttributeValue attribute carries the actual value of the attribute). This value can be used in conjunction with DN values carried in RSVP objects [IDENT]. The attribute name is used to specify a comparison between a list of values and a specific set of attributes that the DN pointer is referring to.

Snir, Ramberg, Strassner, Cohen      expires September 2000      68  
Draft-ietf-policy-qos-schema-01.txt      April 2000

For example, suppose a User class has a multi-valued attribute called 'member-of' that lists the names of groups this user belongs to. Suppose this attribute uses caseIgnoreIA5Match matching. A simple condition can be constructed to match the DN carried in an RSVP Identity policy object to a qosPolicyAttributeValue with qpAttributeName = "member-of" and qpAttributeList = "group-A". For example, an Identity policy object carrying the following DN:

"OU=Sales, CN=J. Smith, O=Widget Inc."

will match this condition only if J. Smith belongs to group-a.

The class definition is as follows:

|                   |                                                                     |
|-------------------|---------------------------------------------------------------------|
| NAME              | qosPolicyAttributeValue                                             |
| DESCRIPTION       | This class is used to define an attribute and a list of its values. |
| DERIVED FROM      | qosPolicyValue                                                      |
| TYPE              | Auxiliary                                                           |
| AUXILIARY CLASSES |                                                                     |

|      |                                       |
|------|---------------------------------------|
| OID  | <td>                                  |
| MUST |                                       |
| MAY  | qpAttributeName, qpAttributeValueList |

The corresponding ABNF is:

```
(<qos-oc-18> NAME 'qosPolicyAttributeValue'
DESC 'This class is used to define an attribute and a list of its
values. The attribute type is specified in the qpAttributeName
attribute, and the specific value is specified in the
qpAttributeValueList attribute.'
SUP qosPolicyValue AUXILIARY
MAY (qpAttributeName $ qpAttributeValueList)
)
```

### [8.21.1. The Attribute qpAttributeName](#)

This attribute defines the type of attribute that the values in the qpAttributeValueList attribute correspond to. Its definition is:

|               |                                                                                     |
|---------------|-------------------------------------------------------------------------------------|
| NAME          | qpAttributeName                                                                     |
| DESCRIPTION   | This is the name of an attribute that the list<br>of values should be compared with |
| SYNTAX        | IA5String                                                                           |
| OID           | <td>                                                                                |
| EQUALITY      | CaseIgnoreIA5Match                                                                  |
| MULTI-VALUED  | No                                                                                  |
| DEFAULT VALUE | NULL                                                                                |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 69         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF is:

```
(<qos-at-50> NAME 'qpAttributeName'
DESC 'This is the name of an attribute that the list of values should
be compared with. Default value is NULL.'
SYNTAX IA5String SINGLE-VALUE
EQUALITY CaseIgnoreIA5Match
)
```

### [8.21.2. The Attribute qpAttributeValueList](#)

This attribute contains a list of attribute values. Each value is compared to a value of the attribute specified by the qpAttributeName attribute. The attribute definition is:

|               |                                                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------|
| NAME          | qpAttributeValueList                                                                                         |
| DESCRIPTION   | A list of attribute values. Each value is compared to a value of the attribute specified by qpAttributeName. |
| SYNTAX        | IA5String                                                                                                    |
| OID           | <td>                                                                                                         |
| EQUALITY      | CaseIgnoreMatch                                                                                              |
| MULTI-VALUED  | Yes                                                                                                          |
| DEFAULT VALUE | NULL                                                                                                         |

The corresponding ABNF is:

```
(<qos-at-51> NAME 'qpAttributeValueList'
DESC 'A list of attribute values. Each value is compared to a value of
the attribute specified by qpAttributeName. Default value is NULL.'
SYNTAX IA5String
EQUALITY CaseIgnoreIA5Match
)
```

## 8.22. Class qosPolicyIntegerValue

This class provides a list of integer and integer range values. Integers of arbitrary size can be represented. For a given variable, the set of possible range of integer values allowed is specified via the variable's qpValueConstraints attribute. The class definition is as follows:

|                   |                                             |
|-------------------|---------------------------------------------|
| NAME              | qosPolicyIntegerValue                       |
| DESCRIPTION       | This class is used to define Integer values |
| DERIVED FROM      | qosPolicyValue                              |
| TYPE              | Auxiliary                                   |
| AUXILIARY CLASSES |                                             |
| OID               | <td>                                        |
| MUST              |                                             |
| MAY               | qpIntegerList                               |

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 70         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The corresponding ABNF is:

```
(<qos-oc-19> NAME 'qosPolicyIntegerValue'
DESC 'This class is used to define Integer values'
SUP 'qosPolicyValue' AUXILIARY
MAY (qpIntegerList)
)
```

### 8.22.1. The Attribute qpIntegerList

This attribute provides an unordered list of integers and integer range values. The format of the attribute can take on of the following forms:

1. An integer value.
2. A range of integers. The range is specifies by a start integer and an end integer separated by "-". The range includes all integers between the start and end integers, including the start and end integers. To represent a range of integers that is not bounded, the reserved word INFINITY can be used as the end range integer.

The ABNF definition [[ABNF](#)] is:

```
integer = 1*DIGIT | "INFINITY"
integerrange = integer "-" integer
```

Using ranges, the operators greater-than, greater-than-or-equal-to, less-than and less-than-or-equal-to can also be expressed.

The attribute definition is:

|               |                                                                                 |
|---------------|---------------------------------------------------------------------------------|
| NAME          | qpIntegerList                                                                   |
| DESCRIPTION   | This attribute provides an unordered list of integers and integer range values. |
| SYNTAX        | IA5string                                                                       |
| OID           | <tbid>                                                                          |
| EQUALITY      | caseIgnoreIA5Match                                                              |
| MULTI-VALUED  | YES                                                                             |
| FORMAT        | integer   integerrange                                                          |
| DEFAULT VALUE | NULL                                                                            |

The corresponding ABNF is:

```
(<qos-at-52> NAME 'qpIntegerList'
DESC 'This attribute provides an unordered list of integers and integer
range values. A range of integers is specified by a start integer and
an end integer, separated by "-". The range includes all integers
between start and end integers, including the start and end integers.
To represent a range of integers that is not bounded, the reserved word
INFINITY can be used as the end range integer. Default value is NULL.'
SYNTAX IA5string
EQUALITY caseIgnoreIA5Match
)
```

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 71         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

### **8.23. Class qosPolicyPHBSet**

The qosPolicyPHBSet is an auxiliary class that serves as a named container for qosPolicyPHB objects (defined in the following section). A single PHB set is associated (i.e., referenced) with a QoS domain

using the domain attribute defined in the qosPolicyDomain object. Instances of the qosNamedPolicyContainer class can override the domain's PHB set by referencing another PHB set via the qosPolicyPHBSet class or by attachment of a qosPolicyPHBSet object.

The class is defined as follows:

|                   |                                              |
|-------------------|----------------------------------------------|
| NAME              | qosPolicyPHBSet                              |
| DESCRIPTION       | This class defines a set of PHB definitions  |
| DERIVED FROM      | policy (defined in [ <a href="#">PCIM</a> ]) |
| TYPE              | auxiliary                                    |
| AUXILIARY CLASSES |                                              |
| OID               | <tdb>                                        |
| MUST              |                                              |
| MAY               |                                              |

The corresponding ABNF is:

```
(<qos-oc-20> NAME 'qosPolicyPHBSet'
DESC 'This class serves as a named container for qosPolicyPHB objects,
which provide a set of PHB definitions.'
SUP policy AUXILIARY
)
```

#### **8.24. Class qosPolicyPHB**

The qosPolicyPHB class is an abstract class that extends the Policy class. The purpose of this class is to model a PHB service class. The PHB service class is an abstraction over device-specific parameters. This specification defines one such parameter, a DSCP.

The class definition is as follows:

|                   |                                                         |
|-------------------|---------------------------------------------------------|
| NAME              | qosPolicyPHB                                            |
| DESCRIPTION       | This class defines a single service class in a PHB set. |
| DERIVED FROM      | Policy (defined in [ <a href="#">PCIM</a> ])            |
| TYPE              | Abstract                                                |
| AUXILIARY CLASSES |                                                         |
| OID               | <tdb>                                                   |
| MUST              |                                                         |
| MAY               | qpDSCP                                                  |

The corresponding ABNF is:

```
(<qos-oc-21> NAME 'qosPolicyPHB'
DESC 'This class defines a single service class in a PHB set.'
SUP Policy ABSTRACT
MAY (qpDSCP)
)
```

#### **8.24.1. The attribute qpDSCP**

This attribute is used to represent the service classes specified by a particular PHB. The attribute is defined as follows:

|               |                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------|
| NAME          | qpDSCP                                                                                                         |
| DESCRIPTION   | An integer in the range 0-63, representing the service classes in the domain that are used for classification. |
| SYNTAX        | Integer                                                                                                        |
| OID           | <tbid>                                                                                                         |
| EQUALITY      | IntegerMatch                                                                                                   |
| MULTI-VALUED  | No                                                                                                             |
| DEFAULT VALUE | 0                                                                                                              |

The corresponding ABNF is:

```
(<qos-at-53> NAME 'qpDSCP'
DESC 'An integer in the range 0-63, representing the service classes
in the domain that are used for classification. Default value is NULL.'
SYNTAX Integer SINGLE-VALUE
EQUALITY IntegerMatch
)
```

#### **8.25. Class qosPolicyElementAuxClass**

This class introduces no additional attributes, beyond those defined in the class "PolicyElementAuxClass" from which it is derived. Its role is to "tag" an instance of a class defined outside of the set of policy containers that the policy system uses as being relevant to a QoS policy specification. This tagging can potentially take place at two different levels:

- o Every instance to which a qosPolicyElementAuxClass entry is attached becomes an instance of the class "policy", since the policyElementAuxClass is a subclass of "policy". Thus, a DIT search with the filter "objectClass=policy" will return all such instance. (As noted earlier, this approach does not work for some directory implementations. To accommodate these implementations, policy-related entries SHOULD be tagged with the keyword "POLICY", and the search modified to search instead for the attribute "POLICY".)



- o With the policyKeywords attribute that it inherits from "policy", an instance to which the policyElementAuxClass entry is attached can be tagged as being relevant to a particular type or category of policy, using standard keywords, administrator-defined keywords, or both.

The attribute definition is:

|                   |                                                                                                                                               |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| NAME              | qosPolicyElementAuxClass                                                                                                                      |
| DESCRIPTION       | An auxiliary class used to tag instances of classes defined outside the realm of qos policy as relevant to a particular policy specification. |
| DERIVED FROM      | policyElementAuxClass (defined in [ <a href="#">PCIM</a> ])                                                                                   |
| TYPE              | Auxiliary                                                                                                                                     |
| AUXILIARY CLASSES |                                                                                                                                               |
| OID               | <tdb>                                                                                                                                         |
| MUST              |                                                                                                                                               |
| MAY               |                                                                                                                                               |

The corresponding ABNF is:

```
(<qos-oc-22> NAME 'qosPolicyElementAuxClass'
DESC 'An auxiliary class used to tag instances of classes defined
outside the realm of the QoS policy subtree(s) as nevertheless relevant
to a particular policy specification.'
SUP policyElementAuxClass AUXILIARY
)
```

### **[8.26](#). Class qosPolicyMeter**

Meters measure the temporal properties of the stream of packets selected by a classifier against a traffic profile. QosPolicyMeter is an auxiliary class that models a meter. A meter can be shared between different policy rules. A meter shared by more than one policy rule resides in a repository and is referenced by all sharing rules.

The class is defined as follows:

|                   |                                              |
|-------------------|----------------------------------------------|
| NAME              | qosPolicyMeter                               |
| DESCRIPTION       | This class models a meter                    |
| DERIVED FROM      | policy (defined in [ <a href="#">PCIM</a> ]) |
| TYPE              | auxiliary                                    |
| AUXILIARY CLASSES |                                              |
| OID               | <tdb>                                        |
| MUST              |                                              |

Snir, Ramberg, Strassner, Cohen      expires September 2000      74  
Draft-ietf-policy-qos-schema-01.txt      April 2000

The corresponding ABNF is:

```
(<qos-oc-23> NAME 'qosPolicyMeter'
DESC 'An auxiliary class used to model a meter. QosPolicyMeter
Is either attached to a policy action or attached to a policy
instance and kept in a repository as a shared meter '
SUP policy AUXILIARY
)
```

## **9. Extending the QoS Policy Schema**

The following subsections provide general guidance on how to create a domain-specific schema derived from the QoS Policy Schema by deriving specific classes from the QoS Policy Schema.

### **9.1. Extending qosPolicyValue**

The qosPolicyValue class and its subclasses describe the common value types used in defining QoS policies. When other specific value types are required, such as a floating-point number, the required class SHOULD be derived from the qosPolicyValue class, and an attribute that contains the corresponding value SHOULD be added. Note that in many cases, using the attribute value class allows the definition of non-standard policy atoms without extending the qosPolicyValue class.

### **9.2. Extending qosPolicySimpleCondition**

Policy condition describes a single atomic Boolean condition. For Boolean conditions that are not structured as the ordered triple

<variable - relation - value>,

a new type of condition class SHOULD be defined. An example would be a unary condition.

Subclassing could be done using either the policyCondition or the qosPolicySimpleCondition class as the superclass. Notice that the qosPolicySimpleCondition class is an auxiliary class. This enables it to be attached to the policyRule class instance. Any classes derived

from this class MUST also be auxiliary classes.

Snir, Ramberg, Strassner, Cohen      expires September 2000      75  
Draft-ietf-policy-qos-schema-01.txt      April 2000

### **9.3. Extending qosPolicyAction**

The Qos Policy action classes defined in the QoS Policy Schema includes several types of provisioning actions. These include:

- \* Marking
- \* Policing, shaping and remarking according to a traffic profile.
- \* Signaling RSVP actions, including:
  - \* RSVP policy admission
  - \* RSVP signal control extensions.
  - \* RSVP flow control extensions.

In order to add other actions to a particular qosPolicyAction instance, additional actions SHOULD be added to the qosPolicyAction by deriving a new class and adding the appropriate attributes. Notice that the qosPolicyAction class is an auxiliary class in order to allow attachment to the policyRule class instance. Any classes derived from this class MUST also be auxiliary classes.

## **10. Security Considerations**

See [[PFSCHEMA](#)]. This draft has the same security implications as does the [[PFSCHEMA](#)] draft.

## **11. Acknowledgments**

This document has benefited from the comments and participation of participants of the Policy Framework working group. In particular, we would like to thank Ryan Moats for supplying a preliminary version of the ABNF and Alex Wang for his helpful comments.

## **12. References**

[TERMS]      S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", Internet [RFC 2119](#), March 1997.

[PCIM] J. Strassner, E. Ellesson, B. Moore, "Policy Framework Core Information Model", Internet Draft  
<[draft-ietf-policy-core-info-model-05.txt](#)>

[PFSCHEMA] J. Strassner, E. Ellesson, B. Moore, "Policy Framework LDAP Core Schema", Internet Draft  
<[draft-ietf-policy-core-schema-06.txt](#)>

[COPS] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, "The COPS (Common Open Policy Service) Protocol", [RFC2748](#)

Snir, Ramberg, Strassner, Cohen expires September 2000 76  
Draft-ietf-policy-qos-schema-01.txt April 2000

[COPSRVIP] S. Herzog, J. Boyle, R. Cohen, D. Durham, R. Rajan, A. Sastry, "COPS Usage for RSVP", [RFC2749](#)

[LDAP\_ATTR] M. Wahl, A. Coulbeck, "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", [RFC 2252](#)

[RSVP] Braden, R. ed., "Resource ReSerVation Protocol (RSVP) - Functional Specification.", IETF [RFC 2205](#),  
Proposed Standard, Sep. 1997.

[RSVP\_PREEMP] Shai Herzog, "Signaled Preemption Priority Policy Element", [RFC2751](#)

[DIFF-SERV-ARCH] S. Blake D. Blake, "An Architecture for Differentiated Services", [RFC2475](#)

[PIB] M. Fine, K. McCloghrie, J. Seligson, K. Chan, S. Hahn, A. Smith, "Quality of Service Policy Information Base", Internet Draft <[draft-mfine-cops-pib-01.txt](#)>

[DEREF] R. Moats, J. Maziarski, J. Strassner, "Extensible Match Rules to Dereference Pointer", Internet Draft  
<[draft-moats-ldap-dereference-match-02.txt](#)>

[QOSDEVIM] J. Strassner, W. Weiss, D. Durham, A. Westerinen, "Information Model for Describing Network Device QoS Mechanisms", Internet Draft  
<[draft-ietf-policy-qos-device-info-model-00.txt](#)>

[QOSIM] Y. Snir, Y. Ramberg, J. Strassner, R. Cohen "QoS Policy Information model", internet draft  
<qos-policy-info-model-00.txt>

- [NAME] P. Mockapetris, "Domain names - implementation and specification", [RFC1035](#)
- [IPv6] R. Hinden, S. Deering, "IP Version 6 Addressing Architecture", [RFC2373](#), July 1998
- [ABNF] Crocker, D., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.
- [DNDEF] Wahl, M., Kille, S., and T. Howes, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", [RFC 2253](#), December 1997.
- [IDNET] S. Yadav, R. Yavatkar, R. Pabbati, P. Ford, T. Moore, S. Herzog, "Identity Representation for RSVP", [RFC2752](#), January 2000

Snir, Ramberg, Strassner, Cohen expires September 2000 77  
Draft-ietf-policy-qos-schema-01.txt April 2000

### **[13. Author's Addresses](#)**

Yoram Snir

Cisco Systems  
4 Maskit Street  
Herzliya Pituach, Israel 46766  
Phone: +972-9-970-0085  
Fax: +972-9-970-0219  
E-mail: ysnir@cisco.com

Yoram Ramberg

Cisco Systems  
4 Maskit Street  
Herzliya Pituach, Israel 46766  
Phone: +972-9-970-0081  
Fax: +972-9-970-0219  
E-mail: yramberg@cisco.com

John Strassner

Cisco Systems  
170 West Tasman Drive, Building 15  
San Jose, CA 95134  
Phone: +1-408-527-1069  
Fax: +1-408-527-6351  
E-mail: johns@cisco.com

Ron Cohen

Cisco Systems

4 Maskit Street  
Herzliya Pituach, Israel 46766  
Phone: +972-9-970-0064  
Fax: +972-9-970-0219  
E-mail: ronc@cisco.com

#### **14. Full Copyright Statement**

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

|                                     |                        |            |
|-------------------------------------|------------------------|------------|
| Snir, Ramberg, Strassner, Cohen     | expires September 2000 | 78         |
| Draft-ietf-policy-qos-schema-01.txt |                        | April 2000 |

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The derived value classes should be auxiliary so they can be attached to the qosPolicyConstant class. This means that independent instances of value classes can not be created.

