

Internet Draft  
Expires September 2001  
File: [draft-ietf-rap-frameworkpib-04.txt](#)

M. Fine  
K. McCloghrie  
Cisco Systems  
J. Seligson  
K. Chan  
Nortel Networks  
S. Hahn  
R. Sahita  
Intel  
A. Smith  
Allegro Networks  
F. Reichmeyer  
PFN

March 1, 2001

## **Framework Policy Information Base**

### 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>.



## **1. Glossary**

PRC	Provisioning Class.	A type of policy data.
PRI	Provisioning Instance.	An instance of a PRC.
PIB	Policy Information Base.	The database of policy information.
PDP	Policy Decision Point.	See [ <a href="#">RAP-FRAMEWORK</a> ].
PEP	Policy Enforcement Point.	See [ <a href="#">RAP-FRAMEWORK</a> ].
PRID	Provisioning Instance Identifier.	Uniquely identifies an instance of a PRC.

## **2. Introduction**

[SPPI] describes a structure for specifying policy information that can then be transmitted to a network device for the purpose of configuring policy at that device. The model underlying this structure is one of well-defined provisioning classes and instances of these classes residing in a virtual information store called the Policy Information Base (PIB).

One way to provision policy is by means of the COPS protocol [[COPS](#)] with the extensions for provisioning [[COPS-PR](#)]. This protocol supports multiple clients, each of which may provision policy for a specific policy domain such as QoS, virtual private networks, or security.

As described in [[COPS-PR](#)], each client supports a non-overlapping and independent set of PIB modules. However, some provisioning classes are common to all subject-categories (client-types) and need to be present in each. This document presents a set of PRCs that are common to all clients that provision policy using COPS for Provisioning.

## **3. General PIB Concepts**

### **3.1. Roles**

The policy to apply to an interface may depend on many factors such as immutable characteristics of the interface (e.g., ethernet or frame relay), the status of the interface (e.g., half or full duplex), or user configuration (e.g., branch office or headquarters interface). Rather than specifying policies explicitly for each interface of all devices in the network, policies are specified in terms of interface functionality.

To describe these functionalities of an interface we use the concept of "Roles". A Role is simply a string that is associated with an interface. A given interface may have any number of roles

simultaneously. Provisioning classes have an attribute called a "RoleCombination" which is a lexicographically ordered set of roles. Instances of a given provisioning class are applied to an interface

if and only if the set of roles in the role combination matches the set of the roles of the interface.

Thus, roles provide a way to bind policy to interfaces without having to explicitly identify interfaces in a consistent manner across all network devices. (The SNMP experience with ifIndex has proved this to be a difficult task.) That is, roles provide a level of indirection to the application of a set of policies to specific interfaces. Furthermore, if the same policy is being applied to several interfaces, that policy need be pushed to the device only once, rather than once per interface, as long as the interfaces are configured with the same role combination.

We point out that, in the event that the administrator needs to have unique policy for each interface, this can be achieved by configuring each interface with a unique role.

The PEP reports all its role combinations to the PDP in the initial COPS request (REQ) message and in subsequent request messages generated in response to COPS state synchronization (SSQ) requests and local configuration changes.

The comparing of roles (or role combinations) is case sensitive.

By convention, when formatting the role-combination for exchange within a protocol message, within a PIB/MIB object's value, or as a printed value, the set is formatted in lexicographical order of the role's ASCII values; that is, the role that is first is formatted first. For example, "a+b" and "b+a" are NOT different role-combinations; rather, they are different formatting of the same role-combination, and hence for this example:

- "a+b" is the valid formatting of that role-combination,
- "b+a" is an invalid formatting of that role-combination.

The role-combination of interfaces to which no roles have been assigned is known as the "null" role-combination. (Note the deliberate use of lower-case letters for "null" so that it avoids confusion with the ASCII NULL character that has a value of zero but a length of one.)

In an "install" or an "install-notify" class, the wildcard role-combination "\*" can be used. In addition to providing for interface-specific roles, it also allows for other optimizations in reducing the number of role-combinations for which a policy has to be specified. For example:

Suppose we have three interfaces:

Roles A, B and R1 are assigned to interface I1

Roles A, B and R2 are assigned to interface I2  
Roles A, B and R3 are assigned to interface I3

Then, a PRI of a fictional IfDscpAssignTable that has the following values for its attributes:

```
ifDscpAssignPrid    = 1
ifDscpAssignRoles   = "+A+B"
ifDscpAssignName    = "4queues"
ifDscpAssignDscpMap = 1
```

will apply to all three interfaces, because "\*" matches with R1, R2 and R3. The policies can be assigned to an interface due to more than one wild-carded role combo matching a given interface's role combo string. The PDP should attempt to resolve conflicts between policies before sending policies to the PEP. In the situation where the PDP sends multiple policies to a PEP and they do conflict, either because of an error by the PDP or because of a device-specific conflict, then the PEP MUST reject the installation of the conflicting policies and return an error.

Formally,

- The wildcard Role is denoted by "\*",
- The "\*" Role is not allowed to be defined as part of the role-combination of an interface as notified by the PEP to the PDP; it is only allowed in policies installed/deleted via COPS-PR from the PDP to the PEP.
- For a policy to apply to an interface when the policy's role-combination is "+a+b", then the interface's role-combination:
  - Must include "a" and "b", and
  - Can include zero or more other roles.
- The wildcard character "\*" is listed before the other roles as "\*" is lexicographically before "a"; however, the wildcard matches any zero or more roles, irrespective of lexicographical order. For example: "+b+e+g" would match "a+b+c+e+f+g"

Note that the characters "+" and "\*" MUST not be used in an interface Role. The Framework Role PIB module in [section 4](#) of this document contains the Role and RoleCombination Textual Conventions.

### **3.1.1. An Example**

The functioning of roles might be best understood by an example. Suppose I have a device with three interfaces, with roles as follows:

```
IF1: "finance"
IF2: "finance"
IF3: "manager"
```

Suppose, I also have a PDP with two policies:

P1: Packets from finance department (role "finance") get DSCP 5  
P2: Packets from managers (role "manager") get DSCP 6



To obtain policy, the PEP reports to the PDP that it has some interfaces with role combination "finance" and some with role combination "manager". In response, the PDP downloads policy P1 associated with role combination "finance" and downloads a second policy P2 associated with role combination "manager".

Now suppose the finance person attached to IF2 is promoted to manager and so the system administrator adds the role "manager" to IF2. The PEP now reports to the PDP that it has three role combinations: some interfaces with role combination "finance", some with role combination "manager" and some with role combination "finance+manager". In response, the PDP downloads an additional third policy associated with the new role combination "finance+manager".

How the PDP determines the policy for this new role combination is entirely the responsibility of the PDP. It could do so algorithmically or by rule. For example, there might be a rule that specifies that manager policy takes preference over department policy. Or there might be a third policy installed in the PDP as follows:

P3: Packets from finance managers (role "finance" and role "manager") get DSCP 7

The point here is that the PDP is required to determine what policy applies to this new role combination and to download a third policy to the PEP for the role combination "finance+manager" even if that policy is the same as one already downloaded. The PEP is not required (or allowed) to construct policy for new role combinations from existing policy.

### **3.2. Multiple PIB Instances**

[COPS-PR] supports multiple, disjoint, independent instances of the PIB to represent multiple instances of configured policy. The intent is to allow for the pre-provisioning of policy that can then be made active by a single, short decision from the PDP.

A COPS context can be defined as an independent COPS request state for a particular subject category (client-type).

With the COPS-PR protocol, each of these states is identified by a unique client handle. The creation and deletion of these PIB instances is controlled by the PDP as described in [[COPS-PR](#)].

Although many PIB instances may be configured on a device (the maximum number of these instances being determined by the device itself) only one of them can be active at any given time, the active

one being selected by the PDP. To facilitate this selection, the Framework PIB supports an attribute to make a PIB instance the active one and, similarly, to report the active PIB instance to the

PDP in a COPS request message. This attribute is in the Incarnation Table described below.

Setting the attribute `frwkPibIncarnationActive` to 'true' in one PIB instance MUST ensure that the attribute is 'false' in all other contexts.

### **3.3. Reporting of Device Capabilities**

Each network device providing policy-based services has its own inherent capabilities. These capabilities can be hardware specific, e.g., an ethernet interface supporting input classification, or can be statically configured, e.g., supported queuing disciplines. These capabilities are communicated to the PDP when initial policy is requested by the PEP. Knowing device capabilities, the PDP can send the provisioning instances (PRIs) relevant to the specific device, rather than sending the entire PIB.

The PIB indicates which capabilities the PEP must report to the PDP by means of the PIB-ACCESS clause as described in [[SPPI](#)].

### **3.4. Reporting of Device Limitations**

To facilitate efficient policy installation, it is important to understand a device's limitations in relation to the advertised device capabilities. Limitations may be class-based, e.g., an "install" class is supported as a "notify" or only a limited number of class instances may be created, or attribute-based. Attribute limitations, such as supporting a restricted set of enumerations or requiring related attributes to have certain values, detail implementation limitations at a fine level of granularity.

A PDP can avoid certain installation issues in a proactive fashion by taking into account a device's limitations prior to policy installation rather than in a reactive mode during installation. As with device capabilities, device limitations are communicated to the PDP when initial policy is requested.

Reported device limitations may be accompanied by guidance values that can be used by a PDP to determine acceptable values for the identified attributes.



#### 4. The Framework Role PIB module

```
FRAMEWORK-ROLE-PIB  PIB-DEFINITIONS ::= BEGIN

IMPORTS  MODULE-IDENTITY, TEXTUAL-CONVENTION FROM COPS-PR-SPPI
         SnmpAdminString FROM SNMP-FRAMEWORK-MIB;

frwkRolePib  MODULE-IDENTITY
    SUBJECT-CATEGORIES    { all }
    LAST-UPDATED "200003010400Z"
    ORGANIZATION "IETF RAP WG"
    CONTACT-INFO "Keith McCloghrie
                  Cisco Systems, Inc.
                  170 West Tasman Drive,
                  San Jose, CA 95134-1706 USA
                  Phone: +1 408 526 5260
                  Email: kzm@cisco.com

                  John Seligson
                  Nortel Networks, Inc.
                  4401 Great America Parkway
                  Santa Clara, CA 95054 USA
                  Phone: +1 408 495 2992
                  Email: jseligso@nortelnetworks.com"

    DESCRIPTION
        "The PIB module containing the Role and
        RoleCombination Textual Conventions and other
        required TCs."
    ::= { tbd }

Role ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "A role represents a functionality characteristic or
        capability of a resource to which policies are applied.
        Examples of roles include Backbone_interface,
        Frame_Relay_interface, BGP-capable-router, web-server,
        firewall, etc.
        Valid characters are a-z, A-Z, 0-9, period, hyphen and
        underscore. A role must not start with an underscore."
    SYNTAX SnmpAdminString (SIZE (1..31))

RoleCombination ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "A Display string consisting of a set of roles concatenated
        with a '+' character where the roles are in lexicographic
        order from minimum to maximum."
```

For example,  $a+b$  and  $b+a$  are NOT different role-combinations; rather, they are different formatting of the same (one) role-combination.

Notice the roles within a role-combination are in

Lexicographic order from minimum to maximum, hence, we declare:

a+b is the valid formatting of the role-combination,  
b+a is an invalid formatting of the role-combination.

Notice the need of zero-length role-combination as the role-combination of interfaces to which no roles have been assigned. This role-combination is also known as the null role-combination. (Note the deliberate use of lower case letters to avoid confusion with the ASCII NULL character which has a value of zero but length of one.)"

SYNTAX SnmpAdminString (SIZE (0..255))

PrcIdentifier ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"An OID that identifies a PRC. The value MUST be an OID assigned to a PRC's row definition. An attribute with this syntax can have the value 0.0 to indicate that it currently does not identify a PRC."

SYNTAX OBJECT IDENTIFIER

END

## 5. Summary of the Framework PIB

The Framework PIB comprises of three groups:

### 1. Base PIB classes Group

This contains PRCs intended to describe the PRCs supported by the PEP, PRC and/or attribute limitations and its current configuration.

#### PRC Support Table

As the technology evolves, we expect devices to be enhanced with new PIBs, existing PIBs to add new PRCs and existing PRCs to be augmented or extended with new attributes. Also, it is likely that some existing PRCs or individual attributes of PRCs will be deprecated. The PRC Support Table describes the PRCs that the device supports as well as the individual attributes of each PRC. Using this information the PDP can potentially tailor the policy to more closely match the capabilities of the device. The PRC Support Table instances are specific to the particular Subject Category (Client-Type). That is, the PRC Support Table for Subject Category 'A' will not include instances for classes supported by the Subject Category 'B'.

Note that the COPS client-type [[COPS](#)] used for Framework PIB PRIs sent/received over COPS-PR MUST be the unique SUBJECT-CATEGORY number assigned for the area of policy being managed (eg. QoS, Security etc).



The PEP MUST ignore the attributes that it reports as not Supported in the decision from the PDP. The PEP SHOULD not send duplicate PRC support instances in a COPS Request and the PDP MUST ignore duplicate instances and MUST use the first instance received for a supported PRC in a COPS Request.

#### PIB Incarnation Table

This table contains exactly one row (corresponding to one PRI) per context. It identifies the PDP that was the last to download policy into the device and also contains an identifier to identify the version of the policy currently downloaded. This identifier, both its syntax and value, is meaningful only to the PDPs. It is intended to be a mechanism whereby a PDP, on connecting to a PEP, can easily identify a known incarnation of policy. The incarnation PRC also includes an attribute to indicate which context is the active one at the present time. The incarnation instance is specific to the particular Subject Category (Client-Type).

#### Component Limitations Table

Some devices may not be able to implement the full range of values for all attributes. In principle, each PRC supports a set of errors that the PEP can report to the PDP in the event that the specified policy is not implementable. It may be preferable for the PDP to be informed of the device limitations before actually attempting to install policy, and while the error can indicate that a particular attribute value is unacceptable to the PEP, this does not help the PDP ascertain which values would be acceptable. To alleviate these limitations, the PEP can report some limitations of attribute values and/or classes and possibly guidance values for the attribute in the Component Limitations Table

#### Device Identification Table

This class contains a single provisioning instance that contains device-specific information that is used to facilitate efficient policy installation by a PDP. The instance of this class is reported to the PDP in a COPS request message so that the PDP can take into account certain device characteristics during policy installation.

## 2. Device Capabilities group

This group contains the PRCs that describe the characteristics of interfaces of the device and the Role Combinations assigned to them.

## Interface Capabilities Set Table

The interfaces the PEP supports are described by rows in

[Page 9]

this table (frwkIfCapSetTable). Each row, or instance of this class, associates a unique interface name with a set of capabilities that the interface supports. The unique name is used to form a set of capabilities that the name represents. The capability references can specify instances in relevant capability tables in any PIB. The PEP notifies the PDP of these interface names and capabilities and then the PDP configures the interfaces, per role combination. The unique name (IfCapSetName) is not to be confused with the IfType object in MIB-II [[STD17](#)].

#### Interface Capability and Role Combo Table

The Interface Capabilities Set Table (explained above) describes the interfaces the PEP supports by their capabilities, by assigning the capability sets a unique name. It is possible to tailor the behavior of interfaces by assigning specific roles to the capability sets. This allows interfaces with the same capability sets to be assigned different policies, based on the current roles assigned to them. At the PDP, configuration is done in terms of these interface capability set names (ifCapSetName) and the role combinations assigned to them; The PDP does not deal with individual interfaces on the device. Thus, each row of this class is a <interface capability set name, Role Combo> two-tuple, that indicates the roles that have been assigned to a particular capability set (as identified by IfCapSetName). The ifCapSetName is the grouping attribute used to form a set of role combinations that apply to this capability set.

### 3. Classifier group

This group contains the IP and IEEE 802 Classifier elements. The set of tables consist of a Base Filter table that contains the Index InstanceId and the Negation flag for the filter. This frwkBaseFilterTable is extended to form the IP Filter table and the 802 Filter table [[802](#)]. Filters may also be defined outside this document and used to extend the Base Filter table.

The Extended classes do not have a separate Index value. Instances of the extended classes have the same indices as their base class instance. Inheritance is achieved using the EXTENDS keyword as defined in [[SPPI](#)].



## 6. The Framework PIB Module

```
FRAMEWORK-PIB PIB-DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    Unsigned32, Integer32, MODULE-IDENTITY,  
    MODULE-COMPLIANCE, OBJECT-TYPE, OBJECT-GROUP  
        FROM COPS-PR-SPPI  
    InstanceId, Prid  
        FROM COPS-PR-SPPI-TC  
    RoleCombination, PrcIdentifier  
        FROM FRAMEWORK-ROLE-PIB  
    InetAddress, InetAddressType  
        FROM INET-ADDRESS-MIB  
    TruthValue, PhysAddress  
        FROM SNMPv2-TC  
    SnmpAdminString  
        FROM SNMP-FRAMEWORK-MIB;
```

```
frameworkPib MODULE-IDENTITY
```

```
    SUBJECT-CATEGORIES { all }
```

```
    LAST-UPDATED "200003010400Z"
```

```
    ORGANIZATION "IETF RAP WG"
```

```
    CONTACT-INFO "
```

```
        Michael Fine  
        Cisco Systems, Inc.  
        170 West Tasman Drive  
        San Jose, CA 95134-1706 USA  
        Phone: +1 408 527 8218  
        Email: mfine@cisco.com
```

```
        Keith McCloghrie  
        Cisco Systems, Inc.  
        170 West Tasman Drive,  
        San Jose, CA 95134-1706 USA  
        Phone: +1 408 526 5260  
        Email: kzm@cisco.com
```

```
        John Seligson  
        Nortel Networks, Inc.  
        4401 Great America Parkway  
        Santa Clara, CA 95054 USA  
        Phone: +1 408 495 2992  
        Email: jseligso@nortelnetworks.com"
```

```
DESCRIPTION
```

```
    "A PIB module containing the base set of provisioning  
    classes that are required for support of policies for  
    all subject-categories."
```

::= { tbd }

```
--
-- The root OID for PRCs in the Framework PIB
--

frwkBasePibClasses
    OBJECT IDENTIFIER ::= { frameworkPib 1 }

--
-- Textual Conventions
--

--
-- PRC Support Table
--

frwkPrcSupportTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF FrwkPrcSupportEntry
    PIB-ACCESS      notify
    STATUS          current
    DESCRIPTION
        "Each instance of this class specifies a PRC that the device
        supports and a bit string to indicate the attributes of the
        class that are supported. These PRIs are sent to the PDP to
        indicate to the PDP which PRCs, and which attributes of
        these PRCs, the device supports. This table can also be
        downloaded by a network manager when static configuration is
        used.

        All install and install-notify PRCs supported by the device
        must be represented in this table. Notify PRCs may be
        represented for informational purposes."

    ::= { frwkBasePibClasses 1 }

frwkPrcSupportEntry OBJECT-TYPE
    SYNTAX          FrwkPrcSupportEntry
    STATUS          current
    DESCRIPTION
        "An instance of the frwkPrcSupport class that identifies a
        specific PRC and associated attributes as supported
        by the device."

    PIB-INDEX { frwkPrcSupportPrid }
    UNIQUENESS { frwkPrcSupportSupportedPrc }

    ::= { frwkPrcSupportTable 1 }
```





```
FrwkPrcSupportEntry ::= SEQUENCE {  
    frwkPrcSupportPrid      InstanceId,  
    frwkPrcSupportSupportedPrc PrcIdentifier,  
    frwkPrcSupportSupportedAttrs OCTET STRING  
}
```

frwkPrcSupportPrid OBJECT-TYPE

SYNTAX InstanceId

STATUS current

DESCRIPTION

"An arbitrary integer index that uniquely identifies an instance of the frwkPrcSupport class."

::= { frwkPrcSupportEntry 1 }

frwkPrcSupportSupportedPrc OBJECT-TYPE

SYNTAX PrcIdentifier

STATUS current

DESCRIPTION

"The object identifier of a supported PRC. The value is the OID of the table entry. There may not be more than one instance of the frwkPrcSupport class with the same value of frwkPrcSupportSupportedPrc."

::= { frwkPrcSupportEntry 2 }

frwkPrcSupportSupportedAttrs OBJECT-TYPE

SYNTAX OCTET STRING

STATUS current

DESCRIPTION

"A bit string representing the supported attributes of the class that is identified by the frwkPrcSupportSupportedPrc object.

Each bit of this bit string corresponds to a class attribute, with the most significant bit of the *i*-th octet of this octet string corresponding to the (8\**i* - 7)-th attribute, and the least significant bit of the *i*-th octet corresponding to the (8\**i*)-th class attribute. Each bit specifies whether or not the corresponding class attribute is currently supported, with a '1' indicating support and a '0' indicating no support. If the value of this bit string is *N* bits long and there are more than *N* class attributes then the bit string is logically extended with 0's to the required length."

```
::= { frwkPrcSupportEntry 3 }
```

```
--
--
--
```

#### frwkPibIncarnationTable OBJECT-TYPE

SYNTAX SEQUENCE OF FrwkPibIncarnationEntry

PIB-ACCESS install-notify

STATUS current

##### DESCRIPTION

"This class contains a single provisioning instance per installed context that identifies the current incarnation of the PIB and the PDP or network manager that installed this incarnation. The instance of this class is reported to the PDP in the REQ message so that the PDP can (attempt to) ascertain the current state of the PIB and the active context. A network manager may use the instance to determine the state of the device."

```
::= { frwkBasePibClasses 2 }
```

#### frwkPibIncarnationEntry OBJECT-TYPE

SYNTAX FrwkPibIncarnationEntry

STATUS current

##### DESCRIPTION

"An instance of the frwkPibIncarnation class. Only one instance of this provisioning class is ever instantiated per context"

PIB-INDEX { frwkPibIncarnationPrid }

UNIQUENESS { frwkPibIncarnationName }

```
::= { frwkPibIncarnationTable 1 }
```

#### FrwkPibIncarnationEntry ::= SEQUENCE {

frwkPibIncarnationPrid	InstanceId,
frwkPibIncarnationName	SnmpAdminString,
frwkPibIncarnationId	OCTET STRING,
frwkPibIncarnationLongevity	Unsigned32,
frwkPibIncarnationTtl	Unsigned32,
frwkPibIncarnationActive	TruthValue

```
}
```

#### frwkPibIncarnationPrid OBJECT-TYPE

SYNTAX InstanceId

STATUS current

##### DESCRIPTION

"An index to uniquely identify an instance of this provisioning class."

```
::= { frwkPibIncarnationEntry 1 }
```

**frwkPibIncarnationName OBJECT-TYPE**

SYNTAX SnmpAdminString

STATUS current

## DESCRIPTION

"The name of the PDP that installed the current incarnation of the PIB into the device. By default, it is the zero length string."

::= { frwkPibIncarnationEntry 2 }

**frwkPibIncarnationId OBJECT-TYPE**

SYNTAX OCTET STRING

STATUS current

## DESCRIPTION

"An ID to identify the current incarnation. It has meaning to the PDP/manager that installed the PIB and perhaps its standby PDPs/managers. By default, it is the zero-length string."

::= { frwkPibIncarnationEntry 3 }

**frwkPibIncarnationLongevity OBJECT-TYPE**

SYNTAX Unsigned32 {  
    expireNever(1),  
    expireImmediate(2),  
    expireOnTimeout(3)  
}

STATUS current

## DESCRIPTION

"This attribute controls what the PEP does with the downloaded policy on a Client Close message or a loss of connection to the PDP.

If set to expireNever, the PEP continues to operate with the installed policy indefinitely. If set to expireImmediate, the PEP immediately expires the policy obtained from the PDP and installs policy from local configuration. If set to expireOnTimeout, the PEP continues to operate with the policy installed by the PDP for a period of time specified by frwkPibIncarnationTtl. After this time (and it has not reconnected to the original or new PDP) the PEP expires this policy and reverts to local configuration.

For all cases, it is the responsibility of the PDP to check the incarnation and download new policy, if necessary, on a reconnect. On receiving a Remove-State [[COPS-PR](#)] for the active context, this attribute value MUST be ignored and the PEP should expire the policy in that active context

immediately.

Policy enforcement timing only applies to policies that have been installed dynamically (e.g., by a PDP via COPS)."

::= { frwkPibIncarnationEntry 4 }

## frwkPibIncarnationTtl OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

STATUS current

## DESCRIPTION

"The number of seconds after a Client Close or TCP timeout for which the PEP continues to enforce the policy in the PIB. After this interval, the PIB is considered expired and the device no longer enforces the policy installed in the PIB.

This attribute is only meaningful if  
frwkPibIncarnationLongevity is set to expireOnTimeout."

::= { frwkPibIncarnationEntry 5 }

## frwkPibIncarnationActive OBJECT-TYPE

SYNTAX TruthValue

STATUS current

## DESCRIPTION

"If this attribute is set to TRUE, then the PIB instance to which this PRI belongs becomes the active PIB instance. The previous active instance MUST become inactive and the frwkPibIncarnationActive attribute in that PIB instance MUST be set to false."

::= { frwkPibIncarnationEntry 6 }

--

-- Device Identification Table

--

-- This table supports the ability to export general  
-- purpose device information to facilitate efficient  
-- communication between the device and a PDP

## frwkDeviceIdTable OBJECT-TYPE

SYNTAX SEQUENCE OF FrwkDeviceIdEntry

PIB-ACCESS notify

STATUS current

## DESCRIPTION

"This class contains a single provisioning instance that contains device-specific information that is used to facilitate efficient policy installation by a PDP. The instance of this class is reported to the PDP in a COPS request message so that the PDP can take into account

certain device characteristics during policy installation."

```
::= { frwkBasePibClasses 3 }
```



```
frwkDeviceIdEntry OBJECT-TYPE
    SYNTAX      FrwkDeviceIdEntry
    STATUS      current
    DESCRIPTION
        "An instance of the frwkDeviceId class. Only one instance of
        this provisioning class is ever instantiated."

    PIB-INDEX { frwkDeviceIdPrid }
    UNIQUENESS { frwkDeviceIdDescr }

    ::= { frwkDeviceIdTable 1 }
```

```
FrwkDeviceIdEntry ::= SEQUENCE {
    frwkDeviceIdPrid      InstanceId,
    frwkDeviceIdDescr     SnmpAdminString,
    frwkDeviceIdMaxMsg     Unsigned32,
    frwkDeviceIdMaxContexts Unsigned32
}
```

```
frwkDeviceIdPrid OBJECT-TYPE
    SYNTAX      InstanceId
    STATUS      current
    DESCRIPTION
        "An index to uniquely identify an instance of this
        provisioning class."

    ::= { frwkDeviceIdEntry 1 }
```

```
frwkDeviceIdDescr OBJECT-TYPE
    SYNTAX      SnmpAdminString
    STATUS      current
    DESCRIPTION
        "A textual description of the PEP. This value should include
        the name and version identification of the PEP's hardware
        and software."

    ::= { frwkDeviceIdEntry 2 }
```

```
frwkDeviceIdMaxMsg OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "octets"
    STATUS      current
    DESCRIPTION
        "The maximum message size, in octets, that the device
```

is capable of processing. Received messages with a size in excess of this value must cause the PEP to return an error to the PDP containing the global error code 'maxMsgSizeExceeded'. This is an additional error-avoidance

[Page 17]

mechanism to allow the administrator to have the ability to control the message size of messages sent to the device. The device should send NULL for this attributes if it not defined."

::= { frwkDeviceIdEntry 3 }

frwkDeviceIdMaxContexts OBJECT-TYPE

SYNTAX Unsigned32

UNITS "contexts"

STATUS current

DESCRIPTION

"The maximum number of unique contexts supported by the device. This is an additional error-avoidance mechanism to allow the administrators to have the ability to control the number of contexts installed on the device. The device should send NULL for this attribute if it is not specified."

::= { frwkDeviceIdEntry 4 }

--

-- Component Limitations Table

--

-- This table supports the ability to export information

-- detailing provisioning class/attribute implementation limitations

-- to the policy management system.

frwkComplimitsTable OBJECT-TYPE

SYNTAX SEQUENCE OF FrwkComplimitsEntry

PIB-ACCESS notify

STATUS current

DESCRIPTION

"Each instance of this class identifies a provisioning class or attribute and a limitation related to the implementation of the class/attribute in the device. Additional information providing guidance related to the limitation may also be present. These PRIs are sent to the PDP to indicate which PRCs or PRC attributes the device supports in a restricted manner."

::= { frwkBasePibClasses 4 }

frwkComplimitsEntry OBJECT-TYPE

SYNTAX FrwkComplimitsEntry

STATUS current

DESCRIPTION

"An instance of the frwkCompLimits class that identifies a PRC or PRC attribute and a limitation related to the PRC or PRC attribute implementation supported by the device. [\[COPS-PR\]](#) lists the error codes that MUST be returned (if

applicable)for policy installation that don't abide by the restrictions indicated by the limitations exported. [SPPI] defines an INSTALL-ERRORS clause that allows PIB designers to define PRC specific error codes that can be returned for policy installation. This allows efficient debugging of PIB implementations."

```
PIB-INDEX { frwkCompLimitsPrid }
UNIQUENESS { frwkCompLimitsComponent,
              frwkCompLimitsAttrPos,
              frwkCompLimitsNegation,
              frwkCompLimitsType,
              frwkCompLimitsSubType,
              frwkCompLimitsGuidance }
```

```
::= { frwkCompLimitsTable 1 }
```

```
FrwkCompLimitsEntry ::= SEQUENCE {
    frwkCompLimitsPrid      InstanceId,
    frwkCompLimitsComponent PrcIdentifier,
    frwkCompLimitsAttrPos   Unsigned32,
    frwkCompLimitsNegation   TruthValue,
    frwkCompLimitsType       Unsigned32,
    frwkCompLimitsSubType    Unsigned32,
    frwkCompLimitsGuidance   OCTET STRING
}
```

```
frwkCompLimitsPrid OBJECT-TYPE
    SYNTAX      InstanceId
    STATUS      current
    DESCRIPTION
        "An arbitrary integer index that uniquely identifies an
        instance of the frwkCompLimits class."
```

```
::= { frwkCompLimitsEntry 1 }
```

```
frwkCompLimitsComponent OBJECT-TYPE
    SYNTAX      PrcIdentifier
    STATUS      current
    DESCRIPTION
        "The value is the OID of a PRC (the table entry) which is
        supported in some limited fashion or contains an attribute
        that is supported in some limited fashion with regard to
        it's definition in the associated PIB module. The same OID
        may appear in the table several times, once for each
        implementation limitation acknowledged by the device."
```

```
::= { frwkCompLimitsEntry 2 }
```



**frwkCompLimitsAttrPos OBJECT-TYPE**

SYNTAX Unsigned32

STATUS current

## DESCRIPTION

"The relative position of the attribute within the PRC specified by the frwkCompLimitsComponent. A value of 1 would represent the first columnar object in the PRC and a value of N would represent the Nth columnar object in the PRC. A NULL value indicates that the limit applies to the PRC itself and not to a specific attribute."

::= { frwkCompLimitsEntry 3 }

**frwkCompLimitsNegation OBJECT-TYPE**

SYNTAX TruthValue

STATUS current

## DESCRIPTION

"A boolean value ,if TRUE, negates the component limit exported."

::= { frwkCompLimitsEntry 4 }

**frwkCompLimitsType OBJECT-TYPE**

```
SYNTAX  Unsigned32 {
    priSpaceLimited(1),
    attrValueSupLimited(2),
    attrEnumSupLimited(3),
    attrLengthLimited(4),
    prcLimitedNotify(5)
}
```

STATUS current

## DESCRIPTION

"A value describing an implementation limitation for the device related to the PRC or PRC attribute identified by the frwkCompLimitsComponent and the frwkCompLimitsAttrPos attributes in this class instance."

Values for this object are one of the following:

priSpaceLimited(1) - No more instances than that specified by the guidance value may be installed in the given class. The component identified MUST be a valid PRC. The SubType used MUST be valueOnly(9).

attrValueSupLimited(2) - Limited values are acceptable for the identified component. The component identified MUST be a valid PRC attribute. The guidance OCTET STRING will be

decoded according to the attribute type.

attrEnumSupLimited(3) - Limited enumeration values are legal for the identified component. The attribute identified MUST



be a valid enum type.

attrLengthLimited(4) - The length of the specified value for the identified component is limited. The component identified MUST be a valid PRC attribute of base-type OCTET STRING.

prcLimitedNotify (5) - The component is currently limited for use by request or report messages prohibiting decision installation. The component identified must be a valid PRC."

::= { frwkCompLimitsEntry 5 }

frwkCompLimitsSubType OBJECT-TYPE

```
SYNTAX      Unsigned32 {
                                none(1),
                                lengthMin(2),
                                lengthMax(3),
                                rangeMin(4),
                                rangeMax(5),
                                enumMin(6),
                                enumMax(7),
                                enumOnly(8),
                                valueOnly(9)
                                }
```

STATUS current

DESCRIPTION

"This object indicates the type of guidance related to the noted limitation (as indicated by the frwkCompLimitsType attribute) that is provided in the frwkCompLimitsGuidance attribute.

A value of 'none(1)' means that no additional guidance is provided for the noted limitation type.

A value of 'lengthMin(2)' means that the guidance attribute provides data related to the minimum acceptable length for the value of the identified component. A corresponding class instance specifying the 'lengthMax(3)' value is required in conjunction with this sub-type.

A value of 'lengthMax(3)' means that the guidance attribute provides data related to the maximum acceptable length for the value of the identified component. A corresponding class instance specifying the 'lengthMin(2)' value is required in conjunction with this sub-type.

A value of 'rangeMin(4)' means that the guidance attribute provides data related to the lower bound of the range for the value of the identified component. A corresponding class instance

specifying the 'rangeMax(5)' value is required in conjunction with this sub-type.

A value of 'rangeMax(5)' means that the guidance attribute provides data related to the upper bound of the range for the value of the identified component. A corresponding class instance specifying the 'rangeMin(4)' value is required in conjunction with this sub-type.

A value of 'enumMin(6)' means that the guidance attribute provides data related to the lowest enumeration acceptable for the value of the identified component. A corresponding class instance specifying the 'enumMax(7)' value is required in conjunction with this sub-type.

A value of 'enumMax(7)' means that the guidance attribute provides data related to the largest enumeration acceptable for the value of the identified component. A corresponding class instance specifying the 'enumMin(6)' value is required in conjunction with this sub-type.

A value of 'enumOnly(8)' means that the guidance attribute provides data related to a single enumeration acceptable for the value of the identified component.

A value of 'valueOnly(9)' means that the guidance attribute provides data related to a single value that is acceptable for the identified component.

For example, an implementation of the frwkIpFilter class may be limited in several ways, such as address mask, protocol and Layer 4 port options. These limitations could be exported using this table with the following instances:

Component	Type	Sub Type	Guidance
-----	-----	-----	-----
frwkIpFilterDstAddrMask	attrValueSupLimited	valueOnly	24
frwkIpFilterSrcAddrMask	attrValueSupLimited	valueOnly	24
frwkIpFilterProtocol	attrValueSupLimited	rangeMin	10
frwkIpFilterProtocol	attrValueSupLimited	rangeMax	20

The above entries describe a number of limitations that may be in effect for the frwkIpFilter class on a given

device. The limitations include restrictions on acceptable values for certain attributes.

Also, an implementation of a PRC may be limited in the ways

[Page 22]

it can be accessed. For instance, for a fictitious PRC  
dscpMapEntry, which has a PIB-ACCESS of 'install-notify':

Component	Type	SubType	Guidance
dscpMapEntry	prcLimitedNotify	none	zero-length string."

```
::= { frwkComplimitsEntry 6 }
```

frwkComplimitsGuidance OBJECT-TYPE

SYNTAX OCTET STRING

STATUS current

DESCRIPTION

"A value used to convey additional information related to the implementation limitation. Note that a guidance value will not necessarily be provided for all exported limitations. If a guidance value is not provided, the value must be a zero-length string.

The format of the guidance value, if one is present as indicated by the frwkComplimitsSubType attribute, is described by the following table. Note that the type of guidance value is dictated by the type of the component whose limitation is being exported, interpreted in the context of the frwkComplimitsType and frwkComplimitsSubType values.

Note that numbers are encoded in network byte order.

Base Type	Value
-----	-----
Unsigned32/Integer32	32-bit value.
Unsigned64/Integer64	64-bit Value.
OCTET STRING	octets of data.
OID	32-bit OID components."

```
::= { frwkComplimitsEntry 7 }
```

```
--
-- The device interface capabilities and role combo classes group
--
```

frwkDeviceCapClasses

OBJECT IDENTIFIER ::= { frameworkPib 2 }

```
--
-- Interface Capability Set Table
--
```



## frwkIfCapSetTable OBJECT-TYPE

SYNTAX SEQUENCE OF FrwkIfCapSetEntry

PIB-ACCESS notify

STATUS current

## DESCRIPTION

"This class describes the interfaces that exist on the device. Associated with each interface is a set of capabilities. The capability set is given a unique name that identifies the interface type. These capabilities are used by the PDP to determine policy information to be associated with interfaces of this type."

::= { frwkDeviceCapClasses 1 }

## frwkIfCapSetEntry OBJECT-TYPE

SYNTAX FrwkIfCapSetEntry

STATUS current

## DESCRIPTION

"An instance of this class describes the characteristics of a type of an interface."

PIB-INDEX { frwkIfCapSetPrid }

UNIQUENESS { frwkIfCapSetName,  
frwkIfCapSetCapability }

::= { frwkIfCapSetTable 1 }

FrwkIfCapSetEntry ::= SEQUENCE {

frwkIfCapSetPrid	InstanceId,
frwkIfCapSetName	SnmpAdminString,
frwkIfCapSetCapability	Prid

}

## frwkIfCapSetPrid OBJECT-TYPE

SYNTAX InstanceId

STATUS current

## DESCRIPTION

"An arbitrary integer index that uniquely identifies a instance of the class."

::= { frwkIfCapSetEntry 1 }

## frwkIfCapSetName OBJECT-TYPE

SYNTAX SnmpAdminString

STATUS current

## DESCRIPTION

"The name for the capability set. The capability set name is the unique identifier of an interface type."

::= { frwkIfCapSetEntry 2 }



## frwkIfCapSetCapability OBJECT-TYPE

SYNTAX Prid

STATUS current

## DESCRIPTION

"The complete PRC OID and instance identifier specifying the capability PRC instance for the interface."

::= { frwkIfCapSetEntry 3 }

--

-- Interface Capabilities Set Name and Role Combination Table

--

## frwkIfCapSetRoleComboTable OBJECT-TYPE

SYNTAX SEQUENCE OF FrwkIfCapSetRoleComboEntry

PIB-ACCESS notify

STATUS current

## DESCRIPTION

"Policy for an interface depends not only on the capability set of an interface but also on its roles. This table specifies all the <interface capability set name, role combination> tuples currently on the device."

::= { frwkDeviceCapClasses 2 }

## frwkIfCapSetRoleComboEntry OBJECT-TYPE

SYNTAX FrwkIfCapSetRoleComboEntry

STATUS current

## DESCRIPTION

"An instance of this class describes a combination of an interface capability set name and a role combination."

PIB-INDEX { frwkIfCapSetRoleComboPrid }

UNIQUENESS { frwkIfCapSetRoleComboName,  
frwkIfCapSetRoleComboRoles }

::= { frwkIfCapSetRoleComboTable 1 }

## FrwkIfCapSetRoleComboEntry ::= SEQUENCE {

frwkIfCapSetRoleComboPrid InstanceId,

frwkIfCapSetRoleComboName SnmpAdminString,

frwkIfCapSetRoleComboRoles RoleCombination

}



**frwkIfCapSetRoleComboPrid OBJECT-TYPE**

SYNTAX InstanceId

STATUS current

## DESCRIPTION

"An arbitrary integer index that uniquely identifies a instance of the class."

::= { frwkIfCapSetRoleComboEntry 1 }

**frwkIfCapSetRoleComboName OBJECT-TYPE**

SYNTAX SnmpAdminString

STATUS current

## DESCRIPTION

"The name of the interface capability set. This name must exist in frwkIfCapSetTable."

::= { frwkIfCapSetRoleComboEntry 2 }

**frwkIfCapSetRoleComboRoles OBJECT-TYPE**

SYNTAX RoleCombination

STATUS current

## DESCRIPTION

"A role combination. The PEP requires policy for interfaces with this role combination and of capability set name specified by frwkIfCapSetRoleComboName."

::= { frwkIfCapSetRoleComboEntry 3 }

--

-- The Classification classes group

--

**frwkClassifierClasses**

OBJECT IDENTIFIER ::= { frameworkPib 3 }

--

-- The Base Filter Table

--

**frwkBaseFilterTable OBJECT-TYPE**

SYNTAX SEQUENCE OF FrwkBaseFilterEntry

PIB-ACCESS install

STATUS current

## DESCRIPTION

"The Base Filter class. A packet has to match all fields in an Filter. Wildcards may be specified for those fields that are not relevant."

```
::= { frwkClassifierClasses 1 }
```

## frwkBaseFilterEntry OBJECT-TYPE

SYNTAX FrwkBaseFilterEntry

STATUS current

## DESCRIPTION

"An instance of the frwkBaseFilter class."

PIB-INDEX { frwkBaseFilterPrid }

::= { frwkBaseFilterTable 1 }

## FrwkBaseFilterEntry ::= SEQUENCE {

frwkBaseFilterPrid InstanceId,

frwkBaseFilterNegation TruthValue

}

## frwkBaseFilterPrid OBJECT-TYPE

SYNTAX InstanceId

STATUS current

## DESCRIPTION

"An integer index to uniquely identify this Filter among all the Filters."

::= { frwkBaseFilterEntry 1 }

## frwkBaseFilterNegation OBJECT-TYPE

SYNTAX TruthValue

STATUS current

## DESCRIPTION

"This attribute behaves like a logical NOT for the filter. If the packet matches this filter and the value of this attribute is true, the action associated with this filter is not applied to the packet. If the value of this attribute is false, then the action is applied to the packet."

::= { frwkBaseFilterEntry 2 }

--

-- The IP Filter Table

--

## frwkIpFilterTable OBJECT-TYPE

SYNTAX SEQUENCE OF FrwkIpFilterEntry

PIB-ACCESS install

STATUS current

## DESCRIPTION

"Filter definitions. A packet has to match all fields in a filter. Wildcards may be specified for those fields that are not relevant."

```

INSTALL-ERRORS {
    invalidDstL4PortData(1),
    invalidSrcL4PortData(2)
}
::= { frwkClassifierClasses 2 }

frwkIpFilterEntry OBJECT-TYPE
    SYNTAX      FrwkIpFilterEntry
    STATUS      current
    DESCRIPTION
        "An instance of the frwkIpFilter class."

    EXTENDS { frwkBaseFilterEntry }
    UNIQUENESS { frwkBaseFilterNegation,
        FrwkIpFilterDstAddrType,
        frwkIpFilterDstAddr,
        frwkIpFilterDstAddrMask,
        frwkIpFilterSrcAddrType,
        frwkIpFilterSrcAddr,
        frwkIpFilterSrcAddrMask,
        frwkIpFilterDscp,
        frwkIpFilterProtocol,
        frwkIpFilterDstL4PortMin,
        frwkIpFilterDstL4PortMax,
        frwkIpFilterSrcL4PortMin,
        frwkIpFilterSrcL4PortMax }

    ::= { frwkIpFilterTable 1 }

FrwkIpFilterEntry ::= SEQUENCE {
    frwkIpFilterDstAddrType      InetAddressType,
    frwkIpFilterDstAddr          InetAddress,
    frwkIpFilterDstAddrMask      Unsigned32,
    frwkIpFilterSrcAddrType      InetAddressType,
    frwkIpFilterSrcAddr          InetAddress,
    frwkIpFilterSrcAddrMask      Unsigned32,
    frwkIpFilterDscp             Integer32,
    frwkIpFilterProtocol         Integer32,
    frwkIpFilterDstL4PortMin     Unsigned32,
    frwkIpFilterDstL4PortMax     Unsigned32,
    frwkIpFilterSrcL4PortMin     Unsigned32,
    frwkIpFilterSrcL4PortMax     Unsigned32
}

frwkIpFilterDstAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    STATUS      current

```

#### DESCRIPTION

"The address type enumeration value [[INETADDR](#)] to specify the type of the packet's destination IP address."

::= { frwkIpFilterEntry 1 }



## frwkIpFilterDstAddr OBJECT-TYPE

SYNTAX InetAddress

STATUS current

## DESCRIPTION

"The IP address [[INETADDR](#)] to match against the packet's destination IP address."

::= { frwkIpFilterEntry 2 }

## frwkIpFilterDstAddrMask OBJECT-TYPE

SYNTAX Unsigned32 (0..128)

STATUS current

## DESCRIPTION

"The length of a mask for the matching of the destination IP address. Masks are constructed by setting bits in sequence from the most-significant bit downwards for frwkIpFilterDstAddrMask bits length. All other bits in the mask, up to the number needed to fill the length of the address frwkIpFilterDstAddr are cleared to zero. A zero bit in the mask then means that the corresponding bit in the address always matches."

::= { frwkIpFilterEntry 3 }

## frwkIpFilterSrcAddrType OBJECT-TYPE

SYNTAX InetAddressType

STATUS current

## DESCRIPTION

"The address type enumeration value to specify the type of the packet's source IP address."

::= { frwkIpFilterEntry 4 }

## frwkIpFilterSrcAddr OBJECT-TYPE

SYNTAX InetAddress

STATUS current

## DESCRIPTION

"The IP address to match against the packet's source IP address."

::= { frwkIpFilterEntry 5 }

## frwkIpFilterSrcAddrMask OBJECT-TYPE

SYNTAX           Unsigned32 (0..128)

STATUS           current

DESCRIPTION

    "The length of a mask for the matching of the source IP

address. Masks are constructed by setting bits in sequence from the most-significant bit downwards for frwkIpFilterSrcAddrMask bits length. All other bits in the mask, up to the number needed to fill the length of the address frwkIpFilterSrcAddr are cleared to zero. A zero bit in the mask then means that the corresponding bit in the address always matches."

::= { frwkIpFilterEntry 6 }

frwkIpFilterDscp OBJECT-TYPE

SYNTAX Integer32 (-1 | 0..63)

STATUS current

DESCRIPTION

"The value that the DSCP in the packet can have and match this filter. A value of -1 indicates that a specific DSCP value has not been defined and thus all DSCP values are considered a match."

::= { frwkIpFilterEntry 7 }

frwkIpFilterProtocol OBJECT-TYPE

SYNTAX Integer32 (-1 | 0..255)

STATUS current

DESCRIPTION

"The IP protocol to match against the packet's protocol. A value of -1 means match all."

::= { frwkIpFilterEntry 8 }

frwkIpFilterDstL4PortMin OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

STATUS current

DESCRIPTION

"The minimum value that the packet's layer 4 destination port number can have and match this filter. This value must be equal to or lesser that the value specified for this filter in frwkIpFilterDstL4PortMax."

::= { frwkIpFilterEntry 9 }

frwkIpFilterDstL4PortMax OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

STATUS current

DESCRIPTION

"The maximum value that the packet's layer 4 destination

port number can have and match this filter. This value must be equal to or greater than the value specified for this filter in frwkIpFilterDstL4PortMin."

```
::= { frwkIpFilterEntry 10 }
```

frwkIpFilterSrcL4PortMin OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

STATUS current

DESCRIPTION

"The minimum value that the packet's layer 4 source port number can have and match this filter. This value must be equal to or lesser that the value specified for this filter in frwkIpFilterSrcL4PortMax."

```
::= { frwkIpFilterEntry 11 }
```

frwkIpFilterSrcL4PortMax OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

STATUS current

DESCRIPTION

"The maximum value that the packet's layer 4 source port number can have and match this filter. This value must be equal to or greater that the value specified for this filter in frwkIpFilterSrcL4PortMin."

```
::= { frwkIpFilterEntry 12 }
```

--

-- The IEEE 802 Filter Table

--

-- The IEEE 802 Filter Table supports the specification of IEEE

-- 802-based [\[802\]](#) (e.g., 802.3) information that is used to perform

-- traffic classification.

--

frwk802FilterTable OBJECT-TYPE

SYNTAX SEQUENCE OF Frwk802FilterEntry

PIB-ACCESS install

STATUS current

DESCRIPTION

"IEEE 802-based filter definitions. A class that contains attributes of IEEE 802 (e.g., 802.3) traffic that form filters that are used to perform traffic classification."

```
::= { frwkClassifierClasses 3 }
```

frwk802FilterEntry OBJECT-TYPE

SYNTAX            Frwk802FilterEntry

STATUS            current

DESCRIPTION

    "IEEE 802-based filter definitions. An entry specifies

[Page 31]

(potentially) several distinct matching components. Each component is tested against the data in a frame individually. An overall match occurs when all of the individual components match the data they are compared against in the frame being processed. A failure of any one test causes the overall match to fail.

Wildcards may be specified for those fields that are not relevant."

```
EXTENDS { frwkBaseFilterEntry }
UNIQUENESS { frwkBaseFilterNegation,
              frwk802FilterDstAddr,
              frwk802FilterDstAddrMask,
              frwk802FilterSrcAddr,
              frwk802FilterSrcAddrMask,
              frwk802FilterVlanId,
              frwk802FilterVlanTagRequired,
              frwk802FilterEtherType,
              frwk802FilterUserPriority }
```

```
::= { frwk802FilterTable 1 }
```

```
Frwk802FilterEntry ::= SEQUENCE {
    frwk802FilterDstAddr      PhysAddress,
    frwk802FilterDstAddrMask  PhysAddress,
    frwk802FilterSrcAddr      PhysAddress,
    frwk802FilterSrcAddrMask  PhysAddress,
    frwk802FilterVlanId       Integer32,
    frwk802FilterVlanTagRequired Unsigned32,
    frwk802FilterEtherType     Integer32,
    frwk802FilterUserPriority  BITS
}
```

frwk802FilterDstAddr OBJECT-TYPE

```
SYNTAX      PhysAddress
STATUS      current
```

#### DESCRIPTION

"The 802 address against which the 802 DA of incoming traffic streams will be compared. Frames whose 802 DA matches the physical address specified by this object, taking into account address wildcarding as specified by the frwk802FilterDstAddrMask object, are potentially subject to the processing guidelines that are associated with this entry through the related action class."

```
::= { frwk802FilterEntry 1 }
```





**frwk802FilterDstAddrMask OBJECT-TYPE**

SYNTAX PhysAddress

STATUS current

**DESCRIPTION**

"This object specifies the bits in a 802 destination address that should be considered when performing a 802 DA comparison against the address specified in the frwk802FilterDstAddr object.

The value of this object represents a mask that is logically and'ed with the 802 DA in received frames to derive the value to be compared against the frwk802FilterDstAddr address. A zero bit in the mask thus means that the corresponding bit in the address always matches. The frwk802FilterDstAddr value must also be masked using this value prior to any comparisons.

The length of this object in octets must equal the length in octets of the frwk802FilterDstAddr. Note that a mask with no bits set (i.e., all zeroes) effectively wildcards the frwk802FilterDstAddr object."

::= { frwk802FilterEntry 2 }

**frwk802FilterSrcAddr OBJECT-TYPE**

SYNTAX PhysAddress

STATUS current

**DESCRIPTION**

"The 802 MAC address against which the 802 MAC SA of incoming traffic streams will be compared. Frames whose 802 MAC SA matches the physical address specified by this object, taking into account address wildcarding as specified by the frwk802FilterSrcAddrMask object, are potentially subject to the processing guidelines that are associated with this entry through the related action class."

::= { frwk802FilterEntry 3 }

**frwk802FilterSrcAddrMask OBJECT-TYPE**

SYNTAX PhysAddress

STATUS current

**DESCRIPTION**

"This object specifies the bits in a 802 MAC source address that should be considered when performing a 802 MAC SA comparison against the address specified in the frwk802FilterSrcAddr object.

The value of this object represents a mask that is logically

and'ed with the 802 MAC SA in received frames to derive the value to be compared against the frwk802FilterSrcAddr address. A zero bit in the mask thus means that the corresponding bit in the address always matches. The

frwk802FilterSrcAddr value must also be masked using this value prior to any comparisons.

The length of this object in octets must equal the length in octets of the frwk802FilterSrcAddr. Note that a mask with no bits set (i.e., all zeroes) effectively wildcards the frwk802FilterSrcAddr object."

```
::= { frwk802FilterEntry 4 }
```

#### frwk802FilterVlanId OBJECT-TYPE

SYNTAX Integer32 (-1 | 1..4094)

STATUS current

##### DESCRIPTION

"The VLAN ID (VID) that uniquely identifies a VLAN within the device. This VLAN may be known or unknown (i.e., traffic associated with this VID has not yet been seen by the device) at the time this entry is instantiated.

Setting the frwk802FilterVlanId object to -1 indicates that VLAN data should not be considered during traffic classification."

```
::= { frwk802FilterEntry 5 }
```

#### frwk802FilterVlanTagRequired OBJECT-TYPE

SYNTAX Unsigned32 {  
taggedOnly(1),  
priorityTaggedPlus(2),  
untaggedOnly(3),  
ignoreTag(4)  
}

STATUS current

##### DESCRIPTION

"This object indicates whether the presence of an IEEE 802.1Q VLAN tag in data link layer frames must be considered when determining if a given frame matches this 802 filter entry.

A value of 'taggedOnly(1)' means that only frames containing a VLAN tag with a non-Null VID (i.e., a VID in the range 1..4094) will be considered a match.

A value of 'priorityTaggedPlus(2)' means that only frames containing a VLAN tag, regardless of the value of the VID, will be considered a match.

A value of 'untaggedOnly(3)' indicates that only untagged frames will match this filter component.

The presence of a VLAN tag is not taken into consideration in terms of a match if the value is 'ignoreTag(4)'."

```
::= { frwk802FilterEntry 6 }
```

#### frwk802FilterEtherType OBJECT-TYPE

SYNTAX Integer32 (-1 | 0..'ffff'h)

STATUS current

##### DESCRIPTION

"This object specifies the value that will be compared against the value contained in the EtherType field of an IEEE 802 frame. Example settings would include 'IP' (0x0800), 'ARP' (0x0806) and 'IPX' (0x8137).

Setting the frwk802FilterEtherTypeMin object to -1 indicates that EtherType data should not be considered during traffic classification.

Note that the position of the EtherType field depends on the underlying frame format. For Ethernet-II encapsulation, the EtherType field follows the 802 MAC source address. For 802.2 LLC/SNAP encapsulation, the EtherType value follows the Organization Code field in the 802.2 SNAP header. The value that is tested with regard to this filter component therefore depends on the data link layer frame format being used. If this 802 filter component is active when there is no EtherType field in a frame (e.g., 802.2 LLC), a match is implied."

```
::= { frwk802FilterEntry 7 }
```

#### frwk802FilterUserPriority OBJECT-TYPE

SYNTAX BITS {  
     matchPriority0(0),  
     matchPriority1(1),  
     matchPriority2(2),  
     matchPriority3(3),  
     matchPriority4(4),  
     matchPriority5(5),  
     matchPriority6(6),  
     matchPriority7(7)  
 }

STATUS current

##### DESCRIPTION

"The set of values, representing the potential range of user priority values, against which the value contained

in the user priority field of a tagged 802.1 frame is compared. A test for equality is performed when determining if a match exists between the data in a data link layer frame and the value of this 802 filter component. Multiple

values may be set at one time such that potentially several different user priority values may match this 802 filter component.

Setting all of the bits that are associated with this object causes all user priority values to match this attribute. This essentially makes any comparisons with regard to user priority values unnecessary. Untagged frames are treated as an implicit match."

```
::= { frwk802FilterEntry 8 }
```

```
--
```

```
-- Conformance Section
```

```
--
```

```
frwkBasePibConformance
```

```
    OBJECT IDENTIFIER ::= { frameworkPib 4 }
```

```
frwkBasePibCompliances
```

```
    OBJECT IDENTIFIER ::= { frwkBasePibConformance 1 }
```

```
frwkBasePibGroups
```

```
    OBJECT IDENTIFIER ::= { frwkBasePibConformance 2 }
```

```
frwkBasePibCompliance MODULE-COMPLIANCE
```

```
    STATUS current
```

```
    DESCRIPTION
```

```
        "Describes the requirements for conformance to the
        Framework PIB."
```

```
    MODULE -- this module
```

```
        MANDATORY-GROUPS { frwkPrcSupportGroup,
                            frwkPibIncarnationGroup,
                            frwkDeviceIdGroup,
                            frwkCompLimitsGroup,
                            frwkIfCapSetGroup,
                            frwkIfCapSetRoleComboGroup }
```

```
    OBJECT frwkPibIncarnationLongevity
```

```
    PIB-MIN-ACCESS notify
```

```
    DESCRIPTION "Install support is not required."
```

```
    OBJECT frwkPibIncarnationTtl
```

```
    PIB-MIN-ACCESS notify
```

```
    DESCRIPTION "Install support is not required."
```

OBJECT	frwkPibIncarnationActive
PIB-MIN-ACCESS	notify
DESCRIPTION	"Install support is not required."



```
GROUP    frwkBaseFilterGroup
DESCRIPTION
    "The frwkBaseFilterGroup is mandatory if filtering
    based on traffic components is supported."

GROUP    frwkIpFilterGroup
DESCRIPTION
    "The frwkIpFilterGroup is mandatory if filtering
    based on IP traffic components is supported."

GROUP    frwk802FilterGroup
DESCRIPTION
    "The frwk802FilterGroup is mandatory if filtering
    based on 802 traffic criteria is supported."

::= { frwkBasePibCompliances 1 }

frwkPrcSupportGroup OBJECT-GROUP
OBJECTS {
    frwkPrcSupportSupportedPrc,
    frwkPrcSupportSupportedAttrs }
STATUS   current
DESCRIPTION
    "Objects from the frwkPrcSupportTable."

::= { frwkBasePibGroups 1 }

frwkPibIncarnationGroup OBJECT-GROUP
OBJECTS {
    frwkPibIncarnationName,
    frwkPibIncarnationId,
    frwkPibIncarnationLongevity,
    frwkPibIncarnationTtl,
    frwkPibIncarnationActive }
STATUS   current
DESCRIPTION
    "Objects from the frwkDevicePibIncarnationTable."

::= { frwkBasePibGroups 2 }

frwkDeviceIdGroup OBJECT-GROUP
OBJECTS {
    frwkDeviceIdDescr,
    frwkDeviceIdMaxMsg,
    frwkDeviceIdMaxContexts }
STATUS   current
DESCRIPTION
```

"Objects from the frwkDeviceIdTable."

::= { frwkBasePibGroups 3 }

```
frwkComplimitsGroup OBJECT-GROUP
  OBJECTS {
    frwkComplimitsComponent,
    frwkComplimitsAttrPos,
    frwkComplimitsNegation,
    frwkComplimitsType,
    frwkComplimitsSubType,
    frwkComplimitsGuidance }
  STATUS current
  DESCRIPTION
    "Objects from the frwkComplimitsTable."

  ::= { frwkBasePibGroups 4 }

frwkIfCapSetGroup OBJECT-GROUP
  OBJECTS {
    frwkIfCapSetName,
    frwkIfCapSetCapability }
  STATUS current
  DESCRIPTION
    "Objects from the frwkIfCapSetTable."

  ::= { frwkBasePibGroups 5 }

frwkIfCapSetRoleComboGroup OBJECT-GROUP
  OBJECTS {
    frwkIfCapSetRoleComboName,
    frwkIfCapSetRoleComboRoles }
  STATUS current
  DESCRIPTION
    "Objects from the frwkIfCapSetRoleComboTable."

  ::= { frwkBasePibGroups 6 }

frwkBaseFilterGroup OBJECT-GROUP
  OBJECTS {
    frwkBaseFilterNegation }
  STATUS current
  DESCRIPTION
    "Objects from the frwkBaseFilterTable."

  ::= { frwkBasePibGroups 7 }
```

frwkIpFilterGroup OBJECT-GROUP

[Page 38]

```
OBJECTS {
    frwkIpFilterDstAddrType,
    frwkIpFilterDstAddr,
    frwkIpFilterDstAddrMask,
    frwkIpFilterSrcAddrType,
    frwkIpFilterSrcAddr,
    frwkIpFilterSrcAddrMask,
    frwkIpFilterDscp,
    frwkIpFilterProtocol,
    frwkIpFilterDstL4PortMin,
    frwkIpFilterDstL4PortMax,
    frwkIpFilterSrcL4PortMin,
    frwkIpFilterSrcL4PortMax }
STATUS    current
DESCRIPTION
    "Objects from the frwkIpFilterTable."

::= { frwkBasePibGroups 8 }

frwk802FilterGroup OBJECT-GROUP
OBJECTS {
    frwk802FilterDstAddr,
    frwk802FilterDstAddrMask,
    frwk802FilterSrcAddr,
    frwk802FilterSrcAddrMask,
    frwk802FilterVlanId,
    frwk802FilterVlanTagRequired,
    frwk802FilterEtherType,
    frwk802FilterUserPriority }
STATUS    current
DESCRIPTION
    "Objects from the frwk802FilterTable."

::= { frwkBasePibGroups 9 }

END
```

## 7. Security Considerations

It is clear that this PIB is used for configuration using [[COPS-PR](#)], and anything that can be configured can be misconfigured, with potentially disastrous effect. At this writing, no security holes have been identified beyond those that the COPS base protocol security is itself intended to address. These relate primarily to controlled access to sensitive information and the ability to configure a device - or which might result from operator error, which is beyond the scope of any security architecture.

There are a number of provisioning classes defined in this PIB that have a PIB-ACCESS clause of install (read-create). Such objects may be considered sensitive or vulnerable in some network environments. The support for "Install" decisions sent over [[COPS-PR](#)] in a non-

secure environment without proper protection can have a negative effect on network operations. There are a number of provisioning classes in this PIB that may contain information that may be sensitive from a business perspective, in that they may represent a customer's service contract or the filters that the service provider chooses to apply to a customer's ingress or egress traffic. There are no PRCs that are sensitive in their own right, such as passwords or monetary amounts. It may be important to control even "Notify"(read-only) access to these PRCs and possibly to even encrypt the values of these PRIs when sending them over the network via COPS-PR. The use of IPSEC between the PDP and the PEP, as described in [[COPS](#)], provides the necessary protection against security threats. However, even if the network itself is secure, there is no control as to who on the secure network is allowed to "Install/Notify" (read/change/create/delete) the PRIs in this PIB.

It is then a customer/user responsibility to ensure that the PEP/PDP giving access to an instance of this PIB, is properly configured to give access to the PRIs only to those principals (users) that have legitimate rights to indeed "Install" or "Notify" (change/create/delete) them.

## **8. Author Information and Acknowledgments**

Michael Fine  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706 USA  
Phone: +1 408 527 8218  
Email: [mfine@cisco.com](mailto:mfine@cisco.com)

Keith McCloghrie  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706 USA  
Phone: +1 408 526 5260  
Email: [kzm@cisco.com](mailto:kzm@cisco.com)

John Seligson  
Nortel Networks, Inc.  
4401 Great America Parkway  
Santa Clara, CA 95054 USA  
Phone: +1 408 495 2992  
Email: [jseligso@nortelnetworks.com](mailto:jseligso@nortelnetworks.com)

Kwok Ho Chan  
Nortel Networks, Inc.

600 Technology Park Drive  
Billerica, MA 01821 USA  
Phone: +1 978 288 8175  
Email: [khchan@nortelnetworks.com](mailto:khchan@nortelnetworks.com)



Scott Hahn  
Intel Corp.  
2111 NE 25th Avenue  
Hillsboro, OR 97124 USA  
Phone: +1 503 264 8231  
Email: [scott.hahn@intel.com](mailto:scott.hahn@intel.com)

Ravi Sahita  
Intel Corp.  
2111 NE 25th Avenue  
Hillsboro, OR 97124 USA  
Phone: +1 503 712 1554  
Email: [ravi.sahita@intel.com](mailto:ravi.sahita@intel.com)

Andrew Smith  
Allegro Networks  
6399 San Ignacio Ave.  
San Jose  
CA 95119  
FAX: 415 345 1827  
Email: [andrew@allegronetworks.com](mailto:andrew@allegronetworks.com)

Francis Reichmeyer  
PFN, Inc.  
University Park at MIT  
26 Landsdowne Street  
Cambridge, MA 02139  
Phone: +1 617 494 9980  
Email: [franr@pfn.com](mailto:franr@pfn.com)

Special thanks to Carol Bell and David Durham for their many significant comments.

## 9. References

### [COPS]

Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, R., and A. Sastry, "The COPS (Common Open Policy Service) Protocol" [RFC 2748](#), January 2000.

### [COPS-PR]

K. Chan, D. Durham, S. Gai, S. Herzog, K. McCloghrie, F. Reichmeyer, J. Seligson, A. Smith, R. Yavatkar, "COPS Usage for Policy Provisioning," [draft-ietf-rap-pr-05.txt](#), October 30, 2000.

### [SPPI]

K. McCloghrie, et.al., "Structure of Policy Provisioning

Information," [draft-ietf-rap-sppi-05.txt](#), February 2001.

**[RAP-FRAMEWORK]**

R. Yavatkar, D. Pendarakis, "A Framework for Policy-based Admission Control", [RFC 2753](#), January 2000.

**[SNMP-SMI]**

K. McCloghrie, D. Perkins, J. Schoenwaelder, J. Case, M. Rose and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, [RFC 2578](#), April 1999.

**[INETADDR]**

M. Daniele, B. Haberman, S. Routhier and J. Schoenwaelder "Textual Conventions for Internet Network Addresses" [RFC 2851](#), June 2000

**[802]**

IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture, ANSI/IEEE Std 802, 1990.

**[SNMPFRWK]**

Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", [RFC 2571](#), May 1999

**[STD17]**

K. McCloghrie, M. Rose "Management Information Base for Network Management of TCP/IP-based internets: MIB-II" STD 17, [RFC 1213](#), March 1991

**10. Full Copyright**

Copyright (C) The Internet Society (2000). All Rights Reserved. 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.

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

[Page 42]

HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF  
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Table of Contents

Status of this Memo.....	<a href="#">1</a>
<a href="#">1</a> . Glossary.....	<a href="#">2</a>
<a href="#">2</a> . Introduction.....	<a href="#">2</a>
<a href="#">3</a> . General PIB Concepts.....	<a href="#">2</a>
<a href="#">3.1</a> . Roles.....	<a href="#">2</a>
<a href="#">3.1.1</a> . An Example.....	<a href="#">4</a>
<a href="#">3.2</a> . Multiple PIB Instances.....	<a href="#">5</a>
<a href="#">3.3</a> . Reporting of Device Capabilities.....	<a href="#">6</a>
<a href="#">3.4</a> . Reporting of Device Limitations.....	<a href="#">6</a>
<a href="#">4</a> . The Framework Role PIB module.....	<a href="#">7</a>
<a href="#">5</a> . Summary of the Framework PIB.....	<a href="#">8</a>
<a href="#">6</a> . The Framework PIB Module.....	<a href="#">11</a>
<a href="#">7</a> . Security Considerations.....	<a href="#">39</a>
<a href="#">8</a> . Author Information and Acknowledgments.....	<a href="#">40</a>
<a href="#">9</a> . References.....	<a href="#">41</a>
<a href="#">10</a> . Full Copyright.....	<a href="#">42</a>

