

Network Working Group
Internet Draft
Expires September 2000

M. Fine
K. McCloghrie
Cisco Systems
J. Seligson
K. Chan
Nortel Networks
S. Hahn
Intel
A. Smith
Extreme Networks
Francis Reichmeyer
IPHighway

March 10, 2000

Framework Policy Information Base

[draft-ietf-rap-frameworkpib-00.txt](#)

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.''

To view the current status of any Internet-Draft, please check the ``[id-abstracts.txt](#)'' listing contained in an Internet-Drafts Shadow Directory, see <http://www.ietf.org/shadow.html>.

1. Glossary

PRC	Policy Rule Class. A type of policy data.
PRI	Policy Rule Instance. An instance of a PRC.
PIB	Policy Information Base. The database of policy information.
PDP	Policy Decision Point. See [RAP-FRAMEWORK].
PEP	Policy Enforcement Point. See [RAP-FRAMEWORK].
PRID	Policy Rule 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 policy rule 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 PIB. However, some policy rule classes are common to all client types and replicated in each. This document presents the PIB classes 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. Policy rule classes have an attribute called a "role-

[Page 2]

Framework Policy Information Base

March 2000

combination" which is an unordered set of roles. Instances of a given policy rule class are applied to an interface if and only if the set of roles in the role combination is identical to 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 at connect time or whenever they change.

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

The concept and usage of roles in this document is consistent with that specified in [[POLICY](#)]. Roles are currently under discussion in the IETF's Policy WG; as and when that discussion reaches a conclusion, this PIB will be updated in accordance with that conclusion.

[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 PHB 5

P2: Packets from managers (role "manager") get PHB 6

To obtain policy, the PEP reports to the PDP that it has some interfaces with role combination "finance" and some with role combination

[Page 3]

Framework Policy Information Base

March 2000

"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 PHB 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](#)

Similar to SNMP contexts, [[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 which can then be made active by a single, short decision from the PDP.

With the COPS-PR protocol, each of these instances 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](#)]. The intent is to allow for the pre-provisioning of policy which can then be made active by a single, short decision from the PDP.

Although many PIB instances may be configured on a device (the maximum

[Page 4]

Framework Policy Information Base

March 2000

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 at connect time. This attribute is in the Incarnation Table described below.

Setting the attribute `policyPibIncarnationActive` to 'true' in one PIB instance automatically ensures 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 policy rule 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 POLICY-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.

[Page 5]

Reported device limitations may be accompanied by guidance values that can be used by a PDP to determine acceptable values for the identified attributes. The format of the guidance information must be specified where the errors used to signal implementation limitations are defined.

[4.](#) Summary of the Framework PIB

The Framework PIB comprises four PRCs intended to describe the capabilities of the device and its current configuration.

The 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 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.

PIB Incarnation Table

This table contains exactly one row (corresponding to one PRI). 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 any given time.

Policy Attribute 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. There are two problems with this: 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.

[Page 6]

To alleviate these limitations, the PEP can report some limitations of attribute values in the Attribute Limitations Table.

Policy Device Identification

This class contains a single policy rule 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 at client connect time so that the PDP can take into account certain device characteristics during policy installation.

5. PIB Operational Overview

All PRCs in this Framework PIB have POLICY-ACCESS values of notify or install-notify. Consequently the entire contents of these tables are reported to the PDP as part of each REQ message.

6. The Policy Framework PIB Module

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

```
IMPORTS
```

```
    Unsigned32, Integer32, PolicyInstanceId, MODULE-IDENTITY,  
    MODULE-COMPLIANCE, OBJECT-TYPE,  
        FROM COPS-PR-SPPI  
    TruthValue, TEXTUAL-CONVENTION  
        FROM SNMPv2-TC  
    Role, RoleCombination  
        FROM POLICY-DEVICE-AUX-MIB  
    SnmpAdminString  
        FROM SNMP-FRAMEWORK-MIB  
    OBJECT-GROUP  
        FROM SNMPv2-CONF;
```

```
policyFrameworkPib MODULE-IDENTITY  
    CLIENT-TYPE { all }  
    LAST-UPDATED "200003101800Z"  
    ORGANIZATION "IETF RAP WG"  
    CONTACT-INFO "  
        Michael Fine  
        Cisco Systems, Inc.
```

[Page 7]

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 policy rule classes that are required for support of all policies."

::= { tbd }

--

-- The root OID for PRCs in the Framework PIB

--

policyBasePibClass

OBJECT IDENTIFIER ::= { policyFrameworkPib 1 }

--

-- Textual Conventions

--

--

-- PRC Support Table

--

policyPrcSupportTable OBJECT-TYPE

SYNTAX SEQUENCE OF PolicyPrcSupportEntry

POLICY-ACCESS notify

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."

::= { policyBasePibClass 1 }

policyPrcSupportEntry OBJECT-TYPE

SYNTAX PolicyPrcSupportEntry

STATUS current

DESCRIPTION

"An instance of the policyPrcSupport class that identifies a specific policy class and associated attributes as supported by the device."

INDEX { policyPrcSupportPrid }

UNIQUENESS { policyPrcSupportSupportedPrc }

::= { policyPrcSupportTable 1 }

PolicyPrcSupportEntry ::= SEQUENCE {
 policyPrcSupportPrid PolicyInstanceId,
 policyPrcSupportSupportedPrc OBJECT IDENTIFIER,
 policyPrcSupportSupportedAttrs OCTET STRING,
 policyPrcSupportMaxPris Unsigned32
}

policyPrcSupportPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

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

::= { policyPrcSupportEntry 1 }

policyPrcSupportSupportedPrc OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

STATUS current

DESCRIPTION

"The object identifier of a supported PRC. There may not be more than one instance of the policyPrcSupport class with the same value of policyPrcSupportSupportedPrc."

::= { policyPrcSupportEntry 2 }

policyPrcSupportSupportedAttrs OBJECT-TYPE

SYNTAX OCTET STRING

STATUS current

DESCRIPTION

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

Each bit of this bit mask 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 of this bit mask 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 mask is *N* bits long and there are more than *N* class attributes then the bit mask is logically extended with 0's to the required length."

::= { policyPrcSupportEntry 3 }

policyPrcSupportMaxPris OBJECT-TYPE

SYNTAX Unsigned32

STATUS current

DESCRIPTION

"A non-negative value indicating the maximum numbers of policy rule instances that can be installed in the identified policy rule class. Note that actual number of PRIs that can be installed in a PRC at any given time may be less than this value based on the current operational state (e.g., resources currently consumed) of the device."

::= { policyPrcSupportEntry 4 }

--

-- PIB Incarnation Table

--

policyPibIncarnationTable OBJECT-TYPE

SYNTAX SEQUENCE OF PolicyPibIncarnationEntry

POLICY-ACCESS install-notify

STATUS current

DESCRIPTION

"This class contains a single policy rule instance 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 at client connect time so that the PDP can (attempt to) ascertain the current state of the PIB. A network manager may use the instance to determine the state of the device with regard to existing NMS interactions."

::= { policyBasePibClass 2 }

policyPibIncarnationEntry OBJECT-TYPE

SYNTAX PolicyPibIncarnationEntry

STATUS current

DESCRIPTION

"An instance of the policyPibIncarnation class. Only one instance of this policy class is ever instantiated."

INDEX { policyPibIncarnationPrid }

UNIQUENESS { policyPibIncarnationName }

::= { policyPibIncarnationTable 1 }

PolicyPibIncarnationEntry ::= SEQUENCE {

policyPibIncarnationPrid

PolicyInstanceId,

policyPibIncarnationName

SnmpAdminString,

policyPibIncarnationId

OCTET STRING,

policyPibIncarnationLongevity

INTEGER,

policyPibIncarnationTtl

Unsigned32,

policyPibIncarnationActive

TruthValue

}

policyPibIncarnationPrid OBJECT-TYPE

SYNTAX PolicyInstanceId

STATUS current

DESCRIPTION

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

::= { policyPibIncarnationEntry 1 }

[Page 11]

Framework Policy Information Base

March 2000

policyPibIncarnationName 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."

::= { policyPibIncarnationEntry 2 }

policyPibIncarnationId 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."

::= { policyPibIncarnationEntry 3 }

policyPibIncarnationLongevity OBJECT-TYPE

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

STATUS current

DESCRIPTION

"This attribute controls what the PEP does with the downloaded policy on receipt of 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 `policyPibIncarnationTtl`. 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

[Page 12]

Framework Policy Information Base

March 2000

reconnect.

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

::= { policyPibIncarnationEntry 3 }

policyPibIncarnationTtl OBJECT-TYPE

SYNTAX Unsigned32

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 `policyPibIncarnationLongevity` is set to `expireOnTimeout`."

::= { policyPibIncarnationEntry 4 }

policyPibIncarnationActive 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 becomes inactive and the `policyPibIncarnationActive` attribute in that PIB instance is

automatically set to false."

```
::= { policyPibIncarnationEntry 5 }
```

--

-- Device Identification Table

--

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

--

policyDeviceIdentificationTable OBJECT-TYPE

SYNTAX SEQUENCE OF PolicyDeviceIdentificationEntry

POLICY-ACCESS notify

[Page 13]

Framework Policy Information Base

March 2000

STATUS current

DESCRIPTION

"This class contains a single policy rule 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 at client connect time so that the PDP can take into account certain device characteristics during policy installation."

```
::= { policyDeviceConfig 3 }
```

policyDeviceIdentificationEntry OBJECT-TYPE

SYNTAX PolicyDeviceIdentificationEntry

STATUS current

DESCRIPTION

"An instance of the policyDeviceIdentification class. Only one instance of this policy class is ever instantiated."

INDEX { policyDeviceIdentificationPrid }

UNIQUENESS { policyDeviceIdentificationDescr,
policyDeviceIdentificationMaxMsg }

```
::= { policyDeviceIdentificationTable 1 }
```

```
PolicyDeviceIdentificationEntry ::= SEQUENCE {
    policyDeviceIdentificationPrid      PolicyInstanceId,
    policyDeviceIdentificationDescr     SnmpAdminString,
    policyDeviceIdentificationMaxMsg    Unsigned32
}
```

```
policyDeviceIndentificationPrid OBJECT-TYPE
    SYNTAX      PolicyInstanceId
    STATUS      current
    DESCRIPTION
        "An index to uniquely identify an instance of this
        policy class."
```

```
::= { policyDeviceIdentificationEntry 1 }
```

```
policyDeviceIdentificationDescr OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(0..255))
    STATUS      current
    DESCRIPTION
        "A textual description of the PEP. This
        value should include the name and version
        identification of the PEP's hardware and
```

[Page 14]

Framework Policy Information Base

March 2000

software."

```
::= { policyDeviceIdentificationEntry 2 }
```

```
policyDeviceIdentificationMaxMsg OBJECT-TYPE
    SYNTAX      Unsigned32
    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'."
```

```
::= { policyDeviceIdentificationEntry 3 }
```

--


```
-- Policy Component Limitations Table
--
-- This table supports the ability to export information
-- detailing policy class/attribute implementation limitations
-- to the policy management system.
--
```

```
policyCompLimitsTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF PolicyCompLimitsEntry
    POLICY-ACCESS   notify
    STATUS          current
    DESCRIPTION
        "Each instance of this class identifies a policy class or
        attribute and a limitation related to the implementaion 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."

    ::= { policyDeviceConfig 4 }
```

```
policyCompLimitsEntry OBJECT-TYPE
    SYNTAX          PolicyCompLimitsEntry
    STATUS          current
    DESCRIPTION
        "An instance of the policyCompLimits class that identifies
        a PRC or PRC attribute and a limitation related to the PRC
```

[Page 15]

or PRC attribute implementation supported by the device. All PRIs of this class represent errors that would be returned in relation to the identified component for policy installation requests that don't abide by the restrictions indicated by the error code and, possibly, a provided guidance value."

```
INDEX { policyCompLimitsPrid }
UNIQUENESS { policyCompLimitsComponent,
             policyCompLimitsType,
             policyCompLimitsGuidance }
```

```

 ::= { policyCompLimitsTable 1 }

PolicyCompLimitsEntry ::= SEQUENCE {
    policyCompLimitsPrid          PolicyInstanceId,
    policyCompLimitsComponent     OBJECT IDENTIFIER,
    policyCompLimitsType          Integer32,
    policyCompLimitsGuidance      OCTET STRING
}

policyCompLimitsPrid OBJECT-TYPE
    SYNTAX          PolicyInstanceId
    STATUS          current
    DESCRIPTION
        "An arbitrary integer index that uniquely identifies an
        instance of the policyCompLimits class."

 ::= { policyCompLimitsEntry 1 }

policyCompLimitsComponent OBJECT-TYPE
    SYNTAX          OBJECT IDENTIFIER
    STATUS          current
    DESCRIPTION
        "The object identifier of a PRC or PRC attribute that
        is supported in some limited fashion with regard to it's
        definition in the associated PIB module. The same PRC or
        PRC attribute identifier may appear in the table several
        times, once for each implementation limitation
        acknowledged by the device."

 ::= { policyCompLimitsEntry 2 }

policyCompLimitsType OBJECT-TYPE
    SYNTAX          Integer32

```

```

STATUS          current
DESCRIPTION
    "A value describing an implementation limitation for the
    device related to the PRC or PRC attribute identified by
    the policyCompLimitsComponent data in this class instance."

```

Values for this object are derived from the defined error values associated with the PRC of the identified attribute or the PRC itself. All genericPrc and specificPrc (defined in a PRC INSTALL-ERRORS clause) error codes represent valid limitation type values.

For example, an implementation of the qosIpAce 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:

Prid	Component	Type	Guidance
1	'qosIpAceDstAddrMask'	'valueSupLimited'	0xFFFFFFFF
2	'qosIpAceSrcAddrMask'	'valueSupLimited'	0xFFFFFFFF
3	'qosIpAceProtocol'	'valueSupLimited'	0x06 -- TCP
4	'qosIpAceProtocol'	'valueSupLimited'	0x17 -- UDP
5	'qosIpAceDstL4PortMin'	'invalidDstL4PortData'	
6	'qosIpAceDstL4PortMax'	'invalidDstL4PortData'	
7	'qosIpAcePermit'	'enumSupLimited'	'true'

The above entries describe a number of limitations that may be in effect for the qosIpAce class on a given device. The limitations include restrictions on acceptable values for certain attributes and indications of the relationship between related attributes."

```
::= { policyCompLimitsEntry 3 }
```

policyCompLimitsGuidance OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..64))

STATUS current

DESCRIPTION

"A value used to convey additional information related to the implementation limitation noted by the policyCompLimitsType attribute. The value of this attribute must interpreted in the context of the policyCompLimitsType value. Note that a guidance value will not necessarily be provided for all exported limitations.

Well-known genericPrc error codes that are applicable to all PRCs, such as 'attrValueSupLimited' and 'attrEnumSupLimited', have guidance value semantics as follows:

genericPrc	Guidance Semantics
attrValueSupLimited	Integer32 (4 octets) with supported value
attrEnumSupLimited	Integer32 (4 octets) with supported enumeration
attrMaxLengthExceeded	Integer32 (4 octets) with maximum supported length for attribute

The specificPrc error codes have the semantics of the associated guidance value specified where the installation error is defined if appropriate. Errors for which the semantics of the guidance value are not specified require this value to be treated in an implementation dependent manner."

```
::= { policyCompLimitsEntry 4 }
```

```
--
```

```
-- Conformance Section
```

```
--
```

```
policyBasePibConformance
```

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

```
policyBasePibCompliances
```

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

```
policyBasePibGroups
```

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

```
policyBasePibCompliance MODULE-COMPLIANCE
```

```
    STATUS current
```

```
    DESCRIPTION
```

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

```
MODULE -- this module
```

```
    MANDATORY-GROUPS { policyPrcSupportGroup,  
                        policyDevicePibIncarnationGroup,  
                        policyDeviceIdentificationGroup,  
                        policyCompLimitsGroup }
```

Framework Policy Information Base

March 2000

```
OBJECT      policyDevicePibIncarnationLongevity
MIN-ACCESS  notify
DESCRIPTION "Install support is not required."
```

```
OBJECT      policyDevicePibIncarnationTtl
MIN-ACCESS  notify
DESCRIPTION "Install support is not required."
```

```
OBJECT      policyDevicePibIncarnationActiveContext
MIN-ACCESS  notify
DESCRIPTION "Install support is not required."
```

```
::= { policyBasePibCompliances 1 }
```

```
policyPrcSupportGroup OBJECT-GROUP
```

```
OBJECTS {
    policyPrcSupportSupportedPrc,
    policyPrcSupportSupportedAttrs,
    policyPrcSupportMaxPris
}
STATUS current
DESCRIPTION
    "Objects from the policyPrcSupportTable."
```

```
::= { policyBasePibGroups 1 }
```

```
policyDevicePibIncarnationGroup OBJECT-GROUP
```

```
OBJECTS {
    policyDevicePibIncarnationName,
    policyDevicePibIncarnationId,
    policyDevicePibIncarnationLongevity,
    policyDevicePibIncarnationTtl,
    policyDevicePibIncarnationActiveContext
}
STATUS current
DESCRIPTION
    "Objects from the policyDevicePibIncarnationTable."
```

```
::= { policyBasePibGroups 2 }
```

```
policyDeviceIdentificationGroup OBJECT-GROUP
```

```
OBJECTS {
    policyDeviceIdentificationDescr,
```

```
    policyDeviceIdentificationMaxMsg
}
```

[Page 19]

Framework Policy Information Base

March 2000

```
    STATUS current
    DESCRIPTION
        "Objects from the policyDeviceIdentificationTable."

    ::= { policyBasePibGroups 3 }

policyCompLimitsGroup OBJECT-GROUP
    OBJECTS {
        policyCompLimitsComponent,
        policyCompLimitsType,
        policyCompLimitsGuidance
    }
    STATUS current
    DESCRIPTION
        "Objects from the policyCompLimitsTable."

    ::= { policyBasePibGroups 4 }

END
```

7. Security Considerations

The information contained in a PIB when transported by the COPS protocol [[COPS-PR](#)] may be sensitive, and its function of provisioning a PEP requires that only authorized communication take place. The use of IPSEC between PDP and PEP, as described in [[COPS](#)], provides the necessary protection against these threats.

8. Intellectual Property Considerations

The IETF is being notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

9. Authors' Addresses

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

Kwok Ho Chan
Nortel Networks, Inc.
600 Technology Park Drive
Billerica, MA 01821 USA
Phone: +1 978 288 8175
Email: khchan@nortelnetworks.com

[Page 21]

Framework Policy Information Base

March 2000

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

Andrew Smith
Extreme Networks
10460 Bandley Drive
Cupertino CA 95014 USA
Phone: +1 408 342 0999
Email: andrew@extremenetworks.com

Francis Reichmeyer
IPHighway Inc.
Parker Plaza, 16th Floor
400 Kelby St.
Fort-Lee, NJ 07024
Phone: (201) 585-0800
Email: FranR@iphighway.com

10. 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-cops-pr-02.txt](#), March 2000.

[SPPI] K. McCloghrie, et.al., "Structure of Policy Provisioning Information," [draft-ietf-rap-sppi-00.txt](#), march 2000.

[POLICY] M. Stevens, W. Weiss H. Mahon, B. Moore, J. Strassner, G. Waters, A. Westerinen, J. Wheeler, "Policy Framework", [draft-ietf-policy-framework-00.txt](#), September 1999.

[RAP-FRAMEWORK] R. Yavatkar, D. Pendarakis, "A Framework for Policy-based Admission Control", [draft-ietf-rap-framework-03.txt](#), April 1999.

[Page 22]

Framework Policy Information Base

March 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.

Table of Contents

1	Glossary	2
2	Introduction	2
3	General PIB Concepts	2
3.1	Roles	2
3.1.1	An Example	3
3.2	Multiple PIB Instances	4
3.3	Reporting of Device Capabilities	5
3.4	Reporting of Device Limitations	5
4	Summary of the Framework PIB	6
5	PIB Operational Overview	7

6	The Policy Framework PIB Module	7
7	Security Considerations	21
8	Intellectual Property Considerations	21
9	Authors' Addresses	21
10	References	22