Technical Committee

Addendum To Policy Routing V1.0 for a Policy Constraint MIB

af-cs-0198.000
February 2004

The ATM Forum
Presidio of San Francisco
P.O. Box 29920 (mail)
572B Ruger Street (surface)
San Francisco, CA 94129-0920
Tel: +1-415.561.6275
Fax: +1-415.561.6120

Acknowledgements

During preparation of this document, the Control Signalling working group was chaired by Mickey Spiegel. The minutes at related working group meetings were recorded by Thomas Cornely. The editor of this addendum was Peter Roberts. The editor would like to thank the following contributors for their help with this addendum as well as all participants of the Control Signalling working group for the many days and evenings spent discussing this addendum:

Carl Rajsic
Thomas Cornely
E. Mickey Spiegel
David Ker
Navdeep Dhillon

Table of Contents

1 Introduction

[Informative]

This MIB provides a common means of defining Policy Constraints so that they can later be referenced by other applications. Specific applications are outside the scope of this specification. Example applications include Soft PVCs, SVCs, and Path Test Traces.

This MIB does not define the semantics associated with any Network Service Category or policy. These are considered to be specific to each ATM Service Provider network and are thus beyond the scope of this MIB.

This MIB does not address the configuration of NSCs to be advertised by the network element in PNNI routing.

2 Policy Constraint SNMP MIB

[Normative]

```
ATM-POLICY-CONSTRAINT-MIB DEFINITIONS ::= BEGIN

IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE, Integer32,
        enterprises
                FROM SNMPv2-SMI
        TEXTUAL-CONVENTION, RowStatus, DisplayString
                FROM SNMPv2-TC
        MODULE-COMPLIANCE, OBJECT-GROUP
                FROM SNMPv2-CONF;


atmPolicyConstraintMIB MODULE-IDENTITY
    LAST-UPDATED    "200307080000Z"
    ORGANIZATION    "The ATM Forum."
    CONTACT-INFO
        "The ATM Forum
        2570 West El Camino Real, Suite 304
        Mountain View, CA 94040-1313 USA
        Phone: +1 650-949-670
        Fax:   +1 415-949-6705
        info@atmforum.com"
    DESCRIPTION
        "The MIB module for Policy Contraints of ATM Forum
        Policy Routing.

        The Policy Constraint MIB is organized as two main tables:
        the policyConstraintTable and the policyTable.

        The policyConstraintTable provides the entries that
        can be referenced by other MIB objects to utilize
        a policy constraint.  Each entry in the table contains a set of
        up to six pointers into the policyTable.  The policyTable
        specifies the operators of a policy.  Associated with the
        policyTable are the policyNeNscTable and the policyRpNscTable.
```

```
        These two tables contain the lists of NSCs on which the
        policy operators operate.

        To create a policy, the management station should first
        create an associated instance of the row status in a
        policyEntry, using a value of policyIndex that
        is not currently in use.  The object policyNextPolicyIndex
        can be read to get an available policyIndex.
        It must also, either in the same or in successive PDUs,
        create the associated instances of the Ne-NSC and
        Rp-NSC lists for the policyIndex.  It should also
        specify the values for the policy operators.

        Once the appropriate instance of all the configuration
        objects have been created for the policyEntry,
        policyRpNscEntry, and the policyNeNscEntry (as appropriate),
        the row status of the policyEntry should be set to active
        to activate the policy.

        The policy constraint table can include pointers to
        policies that are notReady but they must exist.
        If such a policy constraint is used for a call establishment
        request, then that policy is not used in the signaled
        policy constraint.
        "
    REVISION        "200307080000Z"
    DESCRIPTION
        "Initial version of the MIB for Policy Constraints."
    ::= { atmfPolicyConstraint 1 }

atmForum        OBJECT IDENTIFIER ::= { enterprises 353 }

atmForumNetworkManagement  OBJECT IDENTIFIER ::= { atmForum 5 }

atmfSignalling   OBJECT IDENTIFIER ::= { atmForumNetworkManagement 9 }

atmfPolicyConstraint  OBJECT IDENTIFIER
                      ::= { atmfSignalling 5 }

policyConstraintMIBObjects  OBJECT IDENTIFIER
                      ::= { atmPolicyConstraintMIB 1 }



-- Textual Conventions

NetworkEntityNetworkServiceCategory ::= TEXTUAL-CONVENTION
    STATUS     current
    DESCRIPTION
        "A Network Entity Network Service Category (Ne-NSC)
        is a Network Service Category (NSC) that applies
        to the entire network entity (including all resources)
        and advertises properties of the network entity.  The
        term network entity refers to a horizontal link, an
```

```
        uplink, a node, a spoke, a bypass or a set of reachable
        ATM addresses.

        Ne-NSC identifier values within the range 65000
        through 65535, inclusive, are well known Ne-NSCs.
        The semantics of well-known Ne-NSCs are defined by
        the ATM Forum.

        The distinguished value of 65536 is used to indicate
        an invalid value and is used to remove Ne-NSC entries
        from Ne-NSC lists."
    REFERENCE
        "ATMF Policy Routing Version 1.0"
    SYNTAX     Integer32 (1..65536)

ResourcePartitionNetworkServiceCategory ::= TEXTUAL-CONVENTION
    STATUS     current
    DESCRIPTION
        "A Resource Partition NSC (Rp-NSC) is an NSC that applies
        to a resource partition of a network entity.  Note that
        association of a set of Rp-NSCs to a resource partition
        mandates that connections specify at least one of these
        Rp-NSCs as part of their associated policy in order to
        have access to resources of that partition.  Those
        resources are then used to determine whether the resource
        partition is acceptable for carrying a given connection.

        The Rp-NSC Identifier value 0 is referred to as Rp-NSC_Bare
        and identifies bare resources.  Rp-NSC identifier values
        within the range 65000 through 65535, inclusive, are well
        known Ne-NSCs.  The semantics of well-known Ne-NSCs are
        defined by the ATM Forum.
        "

    REFERENCE
        "ATMF Policy Routing Version 1.0"
    SYNTAX     Integer32 (0..65535)

PolicyConstraintIndex ::= TEXTUAL-CONVENTION
    STATUS     current
    DESCRIPTION
        "The value of this object identifies a row in the
        policyConstraintTable.  This row identifier can
        be used within other MIBs to apply a policy constraint
        to a connection establishment request.  The distinguished
        value zero signifies that no row has been identified.

        The maximum value for this index is controlled by the
        policyConstraintMaxium object.
        The distinguished value of 0 is used to indicate
        an invalid value."
    SYNTAX     Integer32 (0..65535)

PolicyConstraintPolicyIndex ::= TEXTUAL-CONVENTION
```

```
    STATUS     current
    DESCRIPTION
        "The value of this object identifies the position of
        a policy with a policy constraint.  The policies are
        applied in the order of 1 first and 6 last."
    SYNTAX     Integer32 (1..6)


PolicyIndex ::= TEXTUAL-CONVENTION
    STATUS     current
    DESCRIPTION
        "The value of this object identifies a row in the
        policyTable.  It is used within the
        policyConstraintTable to identify which policy is
        in use in the policy constraint.  The distinguished value zero
        signifies that no policy is defined."
    SYNTAX     Integer32 (0..65535)

PolicyOperator ::= TEXTUAL-CONVENTION
    STATUS     current
    DESCRIPTION
        "The value of this object identifies a row in the
        policyNeNscTable.  It is used to distinguish between the
        Ne-NSC list used for a require and the must avoid part
        of a policy."
    SYNTAX     INTEGER {
                          requires(1),
                          mustAvoid(2)
                          }


policyConstraintBaseGroup       OBJECT       IDENTIFIER       ::=       {
policyConstraintMIBObjects 1 }

policyConstraintMaximum OBJECT-TYPE
    SYNTAX     Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The maximum number of concurrent active policy constraints
        that are allowed by the agent.  A value of 0 for this
        object implies that there is no limit on the number of
        concurrent active policy constraints."
    ::= { policyConstraintBaseGroup 1 }

policyMaximum OBJECT-TYPE
    SYNTAX     Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The maximum number of concurrent active policies
        that are allowed by the agent.  A value of 0 for this
        object implies that there is no limit on the number of
        concurrent active policies."
```

```
    ::= { policyConstraintBaseGroup 2 }

policyNeNSCListMaximum OBJECT-TYPE
    SYNTAX     Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The maximum number of Ne-NSCs that can be included
        in the Ne-NSC list of a policy.  A value of 0 for this
        object implies that there is no limit."
    ::= { policyConstraintBaseGroup 3 }

policyRpNSCListMaximum OBJECT-TYPE
    SYNTAX     Integer32 (0..65535)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The maximum number of Rp-NSC entries that can be included
        in the Rp-NSC list of a policy.  A value of 0 for this
        object implies that there is no limit."
    ::= { policyConstraintBaseGroup 4 }


policyConstraintGroup OBJECT IDENTIFIER
                      ::= { policyConstraintMIBObjects 2 }

policyNextPolicyConstraintIndex OBJECT-TYPE
    SYNTAX      PolicyConstraintIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Coordinate policyConstraintIndex value allocation for
        entries in the policyConstraintTable.
        A GET of this object returns the next available
        policyConstraintIndex to be used to create an entry in the
        policyTable; or zero if no valid policyIndex value
        is available.  This object also returns a value of zero
        when it is the lexicographic successor of a varbind
        presented in an SNMP GETNEXT or GETBULK request, for which
        circumstance it is assumed that policyConstraintIndex
        allocation is unintended.

        Successive GETs will typically return different values,
        Thus avoiding collisions among cooperating management
        clients seeking to create table entries simultaneously."
    ::= { policyConstraintGroup 1 }

policyConstraintTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF PolicyConstraintEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The table whose entries describe the policy constraints
        configured in the agent."
```

```
    ::= { policyConstraintGroup 2}

policyConstraintEntry OBJECT-TYPE
    SYNTAX     PolicyConstraintEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "Each entry in this table specifies a policy constraint.
        The policy constraint consists of up to 6 policies.
        A policy constraint must contain at least one policy
        if it is to be used in another MIB object.

        The order of the policies within the policy constraint
        is important and defines the order in which the policies
        are to be applied during path selection and call
        establishment.

        If a policy is specified in the policy constraint, but that
        policy does not exist as an active row of the policyTable,
        then that policy is ignored when the policy constraint is used
        for call establishment."

    INDEX      { policyConstraintIndex,
                 policyConstraintPolicyIndex}
    ::= { policyConstraintTable 1 }

PolicyConstraintEntry ::=
    SEQUENCE {
          policyConstraintIndex          PolicyConstraintIndex,
          policyConstraintPolicyIndex    PolicyConstraintPolicyIndex,
          policyIndex                    PolicyIndex,
          policyConstraintRowStatus      RowStatus
    }

policyConstraintIndex OBJECT-TYPE
    SYNTAX     PolicyConstraintIndex
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "An arbitrary integer uniquely identifying a policy
        constraint.  Its value can be used within other managed
        objects to apply a policy constraint to the object."
    ::= { policyConstraintEntry 1 }

policyConstraintPolicyIndex OBJECT-TYPE
    SYNTAX     PolicyConstraintPolicyIndex
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "An integer uniquely identifying a policy
        within a policy constraint.  The value of this
        index defines the order in which the policies
        of the policy constraint are applied."
    ::= { policyConstraintEntry 2 }
```

```
policyIndex OBJECT-TYPE
    SYNTAX     PolicyIndex
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "The index into the policyTable for the policy
        to be used in a given position within the
        the policy constraint.  There must be an entry
        in the policyTable for this policy index
        or the set is rejected.

        The distinguished value of zero may be used to indicate
        no policy is to be used in the position."
    ::= { policyConstraintEntry 3 }

policyConstraintRowStatus OBJECT-TYPE
    SYNTAX     RowStatus
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "To create, delete, activate and de-activate a row
        of a policy constraint.
        Only those rows of the PolicyConstraintTable that have
        an active status are considered when the policyConstraintIndex
        is used for call establishment.
        "
    ::= { policyConstraintEntry 4 }


policyConstraintNameTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF PolicyConstraintNameEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The table whose entries define the names for the policy
        constraints."
    ::= { policyConstraintGroup 3}

policyConstraintNameEntry OBJECT-TYPE
    SYNTAX     PolicyConstraintNameEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "Each entry in this table specifies a name of a
        policy constraint."
    INDEX      { policyConstraintIndex }
    ::= { policyConstraintNameTable 1 }

PolicyConstraintNameEntry ::=
    SEQUENCE {
          policyConstraintName              DisplayString,
          policyConstraintNameRowStatus     RowStatus
    }
```

```
policyConstraintName OBJECT-TYPE
        SYNTAX        DisplayString
        MAX-ACCESS    read-create
        STATUS        current
        DESCRIPTION
            "The name of the Policy Constraint.  This is used
            to facilitate management of the policy constraints
            between SNMP and other management interfaces."
        DEFVAL { "" }
        ::= { policyConstraintNameEntry 1 }

policyConstraintNameRowStatus OBJECT-TYPE
        SYNTAX        RowStatus
        MAX-ACCESS    read-create
        STATUS        current
        DESCRIPTION
            "To create, delete, activate and de-activate a name
            of a policy constraint."
        ::= { policyConstraintNameEntry 2 }


policyGroup OBJECT IDENTIFIER ::= { policyConstraintMIBObjects 3 }

policyNextPolicyIndex OBJECT-TYPE
    SYNTAX      PolicyIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Coordinate policyIndex value allocation for entries in
        policyTable.
        A GET of this object returns the next available policyIndex
        to be used to create an entry in the policyTable; or zero if
        no valid policyIndex value is available.  This object also
        returns a value of zero when it is the lexicographic
        successor of a varbind presented in an SNMP GETNEXT or
        GETBULK request, for which circumstance it is assumed
        that policyIndex allocation is unintended.

        Successive GETs will typically return different values,
        Thus avoiding collisions among cooperating management
        clients seeking to create table entries simultaneously."
    ::= { policyGroup 1 }

policyTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF PolicyEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The table whose entries describe the policies
        configured.

        Each policy contains two possible policy operators:
        'require' and 'must avoid'.  The require policy
```

```
        operator can be applied with an Ne-NSC list, an Rp-NSC
        list or both.  The must avoid policy operator can
        be applied with only an Ne-NSc list.

        In order to create a new policy, the policyRowStatus
        should be set to createAndWait.  The status should
        not be set to active until the remaining objects
        of the entry have been specified.

        If an existing entry is to be modified, then the RowStatus
        should be set to notInService, the objects modified,
        and then the RowStatus set to active.

        If the RowStatus is set to indicate that the entry
        is to become active (CreateAndGo or Active), then the
        following rules are checked:
          1) At least one of requireNeNscOperator or
             requireRpNscOperator or mustAvoidNeNscOperator
             must be specified.
          2) If the requireNeNscOperator is specified,
             then at least one Ne-NSC value must
             exist in an active row of the policyNeNscTable for this
             policyIndex and an policyOperator of require.
          3) If the requireRpNscOperator is specified,
             then at least one Rp-NSC value must
             exist in an active row of the policyRpNscTable for this
             policyIndex.
          4) If the mustAvoidNeNscOperator is specified,
             then at least one Ne-NSC value must
             exist in an active row of the policyNeNscTable for this
             policyIndex and an policyOperator
             of mustAvoid.
        If a row is deleted from this table, then the corresponding
        rows of the policyNeNscTable and policyRpNscTable are also
        deleted.  In addition, any entries in the policyConstraintTable
        that reference this policyIndex are removed.  "

    ::= { policyGroup 2}

policyEntry OBJECT-TYPE
    SYNTAX     PolicyEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "An entry representing a policy."
    INDEX      { policyIndex }
    ::= { policyTable 1 }

PolicyEntry ::=
    SEQUENCE {
          policyName                    DisplayString,
          requireNeNscOperator          INTEGER,
          requireRpNscOperator          INTEGER,
          mustAvoidNeNscOperator        INTEGER,
```

```
          policyRowStatus                RowStatus
    }

policyName OBJECT-TYPE
    SYNTAX        DisplayString
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "A textual string describing the policy.  This is used
        to facilitate management of the policies
        between SNMP and other management interfaces."
    DEFVAL { "" }
    ::= { policyEntry 1 }

requireNeNscOperator OBJECT-TYPE
    SYNTAX     INTEGER {
                        noop(1),
                        logicalAND(2),
                        logicalOR(3)
                       }
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "An integer identifying the NSC List operator
        for the Ne-NSC list of the require policy operator.
        The policy operator singleNeNsc is assumed if only one Ne-NSC
        is specified in the associated Ne-NSC list. "
    DEFVAL  { noop }
     ::= { policyEntry 2 }

requireRpNscOperator OBJECT-TYPE
    SYNTAX     INTEGER {
                        noop(1),
                        logicalAND(2),
                        logicalOR(3)
                       }
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "An integer identifying the NSC List operator
        for the Rp-NSC list of the require policy operator.
        The policy operator singleRpNsc is assumed if only one Rp-NSC
        is specified in the associated Ne-NSC list. "
    DEFVAL  { noop }
     ::= { policyEntry 3 }

mustAvoidNeNscOperator OBJECT-TYPE
    SYNTAX     INTEGER {
                        noop(1),
                        logicalAND(2),
                        logicalOR(3)
                       }
    MAX-ACCESS read-create
    STATUS     current
```

```
    DESCRIPTION
        "An integer identifying the NSC List operator
        for the Ne-NSC list of the must avoid policy operator.
        The policy operator singleNeNsc is assumed if only one Ne-NSC
        is specified in the associated Ne-NSC list. "
    DEFVAL  { noop }
     ::= { policyEntry 4 }

policyRowStatus OBJECT-TYPE
    SYNTAX     RowStatus
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "Used to create and delete entries in this table.
        When a new entry is being created or an existing
        entry is being modified, then the RowStatus
        should be set to createAndWait or notInService.  Once
        the objects for this row have been set, then the RowStatus
        should be set to active.  When a row is active, it can be
        used in a policy constraint to effect the establishment
        of a call.  If a policy is used in a policy constraint

        while it is not active, then it shall be ignored
        during call establishment."

    ::= { policyEntry 5 }


policyNeNscTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF PolicyNeNscEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The table whose entries describe the NeNSCs of the Ne-NSC lists
        of a policy."
    ::= { policyGroup 3}

policyNeNscEntry OBJECT-TYPE
    SYNTAX     PolicyNeNscEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "An entry representing the Ne-NSC list for a policy."
    INDEX      { policyIndex,
                 policyOperator,
                 policyNeNscIndex }
    ::= { policyNeNscTable 1 }

PolicyNeNscEntry ::=
    SEQUENCE {
          policyNeNscIndex              Integer32,
          policyOperator                PolicyOperator,
          policyNeNsc      NetworkEntityNetworkServiceCategory,
          policyNeNscRowStatus          RowStatus
```

```
    }

policyNeNscIndex OBJECT-TYPE
    SYNTAX     Integer32 (1..65535)
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "An integer identifying the NeNSC within the Ne_NSC list
        of a policy."
    ::= { policyNeNscEntry 1 }

policyOperator OBJECT-TYPE
    SYNTAX     PolicyOperator
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "An integer identifying whether the Ne-NSC list
        is part of the require or must avoid operator
        of a policy."
    ::= { policyNeNscEntry 2 }


policyNeNsc OBJECT-TYPE
    SYNTAX     NetworkEntityNetworkServiceCategory
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "One of the Ne-NSCs of the Ne-NSC list of the policy.
        The policyNeNscOperator object defines how the list
        is to be used by the policy.
        Setting this object with a value of 65536 is equivalent
        to deleting the object.  Deleting the object will fail
        if there are no other policyNeNsc objects with the same
        policyIndex and policyOperator and the policyRowStatus
        object for the policyIndex has the value active"
    ::= { policyNeNscEntry 3 }

policyNeNscRowStatus OBJECT-TYPE
    SYNTAX     RowStatus
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "Used to create and delete entries in this table.
        When a new entry is being created or an existing
        entry is being modified, then the RowStatus
        should be set to createAndWait or notInService.  Once
        the objects for this row have been set, then the RowStatus
        should be set to active.  When a row is active, it can be
        used in a policy to effect the establishment
        of a call.  If a policyNeNscEntry is used in a policy
        while it is not active, then it shall be ignored
        during the application of the policy."

    ::= { policyNeNscEntry 4 }
```

```
policyRpNscTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF PolicyRpNscEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The table whose entries describe the RpNSCs of the Rp-NSC list
        of a policy."
    ::= { policyGroup 4}

policyRpNscEntry OBJECT-TYPE
    SYNTAX     PolicyRpNscEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "An entry representing the Ne-NSC list for a policy."
    INDEX      { policyIndex,
                 policyRpNscIndex }
    ::= { policyRpNscTable 1 }

PolicyRpNscEntry ::=
    SEQUENCE {
          policyRpNscIndex               Integer32,
          policyRpNsc      ResourcePartitionNetworkServiceCategory,
          policyRpNscRowStatus           RowStatus
    }

policyRpNscIndex OBJECT-TYPE
    SYNTAX     Integer32 (1..65535)
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "An integer identifying the RpNSC within the Rp_NSC list
        of a policy."
    ::= { policyRpNscEntry 1 }


policyRpNsc OBJECT-TYPE
    SYNTAX     ResourcePartitionNetworkServiceCategory
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "One of the Rp-NSCs of the Rp-NSC list of the policy.
        The policyRpNscOperator object defines how the list
        is to be used by the policy.
        Setting this object with a value of 65536 is equivalent
        to deleting the object.  Deleting the object will fail
        if there are no other policyRpNsc objects with the same
        policyIndex and the policyRowStatus
        object for the policyIndex has the value active"
    ::= { policyRpNscEntry 2 }

policyRpNscRowStatus OBJECT-TYPE
```

```
    SYNTAX     RowStatus
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "Used to create and delete entries in this table.
        When a new entry is being created or an existing
        entry is being modified, then the RowStatus
        should be set to createAndWait or notInService.  Once
        the objects for this row have been set, then the RowStatus
        should be set to active.  When a row is active, it can be
        used in a policy to effect the establishment
        of a call.  If a policyRpNSC Entry is used in a policy
        while it is not active, then it shall be ignored
        during the application of that policy"

    ::= { policyRpNscEntry 3 }


policyReferenceTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF PolicyReferenceEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "This table provides pointers to entries in the
        policyConstraintTable that reference the policyIndex.

        This is provided to facilitate management of the
        policies and policy constraints.
        "

    ::= { policyGroup 5}

policyReferenceEntry OBJECT-TYPE
    SYNTAX     PolicyReferenceEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "An entry for the policy reference table.
        "
    INDEX      { policyIndex,
                 policyConstraintIndex }
    ::= { policyReferenceTable 1 }

PolicyReferenceEntry ::=
    SEQUENCE {
          policyReferencePCIndex         PolicyConstraintIndex
    }

policyReferencePCIndex OBJECT-TYPE
    SYNTAX     PolicyConstraintIndex
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "If any of the entries of the PolicyConstraintTable
```

```
        for the specified policyConstraintIndex has a value
        for its policyIndex that matches the policyIndex
        of this entry, then this returns the
        policyConstraintIndex; otherwise, the value zero is
        returned.

        This object should be walked using GETNEXT
        and specifying an initial value of zero for the
        policyConstraintIndex.  If a value of zero is returned,
        then there are no more matching entries for the policyIndex.
        "
    ::= { policyReferenceEntry 1 }



-- conformance information

policyConstraintMIBConformance
        OBJECT IDENTIFIER ::= { policyConstraintMIBObjects 4 }

policyConstraintMIBCompliances
        OBJECT IDENTIFIER ::= { policyConstraintMIBConformance 1 }

policyConstraintMIBGroups
        OBJECT IDENTIFIER ::= { policyConstraintMIBConformance 2 }

-- compliance statements

policyConstraintMIBCompliance MODULE-COMPLIANCE
    STATUS     current
    DESCRIPTION
        "The compliance statement for entities which implement the
        Policy Routing Addendum for Policy Constraint MIB.

        Groups of objects required to support certain functionality
        are identified by the suffix MandatoryGroup.

        Groups of optional objects are identified by the suffix
        OptionalGroup."
    MODULE     -- this module
    MANDATORY-GROUPS
        { policyConstraintMIBMandatoryGroup
          }

    GROUP atmTraceConnAndPathFilterMandatoryGroup2
    DESCRIPTION
        "Required if connection trace or path trace using
        filtering of new connection and party establishment messages
        is supported."

    ::= { policyConstraintMIBCompliances 1 }

-- units of conformance
```

```
policyConstraintMIBMandatoryGroup  OBJECT-GROUP
    OBJECTS {
              policyConstraintMaximum,
              policyMaximum,
              policyNeNSCListMaximum,
              policyRpNSCListMaximum,
              policyIndex,
              policyConstraintRowStatus,
              requireNeNscOperator,
              requireRpNscOperator,
              mustAvoidNeNscOperator,
              policyRowStatus,
              policyNeNsc,
              policyNeNscRowStatus,
              policyRpNsc,
              policyRpNscRowStatus
              }
    STATUS    current
    DESCRIPTION
        "A collection of objects required when policy constraint
        specification is supported."
    ::= { policyConstraintMIBGroups 1 }

policyConstraintMIBOptionalGroup  OBJECT-GROUP
    OBJECTS {
              policyNextPolicyConstraintIndex,
              policyConstraintName,
              policyConstraintNameRowStatus,
              policyNextPolicyIndex,
              policyName,
              policyReferencePCIndex
              }
    STATUS    current
    DESCRIPTION
        "A collection of optional objects used for path and connection
        trace."
    ::= { policyConstraintMIBGroups 2 }

END
```