                     Definitions of Managed Objects
                            for APPN TRAPS

Status of this Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

   This memo defines a portion of the Management Information Base (MIB)
   for use with network management protocols in the Internet community.
   In particular, it defines objects for receiving notifications from
   network devices with APPN (Advanced Peer-to-Peer Network) and DLUR
   (Dependent LU Requester) capabilities. This memo identifies
   notifications for the APPN and DLUR architecture.

Table of Contents

1. Introduction

   This document is a product of the SNA NAU Services MIB Working Group.
   It defines a MIB module for notifications for devices with Advanced
   Peer-to-Peer Networking (APPN) and Dependent LU Requester (DLUR)
   capabilities.

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC 2119 [13].

2. The SNMP Network Management Framework

   The SNMP Management Framework presently consists of five major
   components:

   o   An overall architecture, described in RFC 2271 [1].

   o   Mechanisms for describing and naming objects and events for the
         purpose of management. The first version of this Structure of
         Management Information (SMI) is called SMIv1 and described in
         STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The
         second version, called SMIv2, is described in RFC 1902 [5], RFC
         1903 [6] and RFC 1904 [7].

   o   Message protocols for transferring management information. The
         first version of the SNMP message protocol is called SNMPv1 and
         described in STD 15, RFC 1157 [8]. A second version of the SNMP
         message protocol, which is not an Internet standards track
         protocol, is called SNMPv2c and described in RFC 1901 [9] and
         RFC 1906 [10]. The third version of the message protocol is
         called SNMPv3 and described in RFC 1906 [10], RFC 2272 [11] and
         RFC 2274 [12].

   o   Protocol operations for accessing management information. The
         first set of protocol operations and associated PDU formats is
         described in STD 15, RFC 1157 [8]. A second set of protocol
         operations and associated PDU formats is described in RFC 1905
         [13].

   o   A set of fundamental applications described in RFC 2273 [14] and
         the view-based access control mechanism described in RFC 2275
         [15].

   Managed objects are accessed via a virtual information store, termed
   the Management Information Base or MIB. Objects in the MIB are
   defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A
MIB conforming to the SMIv1 can be produced through the appropriate
translations. The resulting translated MIB must be semantically
equivalent, except where objects or events are omitted because no
translation is possible (use of Counter64). Some machine readable
information in SMIv2 will be converted into textual descriptions in
SMIv1 during the translation process. However, this loss of machine
readable information is not considered to change the semantics of the
MIB.

3. Overview

This document identifies the set of objects for reporting the status
of devices with APPN and DLUR capabilities via notifications.

See the SNANAU APPN MIB [18] and SNANAU DLUR MIB [19] for the objects
for monitoring the configuration and active characteristics of the
devices with APPN and DLUR capabilities. Many objects contained in
the notifications of this MIB are imported from the APPN and DLUR
MIBs. Implementors of this MIB must also implement the APPN MIB.
Implementations that support the appnTrapMibDlurConfGroup and the
appnTrapMibDlurNotifGroup must also implement the DLUR MIB.

The SNANAU APPN MIB allows a management station to collect the
network topology of an APPN network (the network nodes (NNs) in the
network and all of transmission groups (TGs) between the network
nodes) from an APPN device. It also allows the management station to
collect the local topology (TGs to end stations, and locally defined
ports and link stations) from an APPN device. While the SNANAU APPN
MIB has an efficient way to poll the APPN device for updates to the
network topology, using flow reduction sequence numbers (FRSNs) as a
table index; it does not have a mechanism to poll the local topology
tables (appnLocalTgTable, appnPortTable, and appnLsTable) for status
changes.

This MIB provides a mechanism for an APPN device to send
notifications to inform the management station of status changes to
rows of these tables. Status changes include operational state
changes, and for TGs also include control-point to control-point
(CP-CP) session state changes. A notification is defined for each
type of status change for each table.

The port and link operational state objects have intermediate states.
Notifications are only sent for transition to active or inactive
state.

Notifications are only sent for row creation if the state is active
or operational. This is done to avoid sending a notification as the
row is created with an inactive initial state, followed by another
notification as the resource is activated.

Notifications are only sent for row deletion if the last state was
active or operational. In most cases, a resource must be deactivated
before it can be deleted, and the deactivation will cause a
notification to be sent. There is no need for a second notification
to be sent for the row deletion, except for the case where the
deletion occurred without deactivation. In this case, the state of
the object in the notification will indicate an inactve state, since
a deleted resource can no longer be active.

The purpose of the appnLocalTgCpCpStateChangeTrap notification is to
identify the loss or recovery of CP-CP sessions on a TG while the TG
remains operational. Thus this notification is only sent if there is
a change to an appnLocalTgCpCpSession object, but not a change to the
appnLocalTgOperational object. This notification is never sent for
the creation or deletion of a row in the appnLocalTgTable.

Each notification always contains an object which is a count of the
number of times the status of a row in table has changed since the
APPN node was last reinitialized. This enables a management station
to detect that it has missed a notification, if it does not get the
notifications in numerical sequence. If the notifications are not in
sequence, the management station should retrieve the entire table to
get the correct status for all rows.

Similarly, the SNANAU DLUR MIB provides a mechanism for retrieving
the configuration and status of dependent LU server (DLUS) sessions
on a device with DLUR capabilities. This MIB defines a notification
for a DLUR-DLUS session state change of a row in the dlurDlusTable,
in the manner described above. A notification is only sent for a
session state transition to active or inactive. As with the above
notifications, it is only sent on row creation if the initial state
is active; and on row deletion is the last state was active, in which
case the notification indicates that the state is now inactive.

The SNANAU APPN MIB also provides a mechanism for a management
station to collect traffic statistics on intermediate sessions,
primarily for accounting purposes. However, when the session is
terminated, all statistics from the last poll until the session
termination time are lost, since the row for that session is deleted
from the appnIsInTable. This MIB defines a notification so that the
session's final statistics can be sent to a management station. If
the notification is not delivered, the final session statistics are
lost. If this is a concern, polling of the appnIsInTable in the APPN

MIB should be increased to more likely reduce the time between the
last poll and the session termination, thereby reducing the amount of
data lost.

Highlights of the management functions supported by the APPN TRAP MIB
module include the following:

o   A notification for an APPN local TG operational state change.

o   A notification for an APPN local TG CP-CP session state change.

o   A notification for an APPN port operational state change.

o   A notification for an APPN link station operational state
       change.

o   A notification for a DLUR-DLUS session state change.

o   A notification for reporting final APPN intermediate session
       statistics.

This MIB module does not support:

o   Objects to query the configuration or status of APPN nodes on
       demand.

o   Notifications for changes to local topology tables not related
       to status.

3.1. APPN TRAP MIB Structure

The APPN TRAP MIB module contains a group of notifications, and a
group of supporting objects.

The group of notifications consists of the following notifications:

1) appnIsrAccountingDataTrap

This notification is generated by an APPN device when an intermediate
session is terminating, to report the final accounting statistics of
the session.

2) appnLocalTgOperStateChangeTrap

This notification identifies a change to the appnLocalTgOperational
object in a row of the SNANAU APPN MIB appnLocalTgTable.

3) appnLocalTgCpCpStateChangeTrap

This notification identifies a change to the appnLocalTgCpCpSession
object in a row of the SNANAU APPN MIB appnLocalTgTable.

4) appnPortOperStateChangeTrap

This notification identifies a change to the appnPortOperState object
in a row of the SNANAU APPN MIB appnPortTable.

5) appnLsOperStateChangeTrap

This notification identifies a change to the appnLsOperState object
in a row of the SNANAU APPN MIB appnLsTable.

6) dlurDlusStateChangeTrap

This notification identifies a change to the dlurDlusSessnStatus
object in a row of the SNANAU DLUR MIB dlurDlusTable.

The group of supporting objects contains the appnTrapControl object,
which controls whether the APPN device generates each type of
notification. Note that generation of the appnIsrAccountingDataTrap
is not controlled by this object; instead it is controlled by the
appnIsInGlobalCtrAdminStatus object in the SNANAU APPN MIB.

Although APPN notification generation could be controlled solely by
entries in the snmpNotificationMIB, RFC 2273 [9], the appnTrapControl
object exists in this MIB so that implementations are not required to
implement RFC 2273 to control generation of APPN notifications. For
a notification to be generated and sent as a TRAP or INFORM, the
notification type must first be enabled by the appnTrapControl
object. It must also not be disabled by an snmpNotificationMIB
entry. The destination of notifications is not within the scope of
this MIB.

Also contained in this group are objects for the TG, port, link, and
DLUR-DLUS session notifications to indicate the number of times each
of the tables has had a status change of a row since the APPN node
was last reinitialized.

4. Definitions

APPN-TRAP-MIB DEFINITIONS ::= BEGIN

IMPORTS

        Counter32, OBJECT-TYPE, MODULE-IDENTITY,

```
          NOTIFICATION-TYPE
                  FROM SNMPv2-SMI

          MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
                  FROM SNMPv2-CONF

          appnMIB, appnIsInP2SFmdPius, appnIsInS2PFmdPius,
          appnIsInP2SNonFmdPius, appnIsInS2PNonFmdPius,
          appnIsInP2SFmdBytes, appnIsInS2PFmdBytes,
          appnIsInP2SNonFmdBytes, appnIsInS2PNonFmdBytes,
          appnIsInSessUpTime, appnObjects,
          appnLocalTgOperational, appnLocalTgCpCpSession,
          appnPortOperState, appnLsOperState,
          appnCompliances, appnGroups
                  FROM APPN-MIB

          dlurDlusSessnStatus
                  FROM APPN-DLUR-MIB;


appnTrapMIB MODULE-IDENTITY
        LAST-UPDATED  "9808310000Z" -- August 31, 1998
        ORGANIZATION  "IETF SNA NAU MIB WG / AIW APPN MIBs SIG"
        CONTACT-INFO

                  "
                          Bob Clouston
                          Cisco Systems
                          7025 Kit Creek Road
                          P.O. Box 14987
                          Research Triangle Park, NC 27709, USA
                          Tel:    1 919 472 2333
                          E-mail: clouston@cisco.com

                          Bob Moore
                          IBM Corporation
                          4205 S. Miami Boulevard
                          BRQA/501
                          P.O. Box 12195
                          Research Triangle Park, NC 27709, USA
                          Tel:    1 919 254 4436
                          E-mail: remoore@us.ibm.com
                  "
        DESCRIPTION
                  "This MIB module defines notifications to be generated by
                  network devices with APPN capabilities.  It presupposes
                  support for the APPN MIB.  It also presupposes
                  support for the DLUR MIB for implementations
                  that support the DLUR-related groups."
```

    ::= { appnMIB 0 }

    -- **
    -- Notifications
    -- **

    appnIsrAccountingDataTrap NOTIFICATION-TYPE
         OBJECTS   {
                     appnIsInP2SFmdPius,
                     appnIsInS2PFmdPius,
                     appnIsInP2SNonFmdPius,
                     appnIsInS2PNonFmdPius,
                     appnIsInP2SFmdBytes,
                     appnIsInS2PFmdBytes,
                     appnIsInP2SNonFmdBytes,
                     appnIsInS2PNonFmdBytes,
                     appnIsInSessUpTime
                    }
         STATUS current
         DESCRIPTION
             "When it has been enabled, this notification is generated by an
             APPN node whenever an ISR session passing through the node is
             taken down, regardless of whether the session went down
             normally or abnormally. Its purpose is to allow a management
             application (primarily an accounting application) that is
             monitoring the ISR counts to receive the final values of these
             counts, so that the application can properly account for the
             amounts the counts were incremented since the last time the
             application polled them. The appnIsInSessUpTime object
             provides the total amount of time that the session was active.

             This notification is not a substitute for polling the ISR
             counts. In particular, the count values reported in this
             notification cannot be assumed to be the complete totals for
             the life of the session, since they may have wrapped while the
             session was up.

             The session to which the objects in this notification apply is
             identified by the fully qualified CP name and PCID that make up
             the table index. An instance of this notification will contain
             exactly one instance of each of its objects, and these objects
             will all belong to the same conceptual row of the
             appnIsInTable.

             Generation of this notification is controlled by the same
             object in the APPN MIB, appnIsInGlobeCtrAdminStatus, that
             controls whether the count objects themselves are being
             incremented."

```
        ::= { appnTrapMIB 1 }

appnLocalTgOperStateChangeTrap NOTIFICATION-TYPE
      OBJECTS  {
                 appnLocalTgTableChanges,
                 appnLocalTgOperational
                }
      STATUS current
      DESCRIPTION
          "When it has been enabled, this notification makes it possible
          for an APPN topology application to get asynchronous
          notifications of local TG operational state changes,
          and thus to reduce the frequency with which it polls
          for these changes.

          This notification is sent whenever there is a change to
          the appnLocalTgOperational object in a row of the
          appnLocalTgTable.  This notification is only sent for row
          creation if the row is created with a value of 'true' for
          appnLocalTgOperational.  This notification is only sent for
          row deletion if the last value of appnLocalTgOperational was
          'true'.  In this case, the value of appnLocalTgOperational
          in the notification shall be 'false', since the deletion of
          a row indicates that the TG is no longer operational.

          The notification is more than a simple 'poll me now' indication.
          It carries both a count of local TG topology changes, and the
          current operational state itself.  The count of changes allows an
          application to detect lost notifications, either when polling
          or upon receiving a subsequent notification, at which point it
          knows it must retrieve the entire appnLocalTgTable again.
          This is the same count as used in the appnLocalCpCpStateChangeTrap
.
          A lost notification could indicate a local TG CP-CP session state
          change or an operational state change.

          Generation of this notification is controlled by the
          appnTrapControl object."

        ::= { appnTrapMIB 2 }

appnLocalTgCpCpChangeTrap NOTIFICATION-TYPE
      OBJECTS  {
                 appnLocalTgTableChanges,
                 appnLocalTgCpCpSession
                }
      STATUS current
      DESCRIPTION
          "When it has been enabled, this notification makes it possible
```

          for an APPN topology application to get asynchronous
          notifications of local TG control-point to control-point (CP-CP)
          session state changes, and thus to reduce the
          frequency with which it polls for these changes.

          This notification is sent whenever there is a change to
          the appnLocalTgCpCpSession object but NOT the
          appnLocalTgOperational object in a row of the appnLocalTgTable.
          This notification is never sent for appnLocalTgTable row
          creation or deletion.

          The notification is more than a simple 'poll me now' indication.
          It carries both a count of local TG topology changes, and the
          current CP-CP session state itself. The count of changes allows
          an application to detect lost notifications, either when polling
          or upon receiving a subsequent notification, at which point it
          knows it must retrieve the entire appnLocalTgTable again. This
          is the same count as used in the appnLocalTgOperStateChangeTrap.
          A lost notification could indicate a local TG CP-CP session
          state change or an operational state change.

          Generation of this notification is controlled by the
          appnTrapControl object."

     ::= { appnTrapMIB 3 }

appnPortOperStateChangeTrap NOTIFICATION-TYPE
     OBJECTS   {
                   appnPortTableChanges,
                   appnPortOperState
               }

     STATUS current
     DESCRIPTION
          "When it has been enabled, this notification makes it possible
          for an APPN topology application to get asynchronous
          notifications of port operational state changes, and thus to
          reduce the frequency with which it polls for these changes.
          This notification is only sent when a appnPortOperState has
          transitioned to a value of 'active' or 'inactive'.

          This notification is sent whenever there is a appnPortOperState
          object transition to 'inactive' or 'active' state in the
          appnPortTable. This notification is only sent for row creation
          if the row is created with a value of 'active' for
          appnPortOperState. This notification is only sent for
          row deletion if the last value of appnPortOperState was
          'active'. In this case, the value of appnPortOperState

in the notification shall be 'inactive', since the deletion of
a row indicates that the port is no longer active.

The notification is more than a simple 'poll me now' indication.
It carries both a count of port table changes, and the
operational state itself. The count of changes allows an
application to detect lost notifications, either when polling
or upon receiving a subsequent notification, at which point
it knows it must retrieve the entire appnPortTable again.

Generation of this notification is controlled by the
appnTrapControl object."

        ::= { appnTrapMIB 4 }

appnLsOperStateChangeTrap NOTIFICATION-TYPE
        OBJECTS {
                    appnLsTableChanges,
                    appnLsOperState
                 }
        STATUS current
        DESCRIPTION
            "When it has been enabled, this notification makes it possible
            for an APPN topology application to get asynchronous
            notifications of link station operational state changes, and
            thus to reduce the frequency with which it polls for these
            changes. This notification is only sent when a appnLsOperState
            has transitioned to a value of 'active' or 'inactive'.

            This notification is sent whenever there is a appnLsOperState
            object transition to 'inactive' or 'active' state in the
            appnLsTable. This notification is only sent for row creation
            if the row is created with a value of 'active' for
            appnLsOperState. This notification is only sent for
            row deletion if the last value of appnLsOperState was
            'active'. In this case, the value of appnLsOperState
            in the notification shall be 'inactive', since the deletion of
            a row indicates that the link station is no longer active.

            The notification is more than a simple 'poll me now' indication.
            It carries both a count of link station table changes, and the
            operational state itself. The count of changes allows an
            application to detect lost notifications, either when polling
            or upon receiving a subsequent notification, at which point it
            knows it must retrieve the entire appnLsTable again.

            Generation of this notification is controlled by the
            appnTrapControl object."

```
     ::= { appnTrapMIB 5 }

dlurDlusStateChangeTrap NOTIFICATION-TYPE
     OBJECTS  {
               dlurDlusTableChanges,
               dlurDlusSessnStatus
             }
     STATUS current
     DESCRIPTION
         "When it has been enabled, this notification makes it possible
         for an APPN topology application to get asynchronous
         notifications of DLUR-DLUS session changes, and thus to reduce
         the frequency with which it polls for these changes.

         This notification is sent whenever there is a dlurDlusSessnStatus
         object transition to 'inactive' or 'active' state in the
         dlurDlusTable.  This notification is only sent for row creation
         if the row is created with a value of 'active' for
         dlurDlusSessnStatus.  This notification is only sent for
         row deletion if the last value of dlurDlusSessnStatus was
         'active'.  In this case, the value of dlurDlusSessnStatus
         in the notification shall be 'inactive', since the deletion of
         a row indicates that the session is no longer active.

         The notification is more than a simple 'poll me now' indication.
         It carries both a count of DLUR-DLUS table changes, and the
         session status itself.  The count of changes allows an
         application to detect lost notifications, either when polling
         or upon receiving a subsequent notification, at which point it
         knows it must retrieve the entire dlurDlusTable again.

         Generation of this notification is controlled by the
         appnTrapControl object."

     ::= { appnTrapMIB 6 }

-- ****************************************************************
-- Supporting Objects
-- ****************************************************************

appnTrapObjects                OBJECT IDENTIFIER ::= { appnObjects 7 }

appnTrapControl OBJECT-TYPE

     SYNTAX BITS {
               appnLocalTgOperStateChangeTrap(0),
               appnLocalTgCpCpChangeTrap(1),
               appnPortOperStateChangeTrap(2),
```

```
                        appnLsOperStateChangeTrap(3),
                        dlurDlusStateChangeTrap(4)
                        -- add other notification types here
                       }
     MAX-ACCESS read-write
     STATUS current
     DESCRIPTION
          "An object to turn APPN notification generation on and off.
          Setting a notification type's bit to 1 enables generation of
          notifications of that type, subject to further filtering
          resulting from entries in the snmpNotificationMIB.  Setting
          this bit to 0 disables generation of notifications of that
          type.

          Note that generation of the appnIsrAccountingDataTrap is
          controlled by the appnIsInGlobeCtrAdminStatus object in
          the APPN MIB:  if counts of intermediate session traffic
          are being kept at all, then the notification is also enabled."

          ::= { appnTrapObjects 1 }


appnLocalTgTableChanges OBJECT-TYPE
     SYNTAX Counter32
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
          "A count of the number of times a row in the appnLocalTgTable
          has changed status since the APPN node was last reinitialized.
          This counter is incremented whenever a condition is detected
          that would cause a appnLocalTgOperStateChangeTrap or
          appnLocalTgCpCpChangeTrap notification to be sent, whether
          or not those notifications are enabled."

          ::= { appnTrapObjects 2 }


appnPortTableChanges OBJECT-TYPE
     SYNTAX Counter32
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
          "A count of the number of times a row in the appnPortTable
          has changed status since the APPN node was last reinitialized.
          This counter is incremented whenever a condition is detected
          that would cause a appnPortOperStateChangeTrap notification
          to be sent, whether or not this notification is enabled."

          ::= { appnTrapObjects 3 }
```

appnLsTableChanges OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
          "A count of the number of times a row in the appnLsTable
          has changed status since the APPN node was last reinitialized.
          This counter is incremented whenever a condition is detected
          that would cause a appnLsOperStateChangeTrap notification
          to be sent, whether or not this notification is enabled."

      ::= { appnTrapObjects 4 }

dlurDlusTableChanges OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
          "A count of the number of times a row in the dlurDlusTable
          has changed status since the APPN node was last reinitialized.
          This counter is incremented whenever a condition is detected
          that would cause a dlurDlusStateChangeTrap notification
          to be sent, whether or not this notification is enabled."

      ::= { appnTrapObjects 5 }

-- **
-- Conformance information
-- **

-- Tie into the conformance structure in the APPN MIB:
-- appnConformance   OBJECT IDENTIFIER ::= {appnMIB 3 }
--
-- appnCompliances   OBJECT IDENTIFIER ::= {appnConformance 1 }
-- appnGroups   OBJECT IDENTIFIER ::= {appnConformance 2 }

-- Compliance statement
appnTrapMibCompliance MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION
          "The compliance statement for the SNMP entities that
          implement the APPN-TRAP-MIB."

      MODULE -- this module

--   Conditionally mandatory groups
          GROUP appnTrapMibIsrNotifGroup
          DESCRIPTION

```
                "This group is mandatory for APPN nodes supporting
                reporting of final ISR counter values via notifications."

           GROUP appnTrapMibTopoConfGroup
           DESCRIPTION
               "This group is mandatory for APPN nodes supporting
               polling reduction for local topology."

           GROUP appnTrapMibTopoNotifGroup
           DESCRIPTION
               "This group is mandatory for APPN nodes supporting
               polling reduction for local topology."

           GROUP appnTrapMibDlurConfGroup
           DESCRIPTION
               "This group is mandatory for APPN nodes supporting
               polling reduction for the dlurDlusTable."

           GROUP appnTrapMibDlurNotifGroup
           DESCRIPTION
               "This group is mandatory for APPN nodes supporting
               polling reduction for the dlurDlusTable."

           OBJECT appnTrapControl
               MIN-ACCESS  read-only
               DESCRIPTION
                   "An agent is not required to support a set to
                   this object."

       ::= {appnCompliances 2 }

-- Units of conformance
appnTrapMibIsrNotifGroup    NOTIFICATION-GROUP
       NOTIFICATIONS {
                       appnIsrAccountingDataTrap
                     }
       STATUS  current
       DESCRIPTION
           "A notification for reporting the final values of the
           APPN MIB's ISR counters."

       ::= { appnGroups 21 }

appnTrapMibTopoConfGroup  OBJECT-GROUP
       OBJECTS  {
               appnTrapControl,
               appnLocalTgTableChanges,
               appnPortTableChanges,
```

```
                  appnLsTableChanges
                }
        STATUS  current
        DESCRIPTION
            "A collection of objects for reducing the polling
            associated with the local topology tables in the
            APPN MIB.  Nodes that implement this group SHALL
            also implement the appnTrapMibTopoNotifGroup."

        ::= { appnGroups 22 }

appnTrapMibTopoNotifGroup    NOTIFICATION-GROUP
        NOTIFICATIONS {
                        appnLocalTgOperStateChangeTrap,
                        appnLocalTgCpCpChangeTrap,
                        appnPortOperStateChangeTrap,
                        appnLsOperStateChangeTrap

                        }
        STATUS  current
        DESCRIPTION
            "A collection of notifications for reducing the polling
            associated with the local topology tables in the
            APPN MIB.  Nodes that implement this group SHALL
            also implement the appnTrapMibTopoConfGroup."

        ::= { appnGroups 23 }

appnTrapMibDlurConfGroup  OBJECT-GROUP
        OBJECTS  {
                 appnTrapControl,
                 dlurDlusTableChanges
                 }
        STATUS  current
        DESCRIPTION
            "A collection of objects for reducing the polling
            associated with the dlurDlusTable in the DLUR
            MIB.  Nodes that implement this group SHALL also
            implement the appnTrapMibDlurNotifGroup."

        ::= { appnGroups 24 }

appnTrapMibDlurNotifGroup    NOTIFICATION-GROUP
        NOTIFICATIONS {
                        dlurDlusStateChangeTrap
                        }
        STATUS  current
        DESCRIPTION
```

               "A notification for reducing the polling associated
               with the dlurDlusTable in the DLUR MIB. Nodes that
               implement this group SHALL also implement the
               appnTrapMibDlurConfGroup."

          ::= { appnGroups 25 }

END

5. Security Considerations

     Certain management information defined in this MIB may be considered
     sensitive in some network environments. Therefore, authentication of
     received SNMP requests and controlled access to management
     information SHOULD be employed in such environments. An
     authentication protocol is defined in [12]. A protocol for access
     control is defined in [15].

     None of the read-only objects in the APPN TRAP MIB reports a
     password, user data, or anything else that is particularly sensitive.
     Some enterprises view their network configuration itself, as well as
     information about network usage and performance, as corporate assets;
     such enterprises may wish to restrict SNMP access to most of the
     objects in the MIB.

     There is one read-write object in the APPN TRAP MIB, appnTrapControl.
     This object controls the generation of the notifications defined in
     the APPN TRAP MIB.

6. Intellectual Property

     The IETF takes no position regarding the validity or scope of any
     intellectual property or other rights that might be claimed to
     pertain to the implementation or use of the technology described in
     this document or the extent to which any license under such rights
     might or might not be available; neither does it represent that it
     has made any effort to identify any such rights. Information on the
     IETF's procedures with respect to rights in standards-track and
     standards- related documentation can be found in BCP-11 [16]. Copies
     of claims of rights made available for publication and any assurances
     of licenses to be made available, or the result of an attempt made to
     obtain a general license or permission for the use of such
     proprietary rights by implementers or users of this specification can
     be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any
copyrights, patents or patent applications, or other proprietary
rights which may cover technology that may be required to practice
this standard. Please address the information to the IETF Executive
Director.

7. Acknowledgments

This MIB module is the product of the IETF SNA NAU MIB WG and the AIW
APPN/HPR MIBs SIG.

8. References

[1] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for
     Describing SNMP Management Frameworks", RFC 2271, Cabletron
     Systems, Inc., BMC Software, Inc., IBM T. J. Watson Research,
     January 1998.

[2] Rose, M., and K. McCloghrie, "Structure and Identification of
     Management Information for TCP/IP-based Internets", STD 16, RFC
     1155, May 1990.

[3] Rose, M., and K. McCloghrie, "Concise MIB Definitions", STD 16,
     RFC 1212, March 1991.

[4] Rose, M., "A Convention for Defining Traps for use with the
     SNMP", RFC 1215, March 1991.

[5] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser,
     "Structure of Management Information for Version 2 of the Simple
     Network Management Protocol (SNMPv2)", RFC 1902, January 1996.

[6] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual
     Conventions for Version 2 of the Simple Network Management
     Protocol (SNMPv2)", RFC 1903, January 1996.

[7] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser,
     "Conformance Statements for Version 2 of the Simple Network
     Management Protocol (SNMPv2)", RFC 1904, January 1996.

[8] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple
     Network Management Protocol", STD 15, RFC 1157, May 1990.

[9] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser,
     "Introduction to Community-based SNMPv2", RFC 1901, January
     1996.

   [10] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser,
        "Transport Mappings for Version 2 of the Simple Network
        Management Protocol (SNMPv2)", RFC 1906, January 1996.

   [11] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message
        Processing and Dispatching for the Simple Network Management
        Protocol (SNMP)", RFC 2272, January 1998.

   [12] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM)
        for version 3 of the Simple Network Management Protocol
        (SNMPv3)", RFC 2274, January 1998.

   [13] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol
        Operations for Version 2 of the Simple Network Management
        Protocol (SNMPv2)", RFC 1905, January 1996.

   [14] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC
        2273, January 1998.

   [15] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access
         Control Model (VACM) for the Simple Network Management Protocol
         (SNMP)", RFC 2275, January 1998

   [16] Hovey, R., and S. Bradner, "The Organizations Involved in the
        IETF Standards Process", BCP 11, RFC 2028, October 1996.

   [17] Bradner, S., "Key words for use in RFCs to Indicate Requirement
        Levels", BCP 14, RFC 2119, March 1997.

   [18] Clouston, B., and B. Moore, "Definition of Managed Objects for
        APPN", RFC 2455, November 1998.

   [19] Clouston, B., and B. Moore, "Definitions of Managed Objects for
        DLUR", RFC 2232, November 1997.

9. Authors' Addresses

Bob Clouston
Cisco Systems
7025 Kit Creek Road
P.O. Box 14987
Research Triangle Park, NC 27709, USA

Phone: +1 919 472 2333
EMail: clouston@cisco.com

Robert Moore
Dept. BRQA/Bldg. 501/G114
IBM Corporation
P.O.Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709, USA

Phone: +1 919 254 4436
EMail: remoore@us.ibm.com

10. Full Copyright Statement

   Copyright (C) The Internet Society (1998). 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
   HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
   MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.