AlarmTraq

SNMP with Avaya IP Office

Version 3.1

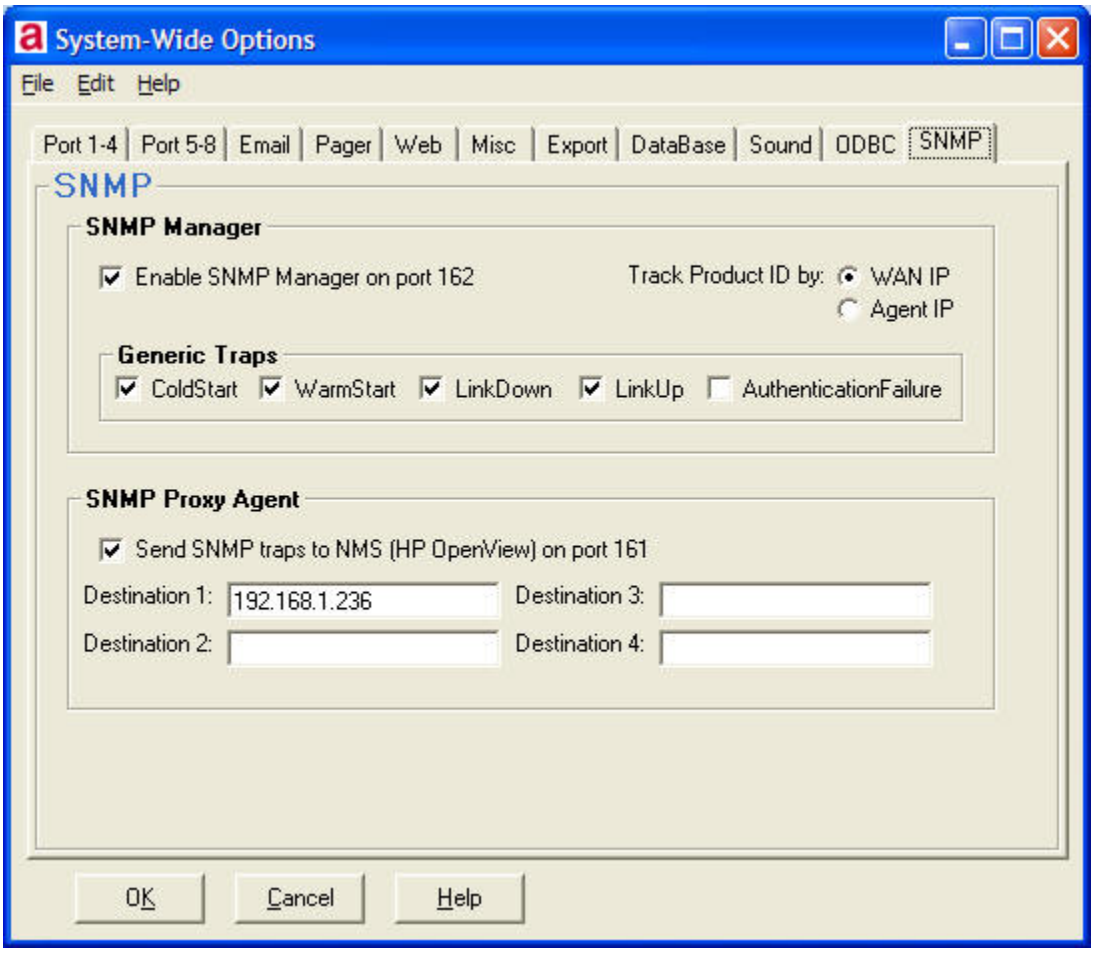
May 17, 2006

Reilly Telecom, Inc. Voice & Data Telecommunications Design & Administration
203-234-9115 | 203-234-9113 Fax | info@reillytele.com

Contents

AlarmTraq SNMP Setup

To setup SNMP click on **Edit>System-Wide Options>SNMP tab**.



SNMP Manager:

Enable SNMP Manager on Port 162: Select this option to automatically load the SNMP manager application to receive traps from systems such as S8700, S8300, S8500, and IP Office. SNMP Manager will run in the background and can be monitored with the SNMP Viewer. To open the SNMP Viewer, click **View>SNMP Trap Viewer.**

Track Product ID by: If you wish to track the WAN IP address for use when assigning a Product ID select WAN IP. To track by Agent IP select Agent IP.

Generic Traps: Select the trap types you want to be processed. If not checked these trap types will be ignored by AlarmTraq.

SNMP Proxy Agent:

Send SNMP trap to NMS (HP OpenView) on port 161: Select this option to automatically send an SNMP trap message to a 3rd party Network Management System (NMS) when an alarm is received via modem or SNMP.

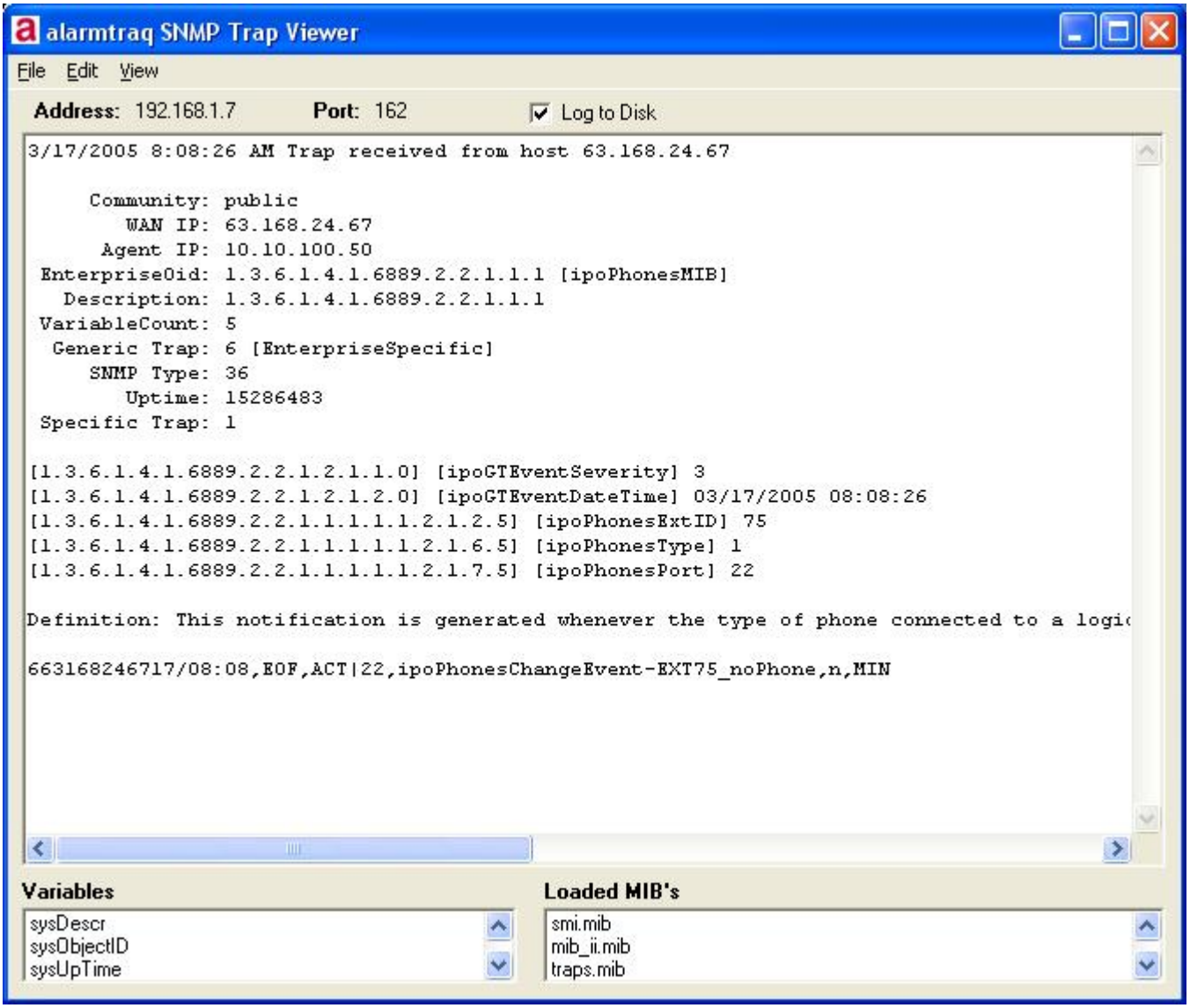
This option allows AlarmTraq to act as a proxy SNMP server to integrate legacy Definity systems such as System-75, G1, G2 and G3 to be monitored by HP OpenView or any 3rd party SNMP manager application.

Destination 1-4: Enter the IP address of the SNMP Manager application you would like to send traps to. This is the IP address of the HP OpenView server or other 3rd party SNMP Manager.

Make sure you DO NOT enter the IP address of the AlarmTraq server in this field, as this will create an endless loop the next time an alarm is received.

AlarmTraq SNMP Viewer:

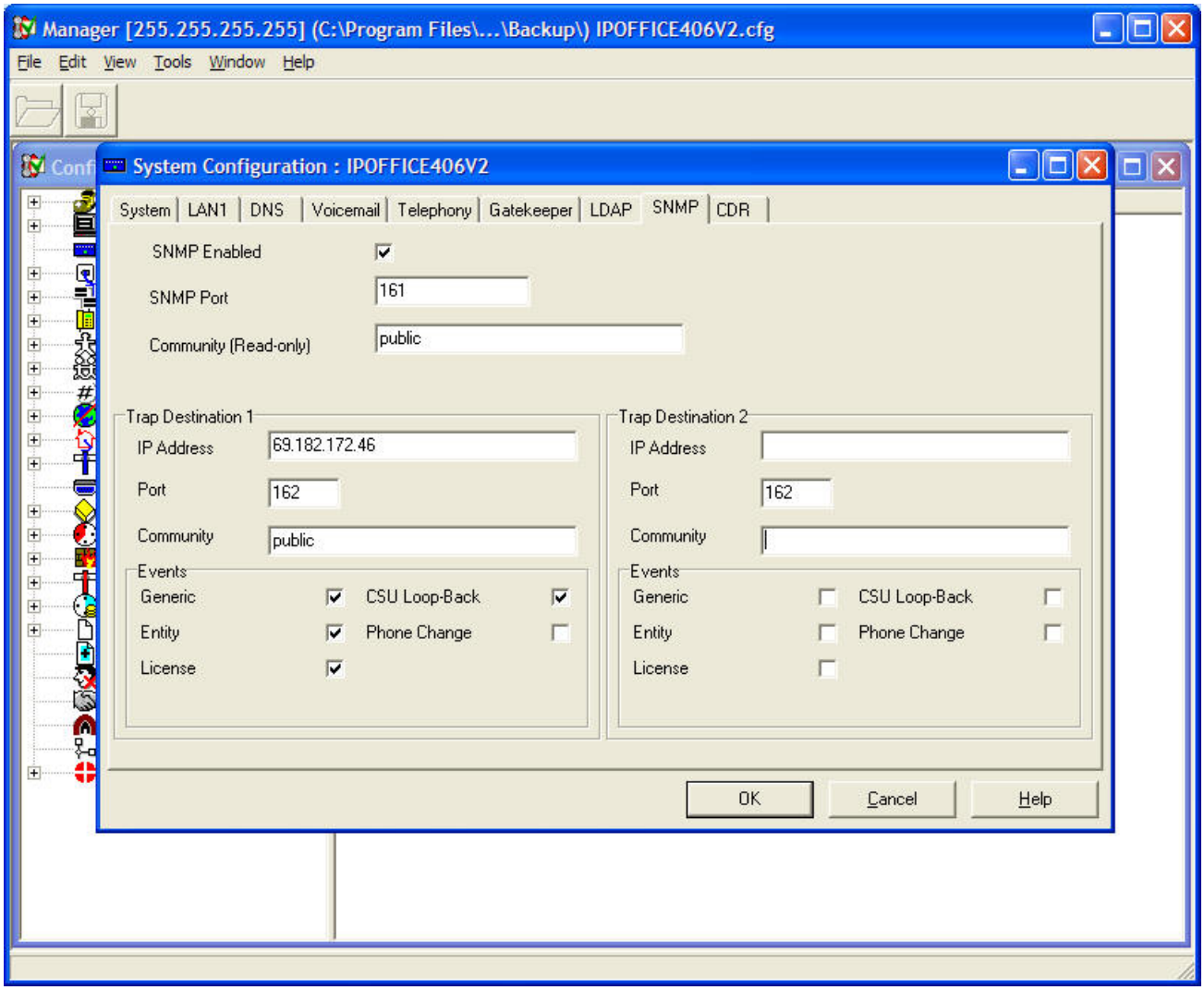
To open the SNMP Viewer, click **View>SNMP Trap Viewer**



To open the SNMP Viewer, click **View>SNMP Trap Viewer**

IP Office SNMP Setup

Open the IP Office Manager application and click on the SNMP tab.



Simple Network Management Protocol (SNMP) allows network devices (SNMP clients and servers) to exchange information. SNMP clients are built into devices such as network routers, server PC, etc. SNMP servers are typically PC application which receive and/or request SNMP information. AlarmTraq is an SNMP server.

The IP Office SNMP client allows the IP Office control unit to respond to SNMP polling and to send information about error conditions to AlarmTraq.

Note: In order for AlarmTraq to interact with an IP Office, the IP Office MIB files, provided on the IP Office Admin CD, must be compiled into AlarmTraq. Refer to the IP Office Installation Manual.

SNMP Enabled: *Default = Off*

Enables support for SNMP by the IP Office control unit.

SNMP Port: *Default = 161*

The port on which the control listens and responds to SNMP polling traffic.

Community (Read-only): *Default = Blank*
The SNMP community, eg. public.

Trap Destination 1/2:
The control unit supports two SNMP traps, to which it can send specified IP Office events.

IP Address: *Default = Blank*
The public IP address of the AlarmTraq server to which trap information is sent.

Port: *Default = 162*
The SNMP transmit port

Community: *Default = Blank*
The SNMP community for the transmitted traps. Must be matched by the receiving SNMP server.

Events: Default = None
Sets which types of IP Office events should be collected and sent by the trap:

Generic: Report on cold starts, warm starts and SNMP authentication failure.

Entity: Report on link up/down changes between IP Office modules (except WAN3), trunks and VCM.

Licence: Report failure to connect with the Licence Key Server.

CSU Loop-Back: Only displays when the system locale is set to ***enu***. Ticking this field enables the sending of CSU loop-back events, which may then be monitored by an SNMP manager application.

Phone Change: Send a trap whenever a phone is removed or moved.

IP Office Alarms

When a trap is received by an IP Office system, the Product ID is derived from the first nine digits of the WAN IP Address plus as many zeros as needed to become a nine-digit number, with the addition of a "**6**" inserted at the beginning in order to identify itself as an IP Office. See the above example:

IP Office Identifier = **6**
WAN IP Address = **63.168.24.67**
Product ID will be **6631682467**

Sample SNMP_TrapLog-MM-DD-YYYY.log (ipoProd403DS)

```
3/15/2005 12:30:28 PM Trap received from host 63.168.24.67

Community: public
WAN IP: 63.168.24.67
Agent IP: 10.10.100.50
EnterpriseOid: 1.3.6.1.4.1.6889.1.2.1.2.2 [ipoProd403DS]
Description: 1.3.6.1.4.1.6889.1.2.1.2.2
VariableCount: 0
Generic Trap: 1 [WarmStart]
SNMP Type: 36
Uptime: 1500
Specific Trap: 0


Definition: A warmStart trap signifies that the sending protocol entity is reinitializing itself such that neither the
agent configuration nor the protocol entity implementation is altered.

663168246715/12:30,EOF,ACT|0,WarmStart-ipoProd403DS,n,WAR

***************** End Trap 3/15/2005 12:30:31 PM ************************************
```

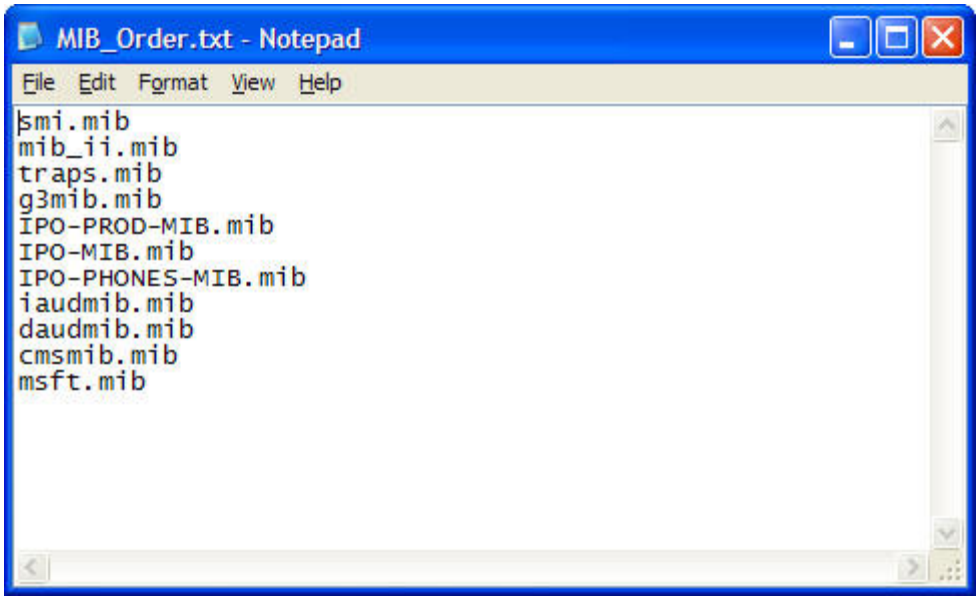
Sample SNMP_TrapLog-MM-DD-YYYY.log (ipoPhonesMIB)

```
3/15/2005 11:18:51 AM Trap received from host 63.168.24.67

Community: public
WAN IP: 63.168.24.67
Agent IP: 10.10.100.50
EnterpriseOid: 1.3.6.1.4.1.6889.2.2.1.1.1 [ipoPhonesMIB]
Description: 1.3.6.1.4.1.6889.2.2.1.1.1
VariableCount: 5
Generic Trap: 6 [EnterpriseSpecific]
SNMP Type: 36
Uptime: 8929612
Specific Trap: 1

[1.3.6.1.4.1.6889.2.2.1.2.1.1.0] [ipoGTEventSeverity] 3
[1.3.6.1.4.1.6889.2.2.1.2.1.2.0] [ipoGTEventDateTime] 03/15/2005 11:18:51
[1.3.6.1.4.1.6889.2.2.1.1.1.1.1.2.1.2.3] [ipoPhonesExtID] 67
[1.3.6.1.4.1.6889.2.2.1.1.1.1.1.2.1.6.3] [ipoPhonesType] 1
[1.3.6.1.4.1.6889.2.2.1.1.1.1.1.2.1.7.3] [ipoPhonesPort] 20

Definition: This notification is generated whenever the type of phone connected to a logical extension entity is
detected as having changed after completion of normal start up of the Agent entity.

663168246715/11:18,EOF,ACT|20,ipoPhonesChangeEvent-EXT67_noPhone,n,MIN

***************** End Trap 3/15/2005 11:18:54 AM ************************************
```

IP Office MIBs

Although MIB (Management Information Base) files are not required for AlarmTraq to operate, MIBs allow AlarmTraq to access management data from supported systems using SNMP.

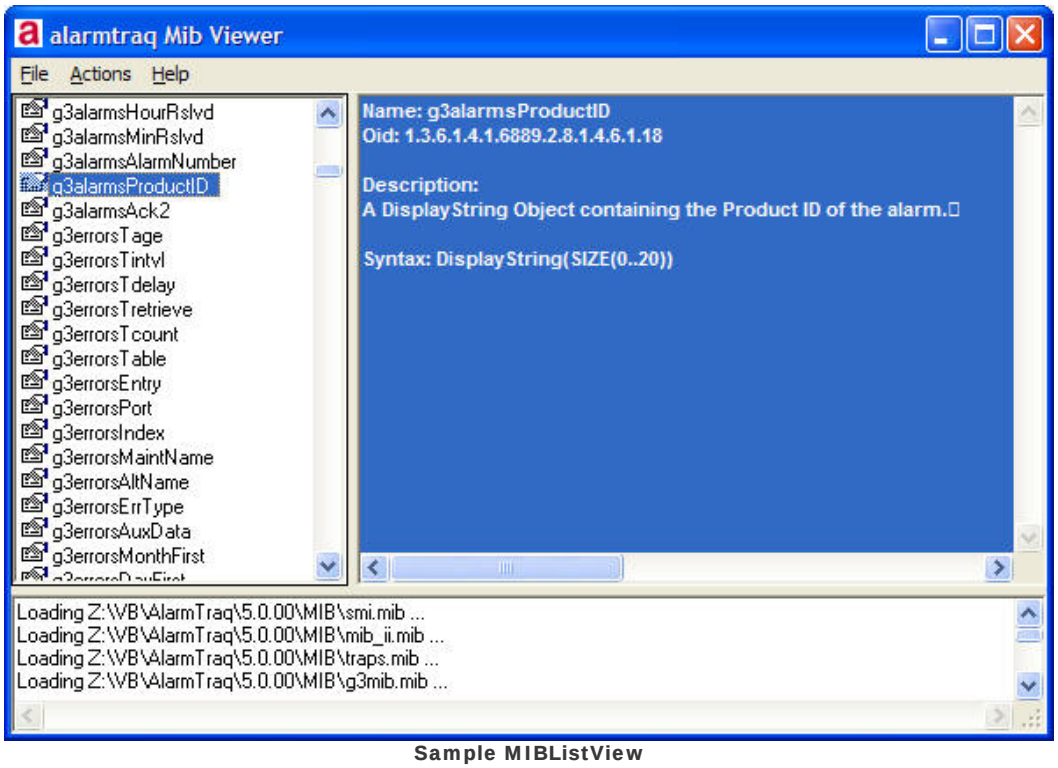
All MIB files that need to be loaded at startup should reside in the "AlarmTraq\MIB\" directory, and the "MIB_Order.txt" file should contain the file names in the order they should be loaded.



Sample MIB_Order.txt file

To Verify the proper operation and loading of the MIBs, a separate program names MIBListView.exe located in the \AlarmTraq directory will show all the variables assigned by the MIBs.

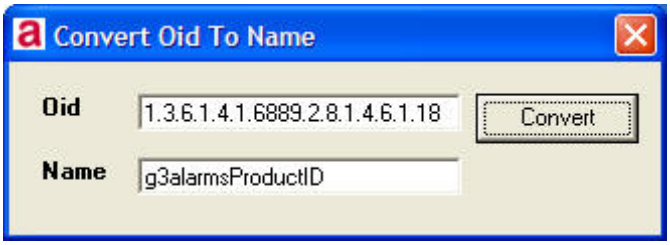
Run "C:\Program Files\AlarmTraq\MIBListView.exe"



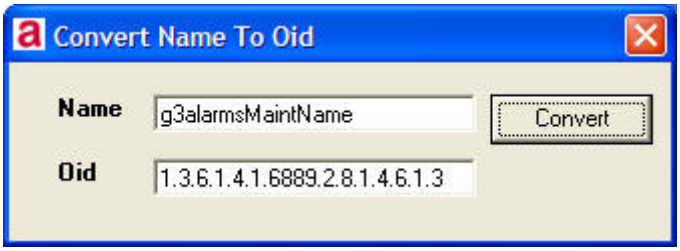
Sample MIBListView

The left column shows the list of variables loaded by the MIBs. The main window shows the Oid (Object Identifier) and description of the variable.

To find the variable name from an Oid, click "Action>Convert Oid To Name" (see below).



To find the OID from a variable name, click "Action>Convert Name To Oid" (see below).



IP Office SNMP Traps

IPO-MIB Traps

ipoGenEntityFailureEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime,
      ipoGTEventEntity
    }
    STATUS current
    DESCRIPTION
      "A ipoGenEntityFailureEvent notification is generated whenever a
      physical entity on the IP Office fails in its operation. It
      signifies that the SNMP entity, acting in an agent role, has
      detected that the state of a physical entity of the system has
      transitioned from the operational to the failed state"
    ::= { ipoGTEvents 1 }
```

ipoGenEntityOperationalEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime,
      ipoGTEventEntity
    }
    STATUS current
    DESCRIPTION
      "A ipoGenEntityOperationalEvent notification is generated whenever
      a physical entity on the IP Office becomes operational again
      after having failed. It signifies that the SNMP entity, acting
      in an agent role, has detected that the state of a physical
      entity of the system has transitioned from the failed to the
      operational state"
    ::= { ipoGTEvents 2 }
```

ipoGenEntityErrorEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime,
      ipoGTEventEntity
    }
    STATUS current
    DESCRIPTION
      "A ipoGenEntityErrorEvent notification is generated whenever a
      physical entity on the IP Office experiences a temporary
      error. It signifies that the SNMP entity, acting in an agent
      role, has detected a transitory error on a physical entity of
      the system."
    ::= { ipoGTEvents 3 }
```

ipoGenEntityChangeEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime,
      ipoGTEventEntity
    }
    STATUS current
    DESCRIPTION
```

```
      "A ipoGenEntityChangeEvent notification is generated whenever
      a physical entity on the IP Office experiences a change itself
      or with other entities associated with it. It signifies that
      the SNMP entity, acting in an agent role, has detected a non
      error/failure change for a physical entity on the system."
   ::= { ipoGTEvents 4 }
```

ipoGenLKSCommsFailureEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime
    }
    STATUS current
    DESCRIPTION
      "A ipoGenLKSCommsFailureEvent notification is generated
      whenever communication with a Licence Key Server fails.  It
      signifies that the SNMP entity, acting in an agent role, has
      detected that the state of the communications between the
      Licence Key Server has transitioned from the operational to
      the failed state."
    ::= { ipoGTEvents 5 }
```

ipoGenLKSCommsOperationalEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime
    }
    STATUS current
    DESCRIPTION
      "A ipoGenLKSCommsOperationalEvent notification is generated
      whenever communication with a Licence Key Server becomes
      operational again after having failed.  It signifies that the
      SNMP entity, acting in an agent role, has detected that the
      state of the communications between the Licence Key Server has
      transitioned from the failed to the operational state."
    ::= { ipoGTEvents 6 }
```

ipoGenLKSCommsErrorEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime
    }
    STATUS current
    DESCRIPTION
      "A ipoGenLKSCommsErrorEvent notification is generated whenever
       a IP Office experiences a temporary error with License Key
       Server communication. It signifies that the SNMP entity,
       acting in an agent role, has detected a transitory error with
       the communication between the License Key Server and Client
       on the system."
    ::= { ipoGTEvents 7 }
```

ipoGenLKSCommsChangeEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime
    }
    STATUS current
    DESCRIPTION
      "A ipoGenLKSCommsChangeEvent notification is generated
      whenever a IP Office experiences a change a non error change
```

```
      License Key Server communication operation.  It signifies that
      the SNMP entity, acting in an agent role, has detected a non
      error/failure change with the License Key Server and Client
      operation on the system."
   ::= { ipoGTEvents 8 }
```

ipoGenLoopbackEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime,
      ipoGTEventEntity,
      ipoGTEventLoopbackStatus
    }
    STATUS current
    DESCRIPTION
      "A ipoGenLoopbackEvent notification is generated whenever a IP
      Office T1 (DS1) interface operating as a CSU actions a loopback
      status change."
   ::= { ipoGTEvents 9 }
```

ipoGenAppFailureEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime,
      ipoGTEventAppEntity
    }
    STATUS current
    DESCRIPTION
      "A ipoGenAppFailureEvent notification is generated whenever
      communication between a IP Office switch and a IP Office
      application fails.  It signifies that the SNMP entity, acting
      in an agent role, has detected that the state of the
      communications between the IP Office switch and a IP Office
      application has transitioned from the operational to the
      failed state.  The IP Office application between which
      communication has been lost is identified by the value of
      ipoGTEventAppEntity."
    ::= { ipoGTEvents 10 }
```

ipoGenAppOperationalEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime,
      ipoGTEventAppEntity
    }
    STATUS current
    DESCRIPTION
      "A ipoGenAppOperationalEvent notification is generated
      whenever communication between a IP Office switch and a IP
      Office application becomes operational again after having
      failed.  It signifies that the SNMP entity, acting in an agent
      role, has detected that the state of the communications
      between the IP Office switch and a IP Office application has
      transitioned from the failed to the operational state.  The IP
      Office application between which communication has been lost
      is identified by the value of ipoGTEventAppEntity."
    ::= { ipoGTEvents 11 }
```

ipoGenAppEvent NOTIFICATION-TYPE

```
    OBJECTS {
```

```
      ipoGTEventSeverity,
      ipoGTEventDateTime,
      ipoGTEventAppEntity,
      ipoGTEventAppEvent
    }
    STATUS current
    DESCRIPTION
      "A ipoGenAppEvent notification is generated whenever a
      application entity of the IP Office system experiences an event.
        It signifies that the SNMP entity, acting as a proxy for
        the application, has detected an event on the application
        entity of the overall IP Office system.
        The event severity varies dependent upon the event condition."
    ::= { ipoGTEvents 12 }
```

ipoGenSogHostFailureEvent NOTIFICATION-TYPE

```
    OBJECTS {
        ipoGTEventSeverity,
        ipoGTEventDateTime,
        ipoGTEventHostAddress
    }
    STATUS current
    DESCRIPTION
        "An ipoGenSogFailureEvent notification is generated whenever a
        previously valid Sub-tending host fails during Small Office
        Gateway operation.
        The ipAddress field indicates the address of the failed host.
        The event severity will always indicate Major."
    ::= { ipoGTEvents 13 }
```

ipoGenSogModeChangeEvent NOTIFICATION-TYPE

```
    OBJECTS {
        ipoGTEventSeverity,
        ipoGTEventDateTime,
        ipoGTEventSOGMode
    }
    STATUS current
    DESCRIPTION
        "An ipoGenSogModeChangeEvent notification is generated whenever
        the Small Office Gateway operating mode changes. This also
        includes entry to the initial mode.
        The ipoGTEventSOGMode field indicates the new operating mode.
        The event severity will be major(2) for a ipoGTEventSOGMode value
        of survivable(1), and minor(3) for a ipoGTEventSOGMode value of
        subTending(2)."
    ::= { ipoGTEvents 14 }
```

IPOPhone Traps

ipoPhonesChangeEvent NOTIFICATION-TYPE

```
    OBJECTS {
      ipoGTEventSeverity,
      ipoGTEventDateTime,
        ipoPhonesExtID,
        ipoPhonesType,
        ipoPhonesPort
      }
```

```
    STATUS current
    DESCRIPTION
      "This notification is generated whenever the type of phone
      connected to a logical extension entity is detected as having
      changed after completion of normal start up of the Agent
      entity.

      Its purpose is to allow a management application to identify
      the removal or switching of phone types on the IP Office
      entity."
    ::= { ipoPhonesMibNotifications 1 }
```

Generic Traps

RFC1157-MIB DEFINITIONS

coldStart TRAP-TYPE

```
    ENTERPRISE  snmp
    DESCRIPTION
                "A coldStart trap signifies that the sending
                protocol entity is reinitializing itself such
                that the agent's configuration or the rotocol
                entity implementation may be altered."
    ::= 0
```

warmStart TRAP-TYPE

```
    ENTERPRISE  snmp
    DESCRIPTION
                "A warmStart trap signifies that the sending
                protocol entity is reinitializing itself such
                that neither the agent configuration nor the
                protocol entity implementation is altered."
    ::= 1
```

linkDown TRAP-TYPE

```
    ENTERPRISE  snmp
    VARIABLES   { ifIndex }
    DESCRIPTION
                "A linkDown trap signifies that the sending
                protocol entity recognizes a failure in one of
                the communication links represented in the
                agent's configuration."
    ::= 2
```

linkUp TRAP-TYPE

```
    ENTERPRISE  snmp
    VARIABLES   { ifIndex }
    DESCRIPTION
                "A linkUp trap signifies that the sending
                protocol entity recognizes that one of the
                communication links represented in the agent's
                configuration has come up."
    ::= 3
```

authenticationFailure TRAP-TYPE

```
    ENTERPRISE  snmp
    DESCRIPTION
                "An authenticationFailure trap signifies that
```

```
                the sending protocol entity is the addressee
                of a protocol message that is not properly
                authenticated.  While implementations of the
                SNMP must be capable of generating this trap,
                they must also be capable of suppressing the
                emission of such traps via an implementation-
                specific mechanism."
        ::= 4
```

egpNeighborLoss TRAP-TYPE

```
        ENTERPRISE  snmp
        VARIABLES   { egpNeighAddr }
        DESCRIPTION
                "An egpNeighborLoss trap signifies that an EGP
                neighbor for whom the sending protocol entity
                was an EGP peer has been marked down and the
                peer relationship no longer obtains."
        ::= 5
```