



# AlarmTraq

## SNMP with Avaya IP Office

### Version 3.2

July 27, 2006

Reilly Telecom, Inc. Voice & Data Telecommunications Design & Administration  
203-234-9115 | 203-234-9113 Fax | [info@reillytele.com](mailto:info@reillytele.com)

© 2003 Reilly Telecom Inc. All Rights Reserved.

# Contents

---

AlarmTraq 3.2 with SNMP (Simple Network Management Protocol) .....	4
AlarmTraq SNMP Setup.....	4
SNMP Manager: .....	4
Enable SNMP Manager on Port 162.....	4
Track Product ID by.....	5
Generic Traps.....	5
SNMP Proxy Agent: .....	5
Send SNMP trap to NMS (HP OpenView) on port 161 .....	5
Destination 1-4.....	5
AlarmTraq SNMP Viewer: .....	5
IP Office 3.2 SNMP Setup.....	6
SNMP Info - Configuration: .....	6
SNMP Enabled.....	6
Community (Read-only) .....	6
SNMP Port:.....	6
Device ID:.....	6
Contact .....	6
Location .....	6
SNMP Info - Alarms:.....	7
IP Address.....	7
Phone Change .....	7
IP Office 3.2 Alarms .....	8
Sample SNMP_TrapLog-MM-DD-YYYY.log (ipoGenTraps).....	8
Sample SNMP_TrapLog-MM-DD-YYYY.log (ipoPhonesMIB) .....	8
IP Office 3.1 Alarms .....	9
Sample SNMP_TrapLog-MM-DD-YYYY.log (ipoProd403DS).....	9
Sample SNMP_TrapLog-MM-DD-YYYY.log (ipoPhonesMIB) .....	9
IP Office MIBs.....	10
IP Office SNMP Traps.....	12
ipoGenEntityFailureEvent .....	12
ipoGenEntityOperationalEvent.....	12
ipoGenEntityErrorEvent .....	12
ipoGenEntityChangeEvent .....	12
ipoGenLKSCommsFailureEvent .....	13
ipoGenLKSCommsOperationalEvent.....	13
ipoGenLKSCommsErrorEvent .....	13
ipoGenLKSCommsChangeEvent .....	13
ipoGenLoopbackEvent .....	14
ipoGenAppFailureEvent .....	14
ipoGenAppOperationalEvent.....	14
ipoGenAppEvent.....	14
ipoGenSogHostFailureEvent .....	15
ipoGenSogModeChangeEvent.....	15
ipoGenColdStartSvcEvent .....	15
ipoGenWarmStartSvcEvent.....	15

ipoGenLinkDownSvcEvent .....	16
ipoGenLinkUpSvcEvent .....	16
ipoGenAuthFailureSvcEvent .....	16
ipoGenEntityFailureSvcEvent .....	17
ipoGenEntityOperationalSvcEvent.....	17
ipoGenEntityErrorSvcEvent .....	17
ipoGenEntityChangeSvcEvent .....	17
ipoGenLKSCommsFailureSvcEvent .....	18
ipoGenLKSCommsOperationalSvcEvent.....	18
ipoGenLKSCommsErrorSvcEvent.....	18
ipoGenLKSCommsChangeSvcEvent .....	19
ipoGenLoopbackSvcEvent .....	19
ipoGenAppFailureSvcEvent .....	19
ipoGenAppOperationalSvcEvent.....	19
ipoGenAppSvcEvent.....	20
ipoGenSogHostFailureSvcEvent .....	20
ipoGenSogModeChangeSvcEvent.....	20
IPOPhone Traps .....	21
ipoPhonesChangeEvent .....	21
ipoPhonesChangeSvcEvent .....	21
coldStart .....	22
warmStart.....	22
linkDown.....	22
linkUp .....	22
authenticationFailure.....	22
egpNeighborLoss .....	22

# AlarmTraq 3.2 with SNMP (Simple Network Management Protocol)

**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.

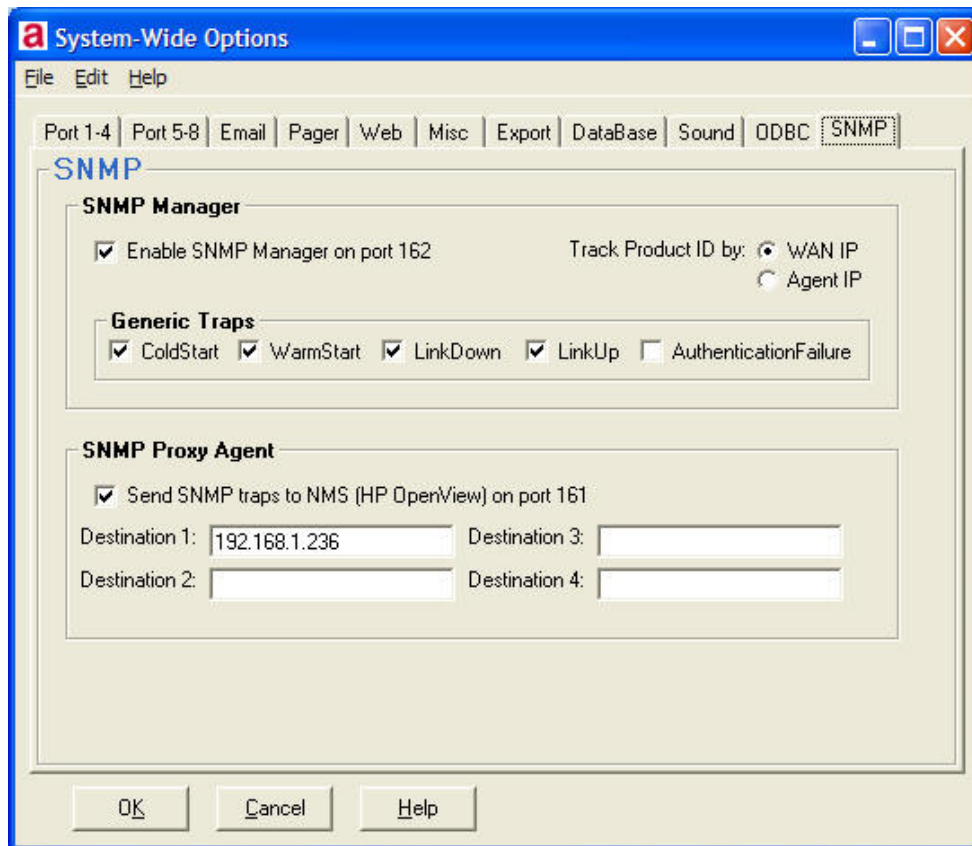
The IP Office SNMP client allows the IP Office control unit to respond to SNMP polling and to send information about error conditions to AlarmTraq.

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.

---

## AlarmTraq SNMP Setup

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



System-Wide Option - SNMP

### 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.

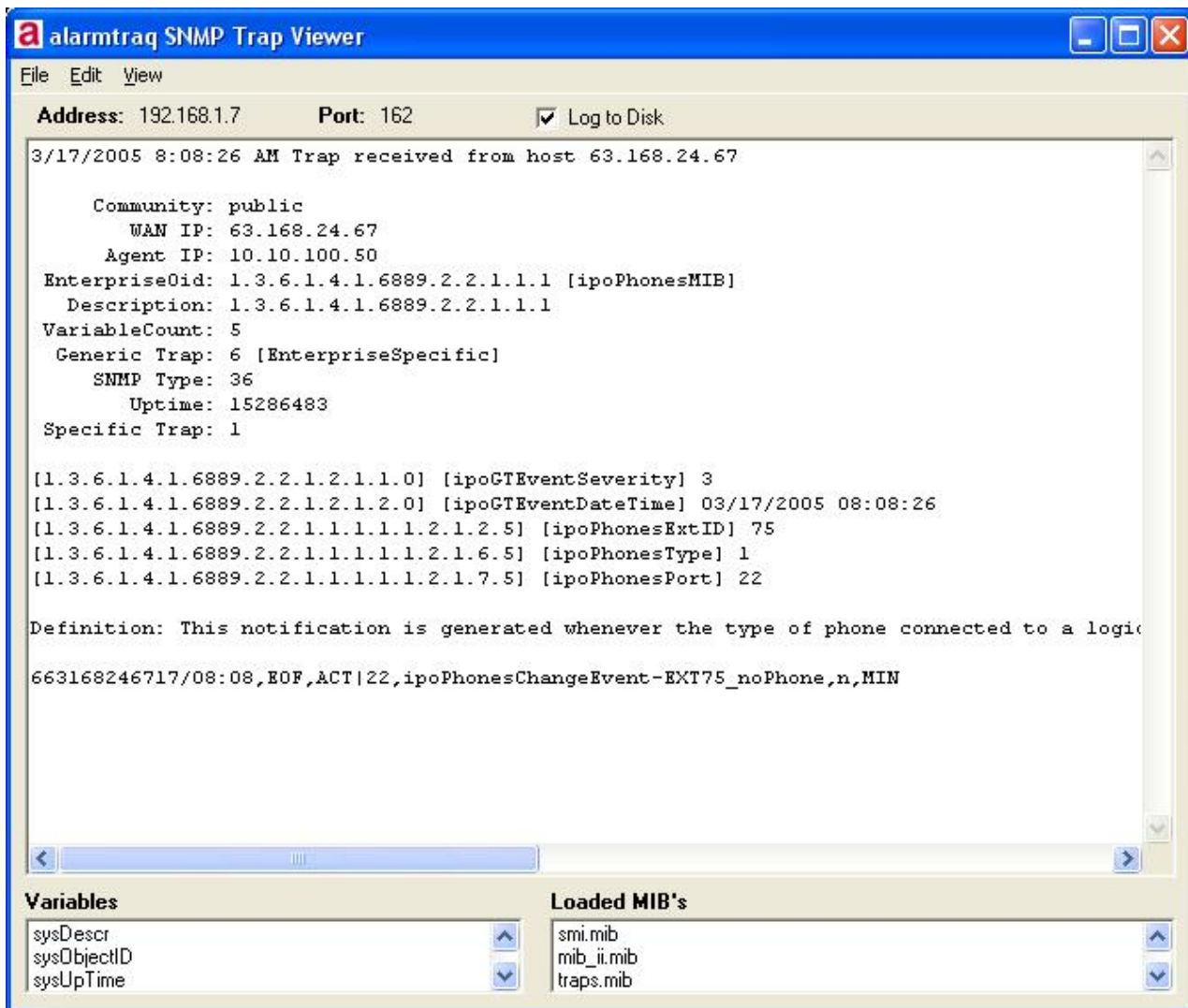
**Destination 1-4:** Enter the IP address of the SNMP Manager application you would like to send traps to. This is the IP address of the HP OpenView server or other 3rd party SNMP Manager.

**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:

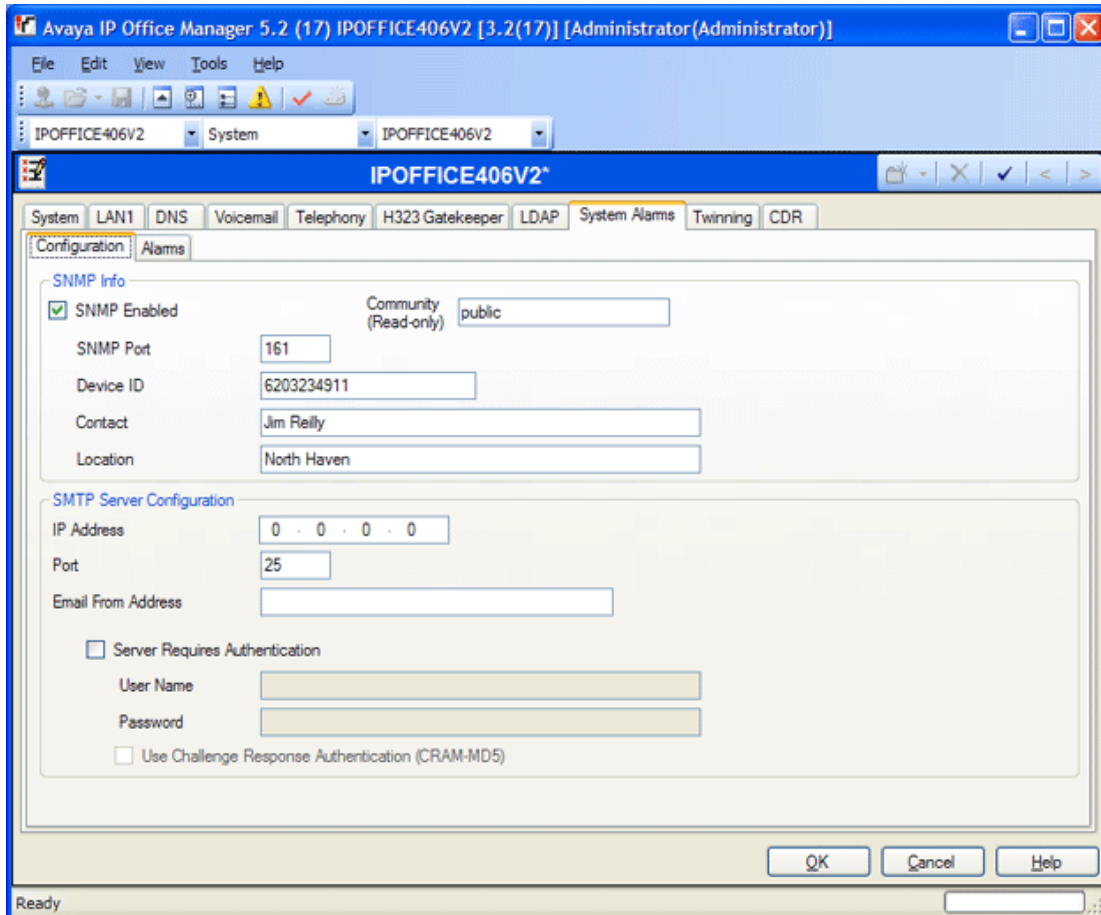
To open the SNMP Viewer, click **View>SNMP Trap Viewer**



# IP Office 3.2 SNMP Setup

## SNMP Info - Configuration:

Open IP Office Manager 5.2 and click on the 'System Alarms' tab.



IP Office Manager 5.2 – System Alarms - Configuration

**SNMP Enabled:** Default = Off  
Enables support for SNMP by the IP Office control unit.

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

**SNMP Port:** Default = 161  
The port on which the control listens and responds to SNMP polling traffic.

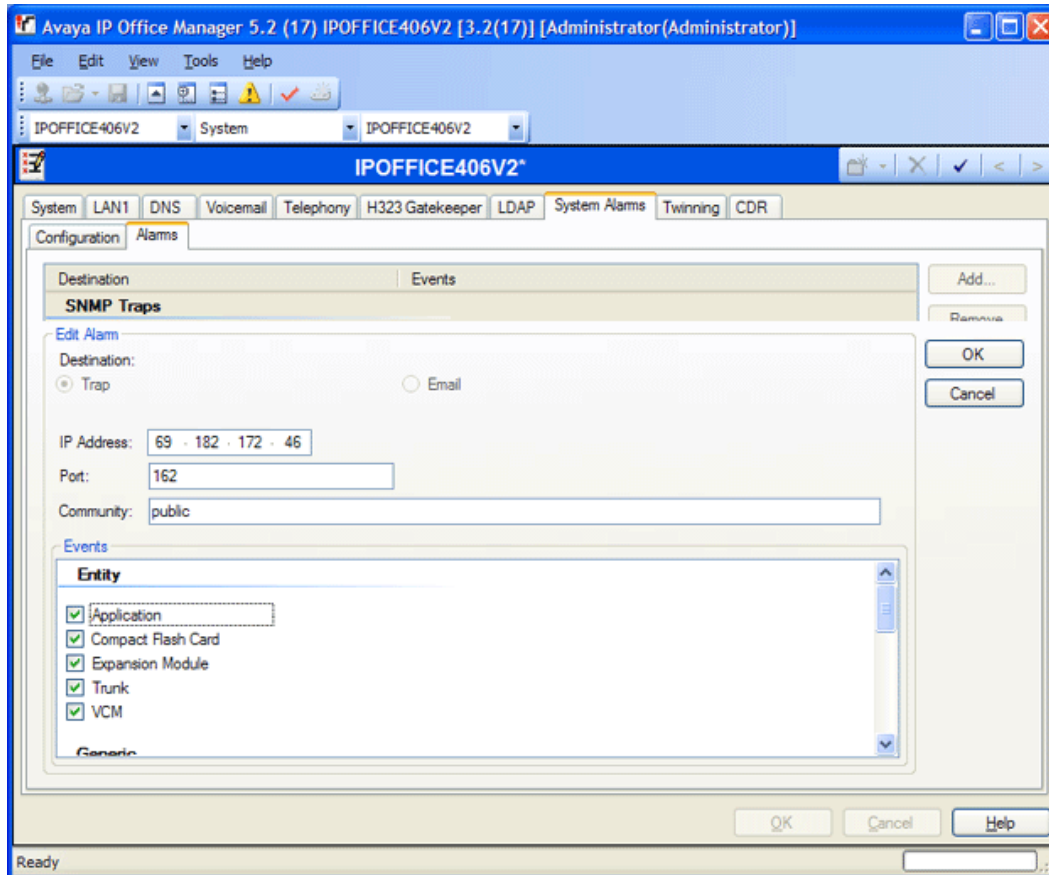
**Device ID:** Default = blank  
Enter the 10-digit Product ID of this system to be used by AlarmTraq. **This number must be in the format 6XXXXXXXXX and be unique in order for AlarmTraq to recognize this system**

**Contact:**  
Enter the name of the contact for this system (reference only, not used by AlarmTraq).

**Location:**  
Enter the name of the location of this system (reference only, not used by AlarmTraq).

## SNMP Info - Alarms:

Click on the 'Alarms' tab.



IP Office Manager 5.2 – System Alarms - Configuration

**IP Address:** Default = Blank

The public IP address of the AlarmTrap 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:

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

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

**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 3.2 Alarms

IP Office 3.2 supports the addition of Device-ID (ipoGTEventDevID) that AlarmTraq uses to establish the Product ID of the system sending the trap.

### Sample SNMP\_TrapLog-MM-DD-YYYY.log (ipoGenTraps)

```
7/27/2006 12:45:06 PM SYS Trap received from host 63.168.24.67
Site based on WAN IP 63.168.24.67

Community: public
WAN IP: 63.168.24.67
Agent IP: 192.168.1.50
EnterpriseOid: 1.3.6.1.4.1.6889.2.2.1.2 [ipoGenTraps]
Description: 1.3.6.1.4.1.6889.2.2.1.2
VariableCount: 4
Generic Trap: 6 [EnterpriseSpecific]
SNMP Type: 36
Uptime: 2186
Specific Trap: 16 [ipoGenWarmStartSvcEvent]

[1.3.6.1.4.1.6889.2.2.1.2.1.9.0] [ipoGTEventStdSeverity] 4
[1.3.6.1.4.1.6889.2.2.1.2.1.2.0] [ipoGTEventDateTime] 07/27/2006 12:45:06
[1.3.6.1.4.1.6889.2.2.1.2.1.10.0] [ipoGTEventDevID] 6203234911
[1.3.6.1.2.1.1.1.0] [sysDescr] IP 406 DS 3.2(17)

Definition: Enterprise version of standard warmstart trap featuring device identification information

Processing alarm as: 620323491127/12:45,EOF,ACT|0,ipoGenWarmStartSvcEvent,n,MAJ
```

### Sample SNMP\_TrapLog-MM-DD-YYYY.log (ipoPhonesMIB)

```
7/27/2006 12:35:25 PM SYS Trap received from host 63.168.24.67
Site based on WAN IP 63.168.24.67

Community: public
WAN IP: 63.168.24.67
Agent IP: 192.168.1.50
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: 8
Generic Trap: 6 [EnterpriseSpecific]
SNMP Type: 36
Uptime: 363752
Specific Trap: 2 [ipoPhonesChangeSvcEvent]

[1.3.6.1.4.1.6889.2.2.1.2.1.9.0] [ipoGTEventStdSeverity] 5
[1.3.6.1.4.1.6889.2.2.1.2.1.2.0] [ipoGTEventDateTime] 07/27/2006 12:35:25
[1.3.6.1.4.1.6889.2.2.1.2.1.10.0] [ipoGTEventDevID] 6203234911
[1.3.6.1.2.1.1.1.0] [sysDescr] IP 406 DS 3.2(17)
[1.3.6.1.4.1.6889.2.2.1.1.1.1.2.1.2.2] [ipoPhonesExtID] 35
[1.3.6.1.4.1.6889.2.2.1.1.1.1.2.1.6.2] [ipoPhonesType] 26
[1.3.6.1.4.1.6889.2.2.1.1.1.1.2.1.7.2] [ipoPhonesPort] 29
[1.3.6.1.4.1.6889.2.2.1.2.1.11.0] [ipoGTEventEntityName] Controller, DS 1

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

Processing alarm as: 620323491127/12:35,EOF,ACT|M:"ipoPhonesExtID 35 changed to 5420 on ipoPhonesPort 29",,n,MIN
```



## IP Office 3.1 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
[1.3.6.1.4.1.6889.2.2.1.1.1.1.2.1.2.3] [ipoPhonesExtID] 67
[1.3.6.1.4.1.6889.2.2.1.1.1.1.2.1.6.3] [ipoPhonesType] 1
[1.3.6.1.4.1.6889.2.2.1.1.1.1.2.1.7.3] [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

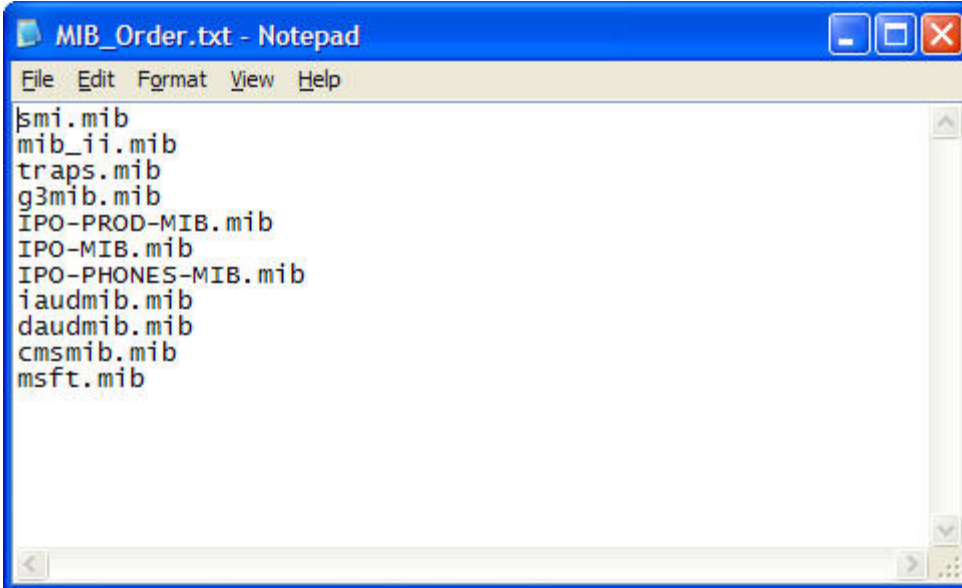
***** End Trap 3/15/2005 11:18:54 AM *****
```

---

## 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.

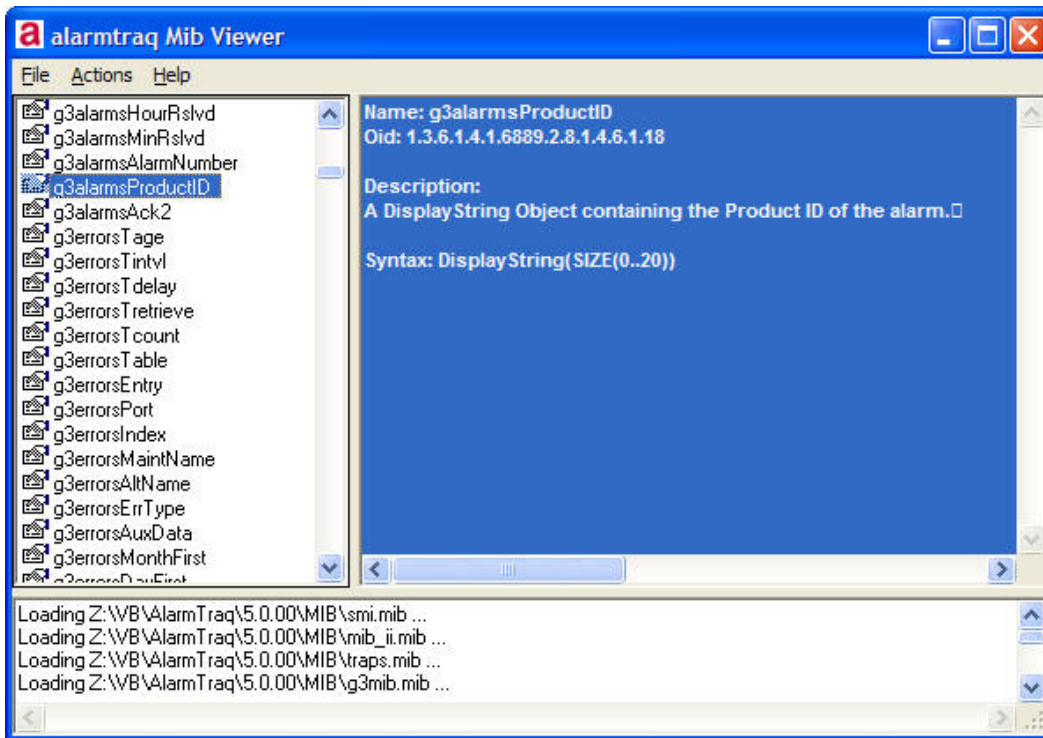


```
MIB_Order.txt - Notepad
File Edit Format View Help
smi.mib
mib_ii.mib
traps.mib
g3mib.mib
IPO-PROD-MIB.mib
IPO-MIB.mib
IPO-PHONES-MIB.mib
iaudmib.mib
daudmib.mib
cmsmib.mib
msft.mib
```

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.

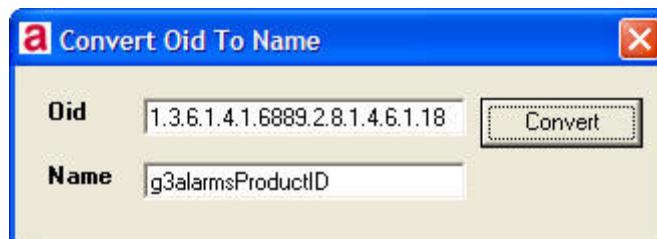
Run "C:\Program Files\AlarmTraq\MIBListView.exe"



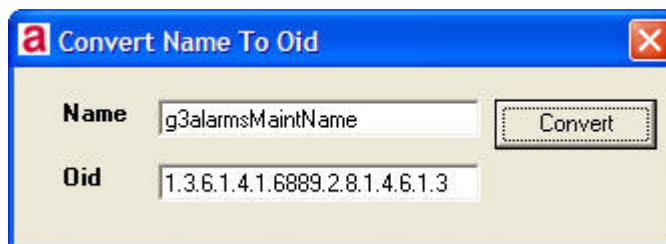
Sample MIBListView

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 3.1*

### **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 }

---

## ***IPO-MIB Traps 3.2***

#### **ipoGenColdStartSvcEvent** NOTIFICATION-TYPE

OBJECTS {

ipoGTEventStdSeverity,  
ipoGTEventDateTime,  
ipoGTEventDevID,  
sysDescr

}

STATUS current

DESCRIPTION

"Enterprise version of standard coldstart trap featuring device identification information. A coldStart trap signifies that the sending protocol entity is reinitializing itself such that the agent's configuration or the protocol entity implementation may be altered."

::= { ipoGTEvents 15 }

#### **ipoGenWarmStartSvcEvent** NOTIFICATION-TYPE

OBJECTS {

ipoGTEventStdSeverity,  
ipoGTEventDateTime,  
ipoGTEventDevID,  
sysDescr

}

STATUS current

DESCRIPTION

"Enterprise version of standard warmstart trap featuring device identification information. A warmStart trap signifies that the sending protocol entity is reinitializing that neither the agent configuration nor the protocol entity implementation is altered."

::= { ipoGTEvents 16 }

#### **ipoGenLinkDownSvcEvent** NOTIFICATION-TYPE

OBJECTS {

ipoGTEventStdSeverity,

ipoGTEventDateTime,

ipoGTEventDevID,

sysDescr,

ifIndex

}

STATUS current

DESCRIPTION

"Enterprise version of standard linkDown trap featuring device identification information. A linkDown trap signifies that the sending protocol entity recognizes a failure in one of the communication links represented in the agent's configuration."

::= { ipoGTEvents 17 }

#### **ipoGenLinkUpSvcEvent** NOTIFICATION-TYPE

OBJECTS {

ipoGTEventStdSeverity,

ipoGTEventDateTime,

ipoGTEventDevID,

sysDescr,

ifIndex

}

STATUS current

DESCRIPTION

"Enterprise version of standard linkUp trap featuring device identification information. 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."

::= { ipoGTEvents 18 }

#### **ipoGenAuthFailureSvcEvent** NOTIFICATION-TYPE

OBJECTS {

ipoGTEventStdSeverity,

ipoGTEventDateTime,

ipoGTEventDevID,

sysDescr

}

STATUS current

DESCRIPTION

"Enterprise version of standard authenticationFailure trap featuring device identification information. 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."

::= { ipoGTEvents 19 }



**ipoGenEntityFailureSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr,
    ipoGTEventEntity,
    ipoGTEventEntityName
}
STATUS current
DESCRIPTION
    "A ipoGenEntityFailureSvcEvent 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 20 }
```

**ipoGenEntityOperationalSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr,
    ipoGTEventEntity,
    ipoGTEventEntityName
}
STATUS current
DESCRIPTION
    "A ipoGenEntityOperationalSvcEvent 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 21 }
```

**ipoGenEntityErrorSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr,
    ipoGTEventEntity,
    ipoGTEventEntityName
}
STATUS current
DESCRIPTION
    "A ipoGenEntityErrorSvcEvent 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 22 }
```

**ipoGenEntityChangeSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr,
    ipoGTEventEntity,
    ipoGTEventEntityName
}
```

```
}
STATUS current
DESCRIPTION
    "A ipoGenEntityChangeSvcEvent 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 23 }
```

#### **ipoGenLKSCommsFailureSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr
}
STATUS current
DESCRIPTION
    "A ipoGenLKSCommsFailureSvcEvent 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 24 }
```

#### **ipoGenLKSCommsOperationalSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr
}
STATUS current
DESCRIPTION
    "A ipoGenLKSCommsOperationalSvcEvent 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 25 }
```

#### **ipoGenLKSCommsErrorSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr
}
STATUS current
DESCRIPTION
    "A ipoGenLKSCommsErrorSvcEvent 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 26 }
```

### **ipoGenLKSCommsChangeSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr
}
STATUS current
DESCRIPTION
    "A ipoGenLKSCommsChangeSvcEvent 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 27 }
```

### **ipoGenLoopbackSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr,
    ipoGTEventEntity,
    ipoGTEventEntityName,
    ipoGTEventLoopbackStatusBits
}
STATUS current
DESCRIPTION
    "A ipoGenLoopbackSvcEvent notification is generated whenever a
    IP Office T1 (DS1) interface operating as a CSU actions a
    loopback status change."
::= { ipoGTEvents 28 }
```

### **ipoGenAppFailureSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr,
    ipoGTEventAppEntity
}
STATUS current
DESCRIPTION
    "A ipoGenAppFailureSvcEvent 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 29 }
```

### **ipoGenAppOperationalSvcEvent** NOTIFICATION-TYPE

```
OBJECTS {
    ipoGTEventStdSeverity,
    ipoGTEventDateTime,
    ipoGTEventDevID,
    sysDescr,
    ipoGTEventAppEntity
}
STATUS current
```

#### DESCRIPTION

"A ipoGenAppOperationalSvcEvent 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.

:: = { ipoGTEvents 30 }

#### **ipoGenAppSvcEvent** NOTIFICATION-TYPE

##### OBJECTS {

ipoGTEventStdSeverity,  
ipoGTEventDateTime,  
ipoGTEventDevID,  
sysDescr,  
ipoGTEventAppEntity,  
ipoGTEventAppEvent

}

STATUS current

##### DESCRIPTION

"A ipoGenAppSvcEvent 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 31 }

#### **ipoGenSogHostFailureSvcEvent** NOTIFICATION-TYPE

##### OBJECTS {

ipoGTEventStdSeverity,  
ipoGTEventDateTime,  
ipoGTEventDevID,  
sysDescr,  
ipoGTEventHostAddress

}

STATUS current

##### DESCRIPTION

"An ipoGenSogFailureSvcEvent 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(4)."

:: = { ipoGTEvents 32 }

#### **ipoGenSogModeChangeSvcEvent** NOTIFICATION-TYPE

##### OBJECTS {

ipoGTEventStdSeverity,  
ipoGTEventDateTime,  
ipoGTEventDevID,  
sysDescr,  
ipoGTEventSOGMode

}

STATUS current

##### DESCRIPTION

"An ipoGenSogModeChangeSvcEvent notification is generated whenever the Small Office Gateway operating mode changes. This also includes entry to the initial mode.

:: = { ipoGTEvents 33 }

---

## IPOPhone Traps

### **ipoPhonesChangeEvent** NOTIFICATION-TYPE

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

STATUS deprecated

#### 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.

**\*\*NOTE:** This notification is deprecated and replaced by ipoPhonesChangeSvcEvent."

:: = { ipoPhonesMibNotifications 1 }

### **ipoPhonesChangeSvcEvent** NOTIFICATION-TYPE

OBJECTS {  
    ipoGTEventStdSeverity,  
    ipoGTEventDateTime,  
    ipoGTEventDevID,  
    sysDescr,  
    ipoPhonesExtID,  
    ipoPhonesType,  
    ipoPhonesPort,  
    ipoGTEventEntityName  
}

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.

Newer implementations of this MIB should put in place this event in favour of ipoPhonesChangeEvent."

:: = { ipoPhonesMibNotifications 2 }

---

## 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 protocol 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