

IPWorks Fault Management Guide for DL380 Platform

USER GUIDE

Copyright

© Ericsson AB 2017, 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Related Information	1
2	Overview	3
2.1	SNMP	3
2.1.1	SNMP Versions Supported by IPWorks	3
2.1.2	SNMP Master Agent	4
2.2	Alarms and Traps	4
2.2.1	Alarms According to Ericsson Alarm IRP MIB	4
2.3	Supported MIBs	6
3	Enabling Fault Management	9
3.1	Configuring Users - SNMPv3	9
3.1.1	Creating SNMPv3 Users Using SNMP Configuration Files	9
3.1.2	Creating SNMPv3 Users Using net-snmp-config	11
3.1.3	Creating Multiple SNMPv3 Users Using snmpusm	12
3.1.4	Changing User Properties and Removing Users	13
3.1.5	Testing SNMPv3 Configuration	14
3.2	Configuring Trap Destinations	15
3.2.1	Sending SNMPv1 Traps	15
3.2.2	Sending SNMPv2 Traps	16
3.2.3	Sending SNMPv3 Traps	16
3.2.4	Configuring Trap Handler	17
4	Fault Management Activities	19
4.1	Alarm Synchronization	19
4.2	Manual Alarm Clearing	20
4.2.1	Prerequisites	20
4.2.2	Manual Alarms Clearing Using snmptrap	20
4.2.3	Using alarmviewer Tool to clear Alarms Manually	22
4.3	Platform Alarm Configuration	23
4.3.1	Disk Usage Monitoring	23
4.3.2	System Load Monitoring	24
4.3.3	Swap Space Monitoring	25
4.3.4	Memory Usage Monitoring	26
4.3.5	Link up and Link down Notifications	26
4.3.6	Interface Traffic Monitoring	28
4.3.7	DPDK Status Monitoring	28
4.3.8	Fixed OID for Ethernet Interface	31
4.3.9	Environment Variables Monitoring (LM-Sensors)	35
	Reference List	47





1 Introduction

This document describes the fault management for host OS of HP ProLiant DL380 Gen9 (G9) and HP ProLiant DL380 Gen10 (G10) on top of which IPWorks VNF is deployed.

Target Groups

This document is intended for personnel working with Ericsson IPWorks. The following prior knowledge is required:

- Intermediate Linux and UNIX skills
- Concepts, terminologies, and telecommunication abbreviations, such as TCP/IP, Packet Data Networks, and Protocol Servers

1.1 Related Information

Trademark information, typographic conventions, and definition and explanation of abbreviations and terminology can be found in the following documents:

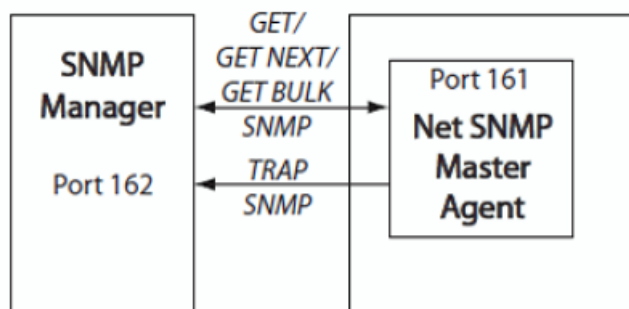
- Trademark Information, Reference [1]
- Typographic Conventions, Reference [2]
- Glossary of Terms and Acronyms, Reference [3]





2 Overview

The G9 and G10 host OS Fault Management is achieved through Net-SNMP. All alarms are forwarded to the SNMP Manager on the Network Management System (NMS) such as OSS-RC through the SNMP agents.



2.1 SNMP

2.1.1 SNMP Versions Supported by IPWorks

The G9 and G10 host OS Fault Management supports all SNMP versions: SNMPv1, SNMPv2, and SNMPv3. Enabling SNMP services between the G9/G10 nodes and the SNMP client requires the configuration of the G9/G10 nodes. The configuration varies for different SNMP versions.

SNMPv1 and SNMPv2c:

SNMP Management uses community-based access control for SNMPv1 and SNMPv2c. User can specify multiple community strings for read and write operations through the “rocommunity” and “rwcommunity” directives in the `/etc/ipworks/common/snmp/snmpd.conf` file.

These community strings only apply to SNMPv1 and SNMPv2c.

SNMPv3:

SNMPv3 provides much more sophisticated security mechanisms compared with SNMPv1 and SNMPv2c, implementing a User-based Security Model (USM) and a View-based Access Control Model (VACM). Section 3.1 on page 9 describes how to implement these security mechanisms in the SNMP configuration file `/etc/ipworks/common/snmp/snmpd.conf`.

IPWorks SNMPv3 supports:

- The MD5 and SHA protocols for authentication.



- Enhanced security features such as encryption with DES and AES privacy protocols. However, Statistics Collection supports only DES for encryption and does not support AES.

2.1.2 SNMP Master Agent

By default, the SNMP Master Agent listens on port 161 and the SNMP Master Agent is configurable according to either of the following ways:

- Listening on a different port by editing the `defaultport 161` directive in the `/etc/ipworks/common/snmp/snmp.conf` file.
- Listening on any domain, such as TCP and UDP, on any port of any address by editing the `agentaddress...` directive in the `/etc/ipworks/common/snmp/snmpd.conf` file.

By default, all the user configuration files for the SNMP Master Agent are stored in `/etc/ipworks/common/snmp` and all the configuration files for the SNMP system are stored in `/var/net-snmp`.

The SNMP Master Agent sends traps to any destination. To be able to send traps, user must configure it through directives “`trapsink`”, “`trap2sink`” and “`trapsess`” to send v1, v2c, and v3 version respectively in the file `/etc/ipworks/common/snmp/snmpd.conf`. For the detailed examples and descriptions about how to use those directives, refer to the `/etc/ipworks/common/snmp/snmpd.conf` file.

2.2 Alarms and Traps

The SNMP Master Agent supports the alternative methods of notifying events to an SNMP manager:

- Alarms defined in ERICSSON-ALARM-IRP-MIB

For more information on the traps and alarms, refer to [IPWorks Alarm List for DL380 Host Management, Reference \[4\]](#).

2.2.1 Alarms According to Ericsson Alarm IRP MIB

The SNMP Master Agent supports generation of alarms according to ERICSSON-ALARM-IRP-MIB to maintain active alarm information.

When the SNMP Master Agent is configured to generate alarms according to ERICSSON-ALARM-IRP-MIB, it performs the following operations:

- Maintains the state of all alarms in a table called `alarmTable`. This enables OSS node to obtain active alarm information from all IPWorks nodes and to synchronize the alarm states at any time.



- Supports alarm synchronization (see Section 4.1 on page 19), automatic alarm clearing, and manual alarm clearing (see Section 4.2 on page 20) for the generated alarms.

The SNMP Master Agent sends an `alarmNew` notification when an alarm is raised and an `alarmCleared` notification when an alarm is cleared. For example,

- If a network interface link downs, the SNMP Agent sends a notification with an `SNMP_TRAP_OID` of `alarmNew(.1.3.6.1.4.1.3881.2.2.0.1)`.
- If a network interface link ups, the SNMP Agent sends a notification with an `SNMP_TRAP_OID` of `alarmClear(.1.3.6.1.4.1.3881.2.2.0.3)`.

For all automatically cleared alarms, the same alarm is not sent with `snmpTrapOID` of `alarmNew` twice. That means, if a specific alarm is already in the active alarm table, the same alarm is ignored when it is raised again. The same alarm is not sent to the trap receiver before it is cleared.

The SNMP Master Agent supports the following alarm attributes, which are defined in `ERICSSON-ALARM-IRP-MIB.txt`:

- `notificationId`
- `alarmId`
- `alarmManagedObjectClass`
- `alarmManagedObjectInstance`
- `alarmEventTime`
- `alarmEventType`
- `alarmProbableCause`
- `alarmPerceivedSeverity`
- `alarmSpecificProblem`
- `alarmAdditionalText`

The SNMP Master Agent does not support the following alarm attributes defined in `ERICSSON-ALARM-IRP-MIB.txt`:

- `alarmAckUser`
- `alarmAckTime`
- `alarmCommentUser`
- `alarmCommentText`

To enable SNMP alarms, configure the appropriate trap destination or destinations as described in Section 3.2 on page 15.

**Note:**

- SNMPv1 trap does not fully support Ericsson IRP MIB because it does not support sysUpTime and snmpTrapOID.

2.3 Supported MIBs

Standard MIB Files:

- ERICSSON-ALARM-IRP-MIB.txt
- HOST-RESOURCES-MIB.txt
- IANAifType-MIB.txt
- IF-MIB.txt
- INET-ADDRESS-MIB.txt
- LM-SENSORS-MIB.txt
- SNMP-FRAMEWORK-MIB.txt
- SNMP-NOTIFICATION-MIB.txt
- SNMP-TARGET-MIB.txt
- SNMPv2-CONF.txt
- SNMPv2-MIB.txt
- SNMPv2-SMI.txt
- SNMPv2-TC.txt
- SNMPv2-TM.txt
- UCD-SNMP-MIB.txt
- NET-SNMP-AGENT-MIB.txt

For all the above MIB files except NET-SNMP-AGENT-MIB.txt, refer to the /opt/ipworks/IPWcommon/mibs directory. For the NET-SNMP-AGENT-MIB.txt file, refer to the /usr/share/snmp/mibs directory.

The SNMP Master Agent supports alarm synchronization, alarm clearing, and alarm standard alignment according to ITU standards.

For alarms, which cannot be cleared automatically, IPWorks supports manual alarm clearing.

OSS-RC can, at any time, read the active alarm information from all IPWorks nodes to synchronize the alarm state. IPWorks nodes provide this information by accessing it from the Alarm Table.

Table 1, Table 2, and Table 3 show which objects defined in IF-MIB, HOST-RESOURCES-MIB and UCD-SNMP-MIB are supported by the SNMP Master Agent.

Table 1 IF MIB

Object	OID	Supported
ifNumber.0	.1.3.6.1.2.1.2.1	Yes
ifTableLastChange	.1.3.6.1.2.1.31.1.5	No



Object	OID	Supported
ifTable	.1.3.6.1.2.1.2.2	Yes
ifXTable	.1.3.6.1.2.1.31.1.1	Yes
ifStackLastChange	.1.3.6.1.2.1.31.1.6	No
ifStackTable	.1.3.6.1.2.1.31.1.2	No
ifTestTable	.1.3.6.1.2.1.31.1.3	No
ifRcvAddressTable	.1.3.6.1.2.1.31.1.4	No

Table 2 HOST-RESOURCES-MIB

Object	OID	Supported
hrSystem	.1.3.6.1.2.1.25.1	Yes
hrMemorySize	.1.3.6.1.2.1.25.2.2	Yes
hrStorageTable	.1.3.6.1.2.1.25.2.3	Yes
hrDeviceTable	.1.3.6.1.2.1.25.3.2	Yes
hrProcessorTable	.1.3.6.1.2.1.25.3.3	Yes
hrNetworkTable	.1.3.6.1.2.1.25.3.4	Yes
hrPrinterTable	.1.3.6.1.2.1.25.3.5	Yes
hrDiskStorageTable	.1.3.6.1.2.1.25.3.6	Yes
hrPartitionTable	.1.3.6.1.2.1.25.3.7	Yes
hrFSTable	.1.3.6.1.2.1.25.3.8	Yes
hrSWOSIndex.0	.1.3.6.1.2.1.25.4.1	No
hrSWRunTable	.1.3.6.1.2.1.25.4.2	Yes
hrSWRunPerfTable	.1.3.6.1.2.1.25.5.1	Yes
hrSWInstalled.*.0	.1.3.6.1.2.1.25.6	Yes
hrSWInstalledTable	.1.3.6.1.2.1.25.6.3	Yes

Table 3 UCD-SNMP-MIB

Object	OID	Supported
prTable	.1.3.6.1.4.1.2021.2	No
Memory.*	.1.3.6.1.4.1.2021.4	Yes
extTable	.1.3.6.1.4.1.2021.8	No
dskTable	.1.3.6.1.4.1.2021.9	Yes
fileTable	.1.3.6.1.4.1.2021.15	Yes
laTable	.1.3.6.1.4.1.2021.10	Yes
systemStats.*.0	.1.3.6.1.4.1.2021.11	Yes
ucdDemoMIBObjects	.1.3.6.1.4.1.2021.2	Yes
logMatch	.1.3.6.1.4.1.2021.16	Yes
version.*.0	.1.3.6.1.4.1.2021.100	Yes



Object	OID	Supported
snmperrs.*.0	.1.3.6.1.4.1.2021.101	Yes
mrTable (D)	.1.3.6.1.4.1.2021.102	No

For more information on the mentioned objects, refer to:

- IF-MIB, Reference [8].
- HOST-RESOURCES-MIB, Reference [9].
- UCD-SNMP-MIB, Reference [10].



3 Enabling Fault Management

To enable the fault management for host OS, trap destinations must be configured for all SNMP versions. For SNMPv3, an SNMPv3 user must be created first.

SNMP Version	SNMPv1	SNMPv2	SNMPv3
Configuring Users	No	No	Yes
Configuring Trap Destinations	Yes	Yes	Yes

3.1 Configuring Users - SNMPv3

This section describes the user access configuration and testing of SNMPv3 interfaces in IPWorks. If SNMPv3 is used, SNMPv3 must be configured before enabling SNMP alarms and traps.

For SNMPv3 user configuration, the SNMP Master Agent uses two configuration files:

- The normal `/etc/ipworks/common/snmp/snmpd.conf` file

Note: This file does not exist until the SNMP Master Agent is started at least once.

- The persistent `/var/net-snmp/snmpd.conf` file

SNMPv3 has two types of users: read-write users (`rwuser`) and read-only users (`rouser`).

There are various ways to configure SNMPv3 users. Two methods are presented as follows:

- By modifying the Net-SNMP configuration files as described in Section 3.1.1 on page 9.
- By using the `net-snmp-config` tool as described in Section 3.1.2 on page 11.

Users can configure multiple users for SNMPv3 access using the procedures outlined earlier. Users can also create multiple users using the `snmpusm` utility as described in Section 3.1.3 on page 12.

Note: To use the DES privacy, users must install the OpenSSL library first.

3.1.1 Creating SNMPv3 Users Using SNMP Configuration Files

3.1.1.1 Creating a User

1. Stop the SNMP Master Agent.



2. Create a user named `ipworks` who has read-write access to the SNMP statistics.

```
# vi /etc/ipworks/common/snmp/snmpd.conf
```

Add the following line to the file:

```
rwuser <username> <authentication type>
```

For example,

```
rwuser ipworks authNoPriv
```

Note:

- The `authNoPriv` requests secure authentication, but not privacy. The SNMP packets themselves are not encrypted.
- The other possibilities are `noauthnopriv` (no authentication and no privacy) and `authpriv` (authentication and privacy).
- To create a user with read-only access, use the command `rouser` instead of `rwuser`. The rest of the procedure and parameters are the same as for `rwuser`.

Save the file and exit.

3. Create an MD5 password.

```
# vi /var/net-snmp/snmpd.conf
```

Add the following line to the file:

```
createUser <username> MD5 <password>
```

For example,

```
createUser ipworks MD5 mysecretpass
```

This creates an MD5 password (`mysecretpass`) for personnel using the username `ipworks`.

Save the file and exit.

4. Create a user with a DES passphrase in addition to an MD5 password.

```
# vi /var/net-snmp/snmpd.conf
```

Add the following line to the file:

```
createUser <username> MD5 <password> DES <passphrase>
```

For example,



```
createUser ipworks MD5 mysecretpass DES mypassphrase
```

Note:

- If users omit the DES mypassphrase, the Net-SNMP sets the DES passphrase to be the same as the MD5 password.
- The Net-SNMP requires the password and passphrase to be at least 8 characters long.
- When the SNMP Master Agent is started, it reads the configuration file, computes secret keys for the added users, and deletes the createUser commands from the file. It then places the secret key in the configuration file.
- This behavior has different consequences. The secret key is based on the engine ID, which for Net-SNMP is based on the IP address. Therefore, do not copy configuration files from one machine to another.

5. Start the SNMP Master Agent.

3.1.1.2

Reconfiguring Net-SNMP

If users change a machine's IP address, users must reconfigure Net-SNMP as follows:

1. Stop the SNMP Master Agent
2. Edit the /var/net-snmp/snmpd.conf file.
3. Delete any entries Net-SNMP has added for users.
4. Add createUser commands to create the users again.
5. Restart the SNMP Master Agent.

When the SNMP Master Agent is started, it reads the configuration file, computes secret keys for the added users, and deletes the createUser commands from the file. It then places the secret key in the configuration file.

3.1.2

Creating SNMPv3 Users Using net-snmp-config

To create a user using net-snmp-config:

1. Stop the SNMP Master Agent.
2. Use the net-snmp-config tool to create the user using either of the following commands:

```
# net-snmp-config --create-snmpv3-user -a <password> <username>
```

or:



```
# net-snmp-config --create-snmpv3-user
```

For example:

```
# net-snmp-config --create-snmpv3-user -a "admin1234" admin
```

If users use the second version of the command above, the `net-snmp-config` tool prompts users to enter the username, the authentication passphrase, and the encryption passphrase, one by one.

Note: The password must be at least eight characters long. The example creates the user `admin` with the password `admin1234` (and uses MD5 and DES for protection)

The `net-snmp-config` command adds the information for the created user to the `/var/net-snmp/snmpd.conf` file in encrypted form, and the information on users' read-write permissions to the `/etc/ipworks/common/snmp/snmpd.conf` file.

For example, the above command adds the line `createUser admin MD5 "admin1234" DES` in the encrypted form to the file `/var/net-snmp/snmpd.conf` and `rwuser admin` to file `/etc/ipworks/common/snmp/snmpd.conf`. This means that the created user has read-write permission. Users can change the user permissions by editing the file `snmpd.conf`.

3. Restart the SNMP Master Agent to let the changes take effect.

To create a user with read-only permissions, use the `net-snmp-config` command as follows:

```
# net-snmp-config --create-snmpv3-user -ro -A <authpass> -X  
<privpass> -a MD5 -x DES <username>
```

For example:

```
# net-snmp-config --create-snmpv3-user -ro -A admin1234 -X  
admin1234 -a MD5 -x DES admin
```

For more information on creating SNMPv3 users using `net-snmp-config`, enter `net-snmp-config /help`.

3.1.3 Creating Multiple SNMPv3 Users Using `snmpusm`

Users can use the `snmpusm` utility to configure multiple users based on the configuration of an existing user. It is achieved by modifying the user database using the SNMP protocol when the SNMP Master Agent is running. The `snmpusm` utility cannot create users from scratch.

To create the first user:

1. Stop the SNMP Master Agent.



2. Create a SNMPv3 user with read and write permissions. This is described in Section 3.1.1 on page 9.
3. Restart the SNMP Master Agent and test the setup.

```
# snmpget -v 3 -u ipworks -l authNoPriv -a MD5 -A my_password localhost sysUpTime.0
```

To create a second user:

1. Start the SNMP Master Agent if it is not running.
2. Clone a second user from the first user using the following command:

```
# snmpusm -v 3 -u ipworks -l authNoPriv -a MD5 -A my_password localhost create ipworks2 ipworks
```

The above command creates user `ipworks2` with the same parameters as the user `ipworks`.

3. Change the second user's password using the following command:

```
# snmpusm -v 3 -u ipworks2 -x DES -X my_passphrase -a MD5 -A my_password localhost passwd my_password new_password
```

Where:

`new_password` is the new password assigned to the second user `ipworks2`. However, one configuration line must be added into the file `/etc/ipworks/common/snmp/snmpd.conf` to allow the user to access SNMPv3 and restart the SNMP Master Agent after editing the `snmpd.conf` file.

For more information on `snmpusm` usage, refer to Net-SNMP Website, Reference [13].

3.1.4 Changing User Properties and Removing Users

- To change SNMP v3 security level for a user:
 - a First find the directive "rwuser" (or rouser) in `/etc/ipworks/common/snmp/snmpd.conf`, for example, `rwuser ipworks authNoPriv`.
 - b Change the security level `authNoPriv` (authentication and no Privacy) for the user "ipworks" to `noauthnopriv` (no authentication and no privacy) or `authpriv` (authentication and privacy). If no security level is specified, the default is `authNoPriv`.
- To change read-write access for a user:
 - a Find the directive "rwuser" (read-write user) or "rouser" (read-only user) in `/etc/ipworks/common/snmp/snmpd.conf` and make the change accordingly.



— To change user authentication and privacy password:

- For authentication password, use the following command:

```
# snmpusm -v 3 -l authPriv -u test1 -a MD5 -A admin1234 -x  
DES -X admin1235 -Ca localhost passwd admin1234 admin1111  
<user>
```
- For privacy password, use the following command:

```
# snmpusm -v 3 -l authPriv -u test1 -a MD5 -A admin1234 -x  
DES -X admin1235 -Cx localhost passwd admin1235 admin2222  
<user>
```

The user "ipworks" is one of the SNMP v3 users that has already been correctly configured and has the write access.

If the <user> is omitted, by default it changes the password of the user specified by the option "-u".

— To remove user "test":

- a Use the following command to remove the user "test".

```
# snmpusm -v 3 -u ipworks -l \  
authNoPriv -a MD5 -A my_password localhost delete test
```

The user "ipworks" is one of the SNMP v3 users that has already been correctly configured and has the write access.

Note:

- When the snmpusm command is used, users must start the SNMP Master Agent first.
- If users want to create a user and change the password through an existing user, the SNMP Master Agent only need to be restarted once. If users want to create a user and change the password through the new user, the SNMP Master Agent needs to be restarted twice because the new user will only be activated after the SNMP Master Agent is restarted.

3.1.5 Testing SNMPv3 Configuration

Users can test the SNMPv3 configuration to verify that the user has obtained results with the specified username and authentication parameters.

Use the following command to verify SNMPv3 configuration.

```
# snmpget -v <version> -u <user> -l <seclevel> -a <authProtocol>  
-A <authPassword> -x <privProtocol> -X <privPassword> <hostname>  
<objectOid or objectname>
```

For example,



```
# snmpget -v 3 -u ipworks -l authNoPriv -a MD5 -A mysecretpass
localhost sysUpTime.0
```

To give a user privacy (encryption) in addition to authentication, use the following command:

```
# snmpget -v 3 -u ipworks -l authPriv -a MD5 -A mysecretpass -x DES
-X mypassphrase localhost sysUpTime.0
```

3.2 Configuring Trap Destinations

This section describes how to configure the destinations for traps and alarms using one or more directives in the `/etc/ipworks/common/snmp/snmpd.conf` file. For more information on SNMP configuration, refer to the Net-SNMP documents in Net-SNMP README files, Reference [14].

Note:

- After editing the `/etc/ipworks/common/snmp/snmpd.conf` file, users must restart the SNMP Master Agent.

```
# /etc/init.d/ipworks.snmpd stop
```

```
# /etc/init.d/ipworks.snmpd start
```

3.2.1 Sending SNMPv1 Traps

Use the `trapsink` directive to send SNMPv1 traps to the SNMP Manager.

The following entry in the `/etc/ipworks/common/snmp/snmpd.conf` file defines the hosts that are to receive SNMPv1 traps or alarms:

```
trapsink <HOST> [<COMMUNITY> [<PORT>]]
```

Where:

<HOST> is either the IP address or the hostname of the machine to which the traps or alarms are to be sent.

Users can specify multiple `trapsink` lines to specify multiple destinations. If users do not specify the <COMMUNITY>, the SNMP Master Agent uses the community string from a preceding trap directive. If users do not specify the <PORT>, the SNMP Master Agent sends traps to the default SNMP trap port (162). The default destination is `localhost`.

For example: Either of the following directives causes the SNMP Master Agent to send SNMPv1 traps to the host with IP address “100.0.0.2” and hostname “ipworks03”:

```
trapsink 100.0.0.2 public 162
or
```



```
trapsink ipworks03 public 162
```

3.2.2 Sending SNMPv2 Traps

Use the `trap2sink` directive to send SNMPv2 traps to the SNMP Manager.

The following entry in the `/etc/ipworks/common/snmp/snmpd.conf` file defines the hosts that are to receive SNMPv2 traps or alarms:

```
trap2sink <HOST> [<COMMUNITY> [<PORT>]]
```

Where:

<HOST> is either the IP address or the hostname of the machine to which the traps or alarms are to be sent.

Users can specify multiple `trap2sink` lines to specify multiple destinations. If users do not specify the <COMMUNITY>, the SNMP Master Agent uses the community string from a preceding trap directive. If users do not specify the <PORT>, the SNMP Master Agent sends traps to the default SNMP trap port (162). The default destination is `localhost`.

For example: Either of the following directives causes the SNMP Master Agent to send SNMPv2 traps to the host with IP address “100.0.0.2” and hostname “ipworks03”:

```
trap2sink 100.0.0.2 public 162
or
trap2sink ipworks03 public 162
```

3.2.3 Sending SNMPv3 Traps

Simply speaking, SNMPv3 traps are SNMPv2 traps with added authentication and privacy capabilities. Use the `trapsess` directive to send SNMPv3 traps to the SNMP Manager. It is a more generic trap configuration token that allows any type of trap destination to be specified with any version of SNMP.

The following entry in the `/etc/ipworks/common/snmp/snmpd.conf` file defines the hosts that are to receive SNMPv3 traps or alarms:

```
trapsess -v <version> -e <localengineid> -u <user> -l <seclevel> -a
<authProtocol> -A <authPassword> <hostname>:<port>
```

Where:

<localengineid> is the authoritative engineId used for SNMPv3 Request messages.

Users can specify multiple `trapsess` lines to specify multiple destinations. If users do not specify the <port>, the SNMP Master Agent sends traps to the default SNMP trap port (162). The default trap destination is `localhost`.



For example:

```
trapssess -v 3 -e 0x0102030405 -u admin \
-l authNoPriv -a MD5 -A admin123 localhost:162
```

The SNMP Master Agent sends a `cold start` trap when it starts up. If enabled, it also sends traps on authentication failures.

3.2.4 Configuring Trap Handler

Trap Handler is a tool used to intercept the SNMP Traps from the SNMP Master Agent. The configuration is flexible. It can be configured to forward the SNMP Traps to another Trap Handler, or to the SNMP Manager. It is usually used to troubleshoot the alarm or trap sending problems, or when the alarms or traps need to be cleared manually.

For SNMPv1 and SNMPv2, Trap Handler does not require the configuration. It can be enabled simply by using command `/etc/init.d/ipworks.snmptrapd start`. For SNMPv3, Trap Handler must be configured with SNMPv3 user information to receive SNMPv3 traps.

SNMPv3 users for the Trap handler can be created by adding a `createUser` entry to file `/var/net-snmp/snmptrapd.conf`. This file will not be present immediately after fresh installation. This file is created automatically by `snmptrapd` when it is started for the first time.

SNMPv3 users for `snmptrapd` can be created using the following steps:

1. Stop any running `snmptrapd`.
2. Edit `/var/net-snmp/snmptrapd.conf` to insert the following line:

```
createUser -e 0x0102030405 myuser \
MD5 mypassword DES myotherpassword
```

This time we explicitly set the `engineID` of the user to `0x0102030405` (which technically is not a recommended value, but it really does not matter).

Where:

- `myuser` is the security name that is configured on `snmpd` to send SNMPv3 traps.
- `mypassword` is the authentication password.
- `myotherpassword` is the encryption password. It can be left blank if it is to be the same as the authentication password, or users do not want to use encryption.

3. Restart the `snmptrapd`.

Use the `snmptrap` command to send the trap demon a `coldStart v3 TRAP` message:



```
# snmptrap -e 0x0102030405 -v 3 \  
-u myuser -a MD5 -A mypassword -l authNoPriv \  
localhost 42 coldStart.0
```

Note:

- snmptrapd will not log SNMPv3 TRAPs sent by a user which has not been configured using the `createUser` directives as above. They are silently dropped by the snmptrapd program.
- SNMPv3 user information on Trap Handler must match with SNMPv3 user information created on the SNMP Master Agent to send v3 Traps.



4 Fault Management Activities

Fault Management activities include alarm synchronization, manual alarm clearing, notification threshold configuration, platform alarm configuration, and trap handler configuration.

4.1 Alarm Synchronization

IPWorks SNMP supports alarm synchronization. The SNMP Master Agent maintains all active alarm information in an alarm table. If the SNMP Manager on the NMS is shut down and then restarted, it still can retrieve the alarm information from the alarm table.

When an alarm is raised, the SNMP Master Agent sends an `alarmNew` notification to an SNMP Manager and adds the alarm to the alarm table.

When an alarm is cleared, the SNMP Master Agent removes the alarm from the alarm table and thus the alarm is no longer accessible. Meanwhile the Master Agent sends an `alarmCleared` notification to the SNMP Manager with the same alarm id as the `alarmNew` notification in the `var-bind` list. Therefore the alarm table always contains the list of alarms active on an IPWorks node at any point.

The SNMP Master Agent generates every notification with a unique `notificationId` and maintains a unique `alarmId` for all new alarms raised. The SNMP Master Agent maintains the same `alarmId` for both the `alarmNew` and its corresponding `alarmCleared` notification.

For example, Table 4 shows an example of an active alarm table with interface down alarm.

Table 4 Sample Alarm Information

<code>notificationId</code>	1
<code>alarmId</code>	1
<code>alarmManagedObjectClass</code>	<code>ipworksDisman</code>
<code>alarmManagedObjectInstance</code>	<code>DN_Prefix=DC=ipworks.com,g3SubNetwork=US,Node_Distinguished_Name=cluster3-b-1,interface=3</code>
<code>alarmEventTime</code>	<code>2017-2-9,3:34:33.0,-5:0</code>
<code>alarmEventType</code>	<code>communicationsAlarm(2)</code>
<code>alarmProbableCause</code>	<code>x733CommunicationsProtocolError(305)</code>
<code>alarmPerceivedSeverity</code>	<code>major(2)</code>
<code>alarmSpecificProblem</code>	<code>linkDown,communication link failure</code>

The alarm table is stored in a persistent file (on disk) to retain the alarm table between SNMP Agent reboots. The persistent file is named `as/var/net-snmp/ipworks_snmpdata.conf`.



To ensure that the persistent file is not deleted, use the `noreset` argument when executing the `/etc/init.d/ipworks.snmpd` script from the command-line prompt:

```
# /etc/init.d/ipworks.snmpd start noreset
```

The OS reboot always start the SNMP Master Agent without deleting the persistent files.

To start SNMP Master Agent without retaining the previous raised alarms, execute the `/etc/init.d/ipworks.snmpd` script from the command line without `noreset` parameter:

```
# /etc/init.d/ipworks.snmpd start
```

4.2 Manual Alarm Clearing

IPWorks SNMP provides the function to clear the alarms from the alarm table manually.

4.2.1 Prerequisites

To clear alarms manually, the `snmpd` and `snmptrapd` applications must be running.

The definitions for these applications are as follows:

snmpd `snmpd` is an SNMP Agent which binds to a port and awaits requests from SNMP management software.

snmptrapd `snmptrapd` is an SNMP application that receives traps (trap receiver) and forwards them to the SNMP Master Agent.

`snmpd` and `snmptrapd` in IPWorks are the SNMP Master Agent and Trap Handler respectively. Users can start the two by executing the command `/etc/init.d/ipworks.snmpd start noreset` and `/etc/init.d/ipworks.snmptrapd start`. For SNMPv3, additional configuration on Trap Handler is required. See Section 3.2.4 Configuring Trap Handler on page 17 for more information.

4.2.2 Manual Alarms Clearing Using `snmptrap`

`snmptrap` is an SNMP application that forms and sends an SNMP TRAP message to a trap receiver.

4.2.2.1 Manual Alarms Clearing for SNMPv2c

For SNMPv2c, the syntax to clear an alarm entry manually is shown as follows:



```
# snmptrap -v 2c [<COMMON OPTIONS>] uptime <trap-oid> [<OID>
<TYPE> <VALUE>]...
```

Where:

- <Common Options>: See Linux Man Page.
- uptime: a system command to obtain the system time.

If the user uses two single quotes ‘ ‘ in place of the uptime string, snmptrap inserts the current system time into the trap.

- <trap-oid>: the SNMP_TRAP_OID of the alarmCleared notification defined in ERICSSON-ALARM-IRP-MIB.txt.
- <OID> <TYPE> <VALUE>:
 - The object identifiers (OIDs) alarmId, alarmManagedObjectClass, and alarmPerceivedSeverity including their types and values must be specified as inputs to the snmptrap command.
 - The alarmId value should be set as the alarmId value of the row to be removed from the alarmTable and managedObjectClass refers to the managedObjectClass of that particular alarm entry.
 - The alarmPerceivedSeverity value is always 5 for manual alarm clearing.

For example,

To clear an alarm Entry for the Link Down with alarmId 5, follow the instructions as follows:

```
# snmptrap -v 2c -c public localhost 42 .1.3.6. 1.4.1.3881.2.2.0.3
.1.3.6.1.4.1.3881.2.1.8.1.1 i 5 .1.3.6.1.4.1.3881.2.1.8.1.2 s
"ipworksDisman" .1.3.6.1.4.1.3881.2.1.8.1.7 i 5
```

Where:

- .1.3.6.1.4.1.3881.2.2.0.3 is the snmpTrapOID for alarmCleared notification.
- .1.3.6.1.4.1.3881.2.1.8.1.1 is the OID for the attribute alarmId.
- .1.3.6.1.4.1.3881.2.1.8.1.2 is the OID for the attribute managedObjectClass.
- .1.3.6.1.4.1.3881.2.1.8.1.7 is the OID for the attribute alarmPerceivedSeverity.
- i stands for integer type.
- s stands for string type.



4.2.2.2 Manual Alarms Clearing for SNMPv3

For SNMPv3, the syntax to clear an alarm entry manually is shown as follows:

```
# snmptrap -e <localengineID> -v 3 -u <user> -a <authentication-protocol> -A <pass-phrase> localhost <uptime>
.1.3.6.1.4.1.3881.2.2.0.3 .1.3.6.1.4.1.3881.2.1.8.1.1 i
<alarmId> .1.3.6.1.4.1.3881.2.1.8.1.2 s "<managedObjectClass>"
.1.3.6.1.4.1.3881.2.1.8.1.7 i 5
```

For example:

To clear an alarm entry 4 using SNMPv3 for user ipworksuser:

```
# snmptrap -e 0x0102030405 -v 3 -u ipworksuser -a MD5 -A
"ipworksuser123" localhost 42 .1.3.6.1.4.1.3881.2.2.0.3
.1.3.6.1.4.1.3881.2.1.8.1.1 i 4 .1.3.6.1.4.1.3881.2.1.8.1.2 s
"ipworksDisman" .1.3.6.1.4.1.3881.2.1.8.1.7 i 5
```

4.2.3 Using alarmviewer Tool to clear Alarms Manually

Compared with the snmptrap, the alarmviewer tool is more convenient for the users to clear alarms manually.

To clear alarms using the alarmviewer:

1. List all the alarms.

```
# alarmviewer
```

```
registered debug token dump, 1
=====all alarm=====
ID TIMESTAMP SEVERITY Description
1 2017-2-22 1:28:42.0 major An interface is down
2 2017-2-22 1:28:42.0 major An interface is down
3 2017-2-22 1:28:43.0 major lmFanSensorsValue.1 threshold limit exceeded
=====
```

2. Follow the output instruction to enter the alarmID(s) and proceed with **Enter**.
For example,

```
Enter alarmID list you want to delete
0-----Exit
a-----All
alarmID list must delimits each alarmID using comma, such as '3,4,12'
:2
Are you sure that related status of these alarm have been ok and clean these alarm:2 !!
(no-cancel, yes-continue): yes
```

3. Confirm that the alarm is cleared.

```
# alarmviewer
```

For example, the following output shows the alarmID 2 is deleted.



```

registered debug token dump, 1
=====all alarm=====
ID TIMESTAMP SEVERITY Description
1 2017-2-22 1:28:42.0 major An interface is down
3 2017-2-22 1:28:43.0 major lmFanSensorsValue.1 threshold limit exceeded
=====

Enter alarmID list you want to delete
0-----Exit
a-----All
alarmID list must delimits each alarmID using comma, such as '3,4,12'
:

```

Enter 0 to exit.

4.3 Platform Alarm Configuration

The SNMP Master Agent is able to generate a few traps itself. The Master Agent checks its own data at regular intervals and sends out traps when certain conditions occur, such as “disk is full”, or “CPU is overloaded” or “the selected process is not running”. It sends these traps to SNMP managers specified in the `snmpd.conf` file.

For example: when starting up, the SNMP Master Agent generates an `SNMPv2-MIB::coldStart` trap, and when shutting down, it generates a `UCD-SNMP-MIB::ucdShutDown` trap.

IPWorks SNMP supports the following general platform alarms.

- Disk usage is more than the configured threshold (Disk Space Utilization)
- System Load Monitoring is more than the configured value (CPU Utilization)
- Memory Utilization goes beyond the configured threshold
- Network Interface UP/DOWN conditions
- Network Traffic Load
- Environment variables (Temperature, Fan Status, Power Supply) exceed a certain limit or are not within the predefined scope.

The SNMP Master Agent sends the traps when it first detects these conditions and not during each evaluation.

4.3.1 Disk Usage Monitoring

The SNMP Master Agent can raise an alarm for a system on which IPWorks is installed when the available disk space falls below a certain percentage (threshold value). Currently, SNMP only monitors the Complete Physical Disk, not slices and partitions. This value can be configured by including directives of the following form in the SNMP Master Agent configuration file `/etc/ipworks/common/snmp/snmpd.conf`:



```
disk <PATH> [<MINPERCENT>%]
```

Where <PATH> is the pathname of any of the disk partitions and <MINPERCENT> is the percentage of available disk space below which the alarm is raised.

For example:

```
disk / 15%
```

The SNMP Master Agent checks the disk mounted at each <PATH> for available disk space. If the disk space available is less than the configured <MINPERCENT> (%), the SNMP Master Agent sends an alarm to the configured SNMP managers.

In the example above, the SNMP Master Agent raises an alarm if the free disk space left on any one of the listed partitions falls below 15%.

Use the following directive to specify all disks available on the system:

```
includeAllDisks <MINPERCENT>%
```

Note: The `includeAllDisks` directive only includes the disks that are mounted when the `snmpd` daemon is started. It cannot include disks that are dynamically loadable, such as with `automount`. Mount all the disks that must be monitored before starting the `snmpd` daemon.

There can only be one `includeAllDisks` directive in the configuration file. This can be used with the `disk` directive. The threshold value given in a `disk` directive overrides the threshold value specified by the `includeAllDisks` directive, irrespective of the order in which the directives appear in the configuration file.

To send an alarm when the disk is full, the SNMP Master Agent uses `/opt/ipworks/IPWcommon/mibs/UCD-SNMP-MIB.txt`. For more information on these two MIB files, refer to UCD-SNMP-MIB, Reference [10]. These two files are installed with IPWorks by default. These alarms are always enabled by default. User can disable them by commenting out the line as follows in the file `/etc/ipworks/common/snmp/snmpd.conf`:

```
monitor -u cjslyxz -r 600 -o dskPath -o dskErrorMsg "dskTable"
dskErrorFlag ! =0
```

Tip: When the occupied disk space is released and the available disk space is more than the configured threshold value, the raised alarm can be automatically cleared.

4.3.2 System Load Monitoring

The IPWorks SNMP Master Agent monitors the average load of the local system, specifying thresholds for the 1 minute, 5 minute and 15 minute averages. If any of these loads exceed the associated maximum value, the SNMP Master Agent sets the corresponding `laErrorFlag` instance to 1 and raises an alarm.



Tip: When the system average load drops below the configured threshold based on the specific period granularity, the raised alarm can be automatically cleared.

User can configure the value of load by including the following directive in the SNMP Master Agent configuration file `/etc/ipworks/common/snmp/snmpd.conf`:

```
load <MAX1> [<MAX5> [<MAX15>]]
```

— Configure the value as follows:

- For 1 minute, the average value: $N \times 1.6$
- For 5 minutes, the average value: $N \times 0.9$
- For 15 minutes, the average value: $N \times 0.7$

For easy use, round the average value to the nearest integer. N is the number of the CPUs, which can be checked using the following commands:

- Check CPU information.

```
#cat /proc/cpuinfo
```

- Check the number of CPUs.

```
#grep -c 'model name' /proc/cpuinfo
```

For example, if N is 8, configure the value as follows:

```
load 13 7 6
```

If user omits `<MAX15>`, the SNMP Master Agent uses `<MAX5>` as both the 5-minute threshold and the 15-minute threshold. If user omits both `<MAX5>` and `<MAX15>`, the SNMP Master Agent uses `<MAX1>` as the value of all three thresholds.

The following directive is also configured to raise an alarm when the CPU average loads exceed the configured thresholds:

```
monitor -u cjslyxz -o laNames -o laErrorMessage "laTable"
laErrorFlag !=0
```

4.3.3

Swap Space Monitoring

The following directive instructs the SNMP Master Agent to monitor the amount of swap space available on the local system. If this falls below the specified threshold (MIN KB), the SNMP Master Agent sets the `memErrorSwapobject` to 1, and raises an alarm.

```
swap <MIN>
```

For example:



```
swap 512000
```

In this example, the SNMP Master Agent raises an alarm if the available swap space falls below 512 MB.

The following directive is also configured to raise an alarm when the available swap space falls below the configured threshold:

```
monitor -u cjslyxz -o memErrorName -o memSwapErrorMsg "memory"  
memSwapError !=0
```

Tip: When the available swap space increases and is more than the configured threshold value, the raised alarm can be automatically cleared.

4.3.4 Memory Usage Monitoring

User can monitor the memory usage of any process using Net-SNMP by adding the following command to the SNMP configuration file `snmpd.conf`:

```
monitor -u cjslyxz -o hrSWRunName \  
"high process memory" hrSWRunPerfMem > 40000000
```

If memory usage of any process exceeds the specified threshold (KB), the SNMP Master Agent raises an alarm.

Tip: When the memory usage drops below the configured threshold value, the raised alarm can be automatically cleared. However, a special case occurs that if you shut down the process related the raised alarm, when restarting the IPWorks server that raised the original alarm, even the memory usage is lower than the configured threshold, the original raised alarm cannot be automatically cleared.

4.3.5 Link up and Link down Notifications

The IPWorks SNMP Master Agent supports the linkup and linkdown notifications for network interfaces being taken up and down.

According to the default configuration, IPWorks SNMP monitors the specific Ethernet interface. Fixed OID corresponds to the specific Ethernet interface. For more information, see Section 4.3.8 Fixed OID for Ethernet Interface on page 31.

For example:

Fixed OID	Representative Ethernet Interface
ifIndex.800	eth0
ifIndex.801	eth1
ifIndex.802	eth2
...	...



Fixed OID	Representative Ethernet Interface
ifIndex.820	bond0
ifIndex.821	bond1

The `snmpd.conf` is as below:

```
notificationEvent linkUpTrap linkUp
notificationEvent linkDownTrap linkDown
monitor -u cjslyxz -r 600 -e linkUpTrap -I -o ifIndex.800 "ifIndex.800 linkU
monitor -u cjslyxz -r 600 -e linkUpTrap -I -o ifIndex.801 "ifIndex.801 linkU
monitor -u cjslyxz -r 600 -e linkUpTrap -I -o ifIndex.802 "ifIndex.802 linkU
monitor -u cjslyxz -r 600 -e linkUpTrap -I -o ifIndex.803 "ifIndex.803 linkU
monitor -u cjslyxz -r 600 -e linkUpTrap -I -o ifIndex.804 "ifIndex.804 linkU
monitor -u cjslyxz -r 600 -e linkUpTrap -I -o ifIndex.805 "ifIndex.805 linkU
monitor -u cjslyxz -r 600 -e linkUpTrap -I -o ifIndex.820 "ifIndex.820 linkU
monitor -u cjslyxz -r 600 -e linkUpTrap -I -o ifIndex.821 "ifIndex.821 linkU
monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.800 "ifIndex.800 lin
monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.801 "ifIndex.801 lin
monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.802 "ifIndex.802 lin
monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.803 "ifIndex.803 lin
monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.804 "ifIndex.804 lin
monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.805 "ifIndex.805 lin
monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.820 "ifIndex.820 lin
monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.821 "ifIndex.821 lin
```

If IPWorks is deployed with single DPDK, IPWorks SNMP monitors `eth0`, `eth1`, `eth2`, `eth3` and `bond0`, `bond1`. The following lines should be commented:

```
# monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.804 "ifIndex.804 1
# monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.805 "ifIndex.805 1
```

If IPWorks is deployed with dual DPDK, IPWorks SNMP monitors `eth0`, `eth1`, `eth4`, `eth5` and `bond0`, `bond1`. Lines including "ifIndex.802" or "ifIndex.803" should be commented:

```
# monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.802 "ifIndex.802 1
# monitor -u cjslyxz -r 600 -e linkDownTrap -I -o ifIndex.803 "ifIndex.803 1
```

If you want IPWorks SNMP to monitor all Ethernet cards, configure `snmpd.conf` as the following directives. This method is not recommended. Because it will raise fake alarms.

```
notificationEvent linkUpTrap linkUp ifIndex ifAdminStatus ifOperStatus
notificationEvent linkDownTrap linkDown ifIndex ifAdminStatus ifOperStatus
monitor -u cjslyxz -r 600 -e linkUpTrap "Generate linkUp" ifOperStatus !=2
monitor -u cjslyxz -r 600 -e linkDownTrap "Generate linkDown" ifOperStatus =
```



4.3.6 Interface Traffic Monitoring

User can monitor the interface traffic usage of any interface using SNMP Master Agent by adding the following command to the SNMP configuration file `snmpd.conf`:

```
monitor -u cjslyxz -D -r 600 -o ifDescr "network traffic"
ifHCInOctets 10000000 2000000000
```

If the total number of octets received on any interface (delta differences between sample values) rises above the upper threshold (50000000), then the SNMP Master Agent raises an alarm. If the total number of octets received on any interface (delta differences between sample values) falls below the lower threshold (10000000), the raised alarm is automatically cleared.

Note: Ensure that adjust the threshold value based on the actual network station.

4.3.7 DDPK Status Monitoring

The IPWorks SNMP Master Agent supports to monitor the status for DDPK interfaces.

1. Fetching the status for DDPK interface.
 - a. Modify the directives of the following form in the SNMP Master Agent configuration file `/etc/ipworks/common/snmp/snmpd.conf`.

```
pass_persist .1.3.6.1.2.1.2.2.1.1.<INFID> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.2.<INFID> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.8.<INFID> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.1.<INFID+1> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.2.<INFID+1> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.8.<INFID+1> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.1.<INFID+2> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.2.<INFID+2> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.8.<INFID+2> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.1.<INFID+3> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.2.<INFID+3> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.8.<INFID+3> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.1.<INFID+4> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.2.<INFID+4> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
pass_persist .1.3.6.1.2.1.2.2.1.8.<INFID+4> /opt/ipworks/IPWcommon/scripts/dpdkStatus.py [<INFID>]
```

Interface ID	DDPK Interfaces
<INFID+1>	Dpdk0
<INFID+2>	Dpdk1
<INFID+3>	Dpdk2
<INFID+4>	Dpdk3



Note: <INFID+N> is a custom interface ID of DPDK. The default value of <INFID> is 700 . Make sure that the value is bigger than the max real interface ID which can be got by below commands:

```
# ip a | grep -v -E "^ " | awk -F ':' 'BEGIN {max = 0} {if ($1>max) max=$1} END {print "Max=", max}'
```

For example, in /etc/ipworks/common/snmp/snmpd.conf

```
pass_persist .1.3.6.1.2.1.2.2.1.1.700 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.2.700 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.8.700 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.1.701 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.2.701 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.8.701 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.1.702 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.2.702 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.8.702 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.1.703 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.2.703 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.8.703 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.1.704 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.2.704 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
pass_persist .1.3.6.1.2.1.2.2.1.8.704 /opt/ipworks/IPWcommon/scripts/dpdkStatus.py
```

- b. Fetch the values of ifIndex, ifDescr and ifOperStatus for DPDK interfaces by snmpwalk or snmpget in IPWorks SNMP Master Agent.

```
#snmpwalk -v 2c -c public localhost ifIndex
#snmpwalk -v 2c -c public localhost ifDescr
#snmpwalk -v 2c -c public localhost ifOperStatus
#snmpget -v 2c -c public localhost ifIndex.<INFID+N>
#snmpget -v 2c -c public localhost ifDescr.<INFID+N>
#snmpget -v 2c -c public localhost ifOperStatus.<INFID+N>
```

For example:

```
#snmpwalk -v 2c -c public localhost ifIndex
```

```
registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
.1.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
.1.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
.1.3.6.1.2.1.2.2.1.1.6 = INTEGER: 6
.1.3.6.1.2.1.2.2.1.1.8 = INTEGER: 8
.1.3.6.1.2.1.2.2.1.1.701 = INTEGER: 701
.1.3.6.1.2.1.2.2.1.1.702 = INTEGER: 702
.1.3.6.1.2.1.2.2.1.1.703 = INTEGER: 703
.1.3.6.1.2.1.2.2.1.1.704 = INTEGER: 704
```

```
#snmpwalk -v 2c -c public localhost ifDescr
```



```

registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.2.1 = STRING: lo
.1.3.6.1.2.1.2.2.1.2.2 = STRING: eth0
.1.3.6.1.2.1.2.2.1.2.4 = STRING: eth1
.1.3.6.1.2.1.2.2.1.2.6 = STRING: eth2
...
.1.3.6.1.2.1.2.2.1.2.701 = STRING: dpdk0
.1.3.6.1.2.1.2.2.1.2.702 = STRING: dpdk1
.1.3.6.1.2.1.2.2.1.2.703 = STRING: dpdk2
.1.3.6.1.2.1.2.2.1.2.704 = STRING: dpdk3
....

#snmpwalk -v 2c -c public localhost ifOperStatus

registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.8.1 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.2 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.4 = INTEGER: up(1)
...
.1.3.6.1.2.1.2.2.1.8.701 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.702 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.703 = INTEGER: down(2)
.1.3.6.1.2.1.2.2.1.8.704 = INTEGER: up(1)

#snmpget -v 2c -c public localhost ifIndex.701

registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.1.701 = INTEGER: 701

#snmpget -v 2c -c public localhost ifDescr.701

registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.2.701 = STRING: dpdk0

#snmpget -v 2c -c public localhost ifOperStatus.701

...
registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.8.701 = INTEGER: up(1)

```

2. Raises link up and link down notifications for the DPDK interfaces as Ethernet cards in the SNMP Agent. For more information, refer to Section 4.3.5 on page 26.

For example:

— IPWorks SNMP monitors all Ethernet cards (including DPDK):



```
#/etc/ipworks/common/snmp/snmpd.conf
notificationEvent linkUpTrap linkUp ifIndex ifAdminStatus ifOperStatus
notificationEvent linkDownTrap linkDown ifIndex ifAdminStatus ifOperStatus
monitor -u cjslyxz -r 600 -e linkUpTrap "Generate linkUp" ifOperStatus !=2
monitor -u cjslyxz -r 600 -e linkDownTrap "Generate linkDown" ifOperStatus ==2
```

— IPWorks SNMP only monitors monitor dpdk0:

- a Execute the following command to show all the interfaces in driver.

```
#snmpwalk -v 2c localhost ifDescr
1.3.6.1.2.1.2.2.1.2.701 = STRING: dpdk0
.1.3.6.1.2.1.2.2.1.2.702 = STRING: dpdk1
.1.3.6.1.2.1.2.2.1.2.703 = STRING: dpdk2
.1.3.6.1.2.1.2.2.1.2.704 = STRING: dpdk3
```

- b Edit the SNMP configuration file as follows:

```
notificationEvent dpdk0_linkUpTrap linkUp ifIndex.701 ifOperStatus.701 ifDescr.701
notificationEvent dpdk0_linkDownTrap linkDown ifIndex.701 ifOperStatus.701 ifDescr.701
monitor -u cjslyxz -t -r 600 -e dpdk0_linkUpTrap "dpdk0 linkUp" -I ifOperStatus.701 != 2
monitor -u cjslyxz -t -r 600 -e dpdk0_linkDownTrap "dpdk0 linkDown" -I ifOperStatus.701 == 2
```

4.3.8 Fixed OID for Ethernet Interface

To ensure hardware alarms can be raised and cleared correctly, fixed OID is used to correspond to the specific Ethernet interface.

1. Fetch the status for Ethernet interface with fixed OID.
 - a. Modify the directives of the following form in the SNMP Master Agent configuration file /etc/ipworks/common/snmp/snmpd.conf.

```
pass_persist .1.3.6.1.2.1.2.2.1.1.<IPWIFID-1> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.2.<IPWIFID-1> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.8.<IPWIFID-1> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.1.<IPWIFID> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.2.<IPWIFID> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.8.<IPWIFID> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.1.<IPWIFID+1> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.2.<IPWIFID+1> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.8.<IPWIFID+1> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.1.<IPWIFID+2> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.2.<IPWIFID+2> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.8.<IPWIFID+2> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.1.<IPWIFID+3> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.2.<IPWIFID+3> /opt/ipworks/IPWcommon
```



```

pass_persist .1.3.6.1.2.1.2.2.1.8.<IPWIFID+3> /opt/ipworks/IPWcommon
...
pass_persist .1.3.6.1.2.1.2.2.1.1.<IPWIFID+19> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.2.<IPWIFID+19> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.8.<IPWIFID+19> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.1.<IPWIFID+20> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.2.<IPWIFID+20> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.8.<IPWIFID+20> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.1.<IPWIFID+21> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.2.<IPWIFID+21> /opt/ipworks/IPWcommon
pass_persist .1.3.6.1.2.1.2.2.1.8.<IPWIFID+21> /opt/ipworks/IPWcommon

```

Interface ID	Ethernet Interfaces
<IPWIFID+0>	eth0
<IPWIFID+1>	eth1
<IPWIFID+2>	eth2
<IPWIFID+3>	eth3
...	...
<IPWIFID+20>	bond0
<IPWIFID+21>	bond1

Note:

- <IPWIFID+N> is a custom interface ID of fixed OID. The default value of <IPWIFID> is 800. Make sure that this value is bigger than the max real interface ID which can be got by below commands:

```
# ip a | grep -v -E "^ " | awk -F ':' 'BEGIN {max = 0} {if ($1>max) max=$1} END {print "Max=", max}'
```

- The value of <IPWIFID> cannot be same as that of <INFID>.

For example, in /etc/ipworks/common/snmp/snmpd.conf.

```

pass_persist .1.3.6.1.2.1.2.2.1.1.799 /opt/ipworks/IPWcommon/scripts
pass_persist .1.3.6.1.2.1.2.2.1.2.799 /opt/ipworks/IPWcommon/scripts
pass_persist .1.3.6.1.2.1.2.2.1.8.799 /opt/ipworks/IPWcommon/scripts
pass_persist .1.3.6.1.2.1.2.2.1.1.800 /opt/ipworks/IPWcommon/scripts
pass_persist .1.3.6.1.2.1.2.2.1.2.800 /opt/ipworks/IPWcommon/scripts
pass_persist .1.3.6.1.2.1.2.2.1.8.800 /opt/ipworks/IPWcommon/scripts
pass_persist .1.3.6.1.2.1.2.2.1.1.801 /opt/ipworks/IPWcommon/scripts
pass_persist .1.3.6.1.2.1.2.2.1.2.801 /opt/ipworks/IPWcommon/scripts
pass_persist .1.3.6.1.2.1.2.2.1.8.801 /opt/ipworks/IPWcommon/scripts
pass_persist .1.3.6.1.2.1.2.2.1.1.802 /opt/ipworks/IPWcommon/scripts
pass_persist .1.3.6.1.2.1.2.2.1.2.802 /opt/ipworks/IPWcommon/scripts

```



```

pass_persist .1.3.6.1.2.1.2.2.1.8.802 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.1.803 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.2.803 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.8.803 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.1.804 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.2.804 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.8.804 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.1.805 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.2.805 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.8.805 /opt/ipworks/IPWcommon/scrip

pass_persist .1.3.6.1.2.1.2.2.1.1.819 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.2.819 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.8.819 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.1.820 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.2.820 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.8.820 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.1.821 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.2.821 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.8.821 /opt/ipworks/IPWcommon/scrip

```

IPWorks supports monitoring Ethernet interface extend from eth0 to eth19. For example, to monitor eth6, add the below directives:

```

pass_persist .1.3.6.1.2.1.2.2.1.1.806 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.2.806 /opt/ipworks/IPWcommon/scrip
pass_persist .1.3.6.1.2.1.2.2.1.8.806 /opt/ipworks/IPWcommon/scrip

```

- b. Fetch the values of ifIndex, ifDescr and ifOperStatus for network interfaces by snmpwalk or snmpget in IPWorks SNMP Master Agent.

```

#snmpwalk -v 2c -c public localhost ifIndex
#snmpwalk -v 2c -c public localhost ifDescr
#snmpwalk -v 2c -c public localhost ifOperStatus
#snmpget -v 2c -c public localhost ifIndex.<IPWIFID+N>
#snmpget -v 2c -c public localhost ifDescr.<IPWIFID+N>
#snmpget -v 2c -c public localhost ifOperStatus.<IPWIFID+N>

```

For example:

```

#snmpwalk -v 2c -c public localhost ifIndex
registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
.1.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
...
.1.3.6.1.2.1.2.2.1.1.800 = INTEGER: 800
.1.3.6.1.2.1.2.2.1.1.801 = INTEGER: 801
.1.3.6.1.2.1.2.2.1.1.802 = INTEGER: 802
.1.3.6.1.2.1.2.2.1.1.803 = INTEGER: 803

```



```
.1.3.6.1.2.1.2.2.1.1.804 = INTEGER: 804
.1.3.6.1.2.1.2.2.1.1.805 = INTEGER: 805
.1.3.6.1.2.1.2.2.1.1.820 = INTEGER: 820
.1.3.6.1.2.1.2.2.1.1.821 = INTEGER: 821

# snmpwalk -v 2c -c public localhost ifDesc
registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.2.1 = STRING: lo
.1.3.6.1.2.1.2.2.1.2.2 = STRING: eth8
...
.1.3.6.1.2.1.2.2.1.2.800 = STRING: eth0
.1.3.6.1.2.1.2.2.1.2.801 = STRING: eth1
.1.3.6.1.2.1.2.2.1.2.802 = STRING: eth2
.1.3.6.1.2.1.2.2.1.2.803 = STRING: eth3
.1.3.6.1.2.1.2.2.1.2.804 = STRING: eth4
.1.3.6.1.2.1.2.2.1.2.805 = STRING: eth5
.1.3.6.1.2.1.2.2.1.2.820 = STRING: bond0
.1.3.6.1.2.1.2.2.1.2.821 = STRING: bond1

snmpwalk -v 2c -c public localhost ifOperStatus
registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.8.1 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.2 = INTEGER: down(2)
...
.1.3.6.1.2.1.2.2.1.8.800 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.801 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.802 = INTEGER: down(2)
.1.3.6.1.2.1.2.2.1.8.803 = INTEGER: down(2)
.1.3.6.1.2.1.2.2.1.8.804 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.805 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.820 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.821 = INTEGER: up(1)

#snmpget -v 2c -c public localhost ifIndex.800
registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.1.800 = INTEGER: 800
#snmpget -v 2c -c public localhost ifDescr.800
registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.2.800 = STRING: eth0
#snmpget -v 2c -c public localhost ifOperStatus.800
...
registered debug token dump, 1
.1.3.6.1.2.1.2.2.1.8.800 = INTEGER: up(1)
```

2. Raise link up and link down notifications for the Ethernet interfaces with fixed OID in the SNMP Agent. For more information, refer to Section 4.3.5 on page 26.



4.3.9 Environment Variables Monitoring (LM-Sensors)

4.3.9.1 General

The Net-SNMP Master Agent monitors the environmental variables, such as Temperature, Fan Status and Power Supply, and raise alarms when they exceed the specified threshold or are not within the predefined scope.

Tip: When the temperature drops below the configured threshold value or the fan status returns to normal level, the raised alarm can be automatically cleared; When the power supply returns to the normal value that stays within the configured range, the raised alarm can be automatically cleared.

IPWorks uses the `ipmitool` API to get hardware Sensors names and values, and set them to the corresponding Net-SNMP LM-Sensors.

Note: Remember that before running the `ipmitool` command, the users must start the `ipmit` utility first by running command `/etc/init.d/ipmi start`.

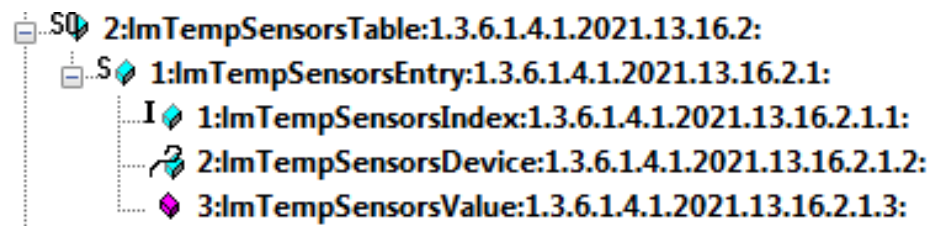
One `SensorsEntry` corresponds to one sensor. Each sensor has an index. For example,

— For the Temperature Sensors:

The Temperature sensors name is set to `lmTempSensorsDevice`.

The Temperature sensors value is set to `lmTempSensorsValue`.

The following screenshot is fetched from the NET-SNMP MIB File: `LM-SENSORS-MIB.txt`.



Use the `snmp` directive to check these values. For example,

```
# snmpwalk -v 2c localhost lmTempSensorsDevice
```

— For the Power Supply Sensors:

The Power Supply sensors name is set to `lmPSSensorsDevice`.

The Power Supply sensors value is set to `lmPSSensorsValue`.

The following screenshot is fetched from the NET-SNMP MIB File: `LM-SENSORS-MIB.txt`.



Use the snmp directive to check these values. For example,

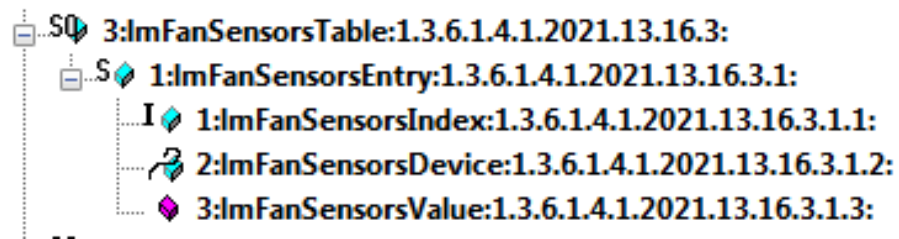
```
# snmpwalk -v 2c localhost lmPSSensorsDevice
```

— For the Fan Sensors:

The Fan sensors name is set to lmFanSensorsDevice.

The Fan sensors value is set to lmFanSensorsValue.

The following screenshot is fetched from the NET-SNMP MIB File: LM-SENSORS-MIB.txt.



Use the snmp directive to check these values. For example,

```
# snmpwalk -v 2c localhost lmFanSensorsDevice
```

4.3.9.2 Temperature Sensors Example

4.3.9.2.1 Fetching the Current Values

Execute the following command to fetch the current temperature values:

```
# ipmitool -v sdr type Temperature
```

For example,

```
Sensor ID           : Temp 1 (0xd)
Entity ID           : 39.1 (External Environment)
Sensor Type (Analog) : Temperature
Sensor Reading      : 21 (+/- 0) degrees C
Status              : ok
Positive Hysteresis  : Unspecified
Negative Hysteresis  : Unspecified
```




```
Minimum sensor range : -127.000
Maximum sensor range : Unspecified
Event Message Control : Entire Sensor Only
Readable Thresholds   : ucr unr
Settable Thresholds   :
Threshold Read Mask   : ucr unr
...
```

4.3.9.2.2 Checking the Thresholds

Log on to iLO platform to get the temperature thresholds:

For G9:

Select **Information > System Information > Temperatures**.

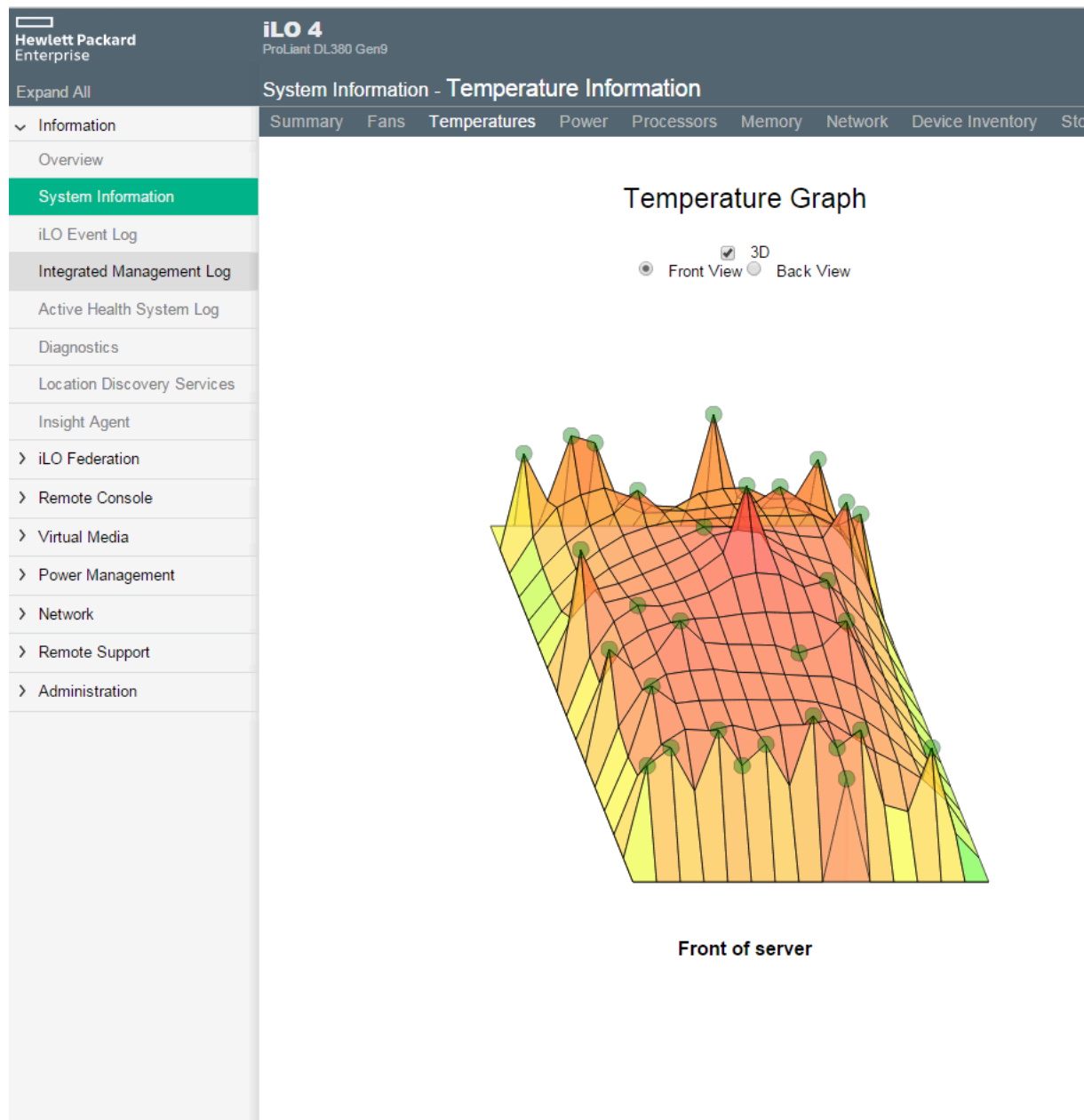


Figure 1 Get Temperature Thresholds on G9
For G10

Select **Power & Thermal > Temperatures**.

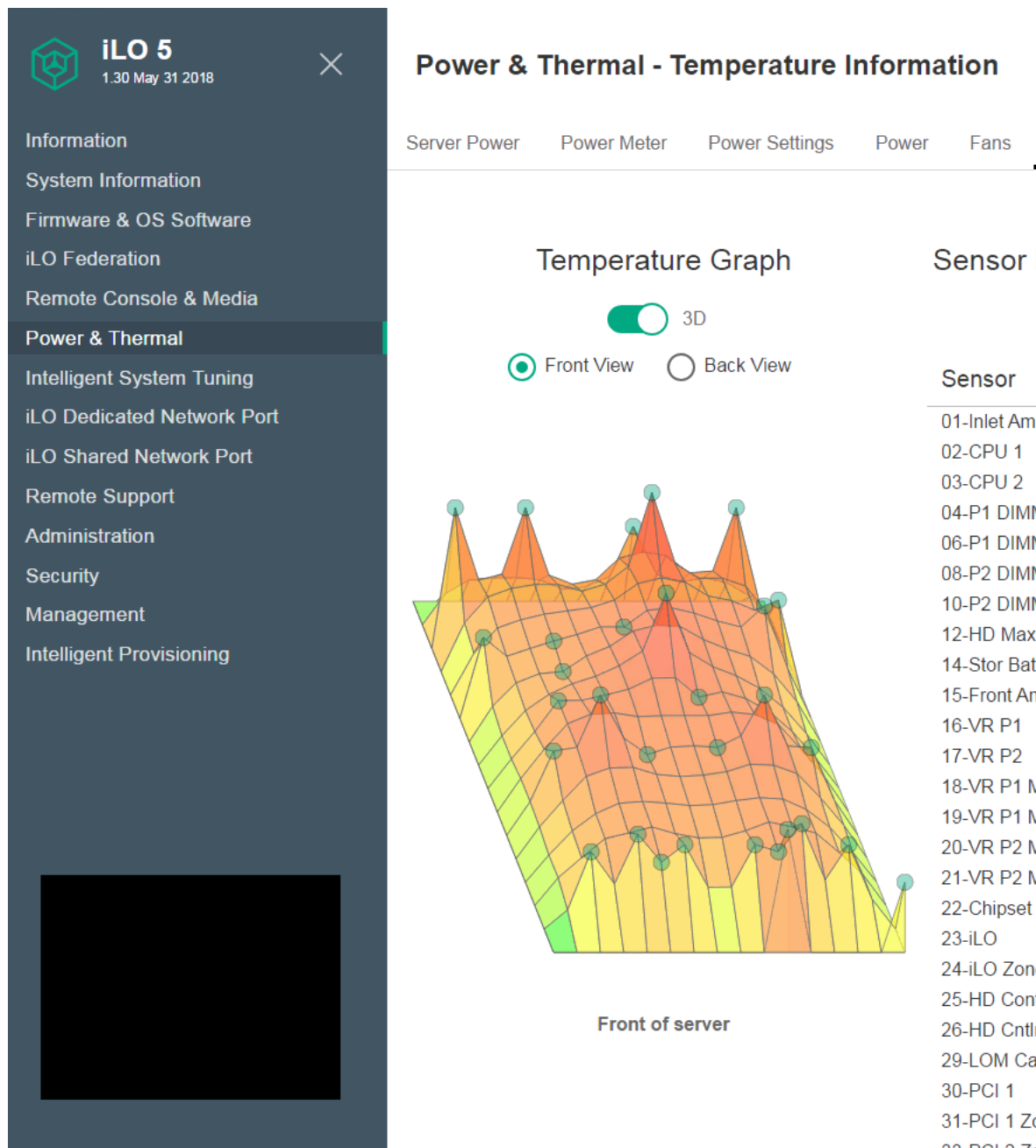


Figure 2 Get Temperature Thresholds on G10

4.3.9.2.3 Checking Alarm Details When an Alarm is Raised

If an alarm is raised, use the following command to check the alarm details:

```
# snmpwalk -v 2c localhost alarmtable
```

Follow the procedure to set the threshold and clear the alarm:



```

registered debug token dump, 1
.1.3.6.1.4.1.3881.2.1.8.1.1.2 = INTEGER: 2
.1.3.6.1.4.1.3881.2.1.8.1.2.2 = STRING: "ipworksDisman"
.1.3.6.1.4.1.3881.2.1.8.1.3.2 = STRING:
"DN_Prefix:DC=ipworks.com,g3SubNetwork=US,Node_Distinguished_Name=ipwm10ss-01"
.1.3.6.1.4.1.3881.2.1.8.1.4.2 = STRING: 2013-8-6,21:8:29.0,+8:0
.1.3.6.1.4.1.3881.2.1.8.1.5.2 = INTEGER: environmentalAlarm(3)
.1.3.6.1.4.1.3881.2.1.8.1.6.2 = INTEGER: x733ThresholdCrossed(351)
.1.3.6.1.4.1.3881.2.1.8.1.7.2 = INTEGER: major(2)
.1.3.6.1.4.1.3881.2.1.8.1.8.2 = STRING: "Temperature Level Threshold Reached"
.1.3.6.1.4.1.3881.2.1.8.1.9.2 = ""
.1.3.6.1.4.1.3881.2.1.8.1.10.2 = STRING:
.1.3.6.1.4.1.3881.2.1.8.1.11.2 = ""
.1.3.6.1.4.1.3881.2.1.8.1.12.2 = ""
.1.3.6.1.4.1.3881.2.1.8.1.13.2 = STRING: "lmTempSensorsvalue.1 threshold limit exceeded "

```

4.3.9.2.4 Configuring the Thresholds

1. Modify the `/etc/ipworks/common/snmp/snmpd.conf` file according to your hardware.

```
# vi /etc/ipworks/common/snmp/snmpd.conf
```

For example,

```
monitor -u cjslyxz -o lmTempSensorsDevice.1 -I lmTempSensorsva
lue.1 lmTempSensorsValue.1 >= 41000
...
```

Where:

- **1** in `lmTempSensorsDevice.1` is the value of corresponding `lmTempSensorsIndex`. The value of `lmTempSensorsIndex` corresponds to the `Sensor ID` displayed when using command `ipmitool -v sdr type Temperature` to fetch the current temperature value (see Section 4.3.9.2.1 on page 36). During the process of configuring the threshold, you can skip the IDs of the disabled sensors. Read the output after executing the command `ipmitool -v sdr type Temperature`, if the `Sensor Reading` shows `Disabled`, it means the sensor is disabled.
- **-u** specifies a security name to be used for scanning the local host, instead of the default `iquerySecName`. The user must be explicitly created and given suitable access rights.
- **-o** defines the additional varbinds to be added to the notification payload when this monitor trigger fires.
- **-I** indicates that the monitored expression must be applied to the specified OID as a single instance. By default, the OID is treated as a wildcarded object, and the monitor expanded to cover all matching instances.

Note: Different hardware has different sensor IDs. The default configuration is for G10. IPWorks monitors the sensors' ID from 1 to 35. On G9, IPWorks monitors the sensors' ID from 1 to 26. So the IDs from 27 to 35 should be commented



2. Restart the SNMP service to let the configuration take effect.

- Clear all alarms in the alarm table:

```
# /etc/init.d/ipworks.snmpd stop
# /etc/init.d/ipworks.snmpd start
```

- Keep all alarms in the alarm table:

```
# /etc/init.d/ipworks.snmpd stop
# /etc/init.d/ipworks.snmpd start noreset
```

4.3.9.3 Power Supply Sensors Example

4.3.9.3.1 Fetching the Current Values

Execute the following command to fetch the current power supply value:

```
# ipmitool -v sdr type 'Power Supply'
```

For example,

```
Sensor ID           : Power Supply 1 (0x3)
Entity ID           : 10.1 (Power Supply)
Sensor Type (Analog) : Power Supply
Sensor Reading      : 95 (+/- 0) Watts
Status              : Lower Non-Critical
Positive Hysteresis  : Unspecified
Negative Hysteresis  : Unspecified
Minimum sensor range : Unspecified
Maximum sensor range : Unspecified
Event Message Control : Entire Sensor Only
Readable Thresholds  : No Thresholds
Settable Thresholds  : No Thresholds
...
```

4.3.9.3.2 Checking the Thresholds

Execute the following command to fetch the threshold of the power supply sensors predefined in the `snmpd.conf` file:

```
# vi /etc/ipworks/common/snmp/snmpd.conf
```

For example,

```
monitor -u cjslyxz -o lmPSSensorsValue.1 -I lmPSSensorsValue.1 lmPSSensorsValue.1 < 10000
monitor -u cjslyxz -o lmPSSensorsValue.1 -I lmPSSensorsValue.1 lmPSSensorsValue.1 > 150000
monitor -u cjslyxz -o lmPSSensorsValue.2 -I lmPSSensorsValue.2 lmPSSensorsValue.2 < 10000
monitor -u cjslyxz -o lmPSSensorsValue.2 -I lmPSSensorsValue.2 lmPSSensorsValue.2 > 150000
```

**Note:**

- The example shows that the current value -Sensor Reading 95*1000 is within the threshold range 10000(min)~150000(max). If the value is less than 10000 or greater than 150000, one alarm is raised.
- Noted that different hardware has different sensor IDs.
 - On HP DL380 G9 server, sensor IDs of Power supply are lmPSSensorsValue.2 and lmPSSensorsValue.4.
 - On HP DL380 G10 server, sensor IDs of Power supply are lmPSSensorsValue.1 and lmPSSensorsValue.2 .

It is recommended to run following commands to confirm the hardware before modifying:

```
snmpwalk -v 2c localhost -c public lmPSSensorsValue
snmpwalk -v 2c localhost -c public lmPSSensorsDevice
```

4.3.9.3.3 Checking Alarm Details When an Alarm is Raised

If an alarm is raised, use the following command to check the alarm details:

```
# snmpwalk -v 2c localhost alarmtable
```

4.3.9.3.4 Configuring the Thresholds

1. Modify the threshold to a higher level in the /etc/ipworks/common/snmp/snmpd.conf file.

```
# vi /etc/ipworks/common/snmp/snmpd.conf
```

For example, if the current value is 155*1000, modify the file as follows:

```
monitor -u cjslyxz -o lmPSSensorsValue.1 -I lmPSSensorsValue.1 lmPSSensorsValue.1 < 10000
monitor -u cjslyxz -o lmPSSensorsValue.1 -I lmPSSensorsValue.1 lmPSSensorsValue.1 > 160000
monitor -u cjslyxz -o lmPSSensorsValue.2 -I lmPSSensorsValue.2 lmPSSensorsValue.2 < 10000
monitor -u cjslyxz -o lmPSSensorsValue.2 -I lmPSSensorsValue.2 lmPSSensorsValue.2 > 160000
```

2. Restart the SNMP service to let the configuration take effect.

- Clear all alarms in the alarm table:

```
# /etc/init.d/ipworks.snmpd stop
```

```
# /etc/init.d/ipworks.snmpd start
```

- Keep all alarms in the alarm table:

```
# /etc/init.d/ipworks.snmpd stop
```



```
# /etc/init.d/ipworks.snmpd start noreset
```

4.3.9.4 Fan Sensors Example

For Fan sensors, the warning threshold levels are described by different levels with one specific number for each level, defined as follows:

- Lower_Non_Recoverable = 0
- Lower_Critical = 1
- Lower_Non_Critical = 2
- Upper_Non_Critical = 3
- Upper_Critical = 4
- Upper_Non_Recoverable = 5

4.3.9.4.1 Fetching the Current Values

```
# ipmitool -v sdr type Fan
```

For example,

```
Sensor ID           : Fan 1 (0x6)
Entity ID           : 7.1 (System Board)
Sensor Type (Analog) : Fan
Sensor Reading       : 13.720 (+/- 0) unspecified

Status             : Lower Non-Critical
Positive Hysteresis  : Unspecified
Negative Hysteresis  : Unspecified
Minimum sensor range : Unspecified
Maximum sensor range : Unspecified
Event Message Control : Entire Sensor Only
Readable Thresholds  : No Thresholds
Settable Thresholds  : No Thresholds
...
```

Note: Check the **Status** value displayed in the output of ipmitool command.

4.3.9.4.2

Check the threshold level of the fan sensors predefined in the snmpd.conf file:

```
# vi /etc/ipworks/common/snmp/snmpd.conf
```

For example,

```
monitor -u cjslyxz -o lmFanSensorsDevice.1 -I lmFanSensorsValue.1 lmFanSensorsValue.1 >= 3
```

The above example shows that the current Status value (Lower Non-Critical=2) does not exceed the predefined threshold level (3).



If the current Status value exceeds the predefined threshold or is not within the predefined scope, an alarm is raised.

Note: Different hardware has different sensor IDs.

- On HP DL380 G9 server, sensor IDs of Fan device are:
 - lmFanSensorsDevice.2
 - lmFanSensorsDevice.4
 - lmFanSensorsDevice.6
 - lmFanSensorsDevice.8
 - lmFanSensorsDevice.10
 - lmFanSensorsDevice.12
- On HP DL380 G10 server, sensor IDs of Power supply are:
 - lmFanSensorsDevice.1
 - lmFanSensorsDevice.2
 - lmFanSensorsDevice.3
 - lmFanSensorsDevice.4
 - lmFanSensorsDevice.5
 - lmFanSensorsDevice.6

4.3.9.4.3 Checking Alarm Details When an Alarm is Raised

If an alarm is raised, use the following command to check the alarm details:

```
# snmpwalk -v 2c localhost alarmtable
```

4.3.9.4.4 Configuring the Thresholds

1. Modify the threshold to a higher level in the `/etc/ipworks/common/snmp/snmpd.conf` file.

```
# vi /etc/ipworks/common/snmp/snmpd.conf
```

For example,

```
monitor -u cjslyxz -o lmFanSensorsDevice.1 -I lmFanSensorsValue.1  
lmFanSensorsValue.1 >= 4
```

2. Restart the SNMP service to let the configuration take effect:

- Clear all alarms in the alarm table:



```
# /etc/init.d/ipworks.snmpd stop
```

```
# /etc/init.d/ipworks.snmpd start
```

— Keep all alarms in the alarm table:

```
# /etc/init.d/ipworks.snmpd stop
```

```
# /etc/init.d/ipworks.snmpd start noreset
```





Reference List

Ericsson Documents

- [1] Trademark Information
- [2] Typographic Conventions
- [3] Glossary of Terms and Acronyms
- [4] IPWorks Alarm List for DL380 Host Management

Standards

- [5] [Agent Extensibility \(AgentX\) Protocol Version 1](#)
- [6] [A Simple Network Management Protocol \(SNMP\)](#)
- [7] [Introduction to Community-based SNMPv2](#)
- [8] [IF-MIB](#)
- [9] [HOST-RESOURCES-MIB](#)
- [10] [UCD-SNMP-MIB](#)
- [11] [DISMAN-EVENT-MIB](#)
- [12] [LM-SENSORS-MIB](#)

Other References

- [13] [Net-SNMP Website](#)
- [14] [Net-SNMP README files](#)