

A large, abstract, curved graphic on the left side of the page, composed of overlapping translucent shapes in shades of purple, blue, and grey.

REMOTE MONITORING OF SUN X64 SYSTEMS USING IPMITOOL AND IPMIEVD

Eric Markwardt, Client Solutions Organization

Sun BluePrints™ OnLine — January 2007

© 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 USA

All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California.

Sun, Sun Microsystems, the Sun logo, Sun BluePrints, Sun Fire, Solaris, and SunDocs are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

AMD, the AMD64 logo, and AMD Opteron are trademarks or registered trademarks of Advanced Micro Devices.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a). DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS HELD TO BE LEGALLY INVALID.



Please
Recycle



Adobe PostScript

Table of Contents

Remote Monitoring of Sun x64 Systems Using ipmitool and ipmievd	1
Integrated Lights Out Manager	1
Interfaces	1
System Event Log	2
Open-Source Utilities ipmitool and ipmievd	3
Ipmitool	3
Ipmievd	4
Installing ipmitool and ipmievd	4
Option 1: Use the Versions of ipmitool and ipmievd Included with the Solaris 10 OS	4
Option 2: Install ipmitool and ipmievd from a CD or CD Image	7
Option 3: Compile and Install ipmitool and ipmievd from the Source Code	9
Testing ipmitool and ipmievd	9
Security	11
Using ipmitool	11
Managing the System Event Log	12
Displaying System Information	13
Performing Actions on a Remote System	16
Using ipmievd	17
Starting ipmievd on Systems Running the Solaris OS	17
Starting ipmievd on Systems Running the Linux OS	18
Runtime Behavior	19
Log File Monitoring	19
Conclusions	20
About the Author	20
Acknowledgements	20
References	20
Ordering Sun Documents	20
Accessing Sun Documentation Online	21

Remote Monitoring of Sun x64 Systems Using ipmitool and ipmievd

The Integrated Lights Out Manager (ILOM), included in Sun's enterprise-class x64 servers, provides an abundance of information on hardware related events, status, and error conditions. This information can be a valuable asset, enabling administrators to proactively monitor systems and quickly respond to situations that might affect system availability. However, aggregating information from multiple servers distributed throughout an organization and responding in an automated manner can be a challenge, especially as the number of managed servers increases.

Administrators can choose from a range of tools to help expedite system maintenance. Fully-featured enterprise system management solutions, such as the Sun™ N1 System Manager, IBM Tivoli, and HP OpenView software, provide many capabilities. But these solutions can be more costly and complex to set up, and may require additional training to gain proficiency. More basic open-source tools, such as `ipmitool` and `ipmievd`, provide a simpler command-line interface and can be used to monitor servers and aggregate any detected errors into a centralized location for further processing. Smaller organizations with fewer managed servers may prefer to use these open-source tools, rather than investing in larger enterprise systems management tools. Furthermore, administrators from any size organization can benefit from tools that enable them to easily query the status of a system with a single command, reboot a remote system that is in a hung state, and aggregate event information into one central repository.

This document focuses on using the open source tools `ipmitool` and `ipmievd` to interface with the ILOM, query hardware-related status of local and remote servers, and automatically aggregate events into a centralized log file.

Integrated Lights Out Manager

The Integrated Lights Out Manager (ILOM) consists of a combination of dedicated hardware and integrated system management software that enables users to manage x64-based Sun servers, such as the Sun Fire™ X4600 server, independently of the operating system. The hardware includes a Service Processor, dedicated hardware that communicates through the system serial port and a dedicated Ethernet port. The software includes both a command line and web-based interface, as well as support for industry-standard management interfaces.

The ILOM is deployed across Sun's enterprise-class x64 server product line, providing the capability to proactively monitor and manage events on these systems. Administrators can use the ILOM to manage both local and remote x64 servers, displaying system information, monitoring system sensor status, and performing power management functions such as rebooting a hung server or powering on a remote system.

Interfaces

The ILOM features a command line interface as well as a web-based graphical user interface (WebGUI). The command line interface enables operation of the ILOM using keyboard commands. Administrators can

connect directly to the system serial port, or connect over the Ethernet. The WebGUI provides a graphical interface for logging in to the service processor and performing system management and monitoring tasks. Both the WebGUI and the command line interfaces are secured, with the WebGUI using Secure Sockets Layer (SSL) and the command line interface utilizing Secure Shell (SSH) to establish encrypted communication across a network.

In addition, the ILOM also supports the industry-standard Simple Network Management Protocol (SNMP) and Intelligent Platform Management Interface (IPMI) management interfaces.

- Simple Network Management Protocol (SNMP)

The ILOM provides an SNMP v3.0 interface, with limited support for SNMP v1 and SNMP v2c. This enables integration with external data center management applications such as the Sun N1 System Manager, IBM Tivoli, and HP OpenView software.

- Intelligent Platform Management Interface (IPMI)

The IPMI, an open-standard management interface specification defined by Intel and other hardware vendors, can be used to query the status of the system hardware, monitor system sensors, and view hardware event logs. IPMI allows for both in-band and out-of-band server management. With out-of-band server management, processing occurs independently of the operating system and the operating system is not responsible for transporting system status data. This provides full IPMI facilities, including communication with the ILOM, when the system or host operating system is down. With in-band server management, kernel drivers communicate with the service processor after the operating system has started. In-band management allows access to many of the facilities of IPMI without the infrastructure costs associated with provisioning the dedicated management port.

System Event Log

The ILOM System Event Log provides non-volatile storage of status information about the system hardware and software. The ILOM monitors the system and logs the following three types of events to the System Event Log:

- Sensor-specific events — Sensors are used to measure voltages, temperatures, fan speeds, chassis intrusion, and the presence of certain devices. The ILOM polls the various sensors on a periodic basis and posts an event to the System Event Log when a threshold is crossed or an event is triggered.
- System BIOS events — Events and error messages generated by the system BIOS during power-on self tests or at other times are logged.
- System management software events — Events and errors that occur within the ILOM software are logged.

Note – For a complete list of the events logged by the ILOM, refer to the *Integrated Lights-Out Manager Supplement* for a specific Sun x64 server.

Events logged in the System Event log can be queried and displayed by the ILOM WebGUI or command line interfaces.¹ In addition, the System Event Log can be queried by IPMI-compliant tools such as `ipmitool` and `ipmievd`. The remainder of this document focuses on using these two tools along with the ILOM to perform remote system monitoring.

Open-Source Utilities `ipmitool` and `ipmievd`

The open-source tools `ipmitool` and `ipmievd` can be used to help manage and monitor Sun's x64-based servers via the IPMI. Pre-built versions of these utilities are included in the Solaris™ 10 Operating System (Solaris 10 OS), are included on the Resource CD which is included with all Sun x64-based servers, and can be obtained from the Sun Download Center. Pre-compiled and source code versions of these tools are also available from <http://ipmitool.sourceforge.net/>. At this time `ipmitool` and `ipmievd` are only available for Solaris and Linux distributions; a Windows-based version is not yet available.²

`ipmitool`

The `ipmitool` utility provides a command line interface to the Service Processor. It works with the local system via a kernel driver (in-band management) or with a local or remote system via the management network (out-of-band management). Multiple interfaces are supported by the `ipmitool` utility, including the `open`, `bmc`, `lan`, and `lanplus` interfaces.

- `open` — utilizes the `OpenIPMI` kernel device driver, which is included in all modern 2.4 and all 2.6 kernels and should be present in recent Linux distribution kernels.
- `bmc` — utilizes the `bmc` device driver provided by the Solaris 10 OS.
- `lan` — communicates with the Service Processor over an Ethernet LAN connection using UDP under IPv4; UDP datagrams contains IPMI messages and Remote Management Control Protocol (RMCP) headers.
- `lanplus` — communicates with the Service Processor over an Ethernet LAN connection using UDP under IPv4. The `lanplus` interface differs from the `lan` interface in that it uses the RMCP+ protocol, allowing for improved authentication and data integrity checks as well as encryption.

The `ipmitool` utility provides a number of capabilities, including the following:

- Reading and displaying sensor lists
- Printing field replaceable unit (FRU) inventory information
- Displaying and setting Service Processor network configuration parameters
- Displaying event logs
- Performing remote chassis power control

By itself `ipmitool` provides no capabilities for automated alerting. However, utilities such as `ipmievd` can be used to detect events sent to the System Event Log and automatically send them to the `syslog` facility.

1. See the *Integrated Light-Out Manager Administration Guide* for details on using the WebGUI or command line interface to view the System Event Log.

2. Cygwin, a Linux-like environment, can be used to run utilities like `ipmievd` and `ipmitool` on a Windows platform. Applications must be rebuilt from source code, and the use and configuration of Cygwin is beyond the scope of this document.

ipmievd

ipmievd is a daemon that listens for hardware-related events which are detected by the Service Processor and sent to the System Event Log. This utility also logs these messages to the `syslog` facility.

The `ipmievd` utility runs in one of the three following modes:

- Asynchronous event notification from the `OpenIPMI` kernel driver using the Event Message Buffer (Linux only)
- Actively polling the contents of the System Event Log for new events via the `bmc` kernel device driver (Solaris 10 OS only)
- Actively polling the contents of the System Event Log for new events over a management network via the `lan` or `lanplus` interfaces

When `ipmievd` receives an event via any of these mechanisms, it logs the event to the `syslog` facility. Traditional `syslog` scanning tools can then be used to watch for logged errors and act upon them accordingly.

Installing ipmitool and ipmievd

Several options exist for installing the `ipmitool` and `ipmievd` utilities. Administrators can use the pre-compiled version included with the Solaris 10 OS release, install from the Resource CD shipped with the Sun system or downloaded from the Sun Download Center, or compile from the latest source code available from a third-party Web site. Factors affecting the decision of which option is most appropriate include the operating system deployed, the desire to use a pre-compiled or self-compiled version, and whether or not the use of the most recent version is preferred.

Option 1: Use the Versions of ipmitool and ipmievd Included with the Solaris 10 OS

The version of `ipmitool` included with the Solaris 10 OS, while not necessarily the most current version available, has been compiled and thoroughly tested by Sun. This version only runs on systems using the Solaris 10 OS, but it can be used to monitor and manage remote systems which are running the Solaris, Linux, or Windows operating systems. On systems running the Solaris 10 OS, `ipmitool` is a part of the `SUNWipmi` package and is installed as `/usr/sfw/bin/ipmitool`.

No pre-built version of `ipmievd` is currently included with the Solaris 10 OS, but the sources are included as a part of the `SUNWipmiS` package. System administrators can compile and install this Sun-supplied version of `ipmievd` for use on their system.

The following steps illustrate how to compile and install `ipmitool` and `ipmievd` on a system running the Solaris 10 OS.

1. Verify the correct package is installed on the system.

- a. First, use the `pkginfo` command to verify that the `SUNWipmiS` package has been installed on the system:

```
# pkginfo SUNWipmiS
system      SUNWipmiS Source for IPMItool utility
```

- b. If the following error message is displayed, it indicates that the `SUNWipmiS` packaged has not been installed:

```
# pkginfo SUNWipmiS
ERROR: information for 'SUNWipmiS' was not found
```

- c. If necessary, use the `pkgadd -d <mount_point>` command to install the `SUNWipmiS` package from the Solaris 10 OS distribution media. The following example assumes `/mnt` is used as the mount point:

```
# pkgadd -d /mnt/Solaris_10/Product SUNWipmiS

Processing package instance <SUNWipmiS> from </mnt/Solaris_10/Product>

Source for IPMItool utility(i386) 11.10.0,REV=2005.01.08.01.09
Copyright 2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Using </> as the package base directory.
## Processing package information.
## Processing system information.
   6 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <SUNWipmiS> [y,n,?] y

Installing Source for IPMItool utility as <SUNWipmiS>

## Installing part 1 of 1.
5710 blocks

Installation of <SUNWipmiS> was successful.
#
```

2. Compile the source code and install the tools.

After verifying that the `SUNWipmiS` package is installed on the system, build the tools `ipmitool` and `ipmievd` from the `ipmitool` source tree under `/usr/sfw/share/src`.

- a. Add `/usr/sfw/bin` and `/usr/ccs/bin` to the user's search path, making sure that `/usr/sfw/bin` precedes `/usr/ccs/bin`. This is necessary to successfully run the `configure` and `gmake` commands in the following steps, even if absolute path names are specified on the command line.

- b. Change directories and use the `configure` command to setup the compilation:

```
# cd /usr/sfw/share/src/ipmitool
# env "LD_FLAGS=-R/usr/sfw/lib -L/usr/sfw/lib" "CPPFLAGS=-I/usr/sfw/include"
./configure --prefix=/usr/sfw
checking build system type... i386-pc-solaris2.10
checking host system type... i386-pc-solaris2.10
checking target system type... i386-pc-solaris2.10
...
config.status: creating config.h
config.status: executing depfiles commands
ipmitool 1.8.4
Interfaces
  lan      : yes
  lanplus  : yes
  open     : no
  imb      : no
  bmc      : yes
  lipmi    : no
Extra tools
  ipmievd  : yes
```

- c. Use the `gmake` command to compile the code:

```
# /usr/sfw/bin/gmake
/usr/sfw/bin/gmake all-recursive
gmake[1]: Entering directory `/usr/sfw/share/src/ipmitool'
Making all in lib
gmake[2]: Entering directory `/usr/sfw/share/src/ipmitool/lib'
if /bin/bash ../libtool --silent --mode=compile gcc -DHAVE_CONFIG_H -I. -I. -I.. -I../
include -I/usr/sfw/include -g -O2 -fno-strict-aliasing -MT helper.lo -MD -MP -MF ".deps/
helper.Tpo" -c -o helper.lo helper.c; \
then mv -f ".deps/helper.Tpo" ".deps/helper.Plo"; else rm -f ".deps/helper.Tpo"; exit 1; fi
...
Making all in control
gmake[2]: Entering directory `/usr/sfw/share/src/ipmitool/control'
gmake[2]: Nothing to be done for `all'.
gmake[2]: Leaving directory `/usr/sfw/share/src/ipmitool/control'
gmake[2]: Entering directory `/usr/sfw/share/src/ipmitool'
gmake[2]: Leaving directory `/usr/sfw/share/src/ipmitool'
gmake[1]: Leaving directory `/usr/sfw/share/src/ipmitool'
```

d. Use the `gmake install` command to install the newly compiled programs:

```
# /usr/sfw/bin/gmake install
Making install in lib
gmake[1]: Entering directory `/usr/sfw/share/src/ipmitool/lib'
gmake[2]: Entering directory `/usr/sfw/share/src/ipmitool/lib'
gmake[2]: Nothing to be done for `install-exec-am'.
gmake[2]: Nothing to be done for `install-data-am'.
gmake[2]: Leaving directory `/usr/sfw/share/src/ipmitool/lib'
gmake[1]: Leaving directory `/usr/sfw/share/src/ipmitool/lib'
Making install in src
gmake[1]: Entering directory `/usr/sfw/share/src/ipmitool/src'
...
gmake[3]: Entering directory `/usr/sfw/share/src/ipmitool/src'
test -z "/usr/sfw/bin" || /usr/sfw/share/src/ipmitool/install-sh -d "/usr/sfw/bin"
  /bin/bash ../libtool --silent --mode=install ../install-sh -c 'ipmitool' '/usr/sfw/bin/
ipmitool'
test -z "/usr/sfw/sbin" || /usr/sfw/share/src/ipmitool/install-sh -d "/usr/sfw/sbin"
  /bin/bash ../libtool --silent --mode=install ../install-sh -c 'ipmievd' '/usr/sfw/sbin/
ipmievd'
gmake[3]: Nothing to be done for `install-data-am'.
gmake[3]: Leaving directory `/usr/sfw/share/src/ipmitool/src'
...
mkdir -p //usr/sfw/share/doc/ipmitool
install -m 644 README COPYING AUTHORS ChangeLog //usr/sfw/share/doc/ipmitool
README installed as /etc/init.d/README
gmake[2]: Leaving directory `/usr/sfw/share/src/ipmitool'
gmake[1]: Leaving directory `/usr/sfw/share/src/ipmitool'
#
```

3. Confirm that the newly-built versions of `ipmitool` and `ipmievd` are installed in the `/usr/sfw/bin` and `/usr/sfw/sbin` directories:

```
# ls -l /usr/sfw/bin/ipmitool /usr/sfw/sbin/ipmievd
-rwxr-xr-x  1 root    root      839220 Sep 22 12:44 /usr/sfw/bin/ipmitool
-rwxr-xr-x  1 root    root      470436 Sep 22 12:44 /usr/sfw/sbin/ipmievd
```

Option 2: Install ipmitool and ipmievd from a CD or CD Image

Both `ipmitool` and `ipmievd` can be installed from the Resource CD that ships with all Sun x64-based servers or from a Resource CD ISO image downloaded from the Sun Download Center. The versions of `ipmitool` and `ipmievd` that are included on the Resource CD are typically more current than the versions bundled with the Solaris OS, since the Resource CD is updated more frequently.

Systems Running the Solaris 10 OS

1. Insert the CD into the drive and copy the appropriate `gzip` file to a temporary location:

```
# cd <mount_point>/support/tools/ipmi
# cp ipmitool-1.8.8-solaris2.10-i386.gz /tmp
```

2. Unzip the temporary file:

```
# cd /tmp
# /usr/bin/gunzip ipmitool-1.8.8-solaris2.10-i386.gz
```

3. Install the resulting package datastream file:

```
# /usr/sbin/pkgadd -d ipmitool-1.8.8-solaris2.10-i386
The following packages are available:
  1 ipmitool      ipmitool - Utility for Intelligent Platform Management Interface
    (IPMI) control
                        (i386) 1.8.8
Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all
Processing package instance <ipmitool> from </tmp/ipmitool-1.8.8-solaris2.10-i386>
ipmitool - Utility for Intelligent Platform Management Interface (IPMI)
control(i386) 1.8.8
Duncan Laurie
Using </opt/ipmitool> as the package base directory.
## Processing package information.
## Processing system information.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing ipmitool - Utility for Intelligent Platform Management Interface (IPMI) control
as <ipmitool>
## Installing part 1 of 1.
/opt/ipmitool/bin/ipmitool
/opt/ipmitool/sbin/ipmievd
/opt/ipmitool/share/man/man1/ipmitool.1
/opt/ipmitool/share/man/man8/ipmievd.8
[ verifying class <none> ]
Installation of <ipmitool> was successful.
#
```

4. Confirm that the newly-built versions of ipmitool and ipmievd are installed in the /usr/ipmitool/bin and /usr/ipmitool/sbin directories:

```
# ls -l /opt/ipmitool/bin/ipmitool /opt/ipmitool/sbin/ipmievd
-rwxr-xr-x  1 root bin 354376 2005-09-08 16:48 /opt/ipmitool/bin/ipmitool*
-rwxr-xr-x  1 root bin 473068 2005-09-08 16:48 /opt/ipmitool/sbin/ipmievd*
```

Install ipmitool and ipmievd on Systems Running the Linux OS

Depending on the Linux distribution and which packages were initially installed, ipmitool and ipmievd may already be installed. If they are not already present, install the appropriate rpm file from the Resource CD for that Linux distribution. This example shows installing the rpm file for Red Hat Enterprise Linux 4 64-bit:

1. Use the `rpm` command to install the `ipmitool` package:

```
# cd <mount_point>/support/tools/ipmi
# rpm -i -v ipmitool-1.8.8-2.rhel4.x86_64.rpm
Preparing packages for installation...
ipmitool-1.8.8-2.rhel4
```

2. Use the `which` command to confirm that `ipmitool` and `ipmievd` are installed on the system:

```
# which ipmitool
/usr/bin/ipmitool

# which ipmievd
/usr/sbin/ipmievd
```

Note – If `ipmitool` or `ipmievd` will be used to monitor the local system via the OpenIPMI driver, this driver must be installed and enabled in the kernel. Refer to the *Open Interface* section of the `ipmitool(1M)` man page for details on the required kernel modules.

Option 3: Compile and Install `ipmitool` and `ipmievd` from the Source Code

If the most recent versions of `ipmitool` and `ipmievd` are preferred, the sources and precompiled binaries for Solaris and several Linux distributions can be obtained from the third-party web site <http://ipmitool.sourceforge.net/>.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Testing `ipmitool` and `ipmievd`

The `ipmitool chassis status` prints out the current status of the chassis, and can be used to easily test if the tool is working correctly. If the `ipmitool` command works correctly, then `ipmievd` should also work without problems since it uses the same drivers and interfaces.

- Local system running the Solaris 10 OS

On Sun x64 servers equipped with a Service Processor (SP) and running the Solaris 10 OS, use `ipmitool` to query the SP via the `bmc` driver that is included with the Solaris OS. The `-I bmc` option specifies the interface to be used, the `-U root` option specifies the user:

```
# ipmitool -I bmc -U root chassis status
System Power          : on
Power Overload        : false
Power Interlock       : inactive
Main Power Fault      : false
Power Control Fault   : false
Power Restore Policy  : always-off
Last Power Event      :
Chassis Intrusion     : inactive
Front-Panel Lockout   : inactive
Drive Fault           : false
Cooling/Fan Fault     : false
```

- Local Linux-based server

On Linux-based servers, `ipmitool` uses the `OpenIPMI` driver to communicate with the SP via the kernel. The `-I open` option specifies the interface to be used:

```
# ipmitool -I open chassis status
System Power          : on
Power Overload        : false
Power Interlock       : inactive
Main Power Fault      : false
Power Control Fault   : false
Power Restore Policy  : always-off
Last Power Event      :
Chassis Intrusion     : inactive
Front-Panel Lockout   : inactive
Drive Fault           : false
Cooling/Fan Fault     : false
```

- Remote servers

To query a remote server from a server running either the Solaris or Linux operating system, use the `lan` or `lanplus` interface. The `-I lan` option specifies the interface to used, the `-U root` option

specifies the user, and the `-H <IP address>` option specifies the IP address (or hostname) of the remote server. When prompted, enter the password of the remote service processor:

```
# ipmitool -I lan -U root -H 129.153.228.111 chassis status
Password: <enter password of remote System Processor>
System Power          : on
Power Overload        : false
Power Interlock       : inactive
Main Power Fault      : false
Power Control Fault   : false
Power Restore Policy  : always-off
Last Power Event      :
Chassis Intrusion     : inactive
Front-Panel Lockout   : inactive
Drive Fault           : false
Cooling/Fan Fault     : false
```

Security

The `ipmitool` utility's `lan` and `lanplus` interface enable an administrator to query platform information and perform power operations, such as reset and power on, on remote systems. While this can simplify administration, and reduce or eliminate physical trips to a remote server for routine maintenance, security should be reviewed before enabling this functionality.

It is strongly advised that the Service Processor's root password be changed from the factory default before connecting the network management Ethernet port to a network for remote access. For increased security, this password should differ from other passwords, such as the root password, on that system. To further reduce vulnerability, the network management interface should only be enabled in secure or trusted environments with a separate, secure management network or in environments where system security is not an issue.

The `ipmitool` utility can be used to change the default password, via the `user set password <userid> <password>` command. In this example, the password for root, `userid 2`, is changed:

```
# ipmitool -I lanplus -H <IP Address> -U root -P <oldpassword> user set password 2
<newpassword>
```

Using ipmitool

The `ipmitool` utility provides the ability to display entries in the System Event Log, print information about field replaceable units (FRUs), display and set the LAN configuration, and display sensor status information on both local and remote systems that support the IPMI specification. In addition, `ipmitool` can also be used to perform remote system power controls, such as powering on or resetting a hung system.

The following examples illustrate just a subset of the many `ipmitool` capabilities. Please see the `ipmitool` man page for a complete listing of supported commands.

Managing the System Event Log

The `ipmitool` utility is commonly used to periodically query the System Event Log to check for any errors that were reported. Pre-defined events can also be sent to the System Event Log, as an aid in testing and debugging. Messages continue to accumulate in the System Event Log until they are explicitly cleared or until they are overwritten when the buffer capacity is reached.

- Query the System Event Log:

Use `ipmitool`'s `sel elist` command to print out the contents of the System Event Log. The following command uses the `lan` interface to display the System Event Log of the remote system at the specified IP address:

```
# ipmitool -I lan -H 129.153.228.111 -U root sel elist
Password:<enter root password of remote System Processor>
100 | 09/20/2006 | 13:22:44 | Power Supply ps0.vinok | State Deasserted
200 | 09/20/2006 | 13:22:46 | Power Supply ps0.pwrok | State Deasserted
300 | 09/20/2006 | 13:23:16 | Power Supply ps0.vinok | State Asserted
400 | 09/20/2006 | 13:23:17 | Power Supply ps0.pwrok | State Asserted
```

If the System Event Log contains a large number of entries, the `last` modifier can be used with the `sel elist` subcommand to show only a specific number of the most recent events. For example, the following command displays the last 2 commands in the System Event Log on a remote server:

```
# ipmitool -I lan -H 129.153.228.111 -U root sel elist last 2
Password:<enter password of remote System Processor>
300 | 09/20/2006 | 13:23:16 | Power Supply ps0.vinok | State Asserted
400 | 09/20/2006 | 13:23:17 | Power Supply ps0.pwrok | State Asserted
```

- Send an event to the System Event Log

Pre-defined events can be sent to the System Event Log using the `event` command. This can be helpful during testing, to verify that `ipmievd` correctly sees entries in the System Event Log and then logs them correctly to the `syslog` facility. For example, the following command sends a test memory event to the System Event Log:

```
# ipmitool -I lan -H 129.153.228.111 -U root event 3
Password:<enter password of remote System Processor>
Sending SAMPLE event: Memory - Correctable ECC
    0 | Pre-Init Time-stamp | Memory #0x53 | Correctable ECC | Asserted
#
```

Please refer to the `ipmitool(1M)` man page for a list of the currently supported pre-defined events.

- Clear the System Event Log

The System Event Log maintains approximately the last 1000 messages in a circular buffer, with the oldest entries replaced by new ones when the buffer fills. While it is not necessary, administrators can use the `clear` command to explicitly clear the messages in the System Event Log:

```
# ipmitool -I lan -H 129.153.228.111 -U root sel clear
Password: <enter password of remote System Processor>
Clearing SEL. Please allow a few seconds to erase.
```

Displaying System Information

The Integrated Lights Out Manager (ILOM) maintains non-volatile storage of information such as sensor data and field replaceable unit (FRU) information. The `ipmitool` utility can be used to query and display this information.

- Display the status of the system LED lights

Use the `led get` command to display the state of the LED lights on a system. This can be useful, particularly on remote systems, to check if a particular failure LED is lit on a system:

```
# ipmitool -I lan -H 129.153.228.111 -U root sunoem led get
Password: <enter password of remote System Processor>
sys.psfail.led | OFF
sys.tempfail.led | OFF
sys.fanfail.led | OFF
bp.power.led | ON
bp.locate.led | OFF
bp.alert.led | OFF
fp.power.led | ON
fp.locate.led | OFF
fp.alert.led | OFF
io.f0.led | OFF
io.hdd0.led | OFF
... output truncated for brevity
```

- Display sensor values

Use the `sdr` command to display sensor values. For example, the following example displays voltage sensor status:

```
# ipmitool -I lan -H 129.153.228.111 -U root sdr type voltage
Password: <enter password of remote System Processor>
mb.v_bat | 06h | ok | 7.0 | 3.12 Volts
mb.v_+3v3stby | 07h | ok | 7.0 | 3.29 Volts
mb.v_+3v3 | 08h | ok | 7.0 | 3.34 Volts
... output truncated for brevity
```


- Display the Service Processor's network configuration

Use the `lan print` command to display the Service Processor's network configuration:

```
# ipmitool -I lan -H 129.153.228.111 -U root lan print 1
Password: <enter password of remote System Processor>
Set in Progress      : Set Complete
Auth Type Support    : NONE MD2 MD5 PASSWORD
Auth Type Enable     : Callback : MD2 MD5 PASSWORD
                    : User      : MD2 MD5 PASSWORD
                    : Operator : MD2 MD5 PASSWORD
                    : Admin   : MD2 MD5 PASSWORD
                    : OEM    :
IP Address Source    : Static Address
IP Address           : 129.153.228.111
Subnet Mask          : 255.255.255.128
MAC Address          : 00:14:4f:0e:2d:8f
SNMP Community String : AMITest
IP Header            : TTL=0x00 Flags=0x00 Precedence=0x00 TOS=0x00
BMC ARP Control      : ARP Responses Disabled, Gratuitous ARP Disabled
Gratuitous ARP Intrvl : 5.0 seconds
Default Gateway IP    : 129.153.228.1
Default Gateway MAC   : 00:00:00:00:00:00
Backup Gateway IP     : 0.0.0.0
Backup Gateway MAC    : 00:00:00:00:00:00
802.1q VLAN ID       : Disabled
802.1q VLAN Priority  : 0
RMCP+ Cipher Suites  : 2,3,0
Cipher Suite Priv Max : XXXXXXXXXXXXXXXXXXXX
                    : X=Cipher Suite Unused
                    : c=CALLBACK
                    : u=USER
                    : o=OPERATOR
                    : a=ADMIN
                    : O=OEM
```

The `lan` command can also be used to configure the IP information. See the `ipmitool(1M)` man page for more information.

- Print a list of all Field Replaceable Unit (FRU) inventory data.

Use the `fru print` command to print a list of all FRU inventory data. This command can be useful for extracting information such as serial numbers, part numbers, network port MAC addresses, memory DIMM sizes, and other information about the system hardware:

```
# ipmitool -I lan _H 129.153.228.111 -U root fru print
Password:<enter password of remote System Processor>
FRU Device Description : Builtin FRU Device (ID 0)
Board Product          : ASSY,SERV PROCESSOR,G1/2
Board Serial           : 1762TH1-0549003395
Board Part Number      : 501-6979-02
Board Extra            : 51
Board Extra            : G1/2_GRASP
Product Manufacturer   : SUN MICROSYSTEMS
Product Name           : ILOM

FRU Device Description : sp.net0.fru (ID 2)
Product Manufacturer   : MOTOROLA
Product Name           : FAST ETHERNET CONTROLLER
Product Part Number    : MPC8248 FCC
Product Serial         : 00:14:4F:0E:2D:8F
Product Extra          : 01
Product Extra          : 00:14:4F:0E:2D:8F

FRU Device Description : mb.fru (ID 4)
Chassis Type           : Rack Mount Chassis
Chassis Part Number    : 541-0251-05
Chassis Serial         : 3183-0544LFF1220
Board Product          : ASSY,MOTHERBOARD,G2
Board Serial           : 1762TH1-0547002891
Board Part Number      : 500-6974-06
Board Extra            : 04
Board Extra            : G2_MB
Product Manufacturer   : SUN MICROSYSTEMS
Product Name           : SUN FIRE X4200
Product Part Number    : 602-3223-01
Product Serial         : 0552AM1862

FRU Device Description : mb.bios.fru (ID 5)
Product Manufacturer   : AMERICAN MEGATRENDS
Product Name           : SYSTEM BIOS
Product Part Number    : AMIBIOS8
Product Version        : 0ABGA036

FRU Device Description : mb.net0.fru (ID 16)
Product Manufacturer   : INTEL
Product Name           : DUAL PORT GIGABIT ETHERNET CONTROLLER (COPPER)
Product Part Number    : 82546EB
Product Serial         : 00:14:4F:01:76:04
Product Extra          : 02
Product Extra          : 00:14:4F:01:76:04
Product Extra          : 00:14:4F:01:76:05
```

```

...
FRU Device Description : pl.d1.fru (ID 13)
Product Manufacturer  : MICRON TECHNOLOGY
Product Name         : 2048MB DDR 400 (PC3200) ECC
Product Part Number  : 36VDDF25672G-40BD2
Product Version      : 0200
Product Serial       : 2A1CDE9B

FRU Device Description : pl.d2.fru (ID 14)

FRU Device Description : pl.d3.fru (ID 15)

```

Performing Actions on a Remote System

The `ipmitool` utility can be used to perform power system controls, such as powering on or resetting a hung server, on remote systems that support the IPMI specification. This functionality can help eliminate trips to remote data centers to service systems, speeding and simplifying system administration.

The following examples illustrate a few applications of `ipmitool` for managing remote systems. For a complete list of all `ipmitool` functionality, please refer to the `ipmitool` documentation.

- Check the cause of the last system restart

Use the `restart_cause` command to display the cause of the last system restart:

```

# ipmitool -I lan -H 129.153.228.111 -U root chassis restart_cause
Password: <enter password of remote System Processor>
System restart cause: chassis power control command
#

```

- Perform a hard reset of a remote system

The `power reset` command is used to perform a hard reset (power off/power on) of a system. This command can be useful for restarting a system that is in a hung state:

```

# ipmitool -I lan -H 129.153.228.111 -U root chassis power reset
Password: <enter password of remote System Processor>
Chassis Power Control: Reset
#

```

Note – Since the `reset` command does not perform a graceful shutdown of the operating system prior to powering down the system, this command should **only** be used when a system is in a hung state.

Commands are also available to perform the equivalent of a five-second host power-button reset and force the power off (`chassis power off`), and perform the equivalent of a brief host power-button press (`chassis power off soft`). Please see the `man` page for complete details on the full set of available `ipmitool` commands.

- Request a PXE boot on the next reboot

The `bootdev pxe` command is used to request that the system perform a PXE boot on the next reboot. This can be used as an alternative to pressing the F12 key on the console during POST to force a PXE boot:

```
# ipmitool -I lan -H 129.153.228.111 -U root chassis bootdev pxe
Password: <enter password of remote System Processor>
Set boot Device to pxe
#
```

Similarly, the keyword `disk` or `cdrom` can be used in place of `pxe` in the previous example to cause the system to boot from the hard disk or CD-ROM, respectively, on the next boot. Please see the `man` page for a complete list of `ipmitool` commands and details on their usage.

Using ipmievd

The following section describes how to configure and start `ipmievd` on systems running the Solaris OS. Information on systems using the Linux OS is included in the section “Starting `ipmievd` on Systems Running the Linux OS” on page 18.

Starting ipmievd on Systems Running the Solaris OS

The following steps describe how to run the `ipmievd` utility on systems running the Solaris OS.

1. Edit the `syslog.conf` file.

Edit the `/etc/syslog.conf` file and add the following lines. This causes messages logged by `ipmievd` to the `LOG_LOCAL4` syslog facility at the info level or higher to be written to the `/var/adm/messages` file:

```
#
# Send messages logged by ipmievd to /var/adm/messages
#
local4.info                /var/adm/messages
```

2. Send the HUP signal to the `syslog` daemon so that it rereads the `syslog.conf` file:

```
# /usr/bin/pkill -HUP syslogd
```

3. Start `ipmievd`.

On systems running the Solaris OS, `ipmievd` can use the `lan`, `lanplus`, or `bmc` interface to periodically poll the System Event Log for new events. Use of the `lan` or `lanplus` interfaces is recommended over the `bmc` interface as they have been observed to be more stable in the author's experience.

To run `ipmievd` with the `lan` interface on a system running the Solaris OS:

```
# /usr/sfw/sbin/ipmievd -I lan -H 129.153.228.111 -U root sel
Password:
```

4. Use the `ps` command to confirm that `ipmievd` is running in the background:

```
# ps -ef | grep ipmievd | grep -v grep
root 8974 1 0 16:09:04 ? 0:00 /usr/sfw/sbin/ipmievd -I lan -H 129.153.228.111 -U root sel
```

The `ipmievd` utility also logs two messages to the `syslog` daemon to indicate that it is running:

```
# grep ipmievd /var/adm/messages
Sep 21 09:04:42 sun-x4200 ipmievd: [ID 702911 local4.notice] Reading sensors...
Sep 21 09:04:42 sun-x4200 ipmievd: [ID 702911 local4.notice] Waiting for events...
```

Starting `ipmievd` on Systems Running the Linux OS

By default, the Linux OS writes the `local4` `syslog` facility messages to the `/var/log/messages` file. Therefore, it is not necessary to edit the `/etc/syslog.conf` before starting `ipmievd`.

Administrators can choose between two different event watching mechanisms on systems running the Linux OS: `OpenIPMI` and the `lan/lanplus` interfaces:

- To start `ipmievd` as a daemon running in the background with the `OpenIPMI` event watching mechanism:

```
# /usr/sbin/ipmievd -I open open
```

- Alternatively, the `lan` or `lanplus` interfaces can be used to poll the System Event Log on a periodic basis. To start `ipmievd` using the `lan` interface:

```
# /usr/sbin/ipmievd -I lan -H 129.153.228.111 -U root sel
```

Note – Use of the `lan` or `lanplus` interface is generally preferred, since some baseboard management controllers do not support the Event Message Buffer required for the `OpenIPMI` event watching mechanism or do not actually deliver events to it, even though they may claim to do so.

In addition, the author experienced difficulties with some version of `ipmievd` when using the `OpenIPMI`, `bmc`, and `lan` interfaces on various platforms. In general, the `lan` and `lanplus` interfaces have been observed to be the most stable and are thus recommended. If the reader experiences difficulties getting the utility to work correctly, trying a different version of `ipmievd` can sometimes solve the problem.

Runtime Behavior

When `ipmievd` is running, it continues to poll the System Event Log on a periodic basis. The default is once every ten seconds. This time interval can be changed by adding the modifier `timeout=<seconds>` to the command line when starting `ipmievd`. The following example starts `ipmievd` and has it poll the System Event Log once every 60 seconds:

```
# /usr/sfw/sbin/ipmievd -I lan -H 129.153.228.111 -U root sel timeout=60
```

Once started, `ipmievd` continues running as a daemon in the background and watches for new events being logged to the System Event Log. If new events are detected, they are immediately logged to `syslog`. The format of the messages appearing in `/var/adm/messages` is similar to the following examples:

```
Sep 21 10:43:57 sun-x4200 ipmievd: [ID 702911 local4.notice] Temperature sensor - Upper Critical going high
Sep 21 10:44:07 sun-x4200 ipmievd: [ID 702911 local4.notice] Voltage sensor - Lower Critical going low
Sep 21 10:44:27 sun-x4200 ipmievd: [ID 702911 local4.notice] Memory sensor - Correctable ECC
```

During testing or debugging, `ipmievd` can also be started with the `nodaemon` option so that it continues to run in the foreground rather than becoming a daemon. When running with this option, `ipmievd` logs all messages to `stderr` rather than to `syslog`:

```
# /usr/sfw/sbin/ipmievd -I lan -H 129.153.228.111 -U root sel nodaemon
Password:
Reading sensors...
Waiting for events...
Temperature sensor - Upper Critical going high
^C
#
```

When started as shown in these examples, `ipmievd` will continue to run until it is explicitly stopped. However, `ipmievd` will not automatically restart after a system reboot. Please refer to your system administration documentation for more information on starting processes automatically upon system boot.

Log File Monitoring

When `ipmievd` is running, any message logged to the System Event Log is automatically forwarded to the `syslog` facility. This can help simplify ongoing system monitoring by consolidating messages in one central repository. Administrators can now use their preferred tools or scripts to watch for any error conditions that are logged by `ipmievd` and to perform log scraping on the messages file.

One possible log watching tool to consider using is `swatch` (Simple WATCHer), a third-party tool that can be configured to watch a log file and take a specific action when certain events occur. This third-party tool is available for download from <http://swatch.sourceforge.net/>.

Conclusions

The open-source tools `ipmitool` and `ipmievd` are useful utilities for interfacing to the ILOM included on Sun's enterprise-class x64 servers. The `ipmitool` utility can be used to query hardware-related status, monitor events that have been logged to the by the ILOM to System Event Log, and perform power related operations such as rebooting or shutting down a remote server. The `ipmievd` utility can be used to detect events sent to the System Event Log and automatically send them to the `syslog` facility, enabling centralized processing. These capabilities can help administrators aggregate hardware-related information from multiple servers distributed throughout an organization and respond in a more automated manner, for easier and more proactive system management.

About the Author

Eric Markwardt has worked for Sun Microsystems, Inc., since 1995 and is currently a Senior Technical Specialist in the Systems Practice of Sun's Client Solutions Organization, focusing on driving customer adoption of Sun's x64-based servers and workstations. Based in the Minneapolis, MN area, Eric participates in technical pre-sales activities at financial services, retail, manufacturing, and health care organizations through the Midwest. Eric has worked in the computer industry since 1988 and holds a B.S. in Computer Science from the University of Minnesota, Duluth.

Acknowledgements

The author would like to thank Marshall Choy and Cari Yuen for their encouragement to write this article and Patrick Boyle, Thomas Harpin, Paul Kirtland, Tim Knight, Chris Kordish, Pierre Reynes, and Victor Pantaleon for their assistance in reviewing the technical content.

References

- [1] *Intelligent Platform Management Interface Specification Second Generation v2.0*, Intel Corp.
http://www.intel.com/design/servers/ipmi/pdf/IPMIv2_0_rev1_0_E3_markup.pdf
- [2] IPMItool overview and documentation, Duncan Laurie.
<http://ipmitool.sourceforge.net/>
- [3] *Integrated Lights-Out Manager (ILOM) Administration Guide*, Sun Microsystems, Inc.,
<http://www.sun.com/products-n-solutions/hardware/docs/pdf/819-1160-13.pdf>
- [4] *Integrated Lights-Out Manager Supplements*, Sun Microsystems, Inc.
Each Sun x64 server has a separate ILOM Supplement document. To access these documents, go to:
http://www.sun.com/products-n-solutions/hardware/docs/Servers/x64_servers/index.html
- [5] Sun Download Center. <http://www.sun.com/download/>

Ordering Sun Documents

The SunDocsSM program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is

`http://docs.sun.com/`

