kontron

» Whitepaper «

IPMI and Open-Source Tools: Sorting Out the Confusion

IPMI and Open-Source Tools: Sorting Out the Confusion. Improving Server Management with the Right IPMI Tools for Your Implementation

The aim of server management is to help service systems via controls and information retrieved from sensors so that problems can be diagnosed early, quickly and, before they become overly critical. IPMI (Intelligent Platform Management Interface) is a standard interface used by many vendors to support all of these server management functions, either locally or remotely. IPMI-based management activities take place in the management controller firmware, instead of using the extra bandwidth from the main processor, which would reduce its production capacity. Using IPMI for server management enables the management of any large numbers of servers without spending the extra time and cost of on-site visits, and administrators can manage many different vendors' systems with the same software, thereby reducing each server's total cost of ownership.

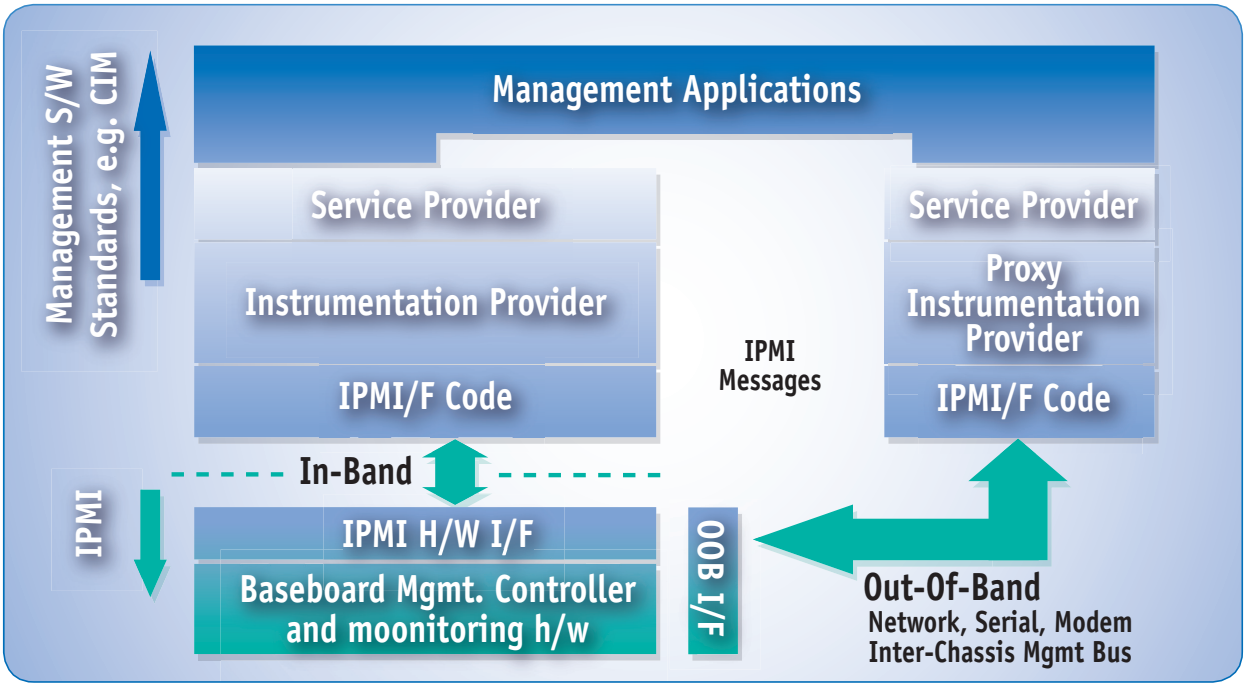
CONTENTS

Server management platform instrumentation involves monitoring system health and performance characteristics such as voltage, temperature, fans and power supplies. Instrumentation also provides important access to hardware asset information, logging, power control, and event alerting. Most OEMs offer some type of instrumentation, but many of these options are not interoperable, unless they are enabled with IPMI. All IPMI firmware implementations contain the ability for remote system reset, even when the OS is not running, and also provide sensor readings for temperature, voltage and fan speed. IPMI provides an advantage over simpler sensor methodologies such as lm_sensors, by including events and controls that can be used to manage server behavior. The IPMI firmware has fan and power controls to automatically handle certain events. An IPMI management agent can wait for events instead of continuously polling sensors, which is the only method available in the absence of an IPMI management controller. Also, the IPMI management controller allows the server to use its watchdog timer to guard against OS hangs and lockups.

software developers are free to choose from several well-established open-source tools, there is often a lack of clarity regarding the strengths and weaknesses of each option. By examining four of the most popular server management tools – openipmi, ipmitool, ipmiutil, and freeipmi – system designers and managers can avoid confusion and choose the most appropriate tool for a specific implementation.

Telecom and IPMI

Demanding physical environments are common to telecom applications and the servers that run them require a close eye on their health, as they are often deployed in remote buildings with no one on-site. Telecom systems need to pass NEBS robustness testing and 99.999% availability requirements, which translates into less than 5 minutes of down-time per year. There can be significant financial penalties for not meeting availability requirements, so IPMI is particularly appropriate for high availability telecom servers.



The primary goal of the IPMI specification is to ensure that a heterogeneous mix of servers can all be managed the same way. However early IPMI-enabled systems often required proprietary management software as part of the implementation. Today, most enterprise-class servers provide some level of IPMI capability and even offer a range of choices among IPMI open-source tools. While customers, administrators, integrators, and

In particular, Kontron's IPMI firmware implementation adds a standard HPM (Hardware Platform Management) interface for automatically flashing new firmware, either locally or remotely. This HPM feature is not part of the IPMI specification, but is defined by the PICMG HPM.1 specification, and is part of some open-source IPMI tools.

Many firmware implementations also include additional layers above IPMI, with other advanced features such as web browser access, KVM application, email notifications, virtual media, and others. In addition, in-band software agents may provide additional SNMP or HPI instrumentation. These additional features are useful in server management, but are not the subject of this article.

IPMI in Action

Understanding how IPMI is used in the overall solution will help determine what IPMI features are important in selecting the best IPMI tool. Below are some typical use cases for solutions that use some form of IPMI management, from basic to advanced. In the list below, each subsequent case includes the features of the all the previous examples. Choosing a use case depends largely on the integrator's knowledge of IPMI capabilities, knowing how important it is having the capabilities, and how much effort is required to implement those capabilities. In the case of telecommunication servers, achieving high availability requirements for a complete solution often involves all of the functions below.

1. *No Instrumentation*
Some integrators may ignore the IPMI functionality, until an error is noticed. At this point, the IPMI System Event Log (SEL) interpretation is key to understanding an error that occurred in the past, so that the root cause of the error can be correctly identified. This is the easiest to implement but has the least functionality.

2. *Local IPMI Instrumentation to Monitor System Health*
This could entail an event monitoring daemon, and a method to ensure that the IPMI firmware log (SEL) does not become full, among other things. Gathering occasional sensor readings and FRU inventory data after an event occurs is often part of this use case.

3. *Increasing Availability by Using IPMI to Protect Against OS Hangs*
Detecting OS hangs and rebooting can be done using the IPMI watchdog timer, since the IPMI BMC firmware is separate from the main CPU that runs the OS. The key feature for this case is how to reliably ensure that the watchdog timer is configured and regularly reset.

4. *Remote IPMI LAN Management for System Monitoring and Control*
The various IPMI functions, including IPMI Serial-Over-LAN access to the console, can be performed remotely if the OS is down, provided that the target system has IPMI LAN and SOL configured beforehand. The two key features for this use case are: (a) the ability to reliably configure IPMI LAN+SOL; and (b) easy-to-use client utilities or scripts for common functions.

5. E*nterprise Management Station, with SNMP as a Baseline and IPMI LAN access*
The key feature for this use case is configuration of the IPMI Platform Event Filter rules and its SNMP alert destination.

6. *Managing a Clustered Server Configuration or HPC Environment*
IPMI functions can also be used in clusters to determine the health of a server, to send a heartbeat via IPMI LAN, or to remotely force a power-down on a clustered node with degraded conditions via IPMI LAN. Also, clustering software can use the FRU asset tag to uniquely mark systems that may have the same IP address.

So to summarize, each use case will contain certain important differentiating features among all the many IPMI features. For example, interpreting IPMI event information accurately and completely from the SEL is important to case 1 and, consequently, to all use cases. Having an existing, reusable event monitoring daemon makes use cases 2 and 4 much easier. Reliably configuring IPMI LAN and SOL would make use cases 4, 5, and 6 easier. Implementing an automatic IPMI PEF and SNMP alert configuration makes use case 5 easier, and relying on flexibility and automation for the IPMI watchdog timer makes use case 3 easier.

IPMI Tool Comparison

Below is an outline of the four open-source tools in question. For a feature comparison chart of these IPMI tools see Appendix A, and then Appendix B for the key differences between each tool.

Advantages of IPMITOOL

The ipmitool package is the most commonly found tool and is quite flexible, so it usually comes with a given Linux distribution. It was started on Solaris, so it is more thoroughly tested on the older Solaris 8 & 9. It has a BSD license, which allows it to be included in either open or proprietary projects, and it has been thoroughly tested on ATCA bladed systems. It has an 'ipmievd' event daemon service, which makes use case 2 easier. The ipmitool package has easily understandable text interpretations for SEL events, but if it does not understand an event, it only shows 'Unknown', which may, in some cases, be due to when the sensor data is an OEM type. However, the raw data can also be accessed in such a case. In summary, ipmitool is desirable because it is usually already distributed with the Linux OS media, and contains the key functionality needed to implement use cases 1, 2, and 4.

Advantages of IPMIUTIL

The ipmiutil package is intended to provide easy-to-use IPMI management. For instance, configuring a variety of IPMI LAN/SOL/PEF servers can be done with one command, making use case 4 easier. It supports managing Windows IPMI servers. It includes an event daemon service, recognizing more IPMI event types, including a severity with all events, and makes use case 1 and 2 easier. It includes a driverless interface for KCS and SSIF, which is useful for out-of-service boot media, and supports detailed watchdog timer configuration and automation, which makes use case 3 easier. PEF functions are supported which makes use case 4 more ideal, while the writing of the FRU asset tag ties it to use case 5. In summary, ipmiutil is the right choice if you need to manage Windows IPMI servers. It has a superset of the features of ipmitool, and it contains the key functionality needed to implement use cases 1, 2, 3, 4, 5, and 6.

Advantages of FREEIPMI

You would want to use freeipmi if you are rolling out an HPC project that requires a granular API for IPMI, or if you need some of the OEM-specific add-ons provided in it. It includes a driverless interface for KCS and SSIF, which is useful for out-of-service boot media. However, it does not come with an existing event daemon, but one could be written with its API. It is not as well suited for proprietary projects because it has a GPL license. The freeipmi package contains the key functionality needed to implement use cases 1, 3, 4, 5, and 6, except for writing FRU data in case 6.

Advantages of OPENIPMI driver & library

You would almost always use the OpenIPMI driver if running Linux, but its library would only be needed if you wanted to create a custom Linux application for an unusual hardware platform, or perhaps for detailed firmware testing. The OpenIPMI driver provides a watchdog module for basic watchdog functions. The Linux OpenIPMI library could be used by a developer to implement any of the use cases, but it is not intended to have those functions already implemented.

IPMI Moving Forward

Server management architects have a powerful weapon in these open source IPMI tools for use in a variety of use cases and server solutions. Understanding server management requirements and what functionality is already implemented in these tools enables integrators to simply pick the best suited tool and quickly put it to good use, as the implementation effort has already been done.

Understanding and using the available IPMI software ensures server monitoring and control to be implemented cleanly across a large number of different vendors' server platforms. This reduces the lead time to introduce new platforms, increases server availability, and provides source access to make custom changes or integrate into a larger framework, when desired.

APPENDIX A – Feature Comparison Chart * Comparing Packages from Common IPMI Open-Source Projects

	ipmitool	ipmiutil	freeipmi	Openipmi
Web Site	ipmitool.sourceforge.net	ipmiutil.sourceforge.net	www.gnu.org/software/freeipmi	sourceforge.net/projects/openipmi
Key Strengths	ATCA blades, IPMI feature coverage, plugins	Server platforms, easy to use, detection, portability	IPMI conformance, ipmi detail*2	Linux driver, Library API, and basic ipmi_ui
OS Support	Linux, Solaris, BSD, *Windows**1	Linux, Windows, Solaris, BSD, EFI	Linux, BSD, Solaris, Windows*1	Linux
Supported Drivers	*Linux:* openipmi, imb, free, *Solaris:* bmc, lipmi, *Any:* lan, lanplus	*Linux:* openipmi, imb, free, valinux, landesk, or driverless KCS/SSIF, *Windows:* Intel imbdrv.sys or MS ipmidrv.sys, *Solaris:* bmc, *BSD:* openipmi or driverless, *Any:* lan, lanplus	*Linux/BSD:* openipmi or driverless KCS/SSIF, *Solaris:* bmc., *Any:* lan, lanplus	*Linux:* openipmi
Target Market	administrators, developers, OEMs	administrators, developers, OEMs	administrators, HPC	Kernel (driver), openhpi (lib)
Linux Distros that have included it	Red Hat, SuSE, Debian, Gentoo, Ubuntu	Red Hat (FC14), SuSE (os11.4), Gentoo, Red Flag, MontaVista	Red Hat (EL6), Debian, Gentoo	Red Hat,SuSE, Debian, Gentoo, MontaVista, etc.
License	BSD	BSD	GPL	GPL for driver & LGPL for library
Watchdog	No	Yes, via wdt, and automated with ipmiutil_wdt svc	Yes	Yes, via /dev/watchdog
PEF control	Some, manual.	Yes, automatic and manual, adds more PEF rules.	Yes, manual via ipmi-pef-config	Write existing only via lib/ui
LEDs	Yes, identify and other LEDs for Sun platforms	Yes, identify and other LEDs for Intel, Sun, Kontron and Fujitsu.	Yes, identify and other LEDs for IBM, Sun, and Fujitsu.	No.
SOL console app	Yes	Yes	Yes, via ipmiconsole 0.7.4 or later	Yes, via sample/solterm
Remote IPMI soft-shutdown	Yes, if chassis power soft is supported by BMC	Yes, (reset -o) with chassis soft-off, else with igetevent -a bridge agent	Yes, if ipmipower --soft is supported by BMC	No
IPMI raw commands	Yes, via raw & i2c	Yes, via cmd	Yes, via ipmi-raw	Yes
LAN parameters	Yes	Yes	Yes	Yes, via ipmi_ui/lanparm
Serial parameters	Yes	Yes	Yes, via bmc-config -v	No
Power control	Yes via power or chassis	Yes, via reset	Yes, via ipmipower	Yes, via ipmi_ui/mc_reset
Sensors	Yes	Yes, also can set thresholds	Yes	Yes, via ipmi_ui/sensors
FRU Inventory Data	Yes, read	Yes, read & write	Yes, read	Yes, via ipmi_ui/fru
SEL	Yes, read/clear	Yes, read/clear plus checksel cron script	Yes	Yes, via ipmi_ui/sel
User parameters	Yes	Yes, with lan/serial	Yes, via bmc-config	No
Channel parameters	Yes	Yes, with lan/serial	Yes, via bmc-config	No
Firmware firewall	Yes	Yes	No	No
PICMG/ATCA extended cmds	Yes	Yes	No	No
Embedded shell	Yes, ipmitool shell	Not needed, use OS shell instead.	No	Yes, impish
OEM-specific Addons	Yes: fwum, sunoem, hpm, kontronoem, ekanalyzer	Yes: oem_intel, fwum, sunoem, hpm, oem_kontron, oem_fujitsu, ekanalyzer	Yes: ipmi-oem for dell, fujitsu, ibm, inventec, quanta, sun, supermicro	No
Discovery of IPMI nodes	No	Yes, idiscover	Yes, via ipmi-detect	Yes, via sample/rmcp_ping
Configuration save/restore	No	Yes, via config	Yes, via bmc-config	No
Event daemon	Yes, ipmievd	Yes, ipmiutil_evt (igetevt)	No	No
Reusable library API	Yes, libipmitool	Yes, libipmiutil or ipmiutil.lib (win)	Yes, libfreeipmi	Yes, libOpenIPMI
Software versions used	ipmitool-1.8.11	ipmiutil-2.6.9	freeipmi-0.8.9	OpenIPMI-2.0.18

*1 = ipmitool and freeipmi support Windows as an IPMI LAN remote client if compiled under cygwin. There is also a Windows native port of ipmitool 1.8.8 from Sun that supports Sun KCS driver. The source to that ipmitool binary from Sun is not available.

*2 = "libfreeipmi is a bit more for those who are familiar with the IPMI protocol in detail, not really abstracting away a lot of IPMI details away." - strategy quote from freeipmi maintainer.

APPENDIX B – Key Differences

This table summarizes the key differences in features and functionalities of the common IPMI package options. More detailed examples follow in Appendix C.

IPMI Feature	ipmitool	Ipmiutil	freeipmi	openipmi
Interpreting IPMI Event information from the SEL	sel [list| elist] obscures some data that it does not recognize	sel [-e] recognizes more data, includes a severity for all events, and shows data that it does not recognize.	ipmi-sel shows data that it does not recognize	Only shows raw hex data, does not attempt to recognize or interpret data.
	ATCA blades, IPMI feature coverage, plugins	Server platforms, easy to use, detection, portability	IPMI conformance, ipmi detail*2	Linux driver, Library API, and basic ipmi_ui
Event monitoring daemon	ipmievd waits for events, and shows them with interpretation	ipmiutil_evt waits for events, and shows them with interpretation	Does not include an event daemon	ipmi_ui can wait for events but only displays them in hex to the screen
Reliably configuring IPMI LAN and SOL on the target	Requires many commands. The bmclanconf script configures 14 of about 50 relevant parameters, which works ok if the factory defaults are set appropriately.	Can configure a working IPMI LAN configuration with just one command, and sets defaults for all relevant parameters.	Linux/BSD: openipmi or driverless KCS/SSIF, Solaris: bmc., Any: lan, lanplus	Linux: openipmi
	Dumps parameters to a file, which can be edited and restored.	administrators, developers, OEMs	administrators, HPC	Kernel (driver), openhpi (lib)
	Requires many commands with specific IPMI knowledge	Red Hat (FC14), SuSE (os11.4), Gentoo, Red Flag, MontaVista	Red Hat (EL6), Debian, Gentoo	Red Hat,SuSE, Debian, Gentoo, MontaVista, etc.
	BSD	BSD	GPL	GPL for driver & LGPL for library
Configuring IPMI PEF rules and SNMP alert destination	Supports showing PEF parameters, but not configuring them, except via raw commands.	Sets PEF parameters as options to the lan configuration, detects OS SNMP configuration, includes default PEF rules and custom rules.	Saves and restores a PEF configuration file that can be edited.	Yes, via /dev/ watchdog
	Uses raw, interactive commands for PEF configuration.	Yes, automatic and manual, adds more PEF rules.	Yes, manual via ipmi-pef-config	Write existing only via lib/ui
IPMI watchdog timer handling	Does not include IPMI watchdog support	Includes a watchdog service via cron daemon; if it cannot launch tasks, the watchdog will expire.	Includes a watchdog daemon; if it gets hung, the watchdog will expire.	Includes an ipmi_watchdog kernel module which enables /dev/ watchdog, it can be automated. If the kernel hangs, the watchdog will expire.
Setting IPMI FRU asset tag	Only supports FRU save/ restore	Supports setting asset tag and FRU save/restore	No	No

See Appendix C online for specific examples of how each tool handles the various functions from the use cases framed above in Appendix B.

GLOSSARY

API	Application Programming Interface
ATCA	Advanced Telecommunications Computing Architecture (http://en.wikipedia.org/wiki/Advanced_Telecommunications_Computing_Architecture)
BSD	Berkeley Software Distribution, referring to either the BSD operating system software (http://en.wikipedia.org/wiki/BSD), or to the license under which BSD is covered (http://en.wikipedia.org/wiki/BSD_licenses).
FRU	IPMI Field Replaceable Unit, detailed inventory information stored in firmware
GPL	GNU Public License, see http://en.wikipedia.org/wiki/GPL
HPC	High Performance Computing - usually CPU-oriented compute clusters, see http://en.wikipedia.org/wiki/High-performance_computing
HPI	Hardware Platform Interface (http://en.wikipedia.org/wiki/Hardware_Platform_Interface)
HPM	The PICMG HPM.1 Hardware Platform Management IPM Controller Firmware Upgrade Specification. See http://www.picmgeu.org/specs/available_specifications.htm, and this article
http://rtcmagazine.com/articles/view/100873	Robust BIOS flash Update with rollover capability (HPM.1); Fail safe field updateable BIOS
IPMI	Intelligent Platform Management Interface (http://www.intel.com/design/servers/ipmi/ipmi.htm)
Over 210 companies have adopted IPMI for their platforms as of September 2010; see http://www.intel.com/design/servers/ipmi/adopterlist.htm for a list.	Console redirection to serial port (VT100)with CMOS setup access, and SOL (Serial over LAN)
KVM	Keyboard, Video, Mouse, a device that enables access to the keyboard, video monitor and mouse, either through a switch or over the network.
LAN	Local Area Network
LED	Light Emitting Diode, usually lights on a system control panel.
lm_sensors	Linux monitoring of hardware sensors, usually raw access to the sensor hardware. See http://www.lm-sensors.org/.
NEBS	Network Equipment Building System, see http://www.telcordia.com/services/testing/nebs/
OEM	Original Equipment Manufacturer, the vendor who manufactures the server
OS	Operating System
PEF	IPMI Platform Event Filter, a method for firmware to take action based on the filter rules.
PICMG	PCI Industrial Computer Manufacturers Group (http://en.wikipedia.org/wiki/PICMG)
SEL	IPMI System Event Log, a non-volatile firmware log
SNMP	Simple Network Management Protocol (http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol)
SOL	IPMI Serial-Over-LAN, a function to redirect the serial console over IPMI LAN
	Standard IPMI Watchdog for all CPU running phase (BIOS execution / OS loading and running).
	IPMI Hardware system monitor (power/voltages), memory and all critical components temperature is monitored.
	Extensive sensors monitoring (around 100 IPMI sensors) and event generation base on thresholds and discrete reading.

About Kontron

Kontron designs and manufactures embedded and communications standards-based, rugged COTS and custom solutions for OEMs, systems integrators, and application providers in a variety of markets. Kontron engineering and manufacturing facilities, located throughout Europe, North America, and Asia-Pacific, work together with streamlined global sales and support services to help customers reduce their time-to-market and gain a competitive advantage. Kontron's diverse product portfolio includes: boards & mezzanines, Computer-on-Modules, HMIs & displays, systems & platforms, and rugged & custom capabilities.

Kontron is a Premier member of the Intel® Embedded Alliance and has been a VDC Platinum Vendor for Embedded Computer Boards 5 years running.

Kontron is listed on the German TecDAX stock exchange under the symbol "KBC".

For more information, please visit: **www.kontron.com**

CORPORATE OFFICES

Europe, Middle East & Africa
Oskar-von-Miller-Str. 1
85386 Eching/Munich
Germany
Tel.: +49 (0)8165/ 77 777
Fax: +49 (0)8165/ 77 385
sales@kontron.com

North America
14118 Stowe Drive
Poway, CA 92064-7147
USA
Tel.: +1 888 294 4558
Fax: +1 858 677 0898
sales@kontron.com

Asia Pacific
17 Building,Block #1,ABP.
188 Southern West 4th Ring Road
Beijing 100070, P.R.China
Tel.: + 86 10 63751188
Fax: + 86 10 83682438
knc@kontron.cn