Using IPMI and CIM
A Partnership for Powerful Platform Management

Tom Slaight & Arvind Kumar

Server Management Architects
Enterprise Platforms Group
Intel Corporation
Agenda

- Introduction to IPMI
- New system directions for IPMI
- IPMI future directions
- IPMI mapping to CIM
- Usage Scenarios
The Instrumentation Challenge

Today’s solution stacks work - 

But not across platforms
IPMI

- Defines a standardized, abstracted, message-based interface to intelligent platform management hardware
- Defines standardized records for describing platform management devices and their characteristics

Promoters: Intel, HP, NEC, Dell

Adopters: 148 and growing

http://developer.intel.com/design/servers/ipmi
Where it fits...

Complements existing management standards

- STANDARD I/F (e.g. CIM-XML)
- STANDARD Schema (e.g. CIM)
- STANDARD SP I/F (e.g. Pegasus C-API, WMI, SAF-HPI)

Management S/W Standards

- Management Applications
- Managed Objects
- Object Interface
- Instrumentation Provider
- IPMI I/F Code

IPMI

IPMI H/W I/F

Baseboard Mgmt. Controller and monitoring h/w
The IPMI Solution

Common Hardware Interfaces and Abstraction -

Enable Cross-platform Management Software
IPMI v1.5 Architecture

- **Remote Mgmt. Card**
- **Bridge Controller**
- **ICMB**
- **Aux. IPMB**
- **Baseboard Mgmt. Controller (BMC)**
- **Satellite Mgmt. Controller**
- **System Interface**
- **Chassis**

**LAN**

**MODEM / Serial**

**RS-232**

**I²C/SMBus**

**SMBus/PCI Mgmt. Bus**

**IPMB (I²C)**

**IPMI Messages**

June 16-19  San Jose, California
IPMI v2.0 Roadmap

<table>
<thead>
<tr>
<th>2003</th>
<th>2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Q2</td>
</tr>
</tbody>
</table>

**IPMI v1.5 Today**
- Monitoring (temp, volt, fan, etc.)
- Control (power on/off/cycle, reset, diag. interrupt)
- System Event Logging
- FRU & SDR Information
- Watchdog Timer
- Serial and LAN access
- Serial and LAN alerts
- Platform Event Filtering
- Serial Port sharing
- Internal/ext’l mgmt busses

**v2.0 Proposed Additions**
- Serial redirection over LAN
- Terminal mode extensions (improved ‘CLI’)
- Alignment with ASF Authentication
- Encryption support
- Modular (blade) support
- IPMI over Web (may be post 2.0)

IPMI continues to evolve valuable new capabilities

DMTF 2003 Global Management Conference

Copyright © 2003 Global Management Conference

June 16-19 San Jose, California
ASF/IPMI Typical Applications

Back-end Servers (Database, App, Storage)

Mid-Tier Servers (File, Application, Storage)

Front-end Servers & Appliances (DNS, Web, Mail, File, Print)

Workstations

Desktop

Mobile

IPMI and ASF are complementary and cooperating technologies
IPMI and CIM

Common interface for in-band and out-of-band platform management
IPMI SW components

1. SDR defines ‘how’ to interpret the raw data provided by the BMC
2. FRUs for Asset and Inventory Management
3. System Event Log for collection of platform events
4. Sensors & Probes get the raw hardware readings

Sensor description can change independent of sensor access. Facilitates customization, H/W and F/W reuse.
IPMI mapping to CIM

- Map IPMI domain to CIM domain
- No attempt to expose IPMI constructs in CIM
- CIM application is unaware of IPMI
- Self describing hardware interfaces make IPMI CIM provider hardware agnostic

CIM is the interface for management applications
Sensors & Devices

**LogicalDevice**
- (See Device Model)
  - AssociatedSensor

**Sensor**
- SensorType : uint16 {enum}
- PossibleStates : string
- CurrentState : string
- PollingInterval : uint64 {units}

**NumericSensor**
- BaseUnits : uint16 {enum}
- UnitModifier : sint32
- RateUnits : uint16 {enum}
- CurrentReading : sint32
- NominalReading : sint32
- NormalMax : sint32
- NormalMin : sint32
- MaxReadable : sint32
- MinReadable : sint32
- Resolution : uint32
- Tolerance : sint32
- Accuracy : sint32 {units}
- IsLinear : boolean
- Hysteresis : uint32
- LowerThresholdNonCritical : sint32
- UpperThresholdNonCritical : sint32
- LowerThresholdCritical : sint32
- UpperThresholdCritical : sint32
- LowerThresholdFatal : sint32
- UpperThresholdFatal : sint32
- SupportedThresholds : uint16 [ ]
- EnableThresholds : uint16 [ ]
- SettableThresholds : uint16 [ ]
- RestoreDefaultThresholds() : uint32

**PowerSupply**
- (See Device Model)
  - AssociatedSupplyCurrentSensor
  - AssociatedSupplyVoltageSensor

**CurrentSensor**

**VoltageSensor**
- **Tachometer**

**TemperatureSensor**

**Processor**
- Role : string
- Family : uint16 {enum}
- OtherFamilyDescription : string
- UpgradeMethod : uint16 {enum}
- MaxClockSpeed : uint32 {units}
- CurrentClockSpeed : uint32 {units}
- DataWidth : uint16 {units}
- AddressWidth : uint16 {units}
- LoadPercentage : uint16 {units}
- Stepping : string
- UniqueID : string
- CPUStatus : uint16 {enum}

**WatchDog**
- MonitoredEntity : uint16 {enum}
- MonitoredEntityDescription : string
- TimeoutInterval : uint32 {units}
- TimerResolution : uint32 {units}
- TimeOfLastExpiration : datetime
- MonitoredEntityOnLastExpiration : uint16 {enum}
- ActionOnExpiration : uint16 {enum}
- KeepAlive() : uint32

**PhysicalElement**
- (See Core Model)

Also See
- CIM_FRU
- CIM_Chassis
- CIM_PhysicalMemory
- CIM_PhysicalElement
- CIM_ComputerSystemPackage
- CIM_MessageLog
- CIM_LogRecord
- Redundancy classes

IPMI provides comprehensive data to populate CIM Model
Implementing Discrete/Digital Sensors

SDR (Type 01/02)

- Sensor Type information defines the kind of sensor (Temperature, Chassis Intrusion, Fan etc.)
- EntityId allows to associate a Sensor with the Device it is monitoring (CIM_AssociatedSensor)
- Event Reading/Type Code defines PossibleStates

IPMI allows straightforward sensor mapping to CIM
Implementing Numeric Sensor

SDR (Type 01/02)

CIM_NumericSensor

- SensorType
- OtherSensorTypeDescription
- PossibleStates
- CurrentState
- BaseUnits
- UnitModifier
- RateUnits
- CurrentReading
- NominalReading
- NormalMax
- NormalMin
- MaxReadable
- MinReadable
- LowerThresholdNonCritical
- UpperThresholdNonCritical
- LowerThresholdCritical
- UpperThresholdCritical
- LowerThresholdFatal
- UpperThresholdFatal
- SupportedThresholds
- EnabledThresholds
- SettableThresholds
- RestoreDefaultThresholds()
IPMI provides strong support for CIM associations
### More CIM Schema population

<table>
<thead>
<tr>
<th>IPMI</th>
<th>CIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Namespace, SystemIdentification, SystemInNamespace, OOBAlertService, ManagementController, DiagnosticTest</td>
</tr>
<tr>
<td>Physical Characteristics</td>
<td>PhysicalFrame, PhysicalPackage, Chassis, Card, PhysicalConnetor, Slot, PackageInConnetor, ConnectorOnPackage, PhysicalMemory etc.</td>
</tr>
<tr>
<td>Devices</td>
<td>CoolingDevice, Fan, PowerSupply, Tachometer, AlarmDevice, Processor, Memory, Chassis,</td>
</tr>
<tr>
<td>Sensors</td>
<td>Voltage, Temperature, Tachometer (FAN speed), Watchdog, ChassisIntrusion</td>
</tr>
<tr>
<td>Field Replaceable Units</td>
<td>CIM_FRU</td>
</tr>
<tr>
<td>Platform Event Trap (PET)</td>
<td>CIM Indication (OOB through Proxy)</td>
</tr>
<tr>
<td>System Event Log (SEL)</td>
<td>MessageLog, LogRecord</td>
</tr>
<tr>
<td>Power Control</td>
<td>Methods on CIM_System</td>
</tr>
<tr>
<td>Boot Options</td>
<td>SettingsData</td>
</tr>
<tr>
<td></td>
<td>(Specific subclassing/enhancements in works)</td>
</tr>
</tbody>
</table>

*The list is only a sample, and not intended to be exhaustive*

**IPMI comprehends a major portion of System & Devices model**
Example Scenario

1. Setup Watchdog, BMC starts monitoring
Example Scenario

Management Applications

STOP

IPMI I/F Code

IPMI H/W I/F

Baseboard Mgmt. Controller and monitoring h/w

CIM Object Manager

Proxy Provider

CIM Object Manager

Proxy Provider

IPMI I/F Code

IPMI I/F Code

2. OS Fails, BMC sends a Platform Event Trap
Example Scenario

3. Proxy Provider converts PET into a CIM_Indication
Example Scenario

4. Boot the system to a Diagnostic Partition
Example Scenario

5. Damaged Disk Partition Recovered, Reboot OS
Example Scenario

Management Applications

- CIM Object Manager
- Instrumentation Provider
- IPMI I/F Code

IPMI H/W I/F

Baseboard Mgmt. Controller and monitoring h/w

Proxy Provider

IPMI I/F Code

OOB I/F

6. OS/Application back up on-line
Summary

- IPMI is the foundation for powerful platform management
- CIM/WBEM is a powerful management model and interface to applications
- IPMI and CIM/WBEM together provide standards based comprehensive management solution for enterprise platforms
## Glossary

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC</td>
<td>Baseboard Management Controller (an IPMI Controller on Baseboard)</td>
</tr>
<tr>
<td>FRU</td>
<td>Field Replaceable Unit. A field replaceable component such as a board, module, fan, power supply, etc.</td>
</tr>
<tr>
<td>ICMB</td>
<td>Intelligent Chassis Management Bus. The ICMB provides a dedicated management bus that enables delivering IPMI messages and alerts between multiple host and peripheral chassis.</td>
</tr>
<tr>
<td>IPMB</td>
<td>Intelligent Platform Management Bus. Name for the architecture, protocol, and implementation of a special bus that interconnects the baseboard and chassis electronics and provides a communications media for system platform management information.</td>
</tr>
<tr>
<td>IPMI</td>
<td>Intelligent Platform Management Interface. IPMI defines a common, abstracted, and self-descriptive interface for platform management hardware that monitors server characteristics such as temperature, voltage, fans, power supplies, and chassis.</td>
</tr>
<tr>
<td>OOB</td>
<td>Out-of-Band. System platform management access that does not involve going through the OS or other software running on the main processors of the managed system.</td>
</tr>
<tr>
<td>PEF</td>
<td>Platform Event Filtering. A feature in IPMI that enables the BMC to generate a selectable action (e.g. power on/off, reset, send Alert, etc.) when a configurable event occurs on the management system.</td>
</tr>
<tr>
<td>SDR</td>
<td>Sensor Data Record. SDRs provide the information that tells management software what sensors, events, management controllers, and FRU information is available from a given IPMI implementation.</td>
</tr>
<tr>
<td>SEL</td>
<td>System Event Log. A non-volatile storage area and associated interfaces for storing system platform event information for later retrieval.</td>
</tr>
</tbody>
</table>
Using IPMI and CIM
A Partnership for Powerful Platform Management

Arvind.Kumar@intel.com
Tom.Slaight@intel.com

Questions?

For Further Information
http://developer.intel.com/design/servers/IPMI
http://www.dmtf.org
Backup
<table>
<thead>
<tr>
<th>Platform Management Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Health and Security Alerts</strong></td>
</tr>
<tr>
<td>• Health Alerts (Temperature, Voltage, Fan, etc. 128 definable alerts)</td>
</tr>
<tr>
<td>• Security Alerts (Chassis Intrusion, LAN Heartbeat, System PWD Violation)</td>
</tr>
<tr>
<td>• BIOS Messages and Alerts</td>
</tr>
<tr>
<td>• OS Hung Watchdog Timer</td>
</tr>
<tr>
<td><strong>Authenticated Remote Control</strong></td>
</tr>
<tr>
<td>• Processor Missing (startup Watchdog)</td>
</tr>
<tr>
<td>• Power up/down/cycle/reset with Boot and Boot Path Options</td>
</tr>
<tr>
<td><strong>Status Info</strong></td>
</tr>
<tr>
<td>• System State</td>
</tr>
<tr>
<td>• System ASF Capabilities</td>
</tr>
<tr>
<td>• Presence Ping/Pong (managed system discovery)</td>
</tr>
<tr>
<td><strong>Monitoring</strong></td>
</tr>
<tr>
<td>• Abstracted Local and Remote Monitoring</td>
</tr>
<tr>
<td>• Health Sensor Present Readings (temperature, voltage, fan, etc.)</td>
</tr>
<tr>
<td><strong>Inventory and Logging</strong></td>
</tr>
<tr>
<td>• FRU/Asset Inventory Data Access</td>
</tr>
<tr>
<td>• System Event Logging</td>
</tr>
<tr>
<td><strong>Extended Out-of-Band Access</strong></td>
</tr>
<tr>
<td>• Serial/Modem Access</td>
</tr>
<tr>
<td>• Text-based Access</td>
</tr>
<tr>
<td>• Inter-chassis Access</td>
</tr>
<tr>
<td>• Multi-user and multi-level Security for remote access</td>
</tr>
<tr>
<td><strong>Extended Alerting</strong></td>
</tr>
<tr>
<td>• Alerts to multiple destinations</td>
</tr>
<tr>
<td>• Alerts and Paging via Modem</td>
</tr>
<tr>
<td><strong>Automatic Actions</strong></td>
</tr>
<tr>
<td>• Platform Event Filtering (PEF) - configurable, event-based automatic recovery and alerts</td>
</tr>
<tr>
<td><strong>Scalability / Extensibility</strong></td>
</tr>
<tr>
<td>• Utilizes Independent BMC</td>
</tr>
<tr>
<td>• “Unlimited” Events and Sensors</td>
</tr>
<tr>
<td>• Extensible Sensor/Event Busses</td>
</tr>
</tbody>
</table>