OPEN
Compute Project

Open Hardware Management Proposed Specification For Remote Machine Management Version 1.01

Revision History

Date	Name	Description
2/24/2014	Hemal Shah, Doug Hughes	Scrub of version 1.0 spec.

Contents

1. Summary

Scale-out computing requires a small and stable set of tools to remotely manage machines. In this specification, the strategy is to define these tools by looking at the needs of a scale-out/stateless machine system administrator to provision and troubleshoot machines. These use cases tease out the actual requirements and prevent non-essential features from entering specifications.

The specification is divided into five sections:

1. Approach
2. Scale-out / Stateless Machine System Administrator (SA) Tasks and recommended best practices
3. Remote Machine Management Specification
4. Commands
5. Glossary/Abbreviations

Upon adoption of the specification, a compliance validation suite will be produced.

Examples of scale-out SA tasks are:

- Initial machine provisioning
- Reset machine BMC to defaults
- Specify boot order
- Fix machine misbehavior
- Recovering a hung machine
- Understanding power and cooling
- Security and authorization
- Inventory management
- Perform machine crash analysis
- Perform firmware management

For each scale-out task, there are recommended tool/functional best practices that will be used with existing standards. As well, there are some infrastructure requirements implicit in these best practices. Please note that when matching the recommendations, some minor aspects may be dropped, or some functions may have more features than required.

For the specification part, the approach is to look at existing standards and leverage them. There are three relevant open specifications: IPMI 2.0, DCMI 1.5 and DMTF/SMASH 2.0. Both standards have features to cover use cases beyond the basic machine management use cases for scale-out computing. Initially, a subset of DCMI 1.5

is used in this specification, since a mixture of mandatory and optional features of DCMI 1.5 cover the scale-out SA's requirements.

This specification uses DCMI 1.5 and IPMI 2.0 to specify how functionality is addressed and leaves the implementation details to the system providers. As long as an implementation passes the compliance validation suite, this specification does not dictate the implementation to be available in a specific system management environment (e.g. in-band or out-of-band).

Special thanks are extended to the many people who helped contribute to this specification though email, conference call and individual calls. As well, representatives of IPMI 2.0/DCMI 1.5 and DMTF were very generous and gracious with their time. This could not have been done without all of you.

2. License

As of April 7, 2011, the following persons or entities have made this Specification available under the Open Web Foundation Final Specification Agreement (OWFa 1.0), which is available at http://www.openwebfoundation.org/legal/the-owf-1-0-agreements/owf-contributor-license-agreement-1-0---copyright-and-patent.

Facebook, Inc.

You can review the signed copies of the Open Web Foundation Agreement Version 1.0 for this Specification at http://opencompute.org/licensing/, which may also include additional parties beyond those listed above.

Your use of this Specification may be subject to other third party rights. THIS SPECIFICATION IS PROVIDED "AS IS." The contributors expressly disclaim any warranties (express, implied, or otherwise), including implied warranties of merchantability, noninfringement, fitness for a particular purpose, or title, related to the Specification. The entire risk as to implementing or otherwise using the Specification is assumed by the Specification implementer and user. IN NO EVENT WILL ANY PARTY BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION OR ITS GOVERNING AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3. Approach

Our approach to delivering and maintaining remote management projects are:

- Categorize the remote management capabilities into two sets:
  1) *Base* – that must be present in all machines that implement this specification and
  2) *Optional* – that may vary from platform to platform but will always have a uniform interface. The intent of this specification is to provide the minimum requirements for remote machine management. Optional features will potentially be the focus of a revision to this standard.

- The Base functions will be modeled against what is absolutely necessary for system administration and management for scale-out and stateless node requirements. All other features will be targeted to an OS agent or orchestration layer to reduce complexity and cost in the BMC.

- Any solution must be uniformly implemented across platforms and processor architectures including but not limited to x86 and ARM.

- Use existing remote management technologies and implementations to determine the best technology to leverage. Identify gaps between Remote Hardware Management and existing specifications:
  - DMTF – SMBIOS, CIM, WBEM, SMASH, WS-Management
  - SNIA – SMI-S
  - Intel with Partnerships – DCMI
  - Intel/HP/Dell/NEC - IPMI

- Provide references to command line tools and API/interfaces for remote machine management functions.

- Coordinate with other Open Compute tracks – especially motherboard, rack and data center designs.

- As part of the every deliverable, a methodology to validate the functionality and maintain validity thereof must be included.

- As it is common to have some non-scale-out platforms in a scaled environment, encourage non-OCP hardware manufacturers to comply with Open Machine Management and to submit those platforms for validation.

- For every referenced tool or function, make sure that security is considered to ensure that systems are not compromised by using these tools.

4. Scale-out – Stateless Machine System Administration Tasks

Remote management's base requirements will be established by using the most flexible and minimal administration for scale-out compute machines – stateless nodes. Stateless compute nodes are defined as:

- Stateless nodes are used in groups where node failure recovery is part of the architecture.
- Administrative requirements are uniform across stateless nodes.
- Pools of stateless nodes typically have very similar (or identical) OS/software distributed across similar pools of hardware so software faults are either epidemic or "random".
- The most common strategy for fixing faulted stateless nodes is either rebuilding the OS/software or replacing the node hardware.

Because of this, scale-out stateless node administration requirements are light and any scale-out platform will need the same capabilities.

In the section below, the requirements needed for scale-out stateless nodes will be discussed. These are capabilities such as machine provisioning, faults, inventory, power and temperature. .

Recommendations are summarized in the table below, in accordance with best practices, and discussed in more detail in the remainder of this chapter.

Base Requirements	Required Specification
Management Node Identifier	Must exist and be unique
Rights and Credentials	Access to Remote Management uses industry Cipher Suites; Recommend these be set to the highest practical level
Boot Order / PXE	Set boot order; and all network ports capable of PXE boot
Serial over LAN	Yes
Power On/Off/Hard Reset	Yes
Power Draw	Present power draw
Temperature	Present intake and CPU
User Levels	User (R/O benign); Operator(R/W – Server Admin); Administrator(R/W All – Includes Config and Admin)
Inventory	Asset Tag; Device ID; System GUID; System Manufacturer; Management Firmware; Management Controller ID
System Log	>= 256 entries; Entries focused on failed/impaired FRUs

4.1 Initial Machine Provisioning

Issues

Initial provisioning of a new node requires knowing the machine's Managed Node ID name to begin configuration. Alternatively, that the node has its configuration pre-set by the vendor or integrator. Frequently, scale-out infrastructures find the machine's remote management interfaces via a variety of means but they are often fragile, such as:

- *MAC address and/or other inventory information feeds/bar codes from the machine's manufacturer*
- *Having someone attach a keyboard-video-mouse (KVM) to perform machine inventory and/or setup the machine*

- *Developing heuristics and searching through IP lease information to find "new" machines that have gotten DHCP leases for their management interfaces.*

Often the default authentication for the interface for each manufacturer is well known. This makes login credentials uniform but many times this information is not changed or inconsistently changed.

Another credential strategy is to have integrators set this to a scale-out consumer's standard. This is not ideal as this can be error prone and makes bringing a new vendor on-board difficult.

Recommendation: Configurable settings for Remote Management functions should be provided. Users can ask their server delivery vendors to alter these standards to their respective defaults. Further, machines with OCP default Remote Management settings shall advertise their availability.

4.2 Management Node Identifier Name

There should be a process that has a unique default name for a machine's BMC and, as part of the configuration, this name is changed to provide a workable and straight forward implementation. Suggestions for default names are:

- *OEM Object Identifier (OID) + serial number*
- *OID + MAC*
- *GUID*

Recommendation: The Management Node Identifier name can be an Enterprise identifier and OEM unique identifier. Example of OEM unique identifiers would be serial number or asset tag. In any event, the combination of the two must result in a unique string.

4.3 Rights and Credentials

There should be consistent initial authentication for the BMC across scale-out machines with two strategies:

1. A standardized password across all OCP machines. Examples could be:
    - Model Number + Serial Number
    - Hash of different read only system identifiers like GUID and MAC.
2. A password that is provided by the scale-out purchaser and is pre-configured.

a. For example, scale-out user XYZ, may desire that all machine coming into their environment have all of their administrator’s account to their remote management interface set to *PasswordXYZ*

For the scale-out and stateless node, requiring LDAP, AD or another directory authentication is too heavy weight for a minimum standard.

It would also be ideal to force a password change after the initial physical installation upon first boot. Not changing the default password makes the platform insecure as it provides an easy way for un-authorized machine access. Although important, this is better left to the orchestration layer rather than including it as part of the standard.

Recommendation: The remote access rights and credentials Management Node Identifier will support authentication, authorization, integrity verification and confidentiality. Best practices dictate that as part of the initial configuration, the default credential must be reset to ensure continued secure operations but this can be accomplished outside the Remote Machine management through scripts/orchestration. Users may work with their server suppliers to change the remote access credentials to their own standard. This is acceptable and tooling around remote management interface configuration should provide for this.

To ensure security access, accepted industry Cipher Suites that are an integration of authentication, integrity verification and confidentiality algorithms will be supported. A summary of these can be found in the DCMI Specification, Section 4.1 Security Access. It is recommended but not mandatory that these be set to the highest practical level.

4.4 Reset machine BMC to defaults

When redeploying a node, there is a need to reset the machine’s Remote Management interface to its default settings.

Recommendation: Resetting the machine’s BMC to defaults is not part of the Base standard. Typically, this is accomplished via removing the power cord. This capability is available in some machines over the network but there are concerns regarding security and vendor proprietary lock-in.

4.5 Correct unknown/wrong boot order or make sure the PXE is first in the boot order

Most scale-out environments leverage PXE for image load and maintenance by loading a maintenance/disposable OS that controls imaging and deployment of a local OS on the system. Without PXE being first in the boot order, the hard drive/cd/etc. will boot with the local OS, and remote maintenance events cannot take place.

There are two options to the boot issue that are widely deployed. This first is to require PXE to always be in the first option in the boot sequence. With that strategy, there is no requirement to have any control of boot options – as less is often best. But, the issues with this are longer boot times and opening to attack by compromising PXE responders. The alternate is to have a remote access method to change boot options. This is possible but will introduce more features into the BMC. It is important to have the mechanism to change the boot order in the BMC and not the host, because the host may not be available during the time of need due to a fault condition.

Ideally, all host network ports will be PXE boot enabled. The advantage to this is that no matter which host network ports are connected; there will always be a successful PXE boot. With the PXE boot and an OS environment, more details about the machine can be discovered and any errors can be corrected.

Recommendation: Include in the Base specification the ability to change the boot order. All network ports on the system must support PXE boot.

4.6 Something odd is happening and there is a need to look at the machine’s serial console

In general, there is a need to understand what is going on with the machine even if the OS is not accessible or available. For example, this is useful for emerging problems or with boot issues after firmware updates. All of this can be accomplished by re-directing the serial console.

Tasks

- A remote serial console is essential for resolving or researching faults or other inconsistent behavior. Examples would be tailing logs, running sar/top and watching boot POST.

- Understanding what resources of a machine are being used for vs. what is installed in the machine can be important. For example, a machine may be operating on 48GB of memory yet there is 64GB of memory physically installed. Inventory is more widely discussed later in the document.

- Monitor the boot process to determine if drivers are loaded or hanging during OS boot.

Recommendation: The Base standard will include a Remote Serial console over LAN that will allow interaction with machines’ pre-boot process and setup, boot process and access to the installed OS through a serial console interface.

4.7 Hung Machine

Issues

Need to hard power off/on/reset without someone visiting the machine or having to purchase smart/ip addressable power strips.

Remotely power on/off/ reset a machine

Need to hard power on, off or hard reset a machine that has a non-responsive operating system.

Recommendation: The Base standard will include the ability to remotely power on, off a machine and perform a “hard reset”. The analogous action would be:

- Power On: turning on a power switch
- Power Off: Turning off the power switch.
- Hard Reset: Similar to a power off/on but without interrupting power.

Signal OS to bring machine down

Having a command to bring down a machine by trying to gracefully shut down is of value. For the graceful shutdown, the complexity comes in defining a common way to pass the OS the message to shut down, a common way to retrieve that message and having corresponding OS support for acting on the message.

Recommendation: Having a function that would first try to bring the machines down from an OS perspective was discussed and it was recommended that this not be part of the Base specification.

4.8 How much power is a server drawing?

Issues

There are a number of reasons to check the power draw for a machine. A few are:

- Understanding minimum and peak workload power draw across a cabinet(s) or group of machines.

- Need to understand glitches for changing workload and/or changes in aggregate workload behaviors.

- Preventing power strip (Multi-Outlet Assembly) circuit breaker tripping

For scale-out nodes, the requirement is to measure present power draw and, if possible, minimum and maximum power draws over a period of time (for example in the past 24 hours).

Recommendation: For the base specification, the present power draw will be immediately available and the daily minimum/maximum power draws will be available for the past 24 hours. The power reading will be for the complete chassis and for all components within it. The sample rate for each system will be one second or less.

Other features, like understanding total kW consumed over a period of time for internal chargeback, power averages over time, historical power information or more extensive logging can be accomplished via an agent or orchestration layer.

4.9 Is the server being cooled effectively?

Issues

As part of the total data center environment, understanding server temperatures is necessary to tune the datacenter and make the most efficient use of cooling. Also, with rising data center temperatures, having good temperature data becomes more important. Finally, during failure situations, understanding temperature is an important data point to know if a server needs to be powered down.

Recommendation: As part of the base standard, the following features should be supported:

- CPU and intake temperatures should be provided.
- These temperatures should be accessible through the remote management interface.
- Temperature readings on the system should have a sampling increment of one (1) second or less.

4.10 Need to provide security around actions

Issue

Many of the remote management actions can be disruptive like power off and boot order changes. User name and password should be used to protect against:

- Malicious attacks (internal or external)
- “User” level administration scripts that inadvertently call commands needing higher privileges.

In general, all accesses to the machines should be governed by credentials, as even the read only attributes may provide enough information to become a security liability.

Recommendation: There will be at least three levels of authentication/authorization:

1) User: Can read benign data like temperature, power state, etc.

2) Operator: Includes User and can change the state of the machine like power down, access the remote serial console, reboot and change boot device order.

3) Administrator: Can alter configuration and rights for Remote Management.

User, Operator and Administrator roles will require:

1) Credentials for all accounts.

2) Credentials to negotiate machine access/accounts.

3) Encryption of commands and responses

4) Ability to toggle Encryption on and off.

5) Security access as described in section 4.3 in the Recommendation section.

Checking for password strength and aging is beyond this specification and best performed by an orchestration layer.

4.11 Need to understand what is in the machine.

Issue

In the event that inventory systems are not correct or incomplete, it is necessary to understand what is in the machine. This is not always possible and some devices are enumerated differently among vendors.

Base Inventory

The BMC will return a minimal amount of information required to remotely configure and administer the machine.

Recommendation: As part of the Base, the base inventory will be:

- Asset Tag
- Device ID
- System GUID

- System Manufacturer
- Management Firmware/Software information
- Management Controller ID

Complete Inventory

In scale-out computing, many nodes have similar or identical hardware information so complete hardware inventory is less critical.

Recommendation: For a complete/extended hardware inventory, remotely managed OCP machines can have an OS based agent that returns more extensive machine information. This agent's specification is out of scope.

4.12 How do I know and diagnose hardware problems?

Issue

In scale-out environments, machines will appear to fault randomly. Although machine failures are architected into scale-out environments, often these faults are on the leading edge of issues. Even a 1% problem over many 10s/100s of thousands of machine is significant.

It is valuable to have support for discriminating hardware failures from software failures so information collected is focused on identifying failed FRUs.

Recommendation: As part of the Base features, a log capable of at least 256 entries should exist that can contain information about hardware faults like NMI, multi-parity error, temperature, power and other machine diagnostic/state information.

Events formats will follow the Platform Event Trap Format (PET) standard until superseded by the OCP Event/Alerts Specification.

4.13 Discussed but not included

The following items were discussed but not included in the Base specification.

- A watch-dog timer and/or server heart beat feature was discussed. This feature is left to a Remote Management agent or orchestration layer.

- Scalable updating of firmware was discussed but will be specified in its separate project.

- This specification explicitly does not cover a single management controller accessing multiple hosts, or multiple management controllers accessing multiple nodes. This will be addressed in later version of the specification.

- One aspect of remote machine management is making the machine known to inventory stores for automation. There were many ideas on this topic. For example, putting a Quick Response Scan (QRS) code shipping materials to eliminate data entry errors or common schemas for sending machine information. Ultimately, this is about supply chain management and would be a fruitful topic for another specification.

- PXE provided many discussions with topics such as PXE failure behavior (retry, reboot, wait), replacing tftp with other transports and general configuration. These new or additional features are BIOS related and may be discussed in a future OCP efforts.

5. Remote Machine Management Specifications

The approach of OCP Remote Machine Management leverages existing specifications. The relevant remote server management specifications are IPMI 2.0, DCMI 1.5 and DMTF's SMASH 2.0. These specifications encompass most of the functionality specified in the previous section. These specifications have more features than required for scale-out systems. Understanding this, the Remote Machine Management specification will accept a "subset" version as a trade-off to quicker or more robust implementation.

After reviewing both standards, for this version of the specification, IPMI 2.0/DCMI 1.5 is being used as a basis for the following reasons:

- For out of band support, when surveying the market for existing chipsets that would support either standard there was IPMI 2.0/DCMI 1.5 silicon available
- For in-band support, there are IPMI drivers available for various operating systems including Linux and Windows. Almost all OEMs either supported IPMI 2.0/DCMI 1.5 or had add-in technology to support this so scale-out scripts can be re-used for non-scale-out maintenance and configuration activities.

5.1 Implementations

5.1.1 Uniform Implementation

Compliance with the standard will be determined through documenting conformance to the validation suite to be created after the final version of this specification is complete

5.1.2 Machine Access

The specification makes a distinction between two specific ways to access the machine management specification:

1. Remote Access: Machine management features are accessed via the network and are included in this specification. For Remote Access there are three models:
    a. No OS dependence: This is typically implemented with a BMC.
    b. Hybrid between hardware and OS: This is a combined approach and may be useful for those who have a relatively uniform OS environment.
    c. OS dependence: Having an OS dependent implementation may seem impractical at this time for the current server population. For example, it would be difficult to capture console boot information or hard power

on purely based on the OS without some additional hardware features. To allow for unique and/or innovative OS dependent implementations and expand this specification for future versions, this model is included in the specification.

Note: OS refers to both standard operating systems as well as hypervisors.

2. Local Access: By definition, this specification is about Remote Management of Machines so there is no discussion on accessing management features locally.

5.1.3 Compliance

After this specification has been approved, a Compliance Suite will be created that will exercise the implementation to ensure interoperability. Compliance will only be determined by a ***Pass*** from the Compliance Suite. When there are differences between the compliance suite and specification, the specification takes precedence. This will be developed in conjunction with the OCP Compliance efforts.

5.1.2.1 Compliance for No-OS solutions

When there is no OS involvement, compliance is determined on a machine basis for all OSs. The BMC and firmware (if applicable) on a motherboard will be certified. For example, FOO BMC Model 4.14 using firmware v 1.2 on BAR's model YYY motherboard.

5.1.2.2 Compliance for Hybrid and OS-only solutions

Compliance will be determined on a hardware platform, OS version basis and agent. For example, compliance would be certified for FOO Computer Model 001 under CentOS v6.3 using IPMI.sys v1.234.

5.2 Standards implementation

- Overall, Interface and protocol needs to be consistent with measurements, such as temperature and electricity, within 1% of tolerance.

5.2.1 IPMI

When citing IPMI, the following document is referenced.

- IPMI v2.0 rev. 1.0 specification markup for IPMI v2.0/v1.5 errata revision 4 (http://download.intel.com/design/servers/ipmi/IPMI2_0E4_Markup_061209.pdf)

5.2.2 DCMI

When citing DCMI, the following document is referenced.

- DCMI Specification version 1.5 (http://www.intel.com/Assets/PDF/prodspec/DCMI_Spec_V1_5_Rev.pdf)

5.2.3 Platform Event Trap (PET)

When citing PET or Platform Event Trap, the following document is referenced. IPMI 2.0 provides additional specifications.

http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/platform-event-trap.pdf

5.3 Specification for DCMI/IPMI Implementations

To avoid replicating the information in the standard, they are referenced by section below. The DCMI 1.5 specification often references the IPMI specifications and this is understood.

5.3.1 General Requirements - DCMI 1.5

- Section 5.3.1.2 LAN Interface Requirements –
    - All requirements in Section 5.3.1.2 apply with the exception of gratuitous ARP.
    - The implementation providing support for a different management VLAN is optional.
    - The definition of “Standby power” will mean that mandatory commands issued to a machine will be executed without regard to the power status of the main chassis.
- Section 5.4.1 Mandatory Requirements (Protocol)
    - All requirements in section 5.4.1 apply with exception of 5.4.1.1 in-band access (where in-band refers to local access)
- Section 5.2 – General Manageability Access Requirements
    - All requirements in section 5.2 apply except references to “in-band” access.
- Section 5.3.1.1 System Interface Requirements

5.3.2 Security

- Security: DCMI 1.5- Section 4.1 Security Access
    - In Table 4-1, the cipher suite ID 17 is mandatory and recommended except where prohibited.
    - In Table 4-1, the cipher suite with ID 8 is mandatory.
- User Privilege: DCMI 1.5 Section 4.2
    - Please take note of table 4-2 that defines three users levels (User, Operator & Administrator) and their capabilities.

5.3.3 Interface Name and Discovery

- DCMI 1.5
  - Section 5.5.1 In-Band Discovery Requirement
    - Applies only if a local (i.e. in-band) interface to the management controller is implemented
  - Section 5.5.2.1 DHCP enabled management controller
  - Section 5.5.2.3 RMCP Ping / Pong

Note: 5.5.2.2 is not included as static IPs is uncommon in scale-out computing.

5.3.4 Event Logging

- DCMI 1.5 Section 3.1.3 DCMI Logging
- PET 1.0 – SNMP Trap Format

5.3.5 Power

- Power on / off / (Hard) Reset: DCMI 1.5 Section 3.1.2
- Chassis Power Draw: DCMI 1.5 Section 6.6.1 Get Power Reading
  - Support for Mode 01h System Power Statistics for reading current power draw is the only mode required.

5.3.6 Inventory

- DCMI 1.5 Section 3.1.1 Identification
- DCMI 1.5 Section 6.4 Identification and Discovery Support

5.3.7 Temperature

- Sensor messages DCMI 1.5 Section 3.1.5 compliant
- Command described in DCMI 1.5 Section 6.7.3 Get Temperature Readings

5.3.8 Boot Control

- Standard: DCMI 1.5 Section 6.7.4

5.4 Commands Support Requirements

The following commands represent a subset of the DCMI 1.5 commands as referenced in Table 6.1 and the respective specifications later in the standards document. Command completion codes are specified in DCMI 1.5 Section 8 – Completion Codes.

In the DCMI standard, there are two sets of command:

- Mandatory - required for DCMI certification

- Optional – not required for DCMI certification but specifies a mandatory implementation and behavior.

This specification takes both DCMI Mandatory and Optional command and makes them part of the Base specification.

In addition to the commands in the table below:

  - All IPMI 2.0 RMCP+ “Session Activation/Termination commands” must be supported as required by ipmitool (including GetChannelAuthenticationCapabilities, OpenSession, RAKP[1-4], CloseSession).
  - All SOL-related session commands must be supported as required by ipmitool (including ActivatePayload, and Set/GetSOLConfigurationParameters)

DCMI Capabilities & Discovery Info	**User**	**Operator**	**Admin**
Get DCMI Capabilities Info	Get	Get	Get
Set & Get DCMI Configuration Parameters	Get Only	Get Only	Set/Get
Set & Get Management Controller Identifier String	Get Only	Get Only	Set/Get
Platform & Asset Identification Commands			
Set & Get Asset Tag	Get Only	Get/Set	Get/Set
Get Device ID	Get	Get	Get
Get System GUID	Get	Get	Get
Boot Control			
Set & Get System Boot Options	n/a	Set/Get	Set/Get
Sensor & SDR Commands			
Get DCMI Sensor Info	n/a	Get	Get
Get Sensor Reading	Get	Get	Get
Logging Command			
Get SEL info	Get	Get	Get
Get SEL Entry	Get	Get	Get
Clear SEL	n/a	Clear	Clear
Power Management			
Get Power Reading	Get	Get	Get
Thermal Management			
Get Temperature Readings	Get	Get	Get
Remote Management			
Set & Get LAN Configuration Parameters	n/a	Get	Set/Get
Set & Get Channel Access	Get	Get	Set/Get
Set & Get User Access	n/a	Get	Set/Get
Set & Get User Name	n/a	Get	Set/Get
Set User Password	n/a	n/a	Set/Get
Set & Get User Payload Access	n/a	Get	Set/Get
Chassis Commands			
Get Chassis Capabilities	Get	Get	Get
Get Chassis Status	Get	Get	Get
Chassis Control	n/a	Set	Set
Chassis Identify	n/a	Set	Set
Get ACPI Power State	Get	Get	Get

5.5 BMCs

The details around the BMC design and placement on the motherboard may vary between motherboards. BMC and board designers can differentiate themselves by optimizing on power.

5.6 Network Interface

The interface to the BMC must be through a network port that is shared by the BMC and the host.

6. Command Reference

The commands below will serve as a reference to the executing the OCP Machine Management functions.   Once the OCP Machine Management specification is approved, `ipmitool` and/or freeipmi will be used for a significant portion of the compliance suite. The commands below serve as examples and are not meant to be exhaustive.

6. 1 Chassis Power management

```
ipmitool: ipmitool -I lanplus -U <username> -P <password> -H < host's OOB > power [status|on|off|cycle|reset]

freeipmi: ipmi-power -h < host's OOB > -u <username> -p <password> [ --on|--off|--cycle|--reset|--stat ]
```

6.2 Remote Serial Console

```
ipmitool: ipmitool -I lanplus -U <username> -P <password> -H < host's OOB > sol activate

freeipmi: ipmiconsole -h < host's OOB > -u <username> -p <password>
```

6.3 Collect system info

6.3.1 user management

```
ipmitool: ipmitool -I lanplus -U <username> -P <password> -H < host's OOB > [list|set name|set password|disable|enable|priv]

freeipmi: bmc-config –checkout
```

6.3.2 sel (Print System Event Log)

```
ipmitool: ipmitool -I lanplus -U <username> -P <password> -H < host's OOB > sel

freeipmi: ipmi-sel -h < host's OOB > -u <username> -p <password>
```

6.3.3. sdr (Print Sensor Data Repository entries and readings)

```
ipmitool: ipmitool -I lanplus -U <username> -P <password> -H < host's OOB > sdr

freeipmi: ipmi-sensors -h < host's OOB > -u <username> -p <password>
```

6.3.4 chassis

```
ipmitool: ipmitool -I lanplus -U <username> -P <password> -H < host's OOB > chassis [bootdev|bootparam]

freeipmi: ipmi-chassis-config -h < host's OOB > -u <username> -p <password> --checkout
```

7. Terms and Abbreviations

Active Directory (AD)

Baseboard Management Controller (BMC) or Management Controller (MC)

A microcontroller or processor that aggregates management parameters from one or more management devices and makes access to those parameters available to local or remote software, or to other management controllers, through one or more management data models.

Common Information Model (CIM)

A specification and schema, defined by the Distributed Management Task Force (DMTF), that provide a common definition of management information for systems, networks, applications and services. For more details, see http://www.dmtf.org/standards/cim.

Data Center Management Interface (DCMI)

A set of specifications, derived from IPMI 2.0, developed by Intel Corporation for the server platform management in large deployments within data centers.

Distributed Management Task Force (DMTF)

A standards organization that *creates standards to enable interoperable IT management.* See www.dmtf.org

Intelligent Platform Management Interface (IPMI)

A set of specifications that define interfaces and protocols originally developed for server platform management by the IPMI Promoters Group: Intel Corporation, Hewlett-Packard Company, NEC Corporation, and Dell Inc.

Light-weight Directory Access Protocol (LDAP)

General Description & Standard: http://tools.ietf.org/html/rfc4511

Media Access Control (MAC)

General Description: http://en.wikipedia.org/wiki/Media_Access_Control

Pre-boot eXecution Environment (PXE)

General Description & Standard:
http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf

Storage Management Initiative Specification (SMI-S)

A set of specifications developed by the Storage Networking Industry Association (SNIA) for the interoperable management of a heterogeneous Storage Area Network (SAN). See http://www.snia.org/tech_activities/standards/curr_standards/smi for more details.

System Management Architecture for Server Hardware (SMASH)

A *set of specifications* that defines *inter*faces and protocols developed by the DMTF for *server platform* management. See http://www.dmtf.org/standards/smash for more details.

Universally Unique Identifier (UUID)

Refers to an identifier originally standardized by the Open Software Foundation (OSF) as part of the Distributed Computing Environment (DCE). UUIDs are created using a set of algorithms that enables them to be independently generated by different parties without requiring that the parties coordinate to ensure that generated IDs do not overlap. Also known as Globally Unique Identifier (GUID).

Web Based Enterprise Management (WBEM)

A *set of management and Internet standard technologies developed* by the DMTF *to unify the management of distributed computing environments.* See http://www.dmtf.org/standards/wbem for more details.

Web Services for Management (WS-Management or WS-Man)

A set of specifications, defining a web services based management protocol developed by the DMTF. See http://www.dmtf.org/standards/wsman for more details.