The Alert Standard Format

ASF Version 2.0

This overivew document introduces the reader to the Distributed Management Task Force (DMTF) Alert Standard Format (ASF) Specification. This note describes the benefits of ASF and explains how to position ASF within a corporate management infrastructure.

Introduction

"System manageability" represents a wide range of technologies for local and remote system access and control. For complete coverage, these technologies must address both operating system-present (OS-present) and operating system-absent (OS-absent) environments. The latter describes scenarios such as when a system is booting or sleeping, or a boot-up is hung.

In the OS-present environment, the DMTF has defined the WBEM (Web-Based Enterprise Management) and DMI (Desktop Management Interface) initiatives and standards. The Alert Standard Format addresses the OS-absent environment. ASF defines remote control and alerting interfaces for computers. The first ASF Specification was released in June 2001, and security was added to the protocols in the 2.0 release in November 2002.

The Alert Standard Format

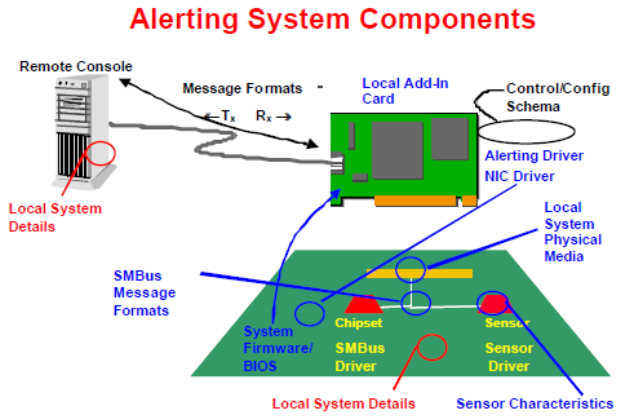
ASF consists of a client or server system (or several systems) heretofore referred to as "client", and a management console that both monitors and controls the client. An ASF-aware computer provides the following interfaces to allow its remote management in an OS-absent scenario:

1. Transmission of messages by the ASF system including system health and security alerts
2. Receipt and processing of remote maintenance requests, sent by the management console
3. Ability to describe the client system's specific capabilities and characteristics to the management console
4. Ability to describe the software used to configure or control the client system in an OS-present state

An additional level of interoperability is needed between the alerting components in a client system:

1. The system firmware must communicate system capabilities to an ASF component's OS-present configuration software.
2. Messaging between the ASF component, the local system host, and local system sensors must be supported.

The following figure provides a graphic of these components:



When an ASF-alerting device (for example, an Ethernet add-in card) is a component of (or added to) a managed client, the alerting device must be configured with the client's specific hardware configuration. This allows it to properly issue alerts and respond to remote maintenance requests. To accomplish this, the client system requires one good boot to an OS-present environment to allow the device's configuration software to run and store system-specific information into the device's non-volatile storage.

In an ACPI-aware (Advanced Configuration and Power Interface) environment, the alerting device's configuration software interrogates the client's configuration data to retrieve necessary information. The following information is collected and placed into the device's non-volatile storage for use in the OS-absent environment:

1. The client's ACPI implementation contains its AS Capabilities, including the IANA Manufacturer ID and System ID.
2. The client's SMBIOS (System Management BIOS) structure-table contains the system GUID (or UUID).
3. The operating system has assigned a TCP/IP address to the ASF-alerting device.
4. The amount of time that the alerting device waits before issuing a system boot-failure alert is configured.
5. The configuration software also provides an interface to allow the system owner to identify the TCP/IP address of the management console to which any alert messages are to be sent.

During this OS-present configuration process, the managed client's optional ASF configuration is also determined and placed into the alerting device's non-volatile storage:

1. If the client includes legacy System Management Bus (SMBus) sensors, the addressing and configuration information for each sensor is recorded.
2. If the client supports remote-control operations, the specific ASF-defined features are stored.

Once the system owner has configured the alerting device, the managed client is enabled to send alert messages and, optionally, respond to remote-control requests from a specified management console.

ASF Network Protocols

ASF uses two network protocols; the first is the Platform Event Trap (PET) to send alerts to the management console, and the second is the Remote Management and Control Protocol (RMCP) to do remote control of the system.

PET

The PET protocol uses a SNMP Trap PDU (IETF's Simple Network Management Protocol Trap Protocol Data Unit) to carry system management information. The alerts cover various low-level system activities. These include:

- Environmental events
- System firmware error & progress events
- CPU error/DOA (dead on arrival) events
- Chassis Intrusion
- OS events
- System heartbeat
- System boot failure

The environmental events defined in the standard include temperature, voltage, and fan problems. The system firmware errors include system memory and hard disk problems, on boot image and option ROM problems, and multi-processor problems. The system firmware progress events provide information on the progress of the boot operation up to the start of the OS boot. The CPU error/DOA events indicate that the host processor has been removed or is not functioning. Chassis intrusion alerts that the system has been tampered with and/or opened. The OS events include OS failure to boot, and OS hang. The system heartbeat event is use to insure that a system is still present in the managed environment.

RMCP

RMCP (Remote Management and Control Protocol) is a UDPbased protocol (User Datagram Protocol) for system control when a managed client is in an *OS-absent* state. In this environment, RMCP packets are exchanged between a management console and a managed client. Typical client control functions include operations such as:

- Reset
- Power-up and power-down
- Reboot to multiple paths

The protocol is intentionally simple, to enable alerting devices' firmware to easily parse the information in the absence of OS-present drivers.

A management console uses RMCP methods as part of a two-tiered approach to managing client systems. The console should always use OS-present methods as the primary method to power down or reset a managed client, so that any shutdown operation is handled in an orderly fashion. Consoles should employ RMCP methods only if the managed client fails to respond to the OS-present methods, since the hardware-based RMCP methods could result in loss of data on the client system.

The Benefits of ASF

ASF allows an IT administrator to proactively and/or reactively respond to a problem on a particular system or set of systems when an operating system is not present. Additionally, ASF technology can make particular tasks like software upgrades or inventory simpler. The following usage examples best illustrate some of these benefits:

1. Invoking ASF technology, the night shift (off-hours) IT Administrator sends a command from their management console, to power on a series of desktop systems. A software patch is sent to these systems utilizing deployment software. Upon notification from the deployment software that the upgrade is complete, the IT Admin again invokes ASF from their console to power down the systems.

1a. An IT Admin receives an ASF "OS Hung" alert during the "power on" process from one of the systems getting the software patch. Again invoking ASF, the Admin sends a "reset" command from their console, and proceeds with the software upgrade.

2. An IT Admin receives a "Chassis intrusion" alert from a server, sitting in a service provider's facility across town. The Admin contacts Security at the facility to follow up.
3. An ASF alert is received by an IT Admin or processed through DMI or CIM, from a server on the opposite side of the campus, suggesting that a fan is not functioning properly. Shortly thereafter, a temperature alert is received. The Admin invokes an ASF "power down" command or a CIM power management service method on the system, preserving the system from failure-induced damage, until technicians can personally attend to the situation.

Closing Remarks

There are many methods available in the marketplace today for managing hardware. ASF was designed specifically to fill the gap of operating system-absent systems management, though it is also operable in operating system-present states. The problem of systems manageability without an operating system, has historically been solved with proprietary and relatively expensive solutions. ASF represents the lowest cost per system, standards-based solution on the market today.