Cisco UCS Monitoring Resource Handbook

Authored by
Eric Williams
Jeff Foster
Jason Shaw

This document is intended to be a central repository of monitoring resources and recommendations for Cisco UCS Manager and the Cisco Standalone C-Series Servers. This content is intended to supplement the 'Demystifying Monitoring for UCS Manager & C-Series' Tech Talk available here: https://communities.cisco.com/docs/DOC-37138

Additional Cisco Monitoring Resources: (Cited within this document)

UCS Manager MIB Reference Guide:
http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/mib/b-series/b_UCS_MIBRef.html

UCS Manager Fault Reference Guide:
http://www.cisco.com/en/US/docs/unified_computing/ucs/ts/faults/reference/UCSFaultsRef.pdf

C-Series MIB Reference Guide:
http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/mib/c-series/b_UCS_Standalone_C-Series_MIBRef.pdf

C-Series Fault Reference Guide:
http://www.cisco.com/en/US/docs/unified_computing/ucs/c/sw/fault/reference/guide/CIMC_Fault_codes.pdf

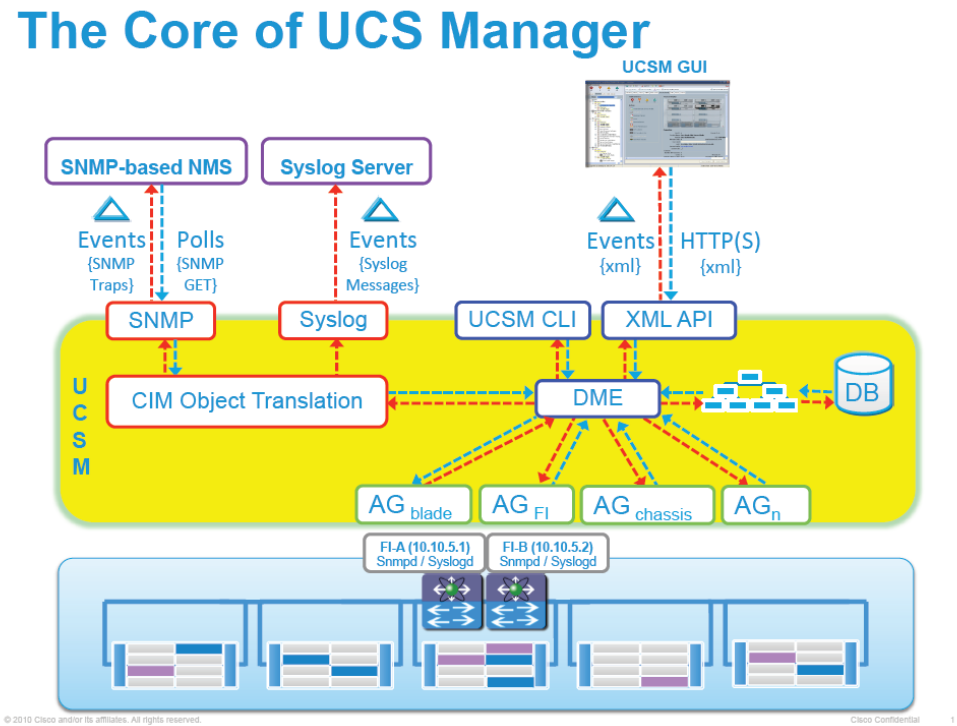
Monitoring UCS Manager with Syslog:
http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/ucsm_syslog/b_Monitoring_Cisco_UCSM_Using_Syslog.pdf

---

UCSM and Standalone C-Series Monitoring Overview:

UCS Manager Monitoring Background:

The core of UCS Manager is made up three core elements, which are the Data Management Engine (DME), Application Gateway (AG), and user accessible northbound interface (SNMP, Syslog, XMLAPI and UCS CLI). With UCS Manager there are three main ways of monitoring UCS servers, which are XML API, SNMP, and syslog. Both SNMP and Syslog are interfaces only used for monitoring as they are "**read-only**" in nature, not allowing an end user to change the configuration. Alternatively, the UCS XML API is a monitoring that is "**read-write**" in nature, which does allow an end user to both monitor UCS, as well as change the configuration if needed.



Data Management Engine (DME) - The DME is the center of the UCS Manager universe, or the "queen bee" of the entire system. It is the maintainer of the UCS XML database which houses the inventory database of all physical elements (blade / rack mount servers, chassis, IO modules, fabric interconnects, etc.), the logical configuration data for profiles, policies, pools, vNIC / vHBA templates, and the various networking related configuration details (VLANs, VSANs, port channels, network uplinks, server downlinks, etc). It maintains the current health and state of all components of all physical and logical elements in a UCS Domain, and maintains the transition information of all Finite State Machine (FSM) tasks occurring. The inventory, health, and configuration data of managed end points stored in the UCS XML Database are always showing current data, delivered in near real time. As fault conditions are raised and cleared on end points, the DME will create, clear, and remove faults in the UCS XML database as those fault conditions are raised or mitigated. The faults stored in the UCS XML database only are the ones actively occurring, as the DME by default does not store a historical log of all faults that have occurred on a UCS Domain.

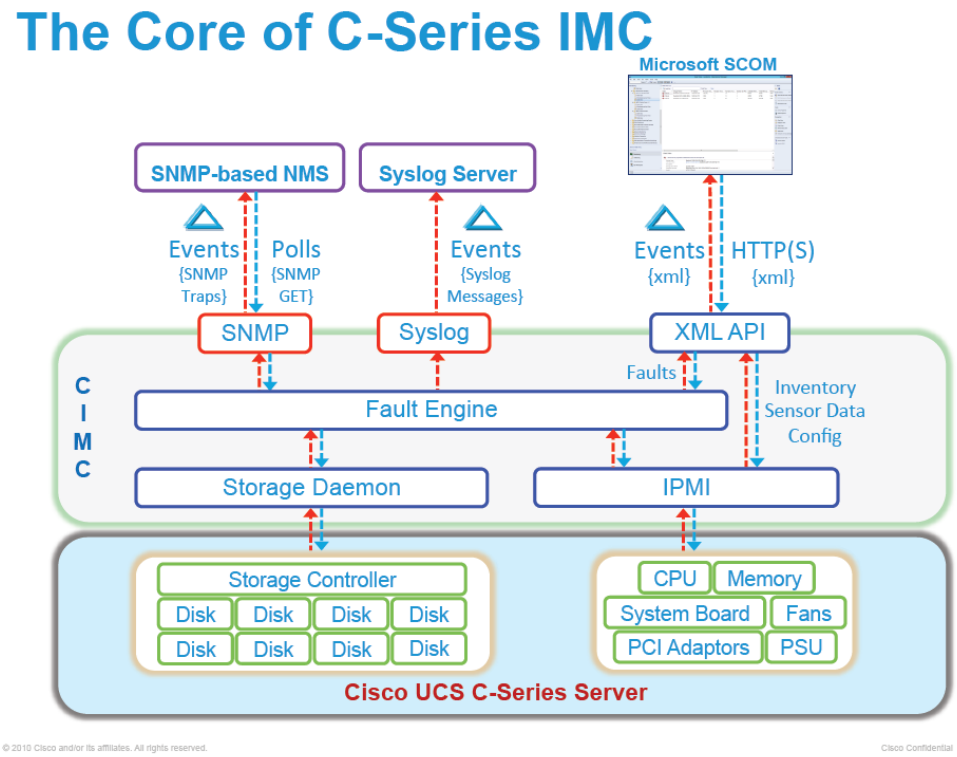
Application Gateway (AG) - The AG's are the software agents, or "worker bees", that communicate directly with the end points to provide the health and state of the end points to the DME. AG's manage configuration changes from the current state to the desired state during FSM transitions when changes are made to the UCS XML database. AG managed end points include servers, chassis, IO Modules, fabric extenders, fabric interconnects, and NXOS. The server AG's actively monitor the server through the IPMI and SEL logs via the Cisco Integrated Management Controller (CIMC) to provide the DME with the health, state, configuration, and potential fault conditions of a device. The IO Module AG and chassis AG communicate with the Chassis Management Controller (CMC) to get information about the health, state, configuration, and fault conditions visible by the CMC. The fabric interconnect / NXOS AG

communicates directly with NXOS to get information about the health, state, configuration, statistics, and fault conditions visible by NXOS on the fabric interconnects. All AG's provide the inventory details to DME about end point during the various discovery processes. The AG's perform the state changes necessary to configure an end point during FSM triggered transitions, monitors the health and state of the end points, and notifies the DME of any faults or conditions.

Northbound interfaces:
The northbound interfaces include SNMP, Syslog, CLI and XML API. The XML API present in the Apache webserver layer used to send login, logout, query, and configuration requests via HTTP or HTTPS. SNMP and Syslog are both consumers of data from the DME. SNMP informs and traps are translated directly from the fault information stored in the UCS XML database. Inversely, SNMP GET requests are sent through the same object translation engine in reverse, where the DME receives a request from the object translation engine and the data is translated from XML data from the DME to a SNMP response. Syslog messages use the same object translation engine as SNMP, where the source of the data (faults, events, audit logs) is translated from XML into a UCS Manager formatted syslog message.

Standalone C-Series Monitoring Background:

Monitoring support for our Standalone C-Series Servers has evolved with each release. The features and capabilities of the current CIMC release, v1.5 supports our M3 Platforms including the C220 M3, C240 M3, C22 M3, C24 M3 and C420 M3 as well as our C260 M2 and C460 M2. While earlier versions of our CIMC supported Syslog and SNMP, the Fault Engine added support for SNMP v3 in CIMC v1.5. We have documented the internals of our monitoring subsystem in the graphic included below.



Fault Engine Overview:

While Cisco Standalone C-Series Servers do not support the DME/AG architecture described above in the UCS Manager section, many of the same concepts can be applied to the monitoring subsystem for Standalone Servers. The Fault Engine

has become a central repository and clearinghouse for fault data as it is passed along to monitoring endpoints. The Fault engine acts as a master repository for events within the system which initiates alerts (SNMP Traps, Syslog messages, XML API events, etc.) but can also be queried via SNMP (GETs) or the XML API. This durability of fault information means provides customers a mechanism to not only receive fault data, but also use these interfaces to query system health data.

Within the system, the Fault Engine regularly polls component health status in the form of sensor data using IPMI and the Storage Daemon and these values are compared to threshold reference points. If a sensor value is outside one of the threshold values, an entry is created in the fault engine and notifications are sent as appropriate.

As discussed earlier, multiple notification types are supported including SNMP (Traps and Informs), Syslog (Messages) and XML API (Event Subscription) and fault queries are supported through SNMP GET and XML API queries. Cisco has developed a number of integrations for 3rd Party Management solutions that leverage queries of the Fault Engine data to drive notifications in these management tools. The Fault Engine retains faults until they are mitigated or until the IMC is rebooted.

UCS Manager Best Practices:

The recommendation for monitoring a UCS Manager environment would be to monitor all faults of either severity critical or major and that are not of type "FSM". FSM related faults are transient in nature, as they are triggered when a FSM transition is occurring in UCS Manager. Generally speaking, FSM related faults will resolve themselves automatically as most are triggered after a task fails the first time, but will be successful on a subsequent try. An example of a FSM task failure would be when a FSM task waiting for a server to finish BIOS POST fails during a service profile association. This particular condition can happen when a server with many memory DIMMs takes longer to successfully finish POST than the default timeout of the FSM task. This timeout would raise a FSM fault on this task, but by default would keep retrying up to the defined FSM task retry limit. If a subsequent retry is successful, the FSM task fault raised will be cleared and removed. However, if subsequent retries are unsuccessful and the retry limit is hit, the FSM task will be faulted and another fault will be raised against the affected object. In this example, a configuration failure would be raised against the service profile, as the association process would have failed because the server did not perform a successful BIOS POST.

If you are looking for a list of the most critical faults codes to monitor, refer to the "Syslog Messages to Monitor" section in Chapter 3 of the "Monitoring UCS Manager with Syslog" guide below. The fault codes listed are the same codes for all interfaces (SNMP, syslog, or XML API).
http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/ucsm_syslog/b_Monitoring_Cisco_UCSM_Using_Syslog.pdf

C-Series Standalone Best Practices:

Filtering: As referenced above, the faults for our Standalone C-Series Servers are consistent with faults for UCS Manager. The concept of FSM (Finite State Machine) does not exist with Standalone C-Series, there is no reason to filter out FSM State changes when monitoring these systems. The recommendation is that filters not be applied to Standalone C-Series Servers as all raised faults are relevant to customers who are interested in monitoring/alerting capabilities. At present, there are approximately 85 faults that are included in the Fault Database for our Standalone C-Series Servers with CIMC 1.5(3).

SNMP vs. Platform Event Filters (PEF): As monitoring has evolved in these systems, support has been extended to include a number of notification mechanisms, and Cisco is planning to deprecate Platform Event Filters (PEF) and Platform Event Traps (PET) in a future CIMC release. Platform Event Traps are sent as IPMI v1 traps where filters (PEF) can be applied so only certain subsystem traps are sent to the NMS system. The variable bindings that are consistent across UCS Manager and Standalone C-Series servers do not apply to Platform Event Filters as they have their own nomenclature that is defined and maintained by Intel.

XML API Usage: As a more robust XML API has been implemented in Standalone C-Series Servers, this is the preferred mechanism for capturing faults sent by the system. The XML API supports Event Subscription which provides proactive alerting. The XML API also supports queries which can be used to collect data in the fault table on a regular basis.

Cisco UCS MIB Files:

Cisco MIBs are available at the following download site:
http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

All Cisco UCS Manager and Standalone C-Series faults are available with SNMP using the cucsFaultTable table and the CISCO-UNIFIED-COMUTING-FAULT-MIB. The table contains one entry for every fault instance. Each entry has variables to indicate the nature of a problem, such as its severity and type. The same object is used to model all Cisco UCS fault types, including equipment problems, FSM failures, configuration or environmental issues, and connectivity issues. The cucsFaultTable table includes all active faults (those that have been raised and need user attention), and all faults that have been cleared but not yet deleted because of the retention interval.

Important OIDs (Object Identifier):

Trap	Description
cucsFaultActiveNotif The OID for this SNMP trap is .1.3.6.1.4.1.9.9.719.0.1.	This notification is generated by a Cisco UCS instance whenever a fault is raised.
cucsFaultClearNotif The OID for this SNMP trap is .1.3.6.1.4.1.9.9.719.0.2.	This notification is generated by a Cisco UCS instance whenever a fault is cleared.

In Release 1.3 and later, Cisco UCS Manager sends a cucsFaultActiveNotif event notification whenever a fault is raised. There is one exception to this rule: Cisco UCS Manager does not send event notifications for FSM faults. The trap variables indicate the nature of the problem, including the fault type. Cisco UCS Manager sends a cucsFaultClearNotif event notification whenever a fault has been cleared. A fault is cleared when the underlying issue has been resolved.

In Release 1.4 and later, the cucsFaultActiveNotif and cucsFaultClearNotif traps are defined in the CISCO-UNIFIED-COMPUTING-NOTIFS-MIB. All faults can be polled using SNMP GET operations on the cucsFaultTable, which is defined in the CISO-UNIFIED-COMPUTING-FAULT-MIB.

Fault Attributes (Variable Bindings):

Attribute	Description
Fault Instance ID (Table Index)	A unique integer that identifies the fault.
Affected Object DN	The distinguished name of the mutable object that has the fault.
Affected Object OID	The Object identifier (OID) of the mutable object that has the fault.
Creation Time	The time that the fault was created.
Last Modification	The time when any of the attributes were modified.
Code	A code that provides information specific to the nature of the fault.
Type	The fault type.
Cause	The probable cause of the fault.
Severity	The severity of the fault.
Occurrence	The number of times that a fault has occurred since it was created.
Description	A human readable string that contains all information related to the fault.

:

MIB Loading Order & Statistics Collection Details:

More details on MIB load ordering and statistics collection including a comprehensive list of Statistics OID and their corresponding Statistics tables are located in the following MIB Reference Guides:

MIB Reference for Cisco UCS Manager:
http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/mib/b-series/b_UCS_MIBRef.pdf

MIB Reference for Cisco UCS Standalone C-Series Servers:
http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/mib/c-series/b_UCS_Standalone_C-Series_MIBRef.pdf

UCS Manager and Standalone C-Series Faults:

In the Cisco UCS, a fault is a mutable object that is managed by the Cisco UCS Manager. Each fault represents a failure in the Cisco UCS instance or an alarm threshold that has been raised. During the lifecycle of a fault, it can change from one state or severity to another.

Each fault includes information about the operational state of the affected object at the time the fault was raised. If the fault is transitional and the failure is resolved, then the object transitions to a functional state. A fault remains in the Cisco UCS Manager until the fault is cleared and deleted according to the settings in the fault collection policy.

You can view all faults in the Cisco UCS instance from either the Cisco UCS Manager CLI or the Cisco UCS Manager GUI. You can also configure the fault collection policy to determine how a Cisco UCS instance collects and retains faults.

Fault Severities for UCS Manager and Standalone C-Series Servers include:

Severity	Description
Cleared	A notification that the condition that caused the fault has been resolved, and the fault has been cleared.
Condition	An informational message about a condition, possibly independently insignificant.
Critical	A service-affecting condition that requires immediate corrective action. For example, this severity could indicate that the managed object is out of service and its capability must be restored.
Info	A basic notification or informational message, possibly independently insignificant.
Major	A service-affecting condition that requires urgent corrective action. For example, this severity could indicate a severe degradation in the capability of the managed object and that its full capability must be restored.
Minor	A non-service-affecting fault condition that requires corrective action to prevent a more serious fault from occurring. For example, this severity could indicate that the detected alarm condition is not currently degrading the capacity of the managed object.
Warning	A potential or impending service-affecting fault that currently has no significant effects in the system. Action should be taken to further diagnose, if necessary, and correct the problem to prevent it from becoming a more serious service-affecting fault.

Types of faults for UCS Manager and Standalone C-Series Servers include:

Type	Description
fsm	An FSM task has failed to complete successfully, or the Cisco UCS Manager is retrying one of the stages of the FSM.
equipment	The Cisco UCS Manager has detected that a physical component is inoperable or has another functional issue.
server	The Cisco UCS Manager is unable to complete a server task, such as associating a service profile with a server.
configuration	The Cisco UCS Manager is unable to successfully configure a component.
environment	The Cisco UCS Manager has detected a power problem, thermal problem, voltage problem, or a loss of CMOS settings.
management	The Cisco UCS Manager has detected a serious management issue, such as one of the following: • Critical services could not be started. • The primary switch could not be identified. • Components in the instance include incompatible firmware versions.
connectivity	The Cisco UCS Manager has detected a connectivity problem, such as an unreachable adapter.
network	The Cisco UCS Manager has detected a network issue, such as a link down.
operational	Cisco UCS Manager has detected an operational problem, such as a log capacity issue or a failed server discovery.

Fault Overview:

The faults in Cisco UCS are stateful, and a fault raised in a Cisco UCS instance transitions through more than one state during its lifecycle. In addition, only one instance of a given fault can exist on each object. If the same fault occurs a second time, the Cisco UCS increases the number of occurrences by one.

A fault has the following lifecycle:

1. A condition occurs in the system and the Cisco UCS raises a fault in the active state.
2. If the fault is alleviated within a short period of time known as the flap interval, the fault severity remains at its original active value but the fault enters the soaking state. The soaking state indicates that the condition that raised the fault has cleared, but the system is waiting to see whether the fault condition reoccurs.
3. If the condition reoccurs during the flap interval, the fault enters the flapping state. Flapping occurs when a fault is raised and cleared several times in rapid succession. If the condition does not reoccur during the flap interval, the fault is cleared.
4. Once cleared, the fault enters the retention interval. This interval ensures that the fault reaches the attention of an administrator even if the condition that caused the fault has been alleviated, and that the fault is not deleted prematurely. The retention interval retains the cleared fault for the length of time specified in the fault collection policy.

5. If the condition reoccurs during the retention interval, the fault returns to the active state. If the condition does not reoccur, the fault is deleted.