Chapter 19

Support for WS-Management and CIM Profiles

If there is anything the nonconformist hates worse than a conformist, it's another nonconformist who doesn't conform to the prevailing standard of nonconformity.

—Bill Vaughan

In Chapter 3 we looked at Common Information Model (CIM) and WS-Management standards. We also discussed several CIM profiles that are defined within the DMTF under the DASH initiative umbrella. In this chapter, we will discuss the native support built into Intel® Active Management Technology (Intel AMT) to support the CIM data model and DASH profiles.

WS-Management Support in Intel® AMT

Intel AMT supports the following standard WS-Management specifications available from DMTF:

- DSP0226: Web Services for Management, version 1.0.0
- DSP0227: WS-Management CIM Binding Specification, version 1.0.0b
- DSP0230: WS-CIM Mapping Specification, version 1.0.0

In Chapter 3, we looked at some of the protocol flows for WS-Management. Intel AMT supports most of the WS-Management key operations, which include Get, Put, Create, Delete, Enumerate and event subscription and delivery. For the data that is accessed using WS-Management, Intel AMT uses the CIM data model. The profiles that are supported by Intel AMT are described in this chapter.

From a WS-Management protocol perspective, there are a few things that need to be kept in mind specific to an Intel AMT implementation.

Return Value from Put Operations

Intel AMT does not return the representation of an object after a Put operation. The client should invoke another Get request to get the new representation. However, if the Put request includes a read/write property, and the Intel AMT device does not return a fault, the client can assume that the value of any changed fields has been updated to the requested value

Enumeration Support

The WS-Enumeration specification indicates that enumeration is a three-part operation: An initial `wsen:Enumerate` is issued to establish the enumeration context and `wsen:Pull` operations are used to iterate over the result set. When the enumeration iterator is no longer required and not yet exhausted, a `wsen:Release` is issued to release the enumerator and associated resources.

Intel AMT allocates a resource that is reserved for the client during the enumeration period. The enumeration flow (enumerate, pull, release) should be completed within 30 seconds. If the enumeration flow does not complete after 30 seconds, the enumeration context may be purged for the use of other clients requesting to enumerate a resource.

Intel AMT Release 3.0 supports a maximum of three concurrent enumeration flows.

Envelope Size

WS-Management protocol defines an optional `<wsman:MaxEnvelopeSize>` tag in the SOAP header that specifies the client's request to limit the length of the response size. Intel AMT requires that if clients specify a maximum response size limitation, then the value for the tag must be at least 50000. If a client specifies a lower value, then it might receive a corresponding WS-Management fault according to the WS-Management specification.

OptionSet Support

Intel AMT does not support the WS-Management Options. If the WS-Management Header OptionSet element is passed with `soap:mustUnderstand=true`, a soap fault element with a fault code of `MustUnderstand` will be returned.

Using Create and Put with Read-Only Properties

When using the Create and Put operations, all `Key` or `Required` properties must be passed. Notice that not all fields of a class are writable; for example, the `CreationClassName` property is never writable in Intel AMT.

Properties that are read-only and are passed in the Create or Put operations will be ignored, and Intel AMT fills its own value for them. Any schema-compliant value may be passed for these properties.

Specifically, this means the client must pass all keys, whether they are read-only or writable; read-only keys will be ignored, but must nevertheless be passed.

Whitespace in XML Elements

Intel AMT treats any information within a simple XML element tag as the tag's value. Numeric fields may contain whitespaces; however, string and string-based elements containing any whitespace, including heading or trailing whitespaces, will be treated as though the whitespaces are part of the field's value.

Class Namespace Usage

Intel AMT implements standard DMTF CIM classes as well as CIM classes specific to Intel AMT.

Classes whose name begins with CIM_ are provided by the DMTF. The exact versions are documented in Intel AMT's class reference documentation. The namespace prefix and ResourceURI prefix that should be used to access these classes is

```
http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2+/
```

For example, `CIM_ComputerSystem` will belong to the namespace

```
http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2+/CIM_
ComputerSystem.xsd
```

and this is also its ResourceURI.

Classes whose names begin with `AMT_` are provided by Intel Corporation. Their namespace prefix is

```
http://intel.com/wbem/wscim/1/amt-schema/1/
```

Currently, the Intel-specific schema is not published as XSD on the Web. Therefore the above namespace prefix can only be used for identification purposes. The schema is provided in the Intel AMT SDK.

Following sections go into more discussions of the data model and CIM classes supported by Intel AMT.

Intel® AMT Data Model

Intel AMT capabilities are exposed by a set of CIM classes that are organized into profiles. Intel AMT supports standard profiles as defined by DMTF. In addition, certain capabilities, for which there are no standard profiles (or none existed at the time of product development), Intel AMT implements extended CIM profiles with its own vendor-defined classes.

DASH Profiles

Here we will review the DASH profiles that are supported by Intel AMT. These profiles are listed in the DMTF DSP0232 (DASH Implementation Requirements) document. Each of the profile is then defined in detail in a separate profile document, also available from DMTF.

The focus here is not to list each and every CIM class and properties that are supported by Intel AMT. Instead, we will focus on the platform capabilities that can be used when a specific profile is used.

Profile Registration

The Profile Registration profile is used to find out what other profiles are implemented by the Intel AMT system. Enumerating all instances of class `CIM_RegisteredProfile` returns the number of instances that corresponds to the supported DASH profiles. Within those instances `RegisteredName` property gives the ASCII string that provides the name of the registered profile. Thus, a management application can programmatically discover the supported capabilities using this profile.

Base Desktop and Mobile

This mandatory DASH profile describes the managed desktop or mobile computer system that contains Intel AMT. Other CIM classes are mostly associated to the main computer system. The central class of this profile is `CIM_ComputerSystem`. Enumerating this class provides an instance of the class that refers to the desktop or mobile system that Intel AMT subsystem is managing. Within this class instance, the following properties are useful:

- *Dedicated* – This is an enumerated integer, with the value 32 corresponding to a Desktop system, and the value of 33 corresponding to a Mobile system. Any Intel AMT implementation will have one of these values. Other values are not used.
- *Name* – This string property has the DNS hostname of the system.
- *EnabledState* – This enumerated integer represents the state of the system. In Intel AMT systems, this property can have three possible values. The value of 2 (enabled) represents that the system is in S0 state, i.e. it is fully powered up. The value of 3 (disabled) represents that the system is in S5 state, i.e. it is powered off. The value of 5 (Not Applicable) is used for all other power states.

Boot Control

The DASH Boot Control profile defines classes and mechanisms for temporarily overriding the boot flow configured in the managed system, as well as permanently changing this configuration. Intel AMT only supports single-use boot configurations that are used to temporarily override the boot flow configured in the managed system on the next boot cycle.

The `ChangeBootOrder()` method in the `CIM_BootConfigSetting` can be called to specify the boot order on next boot. Console application must first discover the existing boot sources on a specific platform by enumerating `CIM_BootSourceSetting` and reading the values of the property `StructuredBootString`. Some of the allowed values are Floppy, Hard-Disk, CD/DVD, Network, PCMCIA, and USB. The existing order can be discovered as well.

Intel AMT provides extension to the DASH Boot Control profile by additional classes that allow further discovery of capabilities and allow additional controls. See "Intel AMT Boot Extensions" below.

Power State Management

The Power State Management profile allows a management console to discover the power management capabilities of the Intel AMT system and perform power control operations on the system. `CIM_PowerManagementCapabilities` class can be enumerated to discover what power states (such as S0, S1, S2, S3, S4, S5 and so on) are supported, and in what ways those could be changed.

`CIM_PowerManagementService` has a method `RequestPowerStateChange()`, which is used for power control operations, such as power on, power off, or reset operations. When `RequestPowerStateChange()` causes a boot cycle, the operation of the boot cycle can be controlled using the boot control classes described in the Boot profile above.

The `EnabledState` property in the `CIM_PowerManagementService` is another useful property that indicates the current power state of the Intel AMT managed system. This provides more granular information than the `EnabledState` property in the `CIM_ComputerSystem` described earlier.

CPU

The CPU profile provides information about the processors in the computer system. The `CIM_Processor` class from this profile can be enumerated to find out how many logical processors are present in the system. For each processor, information such as processor model, family, stepping, clock speed, and status are available. Several management applications use this information to identify the system processing power and other processor capabilities.

FAN

The Fan profile is a very simple profile. Using this profile, a management console can query `CIM_Fan` instances to find out how many cooling fans are present on the system, and what their health state is (running okay or stopped).

Physical Asset

The Physical Asset profile is used to identify the hardware asset information about the system. This profile has several classes that identify different physical components of the platform. The classes from this profile are associated to corresponding logical elements via `CIM_Realizes` association. For example, to get the physical asset information about a Processor package, one must find out the `CIM_Chip` instance that is associated to `CIM_Processor` using `CIM_Realizes` association. Once, the appropriate `CIM_Chip` instance is found, the properties such as Manufacturer, Model, and Version can be found. It can also be found if the chip is a field replaceable unit (FRU) or not.

`CIM_PhysicalMemory`, which is also part of this profile, can be used to identify the memory type, speed, capacity, manufacturer, serial number, and part number.

`CIM_ComputerSystemPackage` has a PlatformGUID, which is typically used by a number of management consoles to uniquely identify a specific computer on the network.

`CIM_Chassis` has the asset tag information, which is programmed by enterprise IT to keep track of the computer asset.

Power Supply

The Power Supply profile is another fairly straightforward profile. A management console can query the instances of `CIM_PowerSupply` to get information of the power supply such as total output power and health state.

Record Log

The Record Log profile is used by Intel AMT to expose the event log entries that is maintained by Intel AMT in the NVRAM. `CIM_LogEntry` provides a standard way to enumerate these log entries. `CIM_RecordLog` aggregates all these log entries and provides administration functions such discovering total size of the log, and a function to clear the log.

Sensors

The Sensors profile allows Intel AMT to expose the voltage or temperature sensors information to a management console. The sensors describe the possible states and current state of the sensor. Current health state of the sensors is also available.

Software Inventory

The Software Inventory profile is used by Intel AMT to describe the identification and version information about the Intel AMT firmware itself.

System Memory

The System Memory profile is used to find out the amount of memory installed and available to the OS on the managed system. The central class of this profile is `CIM_Memory`. It has properties such as `BlockSize` and `NumberOfBlocks` that can be used to compute memory on the system. A management console can also query if the memory is read-only, writable, persistent, or volatile.

Simple Identity Management

The Simple Identity Management profile can be used to create or modify a user account on Intel AMT. The central class for this profile is `CIM_AccountManagementService`. The `CreateAccount` method can be called to create a new user. New user account information is passed using `CIM_Account` as a template. The `CIM_Account.UserID` field is used to identify a new user; Intel AMT ignores `CIM_Account.Name`.

An existing user account can be updated using a Put operation for `CIM_Account`. In this operation, the `Name` field is used to update the entry and the `UserID` field is ignored.

Role-based Authorization

The ability to manage and configure roles for a managed system is represented by the `CIM_RoleBasedAuthorizationService` instance. The `CIM_RoleBasedAuthorizationService` class is the central class of the profile and, through extrinsic methods, serves as the interface for a client to request deletion and modification of existing roles, creation of new roles, and assignment of roles to security principals. The authorized roles on a managed system are represented through instances of `CIM_Role`. Rights granted to a security principal through membership in a role are represented by instances of `CIM_Privilege` that are associated with the instance of `CIM_Role` through the `CIM_MemberOfCollection` association.

The `ActivitiesSupported`, `ActivityQualifiersSupported`, and `QualifierFormatsSupported` properties of the Associated Privilege Management Capability represents the full list of supported activities of the privilege. Note that the `ActivityQualifiersSupported` field identifies the Realms supported in a particular release of Intel AMT. Each realm has a short string associated with it. These strings are used to grant a user access to one or more realms or to see which privileges a user has. See, for example, `CIM_Privilege.ActivityQualifiers`. Table 19.1 lists the realm abbreviations.

Table 19.1 Realm Abbreviations

Realm	Abbreviation
PTAdministrationRealm	ADMIN
NetworkTimeRealm	NETT
HardwareAssetRealm	HAI
RemoteControlRealm	RC
EventManagerRealm	EVTMGR
LocalUN	LOCAPP
EventLogReaderRealm	EVTLOG
StorageAdminRealm	STORA
StorageRealm	STOR
RedirectionRealm	REDIR
AgentPresenceLocalRealm	AGPL
AgentPresenceRemoteRealm	AGPR
CircuitBreakerRealm	CB
GeneralInfoRealm	INFO
FirmwareUpdateRealm	FWUPD
EIT	EIT
EndpointAccessControlRealm	EAC
EndpointAccessControlAdminRealm	EACADM
SecurityAuditLogRealm (Added in Release 4.0)	AUDIT
UserAccessControlRealm (Added in Release 4.0)	UAC
Reserved	RESERVED

Indications Profile

Intel AMT supports the Indications profile for subscriptions and delivery of WS-Management events. `CIM_FilterCollection`, `CIM_FilterCollectionSubscription`, and `CIM_ListenerDestinationWSManagement` are key classes of this profile implemented by Intel AMT.

Each instance of `CIM_FilterCollection` in Intel AMT is a filter that passes a predefined set of events. There are six instances of `CIM_FilterCollection`, defined using the following strings:

- "Intel(r) AMT:FW ProgressEvents"
- "Intel(r) AMT:User"
- "Intel(r) AMT:All"
- "Intel(r) AMT:Platform"
- "Intel(r) AMT:CorePlatform"
- "Intel(r) AMT:Features"

To subscribe for notification of events, a user specifies an instance of `CIM_FilterCollection` to define the desired subset of events. A `CIM_ListenerDestinationWSManagement` instance is created for each valid subscription request, as well as an instance of `CIM_FilterCollectionSubscription` as an association between the instance of `CIM_FilterCollection` mentioned in the subscription request and the newly created instance of `CIM_ListenerDestinationWSManagement`.

The number of subscriptions is limited to six. Attempting to create more than six subscriptions without unsubscribing from one of them will fail.

Intel® AMT Extension Profiles

Following are the data profiles that are extensions to the standard CIM. The list discussed here is not exhaustive of all Intel AMT profiles. Since the profiles are being added constantly to Intel AMT implementation, it is advised that reader refers to the SDK for the most current list.

Intel® AMT Boot extensions

Intel AMT provides extension to the DASH Boot Control profile by providing additional classes that allow further discovery of capabilities and allow additional controls.

The `AMT_BootCapabilities` class instance can be retrieved to discover a number of important capabilities about what is available or not available on the Intel AMT system. Each of these is represented by a Boolean property in the `AMT_BootCapabilities` class, including those listed in Table 19.2

Table 19.2 AMT_BootCapabilities Class Properties

Property	Description
IDER	Indicates whether the platform supports IDE Redirection
SOL	Indicates whether the platform supports Serial Over Lan
BIOSReflash	Indicates whether the platform supports update (reflash) of the BIOS upon boot.
BIOSSetup	Indicates whether the platform supports system automatically entering into BIOS Setup upon reboot
BIOSPause	Indicates whether the platform supports pausing the BIOS upon reboot
ForcePXEBoot	Indicates whether the platform supports forcing of PXE Boot (network boot)
ForceHardDriveBoot	Indicates whether the platform supports forcing of boot from the Hard Drive
ForceHardDriveSafeModeBoot	Indicates whether the platform supports forcing of Hard Drive Safe Mode Boot
ForceDiagnosticBoot	Indicates whether the platform supports forcing of a Diagnostic Boot
ForceCDorDVDBoot	Indicates whether the platform supports forcing of CD or DVD Boot
VerbosityScreenBlank	Indicates whether the platform supports a BIOS blank screen (no messages) on next reboot
PowerButtonLock	Indicates whether the platform supports locking the Power Button
ResetButtonLock	Indicates whether the platform supports locking the Reset Button

Table 19.2 AMT_BootCapabilities Class Properties *(continued)*

KeyboardLock	Indicates whether the platform supports locking the Keyboard
SleepButtonLock	Indicates whether the platform supports locking the Sleep Button =
UserPasswordBypass	Indicates whether the platform supports bypassing the password in next reboot
ForcedProgressEvents	Indicates whether the platform forces the BIOS progress events to be reported
VerbosityVerbose	Indicates whether the platform supports verbose messages from BIOS at the next boot
VerbosityQuiet	Indicates whether the platform supports minimal amount (quiet) of messages on the console at the next boot
ConfigurationDataReset	Indicates whether the platform supports all configuration data reset at next boot

`AMT_BootSettingsData` is another class that affects the flags that are to be used on the next system boot, as described in Table 19.3.

Table 19.3 AMT_BootSettingsData Class Properties

Property	Description
UseSOL	When True, Serial over LAN is used on the next boot cycle.
UseSafeMode	When a Hard-drive boot source is chosen (using CIM_ BootConfigSetting) and this property is set to True, the Intel® AMT firmware will boot the platform in safe mode.
ReflashBIOS	When True, the Intel AMT firmware reflashes the BIOS on the next boot cycle.
BIOSSetup	When True, the Intel AMT firmware forces the platform BIOS to enter the CMOS Setup screen on the next boot cycle.
BIOSPause	When True, the BIOS pauses for user input on the next boot cycle.
LockPowerButton	When True, the Intel AMT firmware disables the power button operation for the system, normally until the next boot cycle.
LockResetButton	When True, the Intel AMT firmware disables the reset button operation for the system, normally until the next boot cycle.
LockKeyboard	When True, the Intel AMT firmware disallows keyboard activity during its boot process.
LockSleepButton	When True, the Intel AMT firmware disables the sleep button operation for the system, normally until the next boot cycle.
UserPasswordBypass	When True, the Intel AMT firmware boots the system and bypasses any user or boot password that might be set in the system.
ForcedProgressEvents	When True, the Intel AMT firmware transmits all progress PET events to the alert-sending device.
FirmwareVerbosity	When set to a nonzero value, controls the amount of information the managed system writes to its local display.
ConfigurationDataReset	When True, the Intel AMT firmware forces BIOS to reset its non-volatile configuration data to the managed system's Setup defaults prior to booting the system.
IDERBootDevice	Specifies the device to use when UseIDER is set.
UseIDER	When True, IDER session is started on the next boot cycle.
BootMediaIndex	This property identifies the boot-media index for the managed platform (when a boot source is set using the CIM_ BootConfigSetting.ChangeBootOrder method)

Agent Presence

The Agent Presence Profile allows management of Agent Presence capability configuration.

The `AMT_AgentPresenceCapabilities` class provides the discovery of how many maximum agents can be monitored, how many maximum actions can be taken and what is the minimum guaranteed actions that will always be executed when an event occurs.

`AMT_AgentPresenceWatchdog` is an extension of standard `CIM_watchdog` class. This class is instantiated for each agent that is being monitored by Intel AMT. The description of the agent that is being monitored is available in the `MonitoredEntityDescription` property. This class provides a number of methods. The `RegisterAgent()` method is issued by applications that wish to start reporting their running state. The `AssertPresence()` method is issued periodically by applications to report their running state. The `AssertShutdown()` method is issued by applications to report their termination state. The `AddAction()` method adds an action to the application watchdog. `DeleteAllActions()` removes all actions associated with the watchdog.

A few additional classes are used for configuration of actions and policies.

Event Management and User Notifications

These profiles allow management of event log entries and configurations of Platform Event Traps (PETs), filters, and trap destinations.

`AMT_EventLogEntry` expands `CIM_LogEntry` (described in the standard Record Log Profile above) and provides a specific format of the log entries, which includes fields like EventSeverity, EventType, SensorNumber and EventData among a lot of other useful information.

`AMT_EventSubscriber` classes provide the data structures to store information about the event subscribers. An instance is created for every new subscriber.

`AMT_PETCapabilities` and `AMT_PETFilterSetting` classes are used to configure PET capabilities and filters to appropriate generate PETs on specific events.

Firmware Update Status

Intel AMT uses `CIM_SoftwareInstallationService` to provide the mechanism to update the Intel AMT firmware. The `CIM_ConcreteJob` class is instantiated when the firmware update process is started. This can be used to find out the status of the firmware update.

Redirection

`AMT_RedirectionService` provides description and management of IDE redirection and Serial over LAN (SoL) capabilities for the Intel AMT subsystem. The capabilities provided are primarily discovering that the capability exist and enable/disable it, if needed.

Service Processor

In the case of Intel AMT, the service processor represents the Intel AMT subsystem. The Service Processor profile defines the minimum top-level object model needed to define a service processor subsystem within a managed computer system. Other profiles add additional management objects to this model to provide non-volatile storage, network outbreak containment, firmware update, and other capabilities supported by Intel AMT.

The profile consists of a single instance of `CIM_ComputerSystem` representing the management processor subsystem within a managed computer system. The service processor is associated with the `CIM_ComputerSystem` instance that represents the managed system via the `CIM_SystemComponent` association, according to the Service Processor profile.

Setup and Configuration

`AMT_SetupAndConfigurationService` provides the description and control of setup and configuration capabilities in Intel AMT. This service has the properties that define Intel AMT provisioning mode, provisioning state, a flag indicating if Zero Touch Configuration is enabled. This service has a number of methods that provide provisioning and unprovisioning operations.

System BIOS

`CIM_BIOSElement` provides information about the system BIOS. It includes information such as manufacturer, version, and release date.

System Defense Profile

This profile allows configuration and management of System Defense policies and actions. `AMT_SystemDefenseService` is the central class of this profile, although most of the useful data is in the classes associated to it. The `AMT_GeneralSystemDefenseCapabilities` class provides information about number of supported policies and filters. `AMT_Network-Filter` and `AMT_IPHeaderFilter` provide configuration of filters. There are a few other classes that provide system defense statistics. The classes in this profile are closely interrelated, and the reader is advised to refer to Intel AMT SDK for details.

Third Party Data Storage

The management of Third Party Data Storage is provided using this profile. This profile consists of two services, `AMT_ThirdPartyDataStorageAdmin-istrationService` and `AMT_ThirdPartyDataStorageService`. Both of these services have a number of extrinsic methods. The administration service enables administrators to reconfigure the global parameters that govern allocation and use of third-party nonvolatile storage. It also enables to retrieve various management data, and perform management actions. The storage service provides limited nonvolatile storage services to third-party software applications running either on the local computer system host processor or on a remote system. Third party applications primarily use this interface.

Time Synchronization

The Time Synchronization profile defines classes and mechanisms for retrieving the local time from the Intel AMT device and synchronizing the device's internal clock with an external clock. The central class of this profile is `AMT_TimeSynchronizationService`, which has an extrinsic method `GetLowAccuracy-TimeSynch()` to read Intel AMT's internal clock, and another method `SetHighAccuracy-TimeSynch()` to synchronize the Intel AMT device's internal clock with an external clock.

Summary

Intel AMT supports a number of CIM data profiles that can be easily accessed by a management console that uses WS-Management and CIM profiles for managing the systems on the network. For the management consoles that do not yet incorporate WS-Management natively, Intel AMT SDK provides a higher level library that can be used as a higher level of abstraction.