
Extending Biz-Client Instrumentation using DELL OpenManage™ Client Instrumentation Tool v8.x

This Dell Technical Whitepaper provides information to remotely access asset information, monitor the health state and change the power state of a Dell Client system.

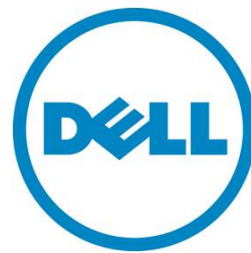
Dell | Product Group

Authors:

Sharmad Naik, Software Engineer

Prasanna J, Software Engineer

Keerthan Kumar, Software Engineer



This document is for informational purposes only and may contain typographical errors and technical inaccuracies. The content is provided as is, without express or implied warranties of any kind.

© 2013 Dell Inc. All rights reserved. Dell and its affiliates cannot be responsible for errors or omissions in typography or photography. Dell, the Dell logo, and PowerEdge are trademarks of Dell Inc. Intel and Xeon are registered trademarks of Intel Corporation in the U.S. and other countries. Microsoft, Windows, and Windows Server are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.

July 2013| Version 1.0

Contents

Introduction	5
CIM and WMI Technology Overview	6
CIM	6
WMI	6
SNMP and MIB Overview	7
SNMP	7
MIB	8
OMCI Overview	8
OMCI Architecture	8
WMI/CIM providers	8
CIM Alert Indications	9
SNMP	10
SNMP Traps	11
Sample Scenarios	11
Asset Management	11
Configuration Management	11
Health Monitoring	12
Management Application Support	12
Enterprise Console	12
Installing OMCI	12
Conclusion	13
Using OMCI	14
CIM	14
Namespace Information	14
Classes	14
General System	14
Bios Management	14
Boot configuration	14
Hardware	15
Indications	16
RAID	16
Sensors	16
Software Info	16
Others	17

Metrics	18
Record Log	18
Operating System	18
Docking station	19
Embedded PCI Devices	19
CPU	19
Service processor	19
SNMP	19
OID Information	19
MIB Groups	20
SNMP Alerts	20
Appendices	22
Appendix A - Sample Scripts	22
Wake on Lan	22
Trusted Platform Module	24
Change Asset Tag	26
Get System Summary	27
Force PXE on Next Boot	29
Get BIOS Settings	31
Trusted Platform Module	33
Clear Chassis Intrusion	34
Shutdown the system	36
Restart the system	38
Get Boot Order	39
Set Boot Order	41

Introduction

The Dell OpenManage™ Client Instrumentation (OMCI) is software that allows remote management application programs to access asset information about the client computer, configure BIOS and boot sequence settings, monitor the health status of the client computer, or change the power state of the computer, such as remotely shutdown of the computer.

OMCI is a standards-based instrumentation component of OpenManage™. OMCI uses the services of the following protocols to provide information remotely:

1. Web-Based Enterprise Management (WBEM), which is a set of systems management technologies developed to unify the management of distributed computing environments. WBEM is based on Internet standards and DMTF open standards: CIM infrastructure and schema, CIM-XML, CIM operations over HTTP, and WS-Management.

Microsoft® Windows® Management Instrumentation (WMI) is Microsoft's implementation of CIM instrumentation. OMCI provides data to WMI, which acts as the common interface to WMI management applications. This helps reduce the total cost of ownership, improves security, and provides a holistic approach to manage all devices including clients, servers, storage, network, and software devices.

Although OMCI has numerous features, its primary purpose is to provide access to information requested by the CIM Object Manager (CIMOM), which in turn provides the information to systems management applications such as OpenManage™ Essentials (OME), Microsoft System Center Configuration Manager (SCCM), Microsoft Operations Manager (MOM) and Windows Remote Management (WinRM).

The CIM schema is defined by the Managed Object Format (MOF) file as distributed by the Distributed Management Task Force (DMTF), which provides a standardized model for describing management information between clients in a management system. The MOF file is not bound to a particular implementation, and it allows the interchange of management information between many different management systems and clients.

2. Simple Network Management protocol(SNMP), is a widely accepted solution to manage devices on IP networks. SNMP is developed and maintained by Internet Engineering Task Force (IETF). OMCI access information and monitor client systems using SNMP. Devices that typically support SNMP include routers, switches, servers, workstations and more. SNMP consists of a set of standards for network management, including an application layer protocol, a database schema, and a set of data objects. SNMP exposes management data in the form of variables on the managed systems, describing the system configuration. Later, these variables can be queried by managing applications.

SNMP itself does not define which information (which variables) a managed system should offer. Rather, SNMP uses an extensible design, where the available information is defined by the management information bases (MIBs). MIBs describe the structure of the management data of a device subsystem; they use a hierarchical namespace containing object identifiers (OID). Each OID identifies a unique variable that can be read via SNMP.

OMCI uses a variety of data sources resident on the client computer:

- Microsoft Win32 WMI providers
- System management basic input/output system (SMBIOS)
- Device drivers
- Operating system registry
- Operating system Application Programming Interfaces (APIs)
- WMI Repository
- Simple Network Management Protocol (SNMP) Agent

CIM and WMI Technology Overview

DMTF is the industry-recognized standards body that leads the development, adoption, unification of management standards (including CIM, and Advance System Format(ASF)), and initiatives for desktop, enterprise, and Internet environments. Working with Dell and other key technology vendors and standards groups, the DMTF enables an integrated, cost-effective approach to management. CIM is a key component of this vision of OMCI.

CIM

CIM is an object-oriented data model for describing management information. Designed to be implementation-neutral, CIM-capable management applications like OME gather information from a variety of CIM objects and devices, including client and server systems, network infrastructure devices, and applications.

CIM helps the detail mapping techniques for improved compatibility with other management protocols. The CIM data model abstracts and describes all elements in a network environment. The CIM “schema” provides the actual data model descriptions and arranges the network into a series of managed objects, all data models interrelated and broadly classified.

Using this approach, CIM goes beyond representing raw data; instead, it describes the enterprise network and allows a much more accurate representation of the network environment and each managed element. There is wide acceptance of CIM by key vendors who have transitioned their products to CIM.

WMI

CIM support is included in Windows® 8, Windows® 7, Windows® Vista and Windows® XP via the WMI infrastructure. WMI supports CIM and Microsoft specific CIM extensions. WMI includes:

- A powerful set of native services such as query-based information retrieval and event notification.
- Extensive scripting capabilities via the Windows Scripting Host (WSH).
- The CIM Object Manager (CIMOM), which is the interface and manipulation point for CIM objects and information.
- WMI plug-in to expose WMI classes to Windows Remote Management(WinRM).

The CIMOM acts as a facilitator in gathering information and manipulating object properties. CIM software components (called “providers”) act as Component Object Manager/Distributed Component Object Manager (COM/DCOM) servers that handle requests from the CIMOM when needed.

The CIMOM stores data in an area called the WMI repository. Management objects stored in the WMI repository can be either physical or logical entities. All CIM objects that the CIMOM has knowledge of reside in a single repository. Users may contribute to the repository by compiling files in the Managed Object Format (MOF).

A variety of management applications support CIM and WMI. Most can collect information directly from the CIMOM interface, but there are alternate methods such as Open Database Connectivity (ODBC), Active Directory Services Interface (ADSI), and the WMI scripting application program interface (API).

WMI provides for user authentication before granting access to CIM data. Access privileges are enforced by DCOM security and the CIMOM. Full or limited access can be granted to users on a per-namespace basis. There is no implementation of class- or property-level security. By default, the administrators group have full local and remote access to WMI.

Note: User Account Control requires an elevation of privileges for local members of the administrators group to have full access to the root\dcim\sysman namespace. See the OMCI User's Guide for more information.

The WMI scripting interface can retrieve information, execute CIM actions (or "methods"), and run VBScript or Jscript scripts that connect to WMI services locally or remotely. Because OMCI is implemented through WMI, most OMCI tasks can be scripted.

SNMP and MIB Overview

IETF is an open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. IETF has created two new working groups in the area of network management. One group is dedicated for the specification and definition of elements to be included in the Management Information Base (MIB). The other group is assigned with the job to define the modifications to the Simple Network Management Protocol (SNMP) to accommodate the needs of the network vendor and operations communities, and to align with the output of the MIB working group.

SNMP

SNMP is used to manage element in growing internet and other networks. SNMP was derived from its predecessor SGMP (Simple Gateway Management Protocol) and was intended to be replaced by a solution based on the CMIS/CMIP (Common Management Information Service/Protocol) architecture.

SNMP is based on the manager/agent model consisting of an SNMP manager, an SNMP agent, a database of management information, managed SNMP devices and the network protocol. The SNMP manager is the point, through which the network administrator performs network management functions. The SNMP agents are the entities that interface to the actual devices being managed. Bridges, routers, switches, or network servers and client like Notebooks and Workstations are examples of managed devices that contain managed objects. These managed objects might be hardware or configuration parameters, performance statistics, and so on, that directly relate to the current operation of the device.

SNMP uses five basic messages (GET, GET-NEXT, GET-RESPONSE, SET, and TRAP) to communicate between the SNMP manager and the SNMP agent. The GET and GET-NEXT messages allow the manager to request information for a specific variable.

Once the agent is receiving a GET or GET-NEXT message, the agent will issue a GET-RESPONSE message to the SNMP manager with either the information requested or an error indication as to why the request cannot be processed.

The SNMP TRAP message allows the agent to spontaneously inform the SNMP manager of an "important" event like Opening of chassis.

MIB

The SNMP manager and agent use an SNMP Management Information Base (MIB) and a relatively small set of commands to exchange information. The SNMP MIB is organized in a tree structure with individual variables, such as point status or description, being represented as leaves on the branches. A long numeric tag or object identifier (OID) is used to distinguish each variable uniquely in the MIB and in SNMP messages.

OMCI Overview

OMCI is a general term that typically refers to the software agent or application that enables a client computer to be managed remotely. Instrumentation is usually vendor-specific, but based on industry standards (such as CIM) defined by the DMTF.

An industry-standard client interface for collecting information and alerting helps to ensure interoperability with management applications. Each application is not designed to understand multiple underlying methodologies for collecting and manipulating client system information or settings on a specific device.

OMCI is the Dell instrumentation package that enables Dell Precision Workstations™, OptiPlex™ and Latitude™ systems to be managed remotely. OMCI contains the underlying driver set that collects system information from a number of different sources on the client computer, including the BIOS, CMOS, System Management BIOS (SMBIOS), System Management Interface (SMI), operating system, APIs, DLLs, and registry settings.

OMCI exposes that information through :

1. The CIMOM interface of the WMI stack.
2. The SNMP protocol using MIB.

In this way, OMCI enables IT administrators to remotely collect asset information, modify CMOS settings, receive proactive notification of potential faults and conditions, and be alerted to potential security breaches.

OMCI Architecture

The OMCI data provider collects the system information data and stores the information in a proprietary XML format. The data manager is a service that loads these providers based on request.

WMI/CIM providers

The OMCI CIM provider layer abstracts the interface to different CIMOM implementations. The input is a combination of XML and XSL data in proprietary form, while the output is a CIM object instance based on the Management Profiles. The WSMAN that serves as the channel protocol requests the data from CIMOM and transmits it to the console.

The OMCI architecture has several layers that are integrated with the Microsoft WMI stack:

- WMI application layer — Consists of management applications, such as Dell OpenManage™ Essentials and other standards-based management tools, and WMI applications, such as Microsoft System Center Configuration Manager, LANDesk, and WMI Tools. The applications in this layer are consumers of the system's management data supplied by OMCI. These applications request client information and alerts through WSMAN/CIM Object Manager (CIMOM).
- OMCI CIM Provider — Lies beneath the CIMOM and contains two CIM providers, which is registered with the CIMOM:
 - The instance/method provider implements an interface that enables utility operations such as create, delete, modify, and query.
 - The indication provider implements an interface for WMI indications (or events).

When the CIMOM receives a request for information, it routes the request to the appropriate provider. Both Dell and Microsoft providers exist in this layer, and they provide information on system devices. The providers send management application requests from the CIMOM to the data router.

- OMCI Data Manager — A service that loads the data provider based on request from the upper layer.
- OMCI Event Manager — A service that monitors hardware and event provider indicators to the upper layer.
- Data Provider — Collects system information like hardware, drivers, and operating system data, and stores them in the proprietary XML format.

For example, a management console in the WMI application layer requests the available processor information on a client system. The WMI application layer makes the request over the network to the CIMOM on the client system. The CIMOM passes the request to the OMCI CIM provider and data manager. The data manager loads the corresponding data provider, which gets the information and stores it in a proprietary format. The information is then returned (through the same path in reverse) to the management console.

CIM Alert Indications

All alerts that are generated by OMCI are delivered in three ways: write to the OS Event log, write the log into DCIM_LogEntry class, and a CIM Indication is generated for the alert. All of the alerts written to the Windows Event Log have a *Source* value of *OpenManage Client Instrumentation*. The alerts supported by OMCI are:

Table 1.

Windows Event Log ID	OMCI Event	Description	Severity Level
1253	ChassisIntrusion	System Chassis Intrusion alert.	WARNING_DEGRADED
1403	EccMemory	A memory checksum failure has occurred.	MINOR
2030	NumberOfDisksIncreased	A hard drive has been added.	WARNING_DEGRADED
2031	NumberOfDisksDecreased	A hard drive has been removed.	WARNING_DEGRADED
2032	MemorySizeIncreased	The memory size has	WARNING_DEGRADED

		been increased.	
2033	MemorySizeDecreased	The memory size has been decreased.	WARNING_DEGRADED
2034	DiskCapacity	One of the hard drives is running out of free space.	WARNING_DEGRADED
2035	NumberOfProcessorIncreased	A processor has been added.	WARNING_DEGRADED
2036	NumberOfProcessorDecreased	A processor has been removed.	WARNING_DEGRADED
2037	SMART	A hard drive condition has occurred that may eventually lead to a drive failure.	MAJOR
2038	DiskSizeIncreased	The size of at least one hard drive has increased.	WARNING_DEGRADED
2039	DiskSizeDecreased	The size of at least one hard drive has decreased.	WARNING_DEGRADED
1801	RAIDControllerFailure	The RAID Controller has Failed	CRITICAL
1802	RAIDControllerOffline	The RAID Controller is Offline	MINOR
1803	RAIDControllerPowerOff	The RAID Controller is Off	MINOR
1811	PhysicalDiskDegraded	The Physical Disk is degraded	WARNING_DEGRADED
1813	PhysicalDiskFailed	The Physical Disk is failed	CRITICAL
1821	VirtualDiskDegraded	The Virtual Disk is degraded	WARNING_DEGRADED
1822	VirtualDiskRebuilding	The Virtual Disk is rebuilding	MINOR
1823	VirtualDiskFailed	The Virtual Disk is failed	CRITICAL

For more information please refer to the whitepaper on “OMCI Health monitoring and Indications” at the following location:

http://en.community.dell.com/techcenter/extras/m/white_papers/20150764.aspx

SNMP

The OMCI SNMP Agent acts as an interface between the SNMP manager like the MIB browser and the Dell Client System being managed.

When the SNMP manager issues a GET or GET-NEXT message, the SNMP agent will send the request to the data router. With the information response received from the data router, the OMCI SNMP agent will issue the GET-RESPONSE message back to the SNMP manager.

The MIB file (name:10892.mib) is provided in the OMCI installation package for enabling the SNMP feature during installation. Please refer the Installation Guide for information on enabling SNMP. The

MIB provides instrumentation data that allows you to monitor the health of a system with SNMP management applications. OMCI uses the same MIB file within OMSA (OpenManage™ Server Administrator). The MIB provides:

- Information about the status of temperatures, voltages, and memory at key points in the system.
- Rapid access to detailed fault and performance information gathered by industry standard systems management agents.
- Version information for BIOS, firmware, and operating system.

For more information on SNMP kindly refer the SNMP reference guide and for information on using MIB file with Dell OMCI to manage Client Systems, please refer to the whitepaper on “Managing Dell Client System using SNMP with Dell OMCI” at the following location:

http://en.community.dell.com/techcenter/extras/m/white_papers/20101707.aspx

SNMP Traps

OMCI can help monitor the chassis intrusion of various client systems in the network. Once the MIB file is loaded in the MIB browser, if a chassis intrusion event occurs on the system then the event will show up in the trap receiver tab in the browser.

For more information please refer to the whitepaper on “Managing Dell Client System using SNMP with Dell OMCI” at the following location:

http://en.community.dell.com/techcenter/extras/m/white_papers/20101707.aspx

Sample Scenarios

Asset Management

A company has several hundred deployed Dell computers. Due to changes in the business and IT staff, inventory information has not been maintained accurately. The CIO requests a plan for identifying which systems can be migrated to Windows® 8. This requires an assessment of the deployed systems to determine the size, scope, and financial impact of such a project.

The man-hours required for manually collecting this information would be substantial. Deploying IT staff to each client system is expensive, not only in IT staff time, but in end-user interruptions.

Using OMCI software on each Dell client system, an IT manager can quickly collect information remotely without dispatching IT staff to each system. Using a tool such as Dell OpenManage™ Essential, the IT manager queries each client system over the network and collects relevant information such as CPU type and speed, amount of memory, hard-drive capacity, BIOS version, and current operating system version. Once collected, the information can be analyzed to determine which systems can be upgraded to Windows® 8.

Configuration Management

An IT department is planning to standardize the client platform and manage each system through its life cycle. As part of this effort, the department acquired a suite of tools, and is planning to automate the deployment of a new client operating system using the Preboot Execution Environment (PXE).

The challenge for the department is to modify the boot order setting in the BIOS of each Dell computer without visiting the desktop. With OMCI installed on each client system, IT has several options for remotely modifying the boot order. Using the management console, IT can remotely modify BIOS settings on all Dell OptiPlex™, Dell Precision Workstations™, and Dell Latitude™ systems. Another option is to write a VB script that alters the boot order setting (see “[Get/Set Boot Order](#)” in “[Appendix A - Sample Scripts](#)” for an example script on changing the boot order). The script can be delivered remotely over the network and run on each client system.

Standardized configurations can provide significant cost savings for companies of all sizes. Many organizations deploy standardized client systems, but few manage the system configuration throughout the life of the computer. With OMCI installed on each client system, the IT department can lock down legacy ports like USB or e-SATA ports to prevent use of unauthorized peripherals, or enable Wake On LAN (WOL) so the system can be revived from a sleep state during non-peak hours to perform systems management tasks.

Health Monitoring

A user receives “read” error messages while trying to access certain files on the client-system hard drive. The user reboots the system and the files now appear to be accessible. The user disregards the initial problem because it appears to have resolved itself.

Meanwhile, OMCI queries the hard drive with the problem for a predicted failure and passes a SMART alert to the management console. (OMCI also displays the SMART error to the local user.) The alert indicates that the hard drive is experiencing excessive read/write errors. IT contacts the user and recommends an immediate backup of critical data files. A service technician is dispatched with a replacement drive. The hard drive is replaced before it fails, avoiding user downtime, a Help Desk call, and a technician trip to the desktop to diagnose the problem.

Note : Enabling SMART error will need a BIOS configuration change in System Configuration category in BIOS/F2 screen.

Management Application Support

Enterprise Console

Most leading enterprise management tools and console providers natively support CIM/WMI and do not require custom integration to manage Dell systems instrumented with OMCI. Instead, the management application simply subscribes with the CIMOM, and client-system alerts are forwarded to the management application regardless of the specific client implementation.

Also using the provided MIB file, IT administrators can easily query for specific information on various group and watch for SNMP traps on the client system.

Installing OMCI

OMCI can be installed locally using the OMCI installation wizard, or remotely via a silent or administrative installation.

- **Local Installation** — The OMCI installation wizard allows IT staff to install the complete package or to do a custom installation of specific components required for a particular environment. OMCI can

also be modified, repaired, or removed through the Add/Remove Programs dialog in the Windows operating system environment.

- **Silent Installation** — OMCI provides a series of command-line switches for silent installation. This allows IT staff to remotely install the product using an electronic software distribution tool with no end user interaction. To obtain information on performing a silent install, refer to the OMCI Quick Installation Guide.
- **Administrative Installation** — The IT Manager can use a server image to deploy OMCI to client computers on a network by using a login script, Windows 2000 system policies, or other methods. The Administrative Installation also provides access directly to the MSI file that can be used to install OMCI. To obtain information on performing an administrative install, refer to the OMCI Quick Installation Guide.

Conclusion

Dell OMCI is available on Dell OptiPlex, Dell Precision Workstation, and Dell Latitude client systems. Using OMCI, system administrators can remotely manage assets, monitor system health, and inventory deployed systems in the enterprise.

OMCI interfaces with leading enterprise management consoles that support industry standards. This approach helps to ensure that Dell systems can be managed by a broad array of existing enterprise management tools. OMCI also interfaces with Dell OpenManage™ Essentials, Microsoft SCCM and SCOM.

Close integration with WMI allows customers to take advantage of the rich scripting capabilities for collecting information and customizing system settings. OpenManage Client Instrumentation is provided at no additional charge to Dell customers.

Additional information and product downloads can be obtained at dell.com/OMConnectionsClient under OpenManage Client Instrumentation. See the DMTF website at www.dmtf.org for more information on CIM, and the Microsoft development website at www.microsoft.com for more information on WMI.

Using OMCI

CIM

Namespace Information

The namespace for access to the Dell OMCI classes is "root\dcim\sysman".

Classes

OMCI provides information in the above mentioned namespace through a number of classes in the namespace. These classes are listed below in general categories. For detailed information on the properties in the classes, refer to the OMCI Reference Guide available at dell.com/OMConnectionsClient under OpenManage Client Instrumentation.

General System

DCIM_ComputerSystem - This class is used to represent the system. It can be a desktop or a notebook system. Invoking RequestStateChange() method of this class shall change the state of the computer such a shutdown or restart. This method has one input parameter: RequestedState, value 3 is for shutdown and 11 is for restart.

Bios Management

DICM_BIOSElement - This class represent the system BIOS related properties such as Manufacturer, ReleaseDate and Version.

DCIM_BIOSEnumeration - This class represents the BIOS attributes of a BIOS setup in a computer system using set of properties such as AttributesName, CurrentVaue, PossibleValues, PossibleValuesDescription etc., Each BIOS attribute is represented by an instance of the DCIM_BIOSEnumeration class. This class defines properties used for setting parameters in the DellSystem Management BIOS.

DCIM_BIOSPassword - This class instances represent the BIOS passwords such as System(or Boot) and Admin(or Setup). If IsSet property is TRUE, it menas password is currently set, FALSE represent password is not set yet.

DCIM_BIOSService - This class having the SetBIOSAttributes() method to change(set) the values of a group of BIOS attributes. This method has three input parameter: AttributeName[], AttributeValue[] and AuthorizationToken. Invoking this method sets the DCIM_BIOSEnumeration.CurrentValue (and also DICM_BIOSPassword) that corresponds to the name specified by the AttributeName parameter and the values specified by the AttributeValue paramater.

Boot configuration

DCIM_BootSourceSetting - An instance of DCIM_BootSourceSetting represents a source from which a boot image can be loaded during the boot process. The DCIM_BootSourceSetting class has three boot string properties to identify boot devices: BootString, BIOSBootString and StructuredBootString.

DCIM_OrderedComponent - The DCIM_OrderedComponent association is used to indicate the order in which DCIM_BootSourceSetting instances should be attempted for a DCIM_BootConfigSetting instance. Use the DCIM_OrderedComponent.AssignedSequence property to determine the boot order.

DCIM_BootConfigSetting - The DCIM_BootConfigSetting class represents a boot configuration of a computer system. Invoking the ChangeBootOrder() method of this class shall set the order in which the instances of DCIM_BootSourceSetting are associated to a DCIM_BootConfigSetting instance. The method has one input parameter: Source.

DCIM_BootServiceCapabilities - This class represents the capabilities of the DCIM_BootService class.

Please find the detailed information about Boot configuration in the following DMTF profile,
http://www.dmtf.org/sites/default/files/standards/documents/DSP1012_1.0.0.pdf

Hardware

DCIM_Battery - The DCIM_Battery class describes properties of the notebook battery. Such properties include the description of the battery's charge status and the time it takes for the battery charge to be depleted.

DCIM_Card - This class represents the various PCI cards (including PCI riser) like Graphics, Network RAID etc., presently connected in the system and its FRU (Field Replaceable Unit) data.

DCIM_Chassis - This class represents the system's main chassis and its FRU data. The physical aspects of the instance of DCIM_ComputerSystem are represented by an instance of DCIM_Chassis through a DCIM_ComputerSystemPackage association.

DCIM_Chip - This class represents the chips like PCI Embedded, CPU & NIC and its FRU data.

DCIM_DesktopMonitor - This class describes capabilities and management of a desktop monitor (LCD or CRT).

DCIM_DeviceBay - This class describes the properties of system's modular device tray such as CD-ROM, CD-RW, DVD, DVD+RW, DVD+/-RW and Hard Disk.

DCIM_DisplayController - This class describes capabilities and management of a PC display/video controller.

DCIM_EthernetPort - This class describes the properties of network ports (Ethernet and Wireless LAN) interface in a managed system.

DCIM_Fan - This class defines various properties for cooling devices in this system.

DCIM_FlatPanel - This class describes Capabilities and management of the flat panel logical device on notebook computers.

DCIM_Memory - This class describes the system's total memory that includes both available and physical memory, as well as CPU cache memories.

DCIM_PhysicalMemory - This class represents the information about individual physical memory devices in the system.

DCIM_PhysicalPackage - This class represents the physical package information of Fan, CPU, Memory, Desktop monitor, Flat panel, Notebook battery, PCI cards etc., and its FRU data.

DCIM_PowerSupply - This class will display the supported Power Supply information of the managed system.

DCIM_Processor - This class represent the CPUs or processors in a managed system.

DCIM_SerialPort - This class describes the serial port on a main system and on a docking station.

DCIM_Slot - This class represent the slots in a managed system. Slots are Memory, PCI, CPU and Fan.

DCIM_USBPort - This class describes the properties of USB ports on a system.

DCIM_VideoHead - This class represent the display properties of display controller.

Indications

DCIM_AlertIndication - Denotes the occurrence of an alert or notification Event. Indications of this class contain information about an alerting situation such as PerceivedSeverity, ProbableCause, RecommendedAction and Trending. Alert can be retrieved by using query.

EXAMPLE: "SELECT * FROM DCIM_AlertIndication" - This query statement specifies that all supported properties of the DCIM_AlertIndication instance can be delivered to clients that have subscribed to this indication when such an event occurs.

DCIM_AlertIndicationSettingData - This class displays information of entity for which indications are generated.

RAID

The below classes used to monitor the RAID status of the system. Note: Monitoring supports only Intel RAID with OMCI 8.1 or above is installed.

DICM_ControllerView - This class instance represent the properties of Intel RAID Controller in a system.

DCIM_PhysicalDiskView - This class instance represent the available Physical Disk in a system.

DCIM_VirtualDiskView - This class instance represent the properties of configured Virtual Disk in a system.

Sensors

The following classes represent the properties of sensors in a managed system.

DCIM_Sensor - DCIM_Sensor class represent the Battery, Intrusion and Voltage sensors in a system.

DCIM_NumericSensor - DCIM_NumericSensor class represent the Temperature, Fan, Voltage and Current sensors in a system.

Software Info

DCIM_SoftwareInstallationService - This class provides the ability to perform installation or update of software such as BIOS.

DCIM_SoftwareIdentity - The DCIM_SoftwareIdentity.TargetTypes array property shall contain one or more strings that are used to advertise the compatibility with a Software Installation Service.

DCIM_ConcreteJob - This class CIM_ConcreteJob represents the job or task that may be started by the invocation of the DCIM_SoftwareInstallationService:InstallFromURI() method.

Others

DCIM_DHCPProtocolEndpoint - DCIM_DHCPProtocolEndpoint class helps to find current state of DHCP client and ProtocolIF type.

DCIM_IPProtocolEndpoint - DCIM_IPProtocolEndpoint class supports the method by which the IP Address Subnet Mask, and Gateway were assigned to the IPProtocolEndpoint.

DCIM_RemoteServiceAccessPoint - DCIM_RemoteServiceAccessPoint class represents various properties like AccessContext, InfoFormat.

DCIM_NetworkPortConfigurationService - DCIM_NetworkPortConfigurationService class describes various properties to identify current NetworkPortConfigurationService.

DCIM_TimeService - DCIM_TimeService class describes various properties to identify current timeservice instance and ManageTime method for handling request.

DCIM_AccountManagementService - DCIM_AccountManagementService class describes various properties to identify current status of this class.

DCIM_PowerManagementService - This class represents the power management service responsible for controlling the power of a computer system. Currently this class doesn't support the RequestPowerStateChange() method for client systems. To change power state, please use RequestStateChange() method in DICM_ComputerSystem.

DCIM_BootService - CIM_BootService shall be the Central Class of Bootcontoller profile. The instance of DCIM_BootService shall be the Central Instance of this profile.

Metrics

DCIM_BaseMetricValue - A metric instance is represented by an instances of this class or its subclass DCIM_AggregationMetricValue.

DCIM_BaseMetricDefinition - The definition of the metric is provided by an associated instance this class or DCIM_AggregationMetricDefinition class

DCIM_ManagedElement - One or more associated instances of this class provide context of metric.

DCIM_MetricService - This class provides the ability to query for metrics that a server-side implementation is able to collect, as well as whether or not collection of the metric is currently enabled through the ShowMetrics() method.

DCIM_MetricInstance - Each instance of DCIM_BaseMetricValue,DCIM_AggregationMetricValue shall be associated with exactly one instance of DCIM_BaseMetricDefinition and DCIM_AggregationMetricDefinition respectively through the DCIM_MetricInstance association.

DCIM_MetricForME - This class relates a metric to the managed element for which it was measured and Together with associations of the class DCIM_MetricDefForME are used to provide the context in which a metric is captured.

Record Log

DCIM_LogEntry - The DCIM_LogEntry class contains properties describing the information about individual records, such as message text and timestamp.

DCIM_RecordLog - This class represents the container for the log entries, the individual log entries, which are represented by the DCIM_LogEntry instances and also describes the general properties of the log, such as its maximal length and state..

DCIM_UseOfLog - At least one instance of the DCIM_UseOfLog association shall reference an instance of DCIM_RecordLog and an instance of the subclass of DCIM_ManagedSystemElement.

DCIM_LogManagesRecord - All log entries are aggregated under the DCIM_RecordLog instance through the DCIM_LogManagesRecord association.

Operating System

DCIM_OperatingSystem - This class provides the ability to perform basic management of operating systems installed on a managed system.

DCIM_InstalledOS - This class helps to representing the installed operating system. Each instance of DCIM_OperatingSystem shall be associated with exactly one instance of DCIM_ComputerSystem through the DCIM_InstalledOS association.

DCIM_RunningOS - This class helps to representing the running operating system. The instance of DCIM_OperatingSystem that represents the operating system running on the managed system may be associated to the instance of DCIM_ComputerSystem through the DCIM_RunningOS association.

Docking station

DCIM_Chassis - This class represents the system chassis and its FRU data.

DCIM_Docked - This class helps to identify the SUT is connected to docking system or not.

DCIM_Realizes - The Physical Elements could be associated to the logical representation of the managed element through the DCIM_Realizes association.

Embedded PCI Devices

DCIM_PCIDevice - This class displays all the PCI devices in a system. PCI device includes all the embedded PCI devices + devices plug into the PCI slots. Properties are bus number, device number, functional number etc.,

CPU

DCIM_Processor - Instances of this class represents the processor that hosts the processor core and processor characteristics.

DCIM_Memory - Each Instances of this class that represents the cache memory for the processor.

DCIM_ProcessorCapabilities class - This class instances describes the capabilities of the processor and this help to modeling the capabilities of a multi-core or multi-thread processor.

DCIM_AssociatedCacheMemory - Each instance of DCIM_Memory used by the processor shall be associated through an instance of the DCIM_AssociatedCacheMemory to the instance of DCIM_Processor.

Service processor

DCIM_AMTSettings - This class reports the Intel AMT remote monitoring settings.

DCIM_ASFSettings - This class reports the ASF configuration settings.

DCIM_VProSettings - This class reports the Intel Vpro technology settings.

DCIM_PowerManagementCapabilities - This class represents the power management capabilities of a computer system. The PowerStatesSupported property array is used to represent the power states that are supported by the associated computer system.

DCIM_SoftwareInstallationServiceCapabilities - This class define the capabilities of DCIM_SoftwareInstallationService such as the list of the methods supported, the types of software that it is capable of installing and the supported installation options such as install, update, repair, forced installation and silent mode installation.

DCIM_AssociatedPowerManagementService - This class associates the DCIM_ComputerSystem instance that represents the target computer system with the DCIM_PowerManagementService instance that represents the service responsible for controlling the power of a computer system. The PowerState property indicates the current power state of the computer system.

SNMP

OID Information

The Client Instrumentation MIB groups are identified by the SNMP OID 1.3.6.1.4.1.674.10892.1.<MIB group number>, where <MIB group number> is the number assigned to the MIB group.

MIB Groups

OMCI provides information through a number of MIB groups identified by the above SNMP OID. These groups are listed below in general categories. For detailed information on the properties in the groups, refer to the OMCI SNMP Reference Guide available at dell.com/OMConnectionsClient under OpenManage Client Instrumentation.

Instrumentation MIB Version Group

Defines version numbers of the Instrumentation MIB

Systems Management Software Group

Defines information about the systems management software and the supported systems management standards.

System State Group

Defines status, state, and redundancy for a system and its components.

Chassis Information Group

Defines chassis types, events, indicators, and basic input/output of a system.

Operating System Group

Defines variables for name, version, service pack, and other information about a system's operating system.

Remote Flash BIOS Group

Defines variables for updating the system's BIOS remotely.

Port Group

Defines variables for major port types such as keyboard, monitor, Universal Serial Bus (USB), and parallel and serial ports.

Device Group

Defines variables for pointing, keyboard, processor, cache, memory, and personal computer interface devices.

Slot Group

Defines variables for voltages, capabilities, states, and settings that are possible for slots.

SNMP Alerts

This section describes the variables that are sent to the management console to provide additional information about a trap or alert generated by some event on your system.

Alert Chassis Intrusion Normal

Chassis intrusion has returned to normal.

Alert Chassis Intrusion Detected

Chassis intrusion has been detected.

Key Acronyms

- ASF - Alert Standard Format
- ASM - Advance System Management
- CIM - Common Information Model
- CIMOM - CIM Object Manager
- COM/DCOM - Component Object Manager/Distributed Component Object Manager
- DMTF - Distributed Management Task Force
- IETF - Internet Engineering Task Force
- MOF - Managed Object Format
- OMCI - OpenManage Client Instrumentation
- SNMP - Simple Network Management Protocol
- SMBIOS - System Management BIOS
- SMBus - System Management Bus
- WMI - Windows Management Instrumentation
- WSH - Windows Scripting Host
- MIB - Management Information Base

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

Appendices

Appendix A - Sample Scripts

Wake on Lan

Below is a sample VB Script that will Enable the Wake On LAN on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_BIOSEnumeration

Property Name: BiosSetupWOL

Property Value: 4

```
' *****
' *** Name: SampleWakeOnLAN.vbs
' *** Purpose: To Enable the Wake On LAN on a Dell OMCI client.
' *** Usage: cscript.exe //nologo SampleWakeOnLAN.vbs <systemname>
' ***
' *** This sample script is provided as an example only, and has not been
' *** tested, nor is warranted in any way by Dell; Dell disclaims any
' *** liability in connection therewith. Dell provides no technical
' *** support with regard to such scripting. For more information on WMI
' *** scripting, refer to applicable Microsoft documentation.
' ***
' *** NOTE: Replace <Password> in line 73 (inside the quotes)
' *** with the correct password if there is any password set in the system.
' *** If both passwords(Admin and Boot) are set please replace it with Admin
' Password.
' *** If there is no password set in the system please leave it as empty.
' *****
```

Option Explicit

```
' *** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strKeyValue
Dim objWMIService
Dim colSystem
Dim objInstance
Dim oInParams
Dim returnValue
Dim strAttributeName(2)
Dim strAttributeValue(2)
Dim strAuthorizationToken

' *** Check that the right executable was used to run the script
' *** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
```

```

        (Wscript.Arguments.Count < 1) Then
        Call Usage()
        WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root\dcim\sysman"
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_BIOSEnumeration"
strKeyValue = "Root/MainSystemChassis/BIOSSetupParent/BiosSetupWOL"

'*** Retrieve the instance of DCIM_BIOSEnumeration class for the TPM
Set objInstance = GetObject("WinMgmts:{impersonationLevel=impersonate," &_
    "AuthenticationLevel=pktprivacy}\\\" & strComputerName & "\" &_
    strNameSpace & ":" & strClassName & "=" & Chr(34) & strKeyValue &
    Chr(34))
WScript.Echo objInstance.CurrentValue(0)
WScript.Echo objInstance.AttributeName

'*** All possible values for WOL are as follows:
'*** 1 = Disable
'*** 2 = Add-in
'*** 3 = On board
'*** 4 = LAN
'*** 5 = PXE boot enable
'*** 6 = LAN or WLAN
'*** 7 = WLAN only

If objInstance.CurrentValue(0) = 1 Then
    '*** Here is where you would perform an action such as writing the
computer
    '*** name out to a text file or enabling WoL with the following code...
    '*** This section will attempt to set the value to 4 (LAN) as this is the
most
    '*** popular for recent Dell systems
    '*** Initialize variables
    '    strClassName = "DCIM_BIOSService"
    '    strAttributeName(0) = objInstance.AttributeName
    '    strAttributeValue(0) = "4"
    '    strAuthorizationToken = "<Password>"

    '    returnValue = 0
    '*** Retrieve the instance of DCIM_BIOSService class
    '    Set objWMIService =
GetObject("WinMgmts:{impersonationLevel=impersonate," &_
    '        "AuthenticationLevel=pktprivacy}\\\" & strComputerName & "\" &_
    '        strNameSpace)
    '    Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)

    '    For each objInstance in ColSystem
    '        Set oInParams =
objInstance.Methods_("SetBIOSAttributes").InParameters.SpawnInstance_
    '        oInParams.AttributeName = strAttributeName
    '        oInParams.AttributeValue = strAttributeValue
    '        oInParams.AuthorizationToken = strAuthorizationToken
    '        Set returnValue = objInstance.ExecMethod_("SetBIOSAttributes",
oInParams)

```

```
'          Exit For
'      Next
End If

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
    WScript.Echo "Enabling Wake On LAN failed."
End If

'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCRLF & _
    "cscript.exe /nologo SampleWakeOnLAN.vbs <systemname>"
WScript.Echo strMessage
End Sub
```

Trusted Platform Module

Below is a sample VB Script that will Enable Trusted Platform Module on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_BIOSEnumeration

Property Name: BiosSetupTPM

Property Value: 1

```
'*****
'*** Name: SampleTrustedPlatformModule.vbs
'*** Purpose: To Enable the Trusted Platform Module on a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleTrustedPlatformModule.vbs <systemname>
'***
'*** This sample script is provided as an example only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides no technical
'*** support with regard to such scripting. For more information on WMI
'*** scripting, refer to applicable Microsoft documentation.
'***
'*** NOTE: Replace <Password> in line 63 (inside the quotes)
'*** with the correct password if there is any password set in the system.
'*** If both passwords(Admin and Boot) are set please replace it with Admin
Password.
'*** If there is no password set in the system please leave it as empty.
'*****
```

Option Explicit

```
'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
```



```

Dim strKeyValue
Dim objWMIService
Dim ColSystem
Dim objInstance
Dim oInParams
Dim returnValue
Dim strAttributeName(2)
Dim strAttributeValue(2)
Dim strAuthorizationToken

'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
    (Wscript.Arguments.Count < 1) Then
    Call Usage()
    WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root\dcim\sysman"
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_BIOSEnumeration"
strKeyValue = "Root/MainSystemChassis/BIOSSetupParent/BiosSetupTPM"

'*** Retrieve the instance of DCIM_BIOSEnumeration class for the TPM
Set objInstance = GetObject("WinMgmts:{impersonationLevel=impersonate," & _
    "AuthenticationLevel=pktprivacy}\\\" & strComputerName & "\" & _
    strNameSpace & ":" & strClassName & "=" & Chr(34) & strKeyValue &
    Chr(34))
WScript.Echo objInstance.CurrentValue(0)
WScript.Echo objInstance.AttributeName

'*** A value of 2 means that it is disabled
If objInstance.CurrentValue(0) = 2 Then
    '*** Here is where you would perform an action such as writing the
    computer
    '*** name out to a text file or enabling WoL with the following code...
    '*** Initialize variables
    '    strClassName = "DCIM_BIOSService"
    '    strAttributeName(0) = objInstance.AttributeName
    '    strAttributeValue(0) = "1"
    '    strAuthorizationToken = "<Password>"

    '    returnValue = 0
    '*** Retrieve the instance of DCIM_BIOSService class
    '    Set objWMIService =
GetObject("WinMgmts:{impersonationLevel=impersonate," & _
    '    "AuthenticationLevel=pktprivacy}\\\" & strComputerName & "\" & _
    '    strNameSpace)
    '    Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)

    '    For each objInstance in ColSystem
    '        Set oInParams =
objInstance.Methods_("SetBIOSAttributes").InParameters.SpawnInstance_
    '        oInParams.AttributeName = strAttributeName
    '        oInParams.AttributeValue = strAttributeValue
    '        oInParams.AuthorizationToken = strAuthorizationToken

```

```
'          Set returnValue = objInstance.ExecMethod_("SetBIOSAttributes",
oInParams)
'          Exit For
'      Next
End If

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
    WScript.Echo "Script for Trusted Platform Module failed."
End If

'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCRLF & _
    "cscript.exe /nologo SampleTrustedPlatformModule.vbs <systemname>"
WScript.Echo strMessage
End Sub
```

Change Asset Tag

Below is a sample VB Script that will help assist Change Asset Tag on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_Chassis

Property Method: ChangeAssetTag

Property Value: New_Tag

```
' *****
' *** Name: SampleChangeAssetTag.vbs
' *** Purpose: To Change Asset Tag on a Dell OMCI client.
' *** Usage: cscript.exe //nologo SampleChangeAssetTag.vbs <systemname>
' ***
' *** This sample script is provided as an example only, and has not been
' *** tested, nor is warranted in any way by Dell; Dell disclaims any
' *** liability in connection therewith. Dell provides no technical
' *** support with regard to such scripting. For more information on WMI
' *** scripting, refer to applicable Microsoft documentation.
' *** Please provide the new asset tag in line 51 and BIOS or Boot password
' *** in line 52 if any one of them is set in BIOS.
' *****
```

Option Explicit

```
' *** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strKeyValue
Dim objInstance
```

```

Dim strPropName
Dim strPropValue
Dim oInParams
Dim objWMIService
Dim returnValue
Dim ColSystem

'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
    (Wscript.Arguments.Count < 1) Then
    Call Usage()
    WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/dcim/sysman"
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_Chassis"
returnValue = 0
'*** Retrieve the instance of DCIM_Chassis class
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate," &_
    "AuthenticationLevel=pktpriacy}\\" & strComputerName & "\" &_
    strNameSpace)
    Set ColSystem=objWMIService.execquery ("Select * from " &
strClassName)

For each objInstance in ColSystem
    Set oInParams=
objInstance.Methods_("ChangeAssetTag").InParameters.SpawnInstance_
    oInParams.NewValue = "New_Tag"
    oInParams.AuthorizationToken = "password"
    Set returnValue = objInstance.ExecMethod_("ChangeAssetTag", oInParams)
Next

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
    WScript.Echo "Operation failed."
End If
'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCRLF & _
    "cscript.exe /nologo SampleChangeAssetTag.vbs <systemname>"
WScript.Echo strMessage
End Sub

```

Get System Summary

Below is a sample VB Script that will help get System Summary on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_Chassis and DCIM_BIOSElement

```
'*****
'*** Name: SampleSystemSummary.vbs
'*** Purpose: To check for System Summary on a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleSystemSummary.vbs <systemname>
'***
'*** This sample script is provided as an example only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides no technical
'*** support with regard to such scripting. For more information on WMI
'*** scripting, refer to applicable Microsoft documentation.
'*****
```

Option Explicit

```
'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strClassNameBIOS
Dim strKeyValueChassis
Dim objInstance
Dim ColSystem
Dim objWMIService
Dim strMessage
'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
    (Wscript.Arguments.Count < 1) Then
    Call Usage()
    WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/dcim/sysman"
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_Chassis"
strClassNameBIOS = "DCIM_BIOSElement"
strKeyValueChassis = "DCIM_Chassis"

'*** Retrieve the instance of DCIM_Chassis class (there should
'*** only be 1 instance).
'*** Establish a connection to the dcim\sysman namespace
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate," & _
    "AuthenticationLevel=pktprivacy}\\\" & strComputerName & "\" & _
    strNameSpace)

Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)
For each objInstance in ColSystem
    if (objInstance.CreationClassName = strKeyValueChassis) Then
        strMessage = "Asset Tag: "
        strMessage = strMessage &
objInstance.Properties_.Item("ElementName").Value
        strMessage = strMessage & vbCRLF & "Service Tag: "
        strMessage = strMessage &
objInstance.Properties_.Item("Tag").Value
```

```

        End if
    Next

    '*** Retrieve all instances of DCIM_BIOSElement (there should
    '*** only be 1 instance).
    Set ColSystem=objWMIService.execquery ("Select * from " & strClassNameBIOS)
    For each objInstance in ColSystem
        strMessage = strMessage & vbCRLF & "BIOS Version: "
        strMessage = strMessage &
        objInstance.Properties_.Item("Version").Value
    Next
    '*** Display the results
    WScript.Echo strMessage

    '*** Sub used to display the correct usage of the script
    Sub Usage()
    Dim strMessage
    strMessage = "incorrect syntax. You should run: " & vbCRLF & _
    "cscript.exe //nologo SampleSystemSummary.vbs <systemname>"
    WScript.Echo strMessage
    End Sub

```

Force PXE on Next Boot

Below is a sample VB Script that will Force PXE on Next Boot on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_BIOSService

Property Name: Force PXE on Next Boot

Property Value: 2

```

'*****
'*** Name: SampleForcePXE.vbs
'*** Purpose: To force PXE Boot on a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleForcePXE.vbs <systemname>
'***
'*** This sample script is provided as an example only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides no technical
'*** support with regard to such scripting. For more information on WMI
'*** scripting, refer to applicable Microsoft documentation.
'***
'*** NOTE: Replace <Password> in line 53 (inside the quotes)
'*** with the desired values if there is any password set in the system.
'*** If both passwords(Admin and Boot) are set please replace it with Admin
'*** Password.
'*** If there is no password set in the system please leave it as empty.
'*****

```

Option Explicit

```

'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strKeyValue
Dim objInstance
Dim strPropName
Dim strPropValue
Dim oInParams
Dim objWMIService
Dim returnValue
Dim ColSystem
Dim strAttributeName(2)
Dim strAttributeValue(2)
Dim strAuthorizationToken

'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
(Wscript.Arguments.Count < 1) Then
Call Usage()
WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/dcim/sysman"
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_BIOSService"
strAttributeName(0) = "Force PXE on Next Boot"
'*** All possible values for Force PXE on Next Boot are as follows:
'*** 1 = Disable
'*** 2 = Enable
strAttributeValue(0) = "2"
strAuthorizationToken = "<Password>"

returnValue = 0
'*** Retrieve the instance of DCIM_BIOSService class
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate," &_
"AuthenticationLevel=pktprivacy}\\\" & strComputerName & "\" &_
strNameSpace)
Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)

For each objInstance in ColSystem
Set oInParams=
objInstance.Methods_("SetBIOSAttributes").InParameters.SpawnInstance_
oInParams.AttributeName = strAttributeName
oInParams.AttributeValue = strAttributeValue
oInParams.AuthorizationToken = strAuthorizationToken
Set returnValue = objInstance.ExecMethod_("SetBIOSAttributes", oInParams)
Next

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
WScript.Echo "Enabling Force PXE on Next Boot failed."

```

```
End If
'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCrLf & _
"cscript.exe /nologo SampleForcePXE.vbs <systemname>"
WScript.Echo strMessage
End Sub
```

Get BIOS Settings

Below is a sample VB Script that will Get BIOS Settings on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_BIOSEnumeration

```
'*****
'***
'*** Name: SampleGetBBIOSSettings.vbs
'*** Purpose: To get the current list of BIOS Settings on a Dell OMCI 8.0
client.
'*** Usage: cscript.exe /nologo SampleGetBIOSSettings.vbs <systemname>
'***
'*** This sample script is provided as an example only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides no technical
'*** support with regard to such scripting. For more information on WMI
'*** scripting, refer to applicable Microsoft documentation.
'*****
'***
'*** Declare variables
Dim objWMIService
Dim strComputerName
Dim strNameSpace
Dim strClassName
Dim ColSystem
Dim fso
Dim txtfile
Dim outString
Dim iTKVal
Dim ArrayItem
Dim objInstance
Dim strSpecialTK1
Dim strSpecialTK2

'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
(Wscript.Arguments.Count < 1) Then
```

```

Call Usage()
WScript.Quit
End If

'*** Initialize variables
Set fso = CreateObject("Scripting.FileSystemObject")
strComputerName = WScript.Arguments(0)
outString = ""
strNameSpace = "root/dcim/sysman"
strClassName = "DCIM_BIOSEnumeration"
strSpecialTK1 = "Auto On Hour"
strSpecialTK2 = "Auto On Minute"

iTKVal = -1
ArrayItem = 0

'*** Establish a connection to the dcim\sysman namespace
'*** Retrieve the instance of DCIM_BIOSEnumeration class
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate," &_
"AuthenticationLevel=pktprivacy}\\\" & strComputerName & "\" &_
strNameSpace)

'*** Set up the output file
set txtfile = fso.CreateTextfile("BIOS_Output.csv", true)
Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)
outString = ""
For each objInstance in ColSystem
'*** Check every instance
iTKVal = objInstance.Properties_.Item("CurrentValue").Value(0)
'*** Auto On Hour and Auto On Minutes don't have PossibleValues and
PossibleValuesDescription properties
if (objInstance.AttributeName <> strSpecialTK1) and
(objInstance.AttributeName <> strSpecialTK2) Then
For Each element in objInstance.Properties_.Item("PossibleValues").Value
if (objInstance.Properties_.Item("PossibleValues").Value(ArrayItem) = iTKVal)
Then
outString = outString & objInstance.AttributeName & ", " &
(objInstance.Properties_.Item("PossibleValuesDescription").Value(ArrayItem))
& VbCrLf
end if
ArrayItem = ArrayItem + 1
Next
else
'*** Auto On Hour and Auto On Minutes token copy the current value directly
outString = outString & objInstance.AttributeName & ", " & iTKVal & VbCrLf
end if
ArrayItem = 0
Next

txtfile.WriteLine outString

'*** Close the file
txtfile.close

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
WScript.Echo "Enabling Trusted Platform Module failed."

```


End If

```
'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCRLF & _
"cscript.exe /nologo SampleGetBIOSSettings.vbs <systemname>"
WScript.Echo strMessage
End Sub

'=====
' End
'=====
```

Trusted Platform Module

Below is a sample VB Script that will Set Admin password on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_BIOSService

Property Name: AdminPwd

Property Value: "new_password"

```
'*****
'*** Name: SampleSetAdminPassword.vbs
'*** Purpose: To Set/Change the BIOS Admin password on a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleSetAdminPassword.vbs <systemname>
'***
'*** This sample script is provided as an example only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides no technical
'*** support with regard to such scripting. For more information on WMI
'*** scripting, refer to applicable Microsoft documentation.
'***
'*** NOTE: Replace <old_password> and <new_password> in line 45 and 44
'*** (inside the quotes) with the desired values.
'*****
```

Option Explicit

```
'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim objInstance
Dim strAttributeName(2)
Dim strAttributeValue(2)
Dim strAuthorizationToken
```

```

Dim returnValue
Dim objWMIService
Dim ColSystem
Dim oInParams

'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
(Wscript.Arguments.Count < 1) Then
Call Usage()
WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/dcim/sysman"
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_BIOSService"
strAttributeName(0) = "AdminPwd"
strAttributeValue(0) = "<new_password>"
strAuthorizationToken = "<old_password>"
returnValue = 0
'*** Retrieve the instance of DCIM_BIOSService class
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate," & _
"AuthenticationLevel=pktprivacy}\\\" & strComputerName & "\" & _
strNameSpace)
Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)

For each objInstance in ColSystem
Set oInParams=
objInstance.Methods_("SetBIOSAttributes").InParameters.SpawnInstance_
oInParams.AttributeName = strAttributeName
oInParams.AttributeValue = strAttributeValue
oInParams.AuthorizationToken = strAuthorizationToken
Set returnValue = objInstance.ExecMethod_("SetBIOSAttributes", oInParams)
Next

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
WScript.Echo "Change admin password failed."
End If

'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCRLF & _
"cscript.exe /nologo SampleSetAdminPassword.vbs <systemname>"
WScript.Echo strMessage
End Sub

```

Clear Chassis Intrusion

Below is a sample VB Script that will Clear Chassis Intrusion on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_BIOSService

Property Name: Chassis Intrusion Status

Property Value: 4

```
*****
'*** Name: SampleChassisIntrusionClear.vbs
'*** Purpose: To clear the chassis intrusion status on a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleChassisIntrusionClear.vbs <systemname>
'***
'*** This sample script is provided as an example only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides no technical
'*** support with regard to such scripting. For more information on WMI
'*** scripting, refer to applicable Microsoft documentation.

'*** NOTE: Replace <Password> in line 57 (inside the quotes)
'*** with the desired values if there is any password set in the system.
'*** If both passwords(Admin and Boot) are set please replace it with Admin
Password.
'*** If there is no password set in the system please leave it as empty.
*****
```

Option Explicit

```
'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strKeyValue
Dim objInstance
Dim strPropName
Dim strPropValue
Dim oInParams
Dim objWMIService
Dim returnValue
Dim ColSystem
Dim strAttributeName(2)
Dim strAttributeValue(2)
Dim strAuthorizationToken

'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
(Wscript.Arguments.Count < 1) Then
Call Usage()
WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/dcim/sysman"
```

```
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_BIOSService"
strAttributeName(0) = "Chassis Intrusion Status"

'*** All possible values for Chassis Intrusion Status are as follows:
'*** 1 = Tripped (Read-only)
'*** 2 = Door open (Read-only)
'*** 3 = Door closed (Read-only)
'*** 4 = Trip reset (Write-only)

strAttributeValue(0) = "4"
strAuthorizationToken = "<Password>"

returnValue = 0
'*** Retrieve the instance of DCIM_BIOSService class
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate," & _
"AuthenticationLevel=pktprivacy}\\\" & strComputerName & "\" & _
strNameSpace)
Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)

For each objInstance in ColSystem
Set oInParams=
objInstance.Methods_("SetBIOSAttributes").InParameters.SpawnInstance_
oInParams.AttributeName = strAttributeName
oInParams.AttributeValue = strAttributeValue
oInParams.AuthorizationToken = strAuthorizationToken
Set returnValue = objInstance.ExecMethod_("SetBIOSAttributes", oInParams)
Next

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
WScript.Echo "Clear chassis intrusion failed."
End If

'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCRLF & _
"cscript.exe /nologo SampleChassisIntrusionClear.vbs <systemname>"
WScript.Echo strMessage
End Sub
```

Shutdown the system

Below is a sample VB Script that will force Shutdown on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_ComputerSystem

```
'*** Name: SampleShutdown.vbs
'*** Purpose: To clear the chassis intrusion status on a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleShutdown.vbs <systemname>
'***
'*** This sample script is provided as an example only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides no technical
'*** support with regard to such scripting. For more information on WMI
'*** scripting, refer to applicable Microsoft documentation.
'*****
```

Option Explicit

```
'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strKeyValue
Dim objInstance
Dim strPropName
Dim strPropValue
Dim oInParams
Dim objWMIService
Dim returnValue
Dim ColSystem

'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
(Wscript.Arguments.Count < 1) Then
Call Usage()
WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/dcim/sysman"
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_ComputerSystem"
returnValue = 0
'*** Retrieve the instance of DCIM_ComputerSystem class
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate," & _
"AuthenticationLevel=pktpriority}\\\" & strComputerName & "\" & _
strNameSpace)
Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)

For each objInstance in ColSystem
Set oInParams=
objInstance.Methods_("RequestStateChange").InParameters.SpawnInstance_
oInParams.RequestedState = 3
Set returnValue = objInstance.ExecMethod_("RequestStateChange", oInParams)
Next

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
WScript.Echo "Operation failed."
End If
```

```
'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCRLF & _
"cscript.exe /nologo SampleShutdown.vbs <systemname>"
WScript.Echo strMessage
End Sub
```

Restart the system

Below is a sample VB Script that will Enable Trusted Platform Module on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_ComputerSystem

```
'*****
'*** Name: SampleRestart.vbs
'*** Purpose: To clear the chassis intrusion status on a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleRestart.vbs <systemname>
'***
'*** This sample script is provided as an example only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides no technical
'*** support with regard to such scripting. For more information on WMI
'*** scripting, refer to applicable Microsoft documentation.
'*****
```

Option Explicit

```
'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strKeyValue
Dim objInstance
Dim strPropName
Dim strPropValue
Dim oInParams
Dim objWMIService
Dim returnValue
Dim ColSystem

'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
(Wscript.Arguments.Count < 1) Then
Call Usage()
WScript.Quit
End If
```

```

'*** Initialize variables
strNameSpace = "root/dcim/sysman"
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_ComputerSystem"
returnValue = 0
'*** Retrieve the instance of DCIM_ComputerSystem class
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate," & _
"AuthenticationLevel=pktprivacy}\\\" & strComputerName & "\\\" & _
strNameSpace)
Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)

For each objInstance in ColSystem
Set oInParams=
objInstance.Methods_("RequestStateChange").InParameters.SpawnInstance_
oInParams.RequestedState = 11
Set returnValue = objInstance.ExecMethod_("RequestStateChange", oInParams)
Next

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
WScript.Echo "System restart failed."
End If

'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCRLF & _
"cscript.exe /nologo SampleRestart.vbs <systemname>"
WScript.Echo strMessage
End Sub

```

Get Boot Order

Below is a sample VB Script that will Get Boot Order on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_BootSourceSetting

```

'*****
'*** Name: SampleGetBootOrder.vbs
'*** Purpose: To get the current Boot Order on a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleGetBootOrder.vbs <systemname>
'***
'*** This sample script is provided as an example only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides no technical
'*** support with regard to such scripting. For more information on WMI
'*** scripting, refer to applicable Microsoft documentation.
'*****

```

Option Explicit

```

'*** Declare variables
Dim objWMIService
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strClassNameOrder
Dim ColSystem
Dim ColOrder
Dim objInstance
Dim objInstanceOrder
Dim strPropName
Dim strPropValue
Dim devStatus

'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
    (Wscript.Arguments.Count < 1) Then
    Call Usage()
    WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/dcim/sysman"
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_BootSourceSetting"
strClassNameOrder = "DCIM_OrderedComponent"
devStatus = 1
'*** Establish a connection to the dcim\sysman namespace
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate," & _
    "AuthenticationLevel=pktpriority}\\\" & strComputerName & "\" & _
    strNameSpace)

'*** Retrieve all instances of DCIM_OrderedComponent
Set ColOrder=objWMIService.execquery ("Select * from " & strClassNameOrder)

'*** Retrieve all instances of DCIM_BootSourceSetting
Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)

wscript.echo "Id" & vbTab & vbTab & vbTab & vbTab & "Name" & vbTab & "Order"
& vbTab & "Status"

For each objInstance in ColSystem
    '*** Need to add code to get boot device info
    '*** Boot order is stored in DCIM_OrderedComponent.AssignedSequence
    property
    '*** Need to find it through Instance ID
    For each objInstanceOrder in ColOrder
        if (InStr(objInstanceOrder.PartComponent,objInstance.InstanceID)> 0)
    Then
        '*** If AssignedSequence is 0 which means it is disabled -
device status is 0
        '*** Otherwise device is enabled - device status is 1
        if(objInstanceOrder.AssignedSequence = 0) Then
            devStatus = 0
        End if
    End if

```



```

                wscript.echo objInstance.InstanceID & vbTab &_
                objInstance.ElementName & vbTab &_
                objInstanceOrder.AssignedSequence & vbTab &_
                devStatus
            End if
        Next
Next

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
    WScript.Echo "An error occurred in obtaining the boot order."
End If

'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCRLF & _
    "cscript.exe /nologo SampleGetBootOrder.vbs <systemname>"
WScript.Echo strMessage
End Sub

'*****

```

Set Boot Order

Below is a sample VB Script that will Set specific Boot Order on a Dell OMCI client. This script can be run locally or remotely using the name of the system.

Namespace: root\dcim\sysman

Class Name: DCIM_BootConfigSetting

```

'*****
'*** Name: SampleSetBootOrder.vbs
'*** Purpose: To set the Boot Order on a Dell OMCI client.
'*** Usage: cscript.exe //nologo SampleSetBootOrder.vbs <systemname>
'***
'*** This sample script is provided as an example only, and has not been
'*** tested, nor is warranted in any way by Dell; Dell disclaims any
'*** liability in connection therewith. Dell provides no technical
'*** support with regard to such scripting. For more information on WMI
'*** scripting, refer to applicable Microsoft documentation.
'***
'*** NOTE: Replace <Password> in line 70 (inside the quotes)
'*** with the desired values if there is any password set in the system.
'*** If both passwords(Admin and Boot) are set please replace it with Admin
Password.
'*** If there is no password set in the system please leave it as empty.
'*** Please provide the new boot order from line 61.
'*** See comments from line 58.
'*****

Option Explicit

```

```

'*** Declare variables
Dim strNameSpace
Dim strComputerName
Dim strClassName
Dim strClassNameOrder
Dim objInstance
Dim objInstanceOrder
Dim strNewOrder()
Dim strOldOrder()
Dim strNewOrderInput(5)
Dim strAuthorizationToken
Dim returnValue
Dim objWMIService
Dim ColSystem
Dim ColSystemOrder
Dim oInParams
Dim arraySize
Dim arrayIndex
Dim i
Dim j

'*** Check that the right executable was used to run the script
'*** and that all parameters were passed
If (LCase(Right(WScript.FullName, 11)) = "wscript.exe" ) Or _
    (Wscript.Arguments.Count < 1) Then
    Call Usage()
    WScript.Quit
End If

'*** Initialize variables
strNameSpace = "root/dcim/sysman"
strComputerName = WScript.Arguments(0)
strClassName = "DCIM_BootConfigSetting"
strClassNameOrder = "DCIM_OrderedComponent"
arraySize = 0
arrayIndex = 0
returnValue = 0
'*** Please provide the new order below to the strNewOrderInput
'*** The order will be array index. The value of array will be the number in
the Instance ID
'*** in DCIM_BootSourceSetting. Following values are just a example.
strNewOrderInput(0)="0"
strNewOrderInput(1)="1"
strNewOrderInput(2)="2"
strNewOrderInput(3)="3"
strNewOrderInput(4)="4"
strNewOrderInput(5)="5"

'*** Admin or Boot password input here
strAuthorizationToken = ""

'*** Retrieve the instance of DCIM_OrderedComponent class
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate," &_
    "AuthenticationLevel=pktprivacy}\" & strComputerName & "\" &_
    strNameSpace)
Set ColSystemOrder=objWMIService.execquery ("Select * from " &
strClassNameOrder)

```

```

'*** Assign the string with contains the instance ID in PartComponent to
strOldOrder array of strings
For each objInstanceOrder in ColSystemOrder
    arraySize = arraySize + 1
Next
'redefine the array for the correct array size
ReDim strOldOrder(arraySize - 1)
ReDim strNewOrder(arraySize - 1)

For each objInstanceOrder in ColSystemOrder
    strOldOrder(arrayIndex) = objInstanceOrder.PartComponent
    arrayIndex = arrayIndex + 1
Next

'*** Rearrange the array of strings with the expect order in strNewOrder
array
For i = 0 to (arrayIndex - 1)
    For j = 0 to (arrayIndex - 1)
        if(InStr(strOldOrder(j),"UEFI")>0) then
            if(InStr(strOldOrder(j),":index-"&strNewOrderInput(i))> 0)
then
                strNewOrder(i) = strOldOrder(j)
                j=arrayIndex
            end if
        else
            if(InStr(strOldOrder(j),":"&strNewOrderInput(i))> 0) then
                strNewOrder(i) = strOldOrder(j)
                j=arrayIndex
            end if
        end if
    Next
Next

'*** Retrieve the instance of DCIM_BootConfigSetting class
'*** Execute ChangeBootOrder method with corresponding parameters
Set ColSystem=objWMIService.execquery ("Select * from " & strClassName)
For each objInstance in ColSystem
    Set oInParams=
objInstance.Methods_("ChangeBootOrder").InParameters.SpawnInstance_
    oInParams.source = strNewOrder
    oInParams.AuthorizationToken = strAuthorizationToken
    Set returnValue = objInstance.ExecMethod_("ChangeBootOrder",
oInParams)
Next

'*** If any errors occurred, let the user know
If Err.Number <> 0 Then
    WScript.Echo "Set boot order failed."
End If

'*** Sub used to display the correct usage of the script
Sub Usage()
Dim strMessage
strMessage = "incorrect syntax. You should run: " & vbCRLF & _
    "cscrip.exe /nologo SampleSetBootOrder.vbs <systemname>"
WScript.Echo strMessage
End Sub

```