



## USER GUIDE



### Cisco IMC PowerTool, Release 1.x

1. [Overview](#)
2. [Management Information Model](#)
3. [System Requirements](#)
4. [Installation](#)
5. [Examples](#)
6. [SamplesRelated Cisco UCS Documentation and Documentation Feedback](#)
7. [Obtaining Documentation and Submitting a Service Request](#)

## 1 Overview

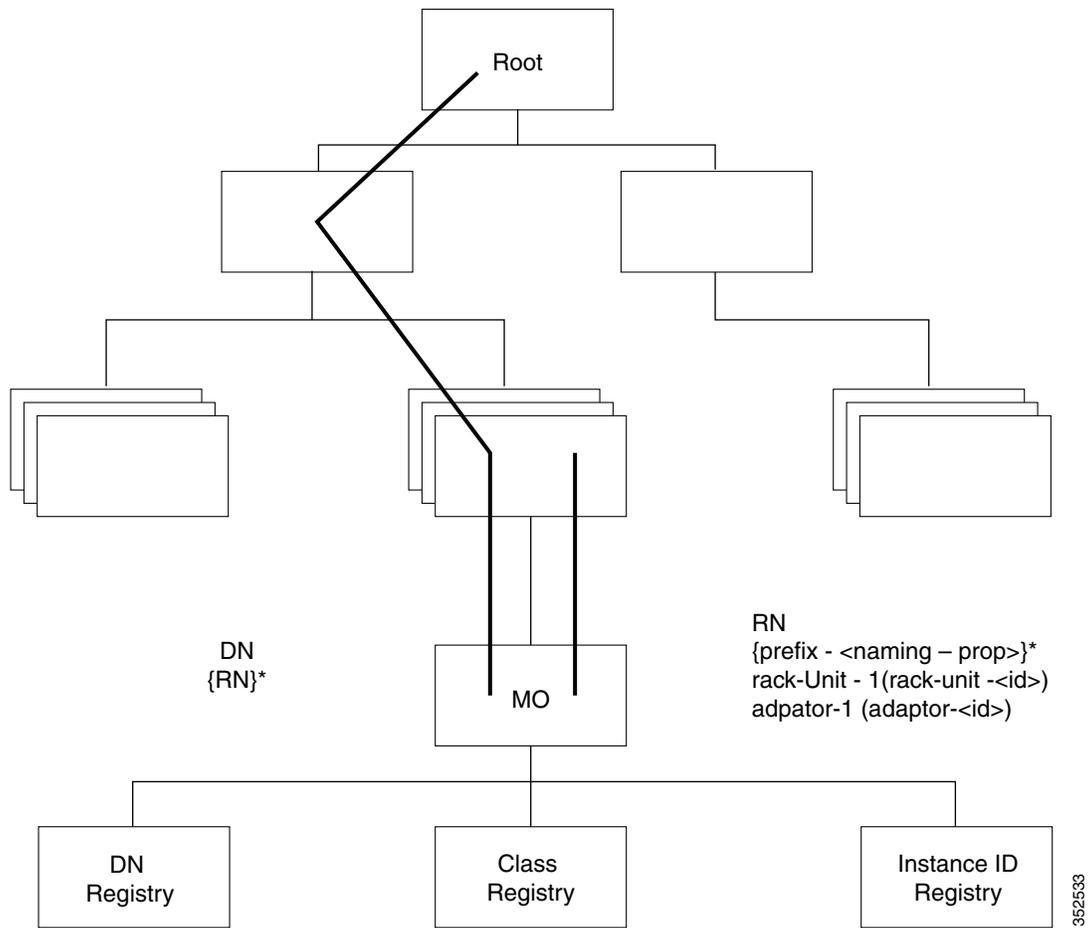
Cisco IMC PowerTool is a PowerShell module which helps automate aspects of Cisco UCS C-Series Stand alone Rack Mount servers and Cisco UCS E-Series servers using XML APIs. Cisco IMC PowerTool enables easy integration with existing IT management processes and tools.

The Cisco IMC PowerTool cmdlets work on the Cisco IMC Management Information Tree (MIT). The cmdlets allow you to create, modify, or delete actions on the Managed Objects (MOs) in the tree.

## 2 Management Information Model

All the physical and logical components that comprise a Cisco C-Series and E-Series servers are represented in a hierarchical Management Information Model (MIM), which is referred to as the Management Information Tree (MIT). Each node in the tree represents a Managed Object (MO), which is uniquely identified by its distinguished name (DN). See [Figure 1](#).

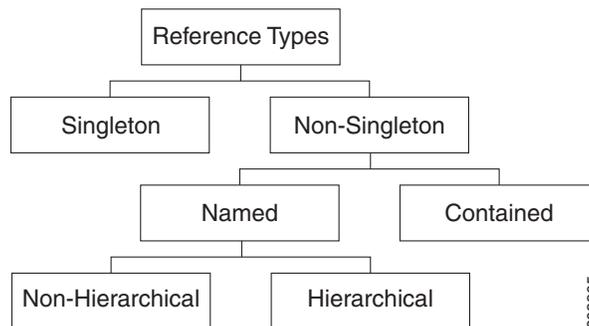
Figure 1 Management Information Model



## Managed Objects

Managed Objects (MOs) (see [Figure 2](#)) are abstractions of Cisco IMC MIT resources, such as CPUs, DIMMs, adapter cards, fans, and power supply units. Managed Objects represent any physical or logical entity that is configured or managed in the Cisco IMC MIT. For example, physical entities such as CPUs, DIMMs, adapter cards and fans and logical entities such as users, communication services like HTTP, SSH are represented as MOs.

Figure 2 Managed Objects



Each MO is uniquely identified in the tree with its distinguished name (DN) and can be uniquely identified within the context of its parent with its relative name (RN). The DN identifies the place of the MO in the MIT. A DN is a concatenation of all the relative names that start from the root to the MO itself. Essentially, DN = [RN]/[RN]/[RN]/.../[RN].

In the following example, DN provides a fully qualified name for adapter-1 in the model.

```
< dn = "sys/rack-unit-1/adaptor-1" />
```

The above written DN is composed of the following RN:

```
topSystem MO: rn="sys" computeRackUnit MO: rn="rack-unit-1" adaptorUnit MO: rn ="adaptor-<id>"
```

A relative name (RN) might have a value of one or more of the MO properties embedded in it. This allows you to differentiate multiple MOs of the same type within the context of the parent. Any properties that form part of the RN as described earlier are referred to as naming properties.

For instance, adaptor MOs reside under a rack unit MO. The adaptor MO contains the adaptor identifier as part of its Rn(adaptor-[Id]), which uniquely identifies each adaptor MO in the context of a rack unit.

## 3 System Requirements

Before installing Cisco IMC PowerTool, ensure that the system meets the following requirements:

- Windows PowerShell 2.0 or above must be installed on the system
- .NET Framework Version 2.0 SP1 or higher

### Cisco IMC

Cisco IMC PowerTool is compatible with the following Cisco IMC releases:

- Release 1.5 and above for the C-Series servers
- Release 2.2(1) and above for the E-Series servers

### Windows PowerShell Requirements

- Cisco IMC PowerTool requires Windows PowerShell v2.0 or higher version to be installed on the system
- Check the Microsoft web site for specific Windows PowerShell requirements

### Installation Requirements

- Before installing Cisco IMC PowerTool, you must do the following:
- Close any instances of PowerShell running with the PowerTool module loaded.
- Uninstall the beta versions of the Cisco IMC PowerTool

### Methods

Methods are Cisco UCS XML APIs used to manage and monitor the system. The following methods are supported:

- Authentication
- aaaLogin—Initial method for logging in.
- aaaRefresh—Refreshes the current authentication cookie.
- aaaLogout—Exits the current session and deactivates the corresponding authentication cookie.
- configResolveDn—Retrieves objects by DN.
- configResolveClass—Retrieves objects of a given class.
- configResolveChildren—Retrieves the child objects of an object.

- configResolveParent—Retrieves the parent object of an object.
- configConfMo—Affects a single managed object (for example, a DN).
- eventSubscribe—To register for events

## Cisco IMC PowerTool Mapping

All but about 10 of the Cisco IMC PowerTool cmdlets are generated from the MO specification. A noun is used in place of the type (Fan instead of EquipmentFan and so on). Get, Add, Set, Remove cmdlets or a subset is generated for the various MO types. All cmdlets support the XML parameter, which dumps the XML request and response on the screen.

**Add Cmdlet**—Uses the ConfigConfMo method with the MO status “created” with the specified property values. If the Force parameter is specified, there is no prompt for confirmation.

**Get Cmdlet**—Uses the ConfigResolveClass method to retrieve MOs. XML API of Cisco UCS C-Series Rack Mount servers do not support any filters. So if any property parameters are specified, PowerTool gets all the instances of the specified class and filters them on the client side using the property values.

**Set Cmdlet**—Uses the ConfigConfMo method with MO status “modified” with the specified property values. If the Force parameter is specified, there is no prompt for confirmation.

**Remove Cmdlet**—Uses the ConfigConfMo method with the MO status “deleted”. If the Force parameter is specified, there is no prompt for confirmation.

This table lists the properties that can be specified for a given verb:

Property	Get	Add	Set
Naming	Yes (Positional)	Yes (Positional)	No
Create-Only	Yes	Yes	No
Read-Write	Yes	Yes	Yes
Operational/ Read-Only	Yes	No	No

This table lists the types that can come down the pipeline for corresponding cmdlets:

Verb/Type	Pipeline Input
Get	Singleton—none non-singleton—Parent Type
Add	Singleton—none non-singleton—Parent Type
Set	MO has naming property—Same type MO has no naming property—Same or Parent Type
Remove/Clear	Same Type

This table lists the methods invoked to generate the required XML requests:

Cmdlet	Method
Add-Imc <sup>1</sup> Set-Imc <sup>1</sup>	ConfigConfMo
Get-Imc <sup>1</sup>	ConfigResolveClass with client side filters
Get-ImcManagedObject -ClassId	ConfigResolveClass
Get-ImcManagedObject -ClassId -Dnlist	ConfigResolveClass (The output is then filtered for the matching Dns)

Get-ImcManagedObject -Dn	ConfigResolveDn
Connect-Imc	AaaLogin
Disconnect-Imc	AaaLogout
Background <sup>1</sup>	AaaRefresh
Get-ImcChild	ConfigResolveChildren

1. This is not a cmdlet. It is a background service.

Get-ImcCmdletMeta is a useful cmdlet to explore the MO types, the corresponding nouns, supported verbs, properties of the MOs, the details of properties including the type (Naming, Read/Write and so on), and the version of Cisco IMC that the property was introduced in.

## 4 Installation

Supports Cisco IMC Version 1.5 or later for C-Series and Cisco IMC version 2.1(1) or later for E-Series.

### Before You Begin

- Ensure that you have PowerShell v2.0 or above installed in your system.
- Close any instances of PowerShell running with the Cisco IMC PowerTool module loaded.

### Installation

**Step 1** Download and launch the installer.

**Step 2** (Optional) Choose **Create Shortcut** to add a shortcut on the desktop.

### Getting Started

**Step 1** From the desktop shortcut, launch **IMC PowerTool**.

**Step 2** View all cmdlets, functions, and aliases supported by Cisco IMC PowerTool.

```
Get-Command -Module CiscoImcPs
Get-Command -Module CiscoImcPs | group CommandType
Get-Command -Module CiscoImcPs | measure
```

**Step 3** Connect to a Cisco UCS C-Series and E-Series Server.

```
$handle = Connect-Imc <ip or hostname> -NotDefault
```



**Note** After logging on, by default, the Cisco UCS handle is added to the default Cisco UCS C-Series and E-Series Server list, unless the `-NotDefault` option is specified. Every cmdlet that operates on a Cisco UCS C-Series and E-Series Server takes the `-Imc` parameter, where the handle can be specified.

**Step 4** Connect to a Cisco UCS C-Series and E-Series Server using a proxy.

```
$proxy = New-Object System.Net.WebProxy
$proxy.Address = "http:\\<url>:<port>"
$proxy.UseDefaultCredentials = $false
$proxy.Credentials = New-Object System.Net.NetworkCredential("<user name>", "<password>")
$handle = Connect-Imc <ip or hostname> -Proxy $proxy
```

**Step 5** Use the following cmdlets:

- a. Get the consolidated status information from the Cisco C-Series and E-Series Server.

```
Get-ImcStatus -Imc $handle
```

- b. Get the inventory summary of the C-Series Unit.

```
Get-ImcRackUnit -Imc $handle
```

- c. Disconnect.

```
Disconnect-Imc -Imc $handle
```

## Default Cisco UCS

If no handle or name is specified, the Cisco UCS C-Series and E-Series Server handle is added to a *DefaultImc* server list unless the `-Imc` parameter is specified. The first cmdlet in the pipeline operates on the default UCS list.

### Connect to UCS C-Series rack server

```
Connect-Imc <ip or hostname>
```

### Get the default rack server.

```
Get-ImcPSSession
```

### Get the status information and CIMC version of the rack server.

```
Get-ImcStatus
```

### Get information about the physical unit of the rack server.

```
Get-ImcRackUnit
```

### Enable HTTP on the rack server.

```
Get-ImcHttp | Set-ImcHttp -AdminState enabled
```

### Disable HTTP on the rack server.

```
Get-ImcHttp | Set-ImcHttp -AdminState disabled
```

### Disconnect the rack server.

```
Disconnect-Imc
```

## Default UCS List with Multiple UCS

Cisco IMC PowerTool cmdlets can work with multiple Cisco UCS C-Series Rack Servers if you specify multiple handles.

### Connect to a Cisco UCS C-Series and E-Series Server.

```
$handle1 = Connect-Imc <ip1> -NotDefault  
$handle2 = Connect-Imc <ip2> -NotDefault  
Get-ImcStatus -Imc $handle1,$handle2  
Disconnect-Imc -Imc $handle1,$handle2
```

**By default, multiple Cisco UCS C-Series handles are not allowed in DefaultImc. You can override this restriction by using the `Set-UcsPowerToolConfiguration` cmdlet.**

```
Get-ImcPowerToolConfiguration  
Set-ImcPowerToolConfiguration -SupportMultipleDefaultImc $true  
Connect-Imc <ip1>  
Connect-Imc <ip2>  
Get-ImcStatus  
Disconnect-Imc
```

### Connect to multiple Cisco C-Series and E-Series Server by using the same login credentials.

```
$user = "<username>"
```

```
$password = "<password>" |
  ConvertTo-SecureString -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential($user, $password)
$servers = @("<Imc1>", "<Imc2>", "<Imc3>")
Connect-Imc $servers -Credential $cred
```

## Credentials to/from a File

```
Connect-Imc <ip1>
Connect-Imc <ip2>
```

**Credentials can be stored to a file. The stored credentials are encrypted with a specified key.**

```
Export-ImcPSSession -LiteralPath C:\work\labs.xml
Disconnect-Imc
```

**A login can be initiated from credentials stored in a file.**

```
Connect-Imc -LiteralPath C:\work\labs.xml
```

**Specify proxy while logging in with credentials stored in a file.**

```
$proxy = New-Object System.Net.WebProxy
$proxy.Address = "http:\\<url>:<port>"
$proxy.UseDefaultCredentials = $false
$proxy.Credentials = New-Object System.Net.NetworkCredential("<user name>", "<password>")
Connect-Imc -LiteralPath C:\work\lab.xml -Proxy $proxy
```

**Log in to an additional system and add the credentials to the file.**

```
Connect-Imc <ip3>
Export-ImcPSSession -Path C:\work\lab.xml -Merge
```

## SSL Handling

When a user connects to a Cisco UCS C-Series and E-Series Server and the server cannot recognize any valid certificates, the connection establishment depends on `InvalidCertificateAction`. `InvalidCertificateAction` is set to `Ignore` by default. By default Cisco IMC PowerTool is configured to establish the connection without taking into account if the certificate is invalid.

You can override this setting by using the `Set-ImcPowerToolConfiguration` cmdlet.

```
Get-ImcPowerToolConfiguration
Set-ImcPowerToolConfiguration -InvalidCertificateAction Fail
```

Following table describes the options to check the validity of the certificate.

	Description
Fail	The cmdlet does not establish connection if the certificate is not valid.
Ignore	The cmdlet establishes a connection without taking into account that the certificate is invalid.
Default	(Windows default) The cmdlet establishes a connection if the certificate is valid.

## Aliases

Some aliases have been defined for convenience. To view the list of all aliases, run the following cmdlet:

```
gal | ? {$_.Name -like "*-Imc*" } | select Name
```

The following table lists the aliases and the corresponding cmdlets:

Aliases	Cmdlets
Add-ImcMo	Add-ImcManagedObject
Disable-ImcLocatorLed	Set-ImcLocatorLed -AdminState off
Enable-ImcLocatorLed	Set-ImcLocatorLed -AdminState on
Get-ImcMo	Get-ImcManagedObject
Remove-ImcMo	Remove-ImcManagedObject
Reset-ImcServer	Set-ImcRackUnit -AdminPower hard-reset-immediate
Restart-ImcServer	Set-ImcRackUnit -AdminPower cycle-immediate
Set-ImcMo	Set-ImcManagedObject
Start-ImcServer	Set-ImcRackUnit -AdminPower up
Stop-ImcServer	Set-ImcRackUnit -AdminPower soft-shut-down

## 5 Examples

The following examples show how to execute the cmdlets:

- [Add User](#)
- [Enable IP Blocking](#)
- [Configure NTP Settings](#)
- [Modify Syslog Settings](#)
- [Configure SoL](#)
- [Set Boot Order](#)
- [vMedia Configuration](#)
- [Get Adapter and Controller Information](#)
- [Transaction Support](#)
- [Managed Object Synchronization](#)
- [Filters](#)
- [HUU Firmware Update](#)
- [Cisco IMC Firmware Update](#)
- [Activate Cisco IMC Firmware](#)
- [Create a Virtual Drive](#)
- [Set a Boot Drive](#)
- [Clear a Boot Drive](#)
- [PowerTool Cmdlet Generation](#)
- [Setting BIOS Password](#)
- [Server Actions](#)

### Add User

```
Get-ImcLocalUser -Id 9 | Set-ImcLocalUser -Name "admin" -pwd "Password" -AccountStatus "active" -Priv "admin"
```



**Note** *Clear-ImcLocalUser* only changes the status to inactive and does not remove the user or data.

## Enable IP Blocking

```
Get-ImcIpBlocking | Set-ImcIpBlocking -Enable "yes"
```

## Configure NTP Settings

```
Get-ImcNtpServer | Set-ImcNtpServer -NtpEnable "yes" -NtpServer1 1.1.1.1 -Force
```

## Modify Syslog Settings

```
Get-ImcSyslog | Set-ImcSyslog -LocalSeverity warning -RemoteSeverity debug -Force
```

## Configure SoL

```
Get-ImcSolif -Dn "sys/rack-unit-1/sol-if" | Set-ImcSolIf -AdminState "enable" -Speed "57600" -Force
```

## Set Boot Order

```
Get-ImcLsbootStorage | Set-ImcLsbootStorage -Order 2 -Force
```

## vMedia Configuration

```
Get-ImcCommVMedia | Set-ImcCommVMedia -AdminState "enabled" -EncryptionState "enabled" -Force
```

## Get Adapter and Controller Information

### # PCI Adapter Properties

```
Get-ImcPciEquipSlot -Id "1"
```

### # Network Adapter Information

```
Get-ImcNetworkAdapterEthIf -Dn "sys/rack-unit-1/network-adapter-L/eth-1"
```

### # Storage Controller Information

```
Get-ImcStorageController -Dn "sys/rack-unit-1/board/storage-SAS-SLOT-4"
```

## Transaction Support

### # Start a transaction

```
Start-ImcTransaction
```

### # Perform an operation.

```
$adaptorHostEthIf = Get-ImcAdaptorUnit | Add-ImcAdaptorHostEthIf -Name AdaptorHostEth
$adaptorHostEthIfModify = $adaptorHostEthIf | Set-ImcAdaptorHostEthIf -PxeBoot enabled
$adaptorEthISCSIProfile = $adaptorHostEthIfModify | Add-ImcAdaptorEthISCSIProfile -InitiatorName AdaptorHostEth
-InitiatorIPAddress xx.xx.xx.xx -InitiatorSubnetMask 255.255.255.0 -DhcpISCSI enabled
$adaptorEthISCSIProfile | Remove-ImcAdaptorEthISCSIProfile
$adaptorHostEthIfModify | Remove-ImcAdaptorHostEthIf
```

### # End a transaction.

```
Complete-ImcTransaction
```

**# Undo a transaction.**

```
Undo-ImcTransaction
```

## Managed Object Synchronization

**# Enable SupportMultipleDefaultImc to connect to multiple CIMC.**

```
Set-ImcPowerToolConfiguration -SupportMultipleDefaultImc $true
```

**# Get credential and store it in a variable.**

```
$secpasswd = ConvertTo-SecureString password -AsPlainText -Force  
$mycreds = New-Object System.Management.Automation.PSCredential ("admin",$secpasswd)
```

**# Connect to different CIMC.**

```
$cimc1 = Connect-Imc xx.xx.xx.xx -Credential $mycreds  
$cimc2 = Connect-Imc xx.xx.xx.xx -Credential $mycreds
```

**# Get local user from different CIMC.**

```
$user1 = Get-ImcLocalUser -Imc $cimc1 -Id 1  
$user2 = Get-ImcLocalUser -Imc $cimc2 -Id 1
```

**# Synchronize a set of MOs from CIMC2 to CIMC1.**

```
Compare-ImcManagedObject $user1 $user2  
Sync-ImcManagedObject (Compare-ImcManagedObject $user1 $user2) -Imc $cimc1
```

## Filters

**# Get SysdebugMEpLog managed object where Type equals to "SEL" or Type equals to "Syslog".**

```
Get-ImcRackUnit | Get-ImcMgmtController | Get-ImcSysdebugMEpLog -Filter '(type -ilike SEL) -or (Type -clike Syslog)'
```

**# Get SysdebugMEpLog managed object where (Type equals to "SEL" or Type equals to "#Syslog") and Id equals to "0" and Type equals to "SEL".**

```
Get-ImcRackUnit | Get-ImcMgmtController | Get-ImcSysdebugMEpLog -Filter '(type -ilike SEL) -or (Type -clike Syslog)' -Id 0 -Type SEL
```

**# Get local user where name is like "admin" (case sensitive).**

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'Name -clike admin'
```

**# Get User where name is like "test\*" (support \* regular expression/case sensitive).**

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'Name -clike test*'
```

**# Get local user where AccountStatus is not equals to inactive.**

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'AccountStatus -cne inactive'
```

**# Get local user where AccountStatus matches 'inacti'.**

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'AccountStatus -cmatch inacti'
```

**# Get local user where AccountStatus matches with 'active' (starts with active/case sensitive).**

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'AccountStatus -cmatch ^active'
```

**# Get local user where AccountStatus does not matches with 'active' (starts with active/case sensitive).**

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'AccountStatus -cnotmatch ^active'
```

**# Get local user where Accountstatus is not like 'active' (starts with active/case sensitive).**

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'AccountStatus -cnotlike active'
```

## HUU Firmware Update

### Create a user credential

```
$user = "<username>"  
$password = "<password>"  
$cred = New-Object System.Management.Automation.PSCredential($user,$password)
```

### Update HUU firmware

```
Set-ImcHuuFirmwareUpdater -AdminState trigger -MapType nfs -RemoteIp 10.105.219.83 -RemoteCredential  
$cred-RemoteShare "/huuIso/ucs-c2x-huu-2.0.3d-1.iso" -StopOnError yes -TimeOut 60 -UpdateComponent  
All-VerifyUpdate no -force -Xml
```

## Cisco IMC Firmware Update

### Create a user credential

```
$user = "<username>"  
$password = "<password>"  
$cred = New-Object System.Management.Automation.PSCredential($user,$password)
```

### Cisco IMC Firmware Update

```
Get-ImcFirmwareUpdatable -Type blade-controller | Set-ImcFirmwareUpdatable -AdminState trigger -Type  
blade-controller -Protocol ftp -RemoteServer "10.65.183.111" -RemotePath  
"/UcseBin/UCSE_CIMC_2.3.1.bin"-RemoteCredential $cred-Force
```

## Activate Cisco IMC Firmware

```
Get-ImcFirmwareBootDefinition -Type "blade-controller" | Get-ImcFirmwareBootUnit |  
Set-ImcFirmwareBootUnit-AdminState trigger -Image backup -ResetOnActivate yes -Force
```

## Create a Virtual Drive

### Create virtual using unused physical drive

```
Get-ImcStorageVirtualDriveCreatorUsingUnusedPhysicalDrive |  
Set-ImcStorageVirtualDriveCreatorUsingUnusedPhysicalDrive -AdminState trigger -size "400 MB" -DriveGroup "[2]"  
-RaidLevel 0 -VirtualDriveName "vd_111" -Force
```

### Create a virtual drive using a virtual drive group

```
Get-ImcStorageVirtualDriveCreatorUsingVirtualDriveGroup |  
Set-ImcStorageVirtualDriveCreatorUsingVirtualDriveGroup -AdminState trigger -VirtualDriveName  
"vd_New"-SharedVirtualDriveId "3" -Size "100 MB" -Force
```

## Set a Boot Drive

### Set a physical drive as a boot Drive

```
Get-ImcStorageLocalDisk -Id 2 | Set-ImcStorageLocalDisk -AdminAction "set-boot-drive" -Force
```

### Set a virtual drive as a boot Drive

```
Get-ImcStorageVirtualDrive -Id 2 | Set-ImcStorageVirtualDrive -AdminAction "set-boot-drive" -Force
```

## Clear a Boot Drive

```
Get-ImcStorageController | Set-ImcStorageController -AdminAction "clear-boot-drive" -Force
```

## PowerTool Cmdlet Generation

### ConvertTo-ImcCmdlet:

Cisco IMC GUI does not support XML logging. To generate Cmdlets ConvertTo-IMC Cmdlet rely on the output of the Get cmdlet and generate Cmdlets to replicate the same object hierarchy.

### Generate cmdlets for the specified MOs.

```
Get-ImcBiosSettings -Hierarchy | ConvertTo-ImcCmdlet
```

### Save the cmdlet output in file.

```
Get-ImcBiosSettings -Hierarchy | ConvertTo-ImcCmdlet -OutputPath "C:/OutputFile.txt"
```

## Setting BIOS Password

```
Get-ImcBiosPassword | Set-ImcBiosPassword -Password "<password>" -Force
```



---

**Note** Setting BIOS password feature is applicable for E-Series servers only.

---

## Server Actions

Added the following new aliases to perform server actions:

Action Description	Cmdlet in PowerTool Release 1.3.1 or earlier	Cmdlet in PowerTool 1.4.1 and above
Power On Server	Get-ImcRackUnit   Set-ImcRackUnit -AdminPower up	Get-ImcRackUnit   Start-ImcServer
Power Off Server	Get-ImcRackUnit   Set-ImcRackUnit -AdminPower soft-shut-down	Get-ImcRackUnit   Stop-ImcServer
Power Cycle Server	Get-ImcRackUnit   Set-ImcRackUnit -AdminPower cycle-immediate	Get-ImcRackUnit   Restart-ImcServer
Hard Reset Server	Get-ImcRackUnit   Set-ImcRackUnit -AdminPower hard-reset-immediate	Get-ImcRackUnit   Reset-ImcServer
Turn On Locator LED	Get-ImcLocatorLed   Set-ImcLocatorLed -AdminState on	Get-ImcLocatorLed   Enable-ImcLocatorLed
Turn Off Locator LED	Get-ImcLocatorLed   Set-ImcLocatorLed -AdminState off	Get-ImcLocatorLed   Disable-ImcLocatorLed

## 6 Samples

Sample scripts developed using Cisco IMC PowerTool will be available in communities.cisco.com shortly.

## 7 Cisco UCS Communities

Cisco UCS Communities is a platform to discuss, share and learn about the Cisco Products and Technologies. For blogs, discussion forums and documents related to UCS integrations with partner ecosystem, visit <https://communities.cisco.com/ucsintegrations>.

## 8 Related Cisco UCS Documentation and Documentation Feedback

For more information, you can access related documents from the following links:

- [Cisco UCS C-Series Documentation Roadmap](#)
- [Cisco IMC XML API Programmer's Guide for Cisco UCS C-Series Servers](#)
- [Cisco UCS E-Series Documentation Roadmap](#)
- [Cisco IMC XML API Programmer's Guide for Cisco UCS E-Series Servers](#)

To provide technical feedback on this document, or to report an error or omission, please send your comments to [ucs-docfeedback@external.cisco.com](mailto:ucs-docfeedback@external.cisco.com). We appreciate your feedback.

## 9 Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: <http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

---

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2015 Cisco Systems, Inc. All rights reserved.

