



## Introduction

---

This chapter contains the following sections:

- [Cisco UCS Manager PowerTool](#) , on page 1
- [Management Information Model](#), on page 1
- [System Requirements](#), on page 8

## Cisco UCS Manager PowerTool

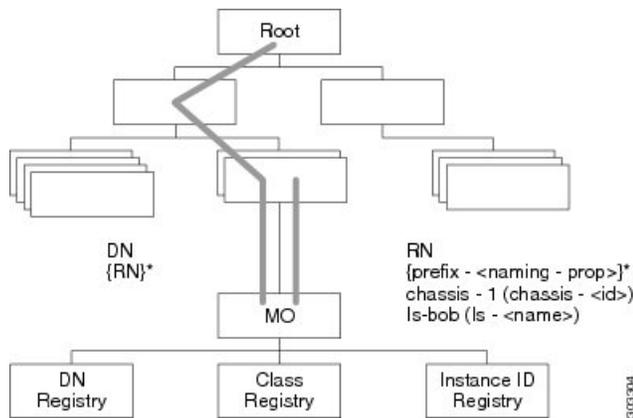
Cisco UCS Manager PowerTool is a PowerShell module which helps automate all aspects of Cisco UCS Manager including server, network, storage, and hypervisor management. PowerTool enables easy integration with the existing IT management processes and tools.

The PowerTool cmdlets work on the UCS Manager's Management Information Tree (MIT), performing create, modify, or delete actions on the Managed Objects (MO) in the tree. The next chapter provides an overview of the Cisco UCS Management Information Model (MIM) and relation of PowerTool cmdlets with it.

The easy way to learn UCS PowerTool configuration is by generating PowerTool cmdlets for performing configuration actions with the GUI using the ConvertTo-UcsCmdlet. For more information, see

## Management Information Model

All the physical and logical components that comprise a Cisco UCS domain are represented in a hierarchical Management Information Model (MIM), referred to as the Management Information Tree (MIT). Each node in the tree represents a Managed Object (MO), uniquely identified by its Distinguished Name (DN). Figure 1 illustrates the MIM.



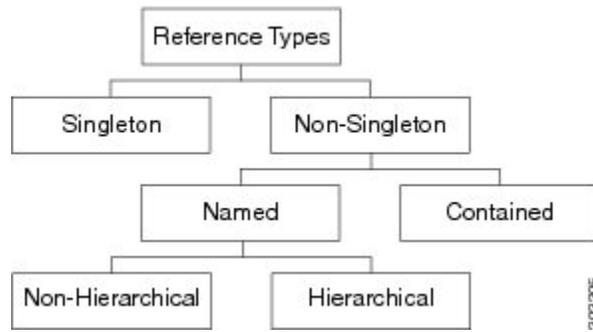
The following illustration shows a sample (partial) MIT for three chassis.

The following illustration shows a sample (partial) MIT for three chassis.

Tree (topRoot)	Distinguished Name
- sys	sys
- chassis-1	sys/chassis-1
- chassis-2	sys/chassis-2
- chassis-3	sys/chassis-3
- blade-1	sys/chassis-3/blade-1
- adaptor-1	sys/chassis-3/blade-1/adaptor-1
- blade-2	sys/chassis-3/blade-2
- adaptor-1	sys/chassis-3/blade-2/adaptor-1
- adaptor-2	sys/chassis-3/blade-2/adaptor-2

## Managed Objects

Managed Objects (MO) are abstractions of Cisco UCS domain resources, such as fabric interconnects, chassis, blades, and rack-mounted servers. Managed Objects represent any physical or logical entity that is configured / managed in the Cisco UCS MIT. For example, physical entities such as Servers, Chassis, I/O cards, Processors and logical entities such as resource pools, user roles, service profiles, and policies are represented as managed objects.



Every managed object 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 starting from the root to the MO itself. Essentially,  $Dn = [Rn]/[Rn]/[Rn]/\dots/[Rn]$ .

In the example below, Dn provides a fully qualified name for adaptor-1 in the model.

```
< dn = "sys/chassis-5/blade-2/adaptor-1" />
```

The above written Dn is composed of the following Rn:

```
topSystem MO: rn="sys" equipmentChassis MO: rn="chassis-<id>" computeBlade MO: rn="blade-<slotId>" adaptorUnit MO: rn="adaptor-<id>"
```

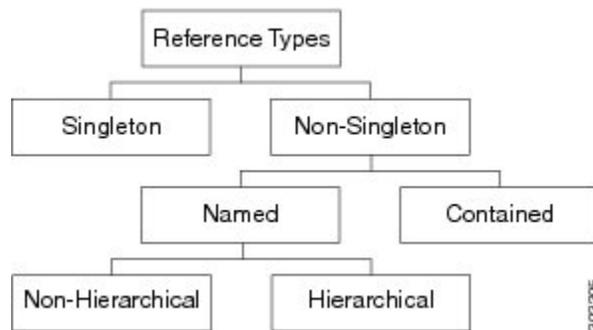
A Relative Name (Rn) may have the value of one or more of the MO's properties embedded in it. This allows in differentiating multiple MOs of the same type within the context of the parent. Any properties that form part of the Rn as described earlier are referred to as Naming properties.

For instance, multiple blade MOs reside under a chassis MO. The blade MO contains the blade identifier as part of its Rn (blade-[Id]), thereby uniquely identifying each blade MO in the context of a chassis.

## References to Managed Objects

The contents of the managed objects are referred during the operation of Cisco UCS. Some of the MOs are referred implicitly (PreLoginBanner during login) or as part of deployment of another MO. The Service Profile MO may refer to a template or a VNIC refers to a number of VLAN MOs.

The references can be classified as the following:



A singleton MO type is found at most once in the entire MIT and is typically referred to implicitly.

Non-Singleton MO type may be instantiated one or more times in the MIT. Often, when an MO refers to another, the reference is made by name. Depending on the type of the referenced MO, the resolution may be hierarchical. For instance, a service profile template is defined in an org. An org may contain suborgs, a sub org may have a service profile template defined with the same name. Now, when a service profile instance refers to a service profile template (by name), the name is looked up hierarchically from the org of the service profile instance up until the root org. The first match is used. If no match is found, the name “default” is looked up in the similar way and the first such match is used.

Reference Type	Example
Singleton	ChassisDiscoveryPolicy PreLoginBanner
Non-Singleton / Named / Non-Hierarchical	CallHomePolicy
Non-Singleton / Named / Hierarchical	BiosPolicy BootPolicy
Non-Singleton / Contained	BootDefinition under LsServer (ServiceProfile) VnicEtherIf under VnicEther

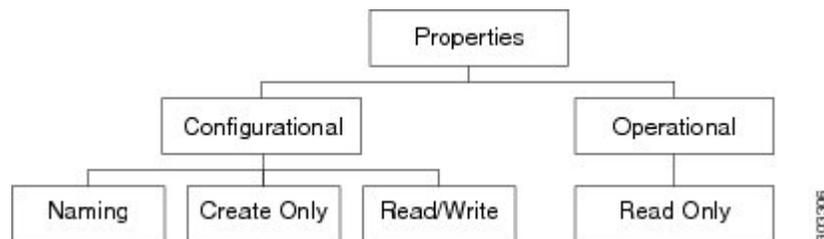
## Properties of Managed Objects

Properties of Managed Objects may be classified as Configuration or Operational.

Configuration properties may be classified as:

- Naming properties: Form part of the Rn. Specify while creating MO, this cannot be modified later.
- Create-Only properties: May be specified only during MO creation and cannot be modified later. If the property is not specified, a default value is assumed.
- Read / Write properties: May be specified during MO creation and can also be modified after.

Operational properties indicate the status of the MO or the system and they are read-only.



The following table lists the examples of the various property types:

Property Type	Example
Naming	Name in LsServer (Service Profile MO)
Create-Only	Type in LsServer
Read/Write	Description in LsServer

Property Type	Example
Read-Only	OperState in LsServer

## Methods

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

- Authentication
- AaaLogin
- AaaRefresh
- AaaLogout
- Configuration
- ConfigConfMo(s)
- LsClone
- Lsinstantiate\*
- FaultAckFaults
- Query
- ConfigResolveDn(s)
- ConfigResolveClass(es)
- ConfigResolveChildren
- Event Monitor
- EventSubscribe

The class query methods ConfigResolveClass(es) and ConfigResolveChildren filters the MOs to match a specific set of MOs and return by the method.

The supported filters are:

- Property Filters:

Supported Filters	Definition
allbits	Match if all specified values present in a multivalued property
anybit	Match if any of the specified values present in a multivalued property
bw	Match if the property's value lies between the two values specified

Supported Filters	Definition
eq	Match if property's value is the same as the specified value
ge	Match if property's value is greater than or equal to the specified value
gt	Match if property's value is greater than the specified value
le	Match if property's value is lesser than or equal to the specified value
lt	Match if property's value is lesser than the specified value
ne	Match if property's value is not equal to the specified value
wcard	Match if property's value matches the pattern specified

- Composite Filters (Acts on subfilters)

Composite Filter	Definition
not	Negates result of subfilter
and	True, if all the subfilters return true
or	True, if any of the subfilters return true

## PowerTool Mapping

Some PowerTool cmdlets are generated from the MO specification. A convenient noun is used as type, for example, ServiceProfile is used instead of LsServer. 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(s) method with MO status as **Created** with the property values as specified. If the ModifyPresent parameter is specified, status changes to **Modified**. If **Force** parameter is specified, no confirmation is prompted.

**Get Cmdlet** — Use the ConfigResolveClass method to retrieve MOs. If any property parameters are specified, they are used to generate *eq* filters. If multiple property parameters are specified, the multiple *eq* filters are combined with *and* filter.

**Set Cmdlet** — Uses the ConfigConfMo(s) method with MO status as **Modified** with the specified property values. If **Force** parameter is specified, no confirmation is prompted.

**Remove Cmdlet** — Uses the ConfigConfMo(s) method with MO status as **Deleted**. If **Force** parameter is specified, no confirmation is prompted.

The following 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

The following table lists the types of pipeline input 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	Same Type

The following table lists the methods invoked to generate the required XML requests:

Cmdlet	Method
Add-Ucs/ Set-Ucs/	ConfigConfMos
Get-Ucs/	ConfigResolveClass with filters
Get-UcsManagedObject -ClassId	ConfigResolveClass
Get-UcsManagedObject –ClassId -Dnlist	ConfigFindDnsByClassId
Get-UcsManagedObject –Dn	ConfigResolveDns
Connect-Ucs	AaaLogin
Disconnect-Ucs	AaaLogout
Background <a href="#">1</a>	AaaRefresh
Copy-UcsServiceProfile	LsClone
Add-UcsServiceProfileFromTemplate	LsInstantiateTemplate, LsInstantiateNTemplate, LsInstantiateNNamedTemplate
Get-UcsChild	ConfigResolveChildren

Cmdlet	Method
Acknowledge-UcsFault	FaultAckFaults
Start-UcsKvmSession	AaaGetNComputeAuthTokenByDn
Watch-Ucs	EventSubscribe (First Watcher)
Clear-UcsStatistics	StatsClearInterval
Get-UcsTransactionImpact	ConfigEstimateImpact

<sup>1</sup> This is not a cmdlet. It is a background service.

## System Requirements

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

- Install Windows PowerShell 3.0 or higher
- .NET Framework Version 4.5 or higher
- Windows PowerShell 4.0 or higher for UCS DSC resource

### Cisco UCS Manager

Cisco UCS Manager PowerTool is compatible with the following Cisco UCS Manager releases:

- Release 4.0
- Release 3.2
- Release 3.1
- Release 3.0
- Release 2.5
- Release 2.2
- Release 2.1
- Release 2.0