



Introduction

This chapter contains the following sections:

- [Overview, page 1](#)
- [Management Information Model, page 2](#)
- [PowerTool Mapping, page 5](#)
- [System Requirements, page 7](#)

Overview

Cisco UCS Central PowerTool is a PowerShell module which helps automate all aspects of Cisco UCS Central management. PowerTool enables easy integration with existing IT management processes and tools.

Cisco UCS Central delivers a common management solution across all Cisco UCS Domains. Cisco UCS Central provides a centralized resource inventory and a repository of policies. Cisco UCS Central simplifies configuration, maintains policy uniformity, resolves contention on global identities, and effectively and consistently manages Cisco UCS domains.

Cisco UCS Central provides a global view of the entire data center through multiple Cisco UCS Manager sessions. Cisco UCS Central can manage Cisco UCS operations for an individual data center or for multiple data centers. Cisco UCS Central facilitates operational management for firmware management, catalog management, configuration backup and restores operations, monitor log, core files, and faults.

Cisco UCS Central is designed for aggregated management functions beyond what Cisco UCS Manager supports today.

Most PowerTool cmdlets work on the UCS Central Management Information Tree (MIT). To create, modify, or delete actions on the Managed Objects (MO) in the tree.

The easy way to learn UCS Central configuration through PowerTool is to generate PowerTool cmdlets, for configuration actions performed with the GUI, using the ConvertTo-UcsCentralCmdlet.

Management Information Model

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



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

Tree (TopRoot)	Distinguish Name
- compute	compute
-sys-1010	compute/sys-1010
- chassis-1	compute/sys-1010/chassis-1
- chassis-2	compute/sys-1010/chassis-2
- chassis-3	compute/sys-1010/chassis-3
- blade-1	compute/sys-1010/chassis-3/blade-1
- adaptor-1	compute/sys-1010/chassis-3/blade-1/adaptor-1
-blade-2	compute/sys-1010/chassis-3/blade-2
-adaptor-1	compute/sys-1010/chassis-3/blade-2/adaptor-1
-adaptor-2	compute/sys-1010/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 either configured or managed in the Cisco UCS Central 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 identified in the tree with its Distinguished Name (Dn). It 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]/.../[Rn].

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

```
< dn = "compute/sys-1010/chassis-5/blade-2/adaptor-1" />
```

This Dn is composed of the following Rn:

```
computeResourceAggrEp MO: rn="compute" computeSystem MO: rn="sys-" equipmentChassis MO:
rn="chassis-" computeBlade MO: rn="blade-" adaptorUnit MO: rn="adaptor-"
```

A Relative Name (Rn) may have the value of one or more of the MO's properties embedded in it. It enables 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 within chassis MO. The blade MO contains the blade identifier as part of its Rn (blade-[Id]). Therefore, uniquely identifying each blade MO in the context of a chassis.

References to Managed Objects

The contents of the managed objects are referred to during the operation of Cisco UCS Central. Some of the MOs are referred to implicitly or as part of deployment of another MO. For example, the service profile MO can refer to a template or a VNIC refer to several VLAN MOs.

The different types of 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 example, a service profile template is defined in an Org. An Org can have sub orgs, a sub org can 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 at the root Org level. 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
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 during MO creation only. It 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 / system and are hence read-only.

Properties of Managed Objects 

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
Read-Only	OperState in LsServer

Concept

Methods are Cisco UCS Central 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 ConfigResolveClasses, ConfigResolveChildren allow a filter to be specified. So that a specific set of MOs are compared and returned by the method.

The supported filters are:

Property Filters:

Supported Filter	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
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 sub filter
and	True, if all the sub filters return true
or	True, if any of the sub filters return true

PowerTool Mapping

All but about 36 of the PowerTool cmdlets are generated from the MO specification. 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 "created" along with the specified property values. If the ModifyPresent parameter is specified, status "created, modified" is specified instead.

Get Cmdlet - Uses 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 an "and" filter.

Set Cmdlet - Uses the ConfigConfMo(s) method with MO status "modified" along with the specified property values. If the Force parameter is specified, there is no prompt for confirmation.

Remove Cmdlet - Uses the ConfigConfMo(s) method with MO status deleted. If the Force parameter is specified, there is no prompt for confirmation.

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 methods invoked to generate the required XML requests:

Cmdlet	Method
Add-UcsCentral Set-UcsCentral	ConfigConfMos
Get-UcsCentral	ConfigResolveClass with filters
Get-UcsCentralManagedObject -ClassId	ConfigResolveClass
Get-UcsCentralManagedObject -ClassId -Dnlist	ConfigFindDnsByClassId
Get-UcsCentralManagedObject -Dn	ConfigResolveDns
Connect-UcsCentral	AaaLogin
Disconnect-UcsCentral	AaaLogout
Background ¹	AaaRefresh
Get-UcsCentralChild	ConfigResolveChildren
Get-UcsCentralServiceProfile	ConfigFindDependencies
Add-UcsCentralServiceProfileFromTemplate	<ul style="list-style-type: none"> • LsInstantiateNTemplate • LsInstantiateTemplate • LsInstantiateNNamedTemplate
Copy-UcsCentralServiceProfile	LsClone
Rename-UcsCentralServiceProfile	ConfigConfRename

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

Get-UcsCentralCmdletMeta cmdlet is used to explore the MO types, the corresponding nouns, supported verbs, properties of the MOs. Also used for the property details including the type: Naming, Read, or Write, and so on. and the version of UCS Central the property was introduced.

System Requirements

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

- Windows PowerShell 3.0 or higher must be installed on the system
- .NET Framework Version 4.5 or higher

Cisco UCS Central

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

- Release 2.0
- Release 1.5
- Release 1.4
- Release 1.3
- Release 1.2

