



FROM USER GUIDE



Cisco UCS Central PowerTool, Release 1.x

- 1 [Overview](#)
- 2 [Management Information Model](#)
- 4 [System Requirements](#)
- 5 [Installation](#)
- 6 [PowerTool Cmdlet Generation](#)
- 7 [Examples](#)
- 8 [Related Cisco UCS Central Documentation and Documentation Feedback](#)
- 9 [Obtaining Documentation and Submitting a Service Request](#)

1 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.

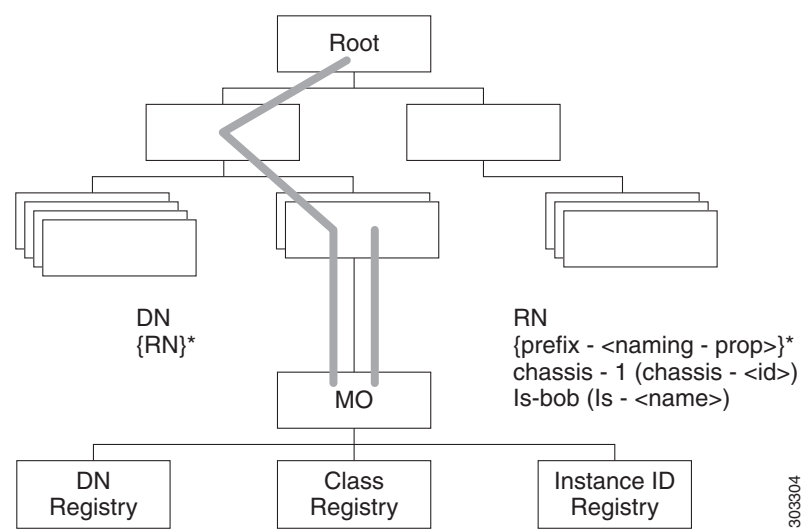
Bulk of the PowerTool cmdlets work on the UCS Central Management Information Tree (MIT), performing create, modify or delete actions on the Managed Objects (MO) in the tree.

One of the easiest ways to learn UCS Central configuration through PowerTool is to generate PowerTool cmdlets, for configuration actions performed with the GUI, using the ConvertTo-UcsCentralCmdlet.

2 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 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.

Figure 1 Management Information Model



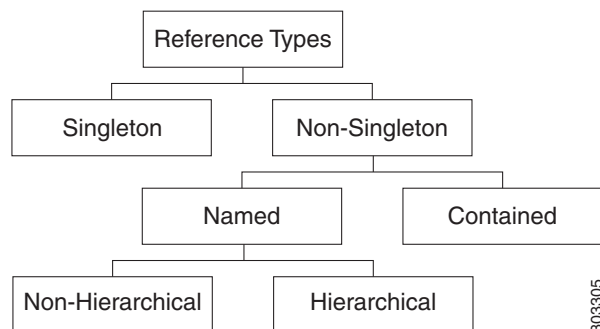
The following illustration shows a sample (partial) MIT for three chassis.

Tree (topRoot)	Distinguished 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 configured / 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.

Figure 2 *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]/.../[Rn].

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

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

The above written Dn is composed of the following Rn:

```
computeResourceAggrEp MO: rn="compute" computeSystem MO: rn="sys-<id>" 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.

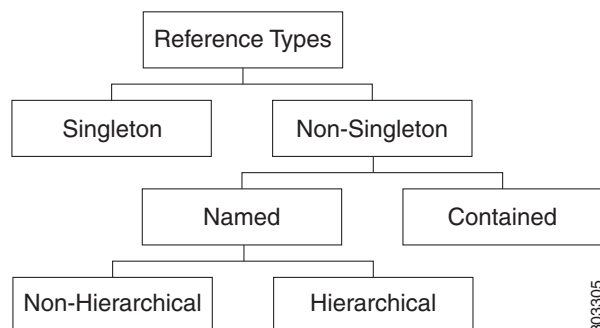
For instance, multiple blade MOs reside under a chassis MO. The blade MO contains the blade identifier as part of its Rn (blade-[Id]), thereby uniquely identifying each blade MO in the context of a chassis.

References to Managed Objects

The contents of the managed objects are referred to during the operation of Cisco UCS Central. Some of the MOs are referred to implicitly 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 different types of references can be classified as shown below:

Figure 3 *References to Managed Objects*



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. In many cases, 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 under an org. Since an org may contain sub-orgs, 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
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

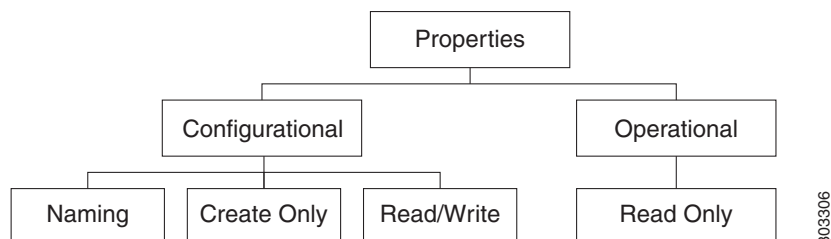
Properties of Managed Objects may be classified as Configuration or Operational.

Configuration properties may be classified as:

- Naming properties: Form part of the Rn. Needs to be specified only during MO creation and 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 subsequently.

Operational properties indicate the current status of the MO / system and are hence read-only.

Figure 4 *Properties of Managed Objects*



The table below 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

Methods

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 (ConfigResolveClass(es), ConfigResolveChildren) allow a filter to be specified so that a specific set of MOs are matched and returned by the method.

The supported filters are:

- Property Filters:

Supported Filter	Definition
allbits	Match if all specified values present in a multi-valued property
anybit	Match if any of the specified values present in a multi-valued 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 sub-filters)

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

3 PowerTool Mapping

All but about 25 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 will be no prompt for confirmation.

Remove Cmdlet - Uses the ConfigConfMo(s) method with MO status “deleted”. If the Force parameter is specified, there will be 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 ¹	ConfigConfMos
Set-UcsCentral ¹	
Get-UcsCentral ¹	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
Add-UcsCentralServiceProfileFromTemplate	LsInstantiateNTemplate

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, the details of properties including the type (Naming, Read/Write and so on), and the version of UCS Central the property was introduced.

4 System Requirements

Before installing Cisco UCS Central 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 later)

Cisco UCS Central

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

- Release 1.2
- Release 1.3

Windows PowerShell Requirements

- Cisco UCS Central PowerTool supports Windows PowerShell 2.0 and later

5 Installation

Before You Begin

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

Installation

Step 1 Download and launch the installer.

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

Getting Started

Step 1 Launch **Cisco UCS Central PowerTool** from the desktop shortcut.

Step 2 View all cmdlets, functions, and aliases supported by the Cisco UCS Central PowerTool.

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

Step 3 Connect to a Ucs Central system.

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



Note If login is successful `Connect-UcsCentral`, by default, adds the UCS Central handle to the Default UCS Central list, unless the `-NotDefault` option is specified. Every cmdlet that operates on Ucs Central takes the `-UcsCentral` parameter, where the handle can be specified.

Step 4 Use the following cmdlets:

- a. Get org objects.

```
Get-UcsCentralOrg -UcsCentral $handle
```

- b. Disconnect.

```
Disconnect-UcsCentral -UcsCentral $handle
```

Default UCS Central

If no handle or name is specified, the UCS Central handle is added to a Default UCS Central list. Unless the `-UcsCentral` parameter is specified, the first cmdlet in the pipeline operates on the default UCS Central list.

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

Get the Default UCS Central list.

```
Get-UcsCentralPSSession
```

Get the set of all chassis objects.

```
Get-UcsCentralChassis
```

Get the object pertaining to chassis 1.

```
Get-UcsCentralChassis -Id 1
```

Get the set of blades, pertaining to chassis 1.

```
Get-UcsCentralChassis -Id 1 | Get-UcsCentralBlade
```

Disconnect.

```
Disconnect-UcsCentral
```

Default UCS Central List with Multiple UCS Central Systems

PowerTool cmdlets can work with multiple UCS Central systems by specifying multiple handles.

Connect to a Cisco UCS Central system.

```
$handle1 = Connect-UcsCentral <ip1> -NotDefault
$handle2 = Connect-UcsCentral <ip2> -NotDefault
Disconnect-Ucs -UcsCentral $handle1,$handle2
```

By default, multiple Cisco UCS Central handles are not allowed in Default UCS Central. This can be overridden using the `Set-UcsCentralPowerToolConfiguration` cmdlet.

```
Get-UcsCentralPowerToolConfiguration
Set-UcsCentralPowerToolConfiguration -SupportMultipleDefaultUcsCentral $true
```

```
Connect-UcsCentral <ip1>
Connect-UcsCentral <ip2>
Get-UcsCentralPSSession
Disconnect-UcsCentral
```

Connect to multiple Cisco UCS Central systems using the same login credentials.

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



```
$password = "<password>" | ConvertTo-SecureString -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential($user, $password)
$servers = @("<ucsc1>", "<ucsc2>", "<ucsc3>")
Connect-UcsCentral $servers -Credential $cred
```

Credentials to/from File

```
Connect-UcsCentral <ip1>
Connect-UcsCentral <ip2>
```

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

```
Export-UcsCentralPSSession -LiteralPath C:\work\labs.xml
Disconnect-UcsCentral
```

Login can be initiated from credentials stored in a file.

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

Login to an additional system and add the credentials to the file.

```
Connect-UcsCentral <ip3>
Export-UcsCentralPSSession -Path C:\work\lab?.xml -Merge
```

SSL Handling

When a user connects to a Cisco UCS Central server and the server cannot recognize any valid certificates, the connection depends on `InvalidCertificateAction`. `InvalidCertificateAction` is set to `Ignore` by default. By default PowerTool is configured to establish the connection without considering the validity of the certificate.

You can override this using the `Set-UcsCentralPowerToolConfiguration` cmdlet.

```
Get-UcsCentralPowerToolConfiguration
Set-UcsCentralPowerToolConfiguration -InvalidCertificateAction Fail
```

Status	Description
Fail	The cmdlet does not establish a connection if the certificate is not valid.
Ignore	The cmdlet establishes the connection without considering the validity of the certificate.
Default	(Windows default) The cmdlet establishes a connection if the certificate is valid.

Aliases

Aliases are the simplified cmdlets which can be used to perform Cisco UCS Central operations.

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

Aliases	UCS Central Operations
Compare-UcsCentralMo	Compare UCS Central Managed Object
Sync-UcsCentralMo	Sync UCS Central Managed Object
Get-UcsCentralMo	Get UCS Central Managed Object
Remove-UcsCentralMo	Remove UCS Central Managed Object
Add-UcsCentralMo	Add UCS Central Managed Object
Set-UcsCentralMo	Set UCS Central Managed Object
Acknowledge-UcsCentralFault	Acknowledge UCS Central Fault

Aliases	UCS Central Operations
Get-UcsCentralUcsDomain	Get UCS Central UCS Domains
Associate-UcsCentralServiceProfile	Associate UCS Central ServiceProfile
Disassociate-UcsCentralServiceProfile	Disassociate UCS Central ServiceProfile
Remove-UcsCentralFex	Remove Fex
Decommission-UcsCentralFex	Decommission Fex
Recommission-UcsCentralFex	Recommission Fex
Acknowledge-UcsCentralFex	Acknowledge Fex
Decommission-UcsCentralServerUnit	Decommission UCS Central ServerUnit
Recommission-UcsCentralServerUnit	Recommission UCS Central ServerUnit
Acknowledge-UcsCentralServerUnit	Acknowledge UCS Central ServerUnit
Remove-UcsCentralCartridge	Remove UCS Central Cartridge Server
Decommission-UcsCentralChassis	Decommission UCS Central Chassis
Recommission-UcsCentralChassis	Recommission UCS Central Chassis
Acknowledge-UcsCentralChassis	Acknowledge UCS Central Chassis
Remove-UcsCentralRackUnit	Remove UCS Central Rack Unit Server
Decommission-UcsCentralRackUnit	Decommission UCS Central Rack Unit Server
Recommission-UcsCentralRackUnit	Recommission UCS Central Rack Unit Server
Acknowledge-UcsCentralRackUnit	Acknowledge UCS Central Rack Unit Server
Decommission-UcsCentralBlade	Decommission UCS Central Blade Server
Recommission-UcsCentralBlade	Recommission UCS Central Blade Server
Acknowledge-UcsCentralBlade	Acknowledge UCS Central Blade Server
Remove-UcsCentralBlade	Remove UCS Central Blade Server
Acknowledge-UcsCentralSlot	Acknowledge UCS Central Slot

6 PowerTool Cmdlet Generation

The ConvertTo-UcsCentralCmdlet enables you to learn about the powertool cmdlets that performs configuration action by generating the cmdlet in one of the following ways:

- Performing actions in GUI
- From a backup file
- From a UCS Central PowerShell object

Generate Cmdlets From UCS Central GUI Actions

Before You Begin

- Install Flash Debugger (not flash player) for your Internet browser. You can download the software from this URL: <https://www.adobe.com/support/flashplayer/downloads.html>.



Note

Flash debuggers are different for different browsers. For example, Internet Explorer has a different flash debugger than Mozilla Firefox.

- Configure mm.cfg file located under user directory. For instance, if there is a user name demoUser on the windows system, this file should be located in this path - "C:\Users\demoUser". This file provides instructions to the Flash Debugger about settings related to log dumping. Modify the following properties as:

```
ErrorReportingEnable=1
TraceOutputFileEnable=1
```

If there is no mm.cfg file user can create one and follow below mentioned link to edit/configure mm.cfg file:

<http://helpx.adobe.com/flash-player/kb/configure-debugger-version-flash-player.html>

Step 1 After the UCS Central GUI launches, at the login page, press **ctrl + alt + shift + L**. The **Logger Console** flash popup is displayed. You need to modify the logger preferences as follows:

- Select the **Trace** option for the property **Destination**.
- Click the **Set Log Preference** button.

Step 2 Enter the username and password for UCS Central and log on to UCS Central.

A log file is created in the log file location depending on the OS. For more information on the exact location of the log files, see <https://helpx.adobe.com/flash-player/kb/configure-debugger-version-flash-player.html>. The name of the log file is the name mentioned in the property TraceOutputFileName in mm.cfg. If this property does not exist, the name will be flashlog.txt.



Note If there is no **Logs** folder, create a **Logs** folder and repeat steps 1 through step 2.

Step 3 Launch UCS Central PowerTool and run the **ConvertTo-UcsCentralCmdlet**.

This cmdlet monitors the configuration logs in the log file.

Step 4 Perform operations in the GUI.

The cmdlets for actions performed in the GUI are generated in the powertool console.

If you are using Google Chrome, perform the following steps to disable the integrated flash player and to install the flash debugger:

Step 1 Install Flash debugger that is compatible with the Netscape browser.

Step 2 Select chrome://plugins and click the "[+] **Details**" icon, located at the top right corner of the screen.

Step 3 Search for the **Shockwave Flash** or **Flash** option.

Step 4 Disable the integrated version installed under the Chrome directory on your local machine. For example, C:\Program Files (x86)\Google\Chrome\Application\39.0.2171.71\PepperFlash\pepflashplayer.dll

Step 5 Enable the Netscape compatible Flash Debugger which you installed in the Systems directory on your local machine. For example, C:\Windows\SysWOW64\Macromed\Flash\NPSWF32_15_0_0_239.dll.

Step 6 Restart the Google Chrome browser.

Generate Cmdlets From Backup file

The following example is used to generate cmdlet from a backup file.

Generate cmdlets from backup file

```
ConvertTo-UcsCentralCmdlet -UcsCentralBackup -LiteralPath "C:\central-mgr.xml" -OutputPath "C:\central-mgr.txt"
```

Generating Cmdlets From UCS Central PowerShell Object

Cmdlets can be generated by piping the UCS Central objects from a Get cmdlet to the ConvertTo cmdlet. In the following example, ConvertTo cmdlet generates the cmdlets required to re-create the 'testSP' service profile object.

Generate cmdlets for the specified managed object

```
Get-UcsCentralServiceProfile -Name testSP | ConvertTo-UcsCentralCmdlet
```

7 Examples

This section show examples to execute the cmdlets. The following examples are included in this section:

- [Org](#)
- [Faults](#)
- [Get Cmdlet -Hierarchy Flag](#)
- [Get Cmdlet -LimitScope Flag](#)
- [Transaction Support](#)
- [UUID Suffix Pools](#)
- [MAC Pools](#)
- [WWNN Pools & Blocks](#)
- [WWPN Pools & Blocks](#)
- [Clone a Service Profile](#)
- [Add Service Profile Using Service Profile Template](#)
- [Rename Service Profile](#)
- [Associate Service Profile](#)
- [Disassociate Service Profile](#)
- [Configuration Backup](#)
- [Import Configuration](#)
- [Tech Support](#)
- [Filters](#)
- [Generic Managed Object Queries](#)
- [Generic Managed Object Cmdlets](#)
- [CCO Integration](#)
- [Export to XML](#)
- [Import from XML](#)
- [KVM](#)
- [Registered UCS Domains](#)
- [Cmdlet Meta Information](#)
- [Add Cmdlet -ModifyPresent Flag](#)
- [Compare UCS Central Managed Object](#)
- [Synchronize UCS Central Managed Object](#)
- [Launch UCS Central Domain Session](#)
- [Get UCS Central Server](#)

Org

Get a list of orgs across Cisco UCS Central systems, in the Default UCS Central list.

```
Get-UcsCentralOrg | select UcsCentral, Name, Dn
```

Get a handle to the root level Org.

```
Get-UcsCentralOrg -Level root
```

Add 5 orgs to UCS.

```
1..5 | % { Add-UcsCentralOrg -Ucs <handle or name> qwerty$_ }
```

Faults

Retrieve faults, group them by severity.

```
Get-UcsCentralFault | group Severity
```

Retrieve critical faults.

```
Get-UcsCentralFault -Severity critical | select UcsCentral, Dn, Cause
```

Get Cmdlet -Hierarchy Flag

Get Managed Object including all children.

```
Get-UcsCentralComputeRequirement -Name sp_name -Hierarchy
```

Get Cmdlet -LimitScope Flag

Get service profile at the root level without descending into org root children.

```
Get-UcsCentralComputeRequirement -Name sp_name -LimitScope
```

Transaction Support

Start a transaction.

```
Start-UcsCentralTransaction
```

Perform an operation.

```
...
```

End a transaction.

```
Complete-UcsCentralTransaction
```

Undo a transaction.

```
Undo-UcsCentralTransaction
```

UUID Suffix Pools

Create a UUID Suffix pool.

```
Add-UcsCentralUuidSuffixPool -Name uuid_pool -Prefix 2EEB8026-9084-11E1
```

Add a block of UUID Suffixes to the suffix pool.

```
Get-UcsCentralUuidSuffixPool -Name uuid_pool | Add-UcsCentralUuidSuffixBlock -From 0000-000000000001 -To 0000-00000000002C
```

Remove a UUID Suffix pool.

```
Get-UcsCentralUuidSuffixPool -Name uuid_pool | Remove-UcsCentralUuidSuffixPool
```

MAC Pools

Create a MAC pool.

```
Add-UcsCentralMacPool -Name mac_pool
```

Add a block of mac to the mac pool.

```
Get-UcsCentralMacPool -Name mac_pool | Add-UcsCentralMacMemberBlock -From 00:25:B5:00:00:00 -To 00:25:B5:00:00:09
```

Remove a mac pool.

```
Get-UcsCentralMacPool -Name mac_pool | Remove-UcsCentralMacPool
```

WWNN Pools & Blocks

Get all WWNN pool in UCS Central.

```
Get-UcsCentralWwnPool -Purpose node-wnn-assignment
```

Create a WWNN pool.

```
$wwnn_pool = Add-UcsCentralWwnPool -Name wwnn_pool -Purpose node-wnn-assignment
```

Add a WWN block to the WWNN pool.

```
$wwnn_pool | Add-UcsCentralWwnMemberBlock -From 20:00:00:24:B5:00:00:00 -To 20:00:00:24:B5:00:00:09
```

Remove a WWNN pool.

```
$wwnn_pool | Remove-UcsCentralWwnPool
```

WWPN Pools & Blocks

Get all WWPN pool in UCS Central.

```
Get-UcsCentralWwnPool -Purpose port-wnn-assignment
```

Create a WWPN pool.

```
$wwpn_pool = Add-UcsCentralWwnPool -Name wwpn_pool -Purpose port-wnn-assignment
```

Remove a WWPN pool.

```
$wwpn_pool | Remove-UcsCentralWwnPool
```

Clone a Service Profile

Copies a service profile.

```
Get-UcsCentralServiceProfile -Name "demoSP" | Copy-UcsCentralServiceProfile -DestinationOrg (Get-UcsCentralOrg -Level root) -NewName "clonedSP"
```

Add Service Profile Using Service Profile Template

Adds new service profiles using an existing service profile template.

```
Get-UcsCentralServiceProfile -Type initial-template -name "SPTemplate" |  
Add-UcsCentralServiceProfileFromTemplate -NamePrefix "SPFromTemplate" -Count 3
```

Rename Service Profile

Renames existing service profile.

```
Get-UcsCentralServiceProfile -name "spName" | Rename-UcsCentralServiceProfile -NewName "spNewName"
```

Associate Service Profile

Associates a service profile to a server pool/blade or rackunit servers.

```
Get-UcsCentralServiceProfile -Name "demoSPName" | Connect-UcsCentralServiceProfile -Blade (Get-UcsCentralBlade -SlotId 1)
```

Disassociate Service Profile

Renames existing service profile.

```
Get-UcsCentralServiceProfile -Name "demoSPName" | Disconnect-UcsCentralServiceProfile -Force
```

Configuration Backup

Remove any previously stored backups in UCS Central.

```
Get-UcsCentralMgmtDataExporter | Remove-UcsCentralMgmtDataExporter
```

The PathPattern can be auto-filled, allowing the cmdlet to be used with multiple Cisco UCS Central domains. Create and download full-state system backup of UCS Central. This creates a binary file that includes a snapshot of the entire system. You can use the file generated from this backup to restore the system during disaster recovery. This file can restore or rebuild the configuration on the new UCS Central VM. You cannot use this file for an import.

```
Backup-UcsCentral -Type full-state -PathPattern  
'C:\Backups\${UcsCentral}-${yyyy}${MM}${dd}-${HH}${mm}-full-state.tgz'
```

Create and download logical backup of UCS Central. This creates an XML file that includes all logical configuration settings such as service profiles, VLANs, VSANs, pools, and policies. You can use the file generated from this backup to import these configuration settings to the UCS Central. You cannot use this file for a system restore.

```
Backup-UcsCentral -Type config-logical -PathPattern  
'C:\Backups\${UcsCentral}-${yyyy}${MM}${dd}-${HH}${mm}-config-logical.tgz'
```

Create and download system backup of UCS Central. This creates an XML file that includes all system configuration settings such as usernames, roles, and locales. You can use the file generated from this backup to import these configuration settings to the UCS Central. You cannot use this file for a system restore.

```
Backup-UcsCentral -Type config-system -PathPattern  
'C:\Backups\${UcsCentral}-${yyyy}${MM}${dd}-${HH}${mm}-config-system.tgz'
```

Create and download config-all backup of UCS Central. This creates an XML file that includes all system and logical configuration settings. You can use the file generated from this backup to import these configuration settings to the UCS Central. You cannot use this file for a system restore. This file does not include passwords for locally authenticated users.

```
Backup-UcsCentral -Type config-all -PathPattern  
'C:\Backups\${UcsCentral}-${yyyy}${MM}${dd}-${HH}${mm}-config-all.tgz'
```

Import Configuration

The import function is available for all configuration, system configuration, and logical configuration files. You can perform an import while the system is up and running.

Import all configuration xml (An XML file that includes all system and logical configuration settings. The current configuration information is replaced with the information in the imported configuration file one object at a time.

```
Import-UcsCentralBackup -LiteralPath 'C:\Backups\config-all.tgz'
```

Import all configuration xml. The information in the imported configuration file is #compared with the existing configuration information. If there are conflicts, the import operation overwrites the information on the Cisco UCS Central domain with the information in the import configuration file.

```
Import-UcsCentralBackup -LiteralPath 'C:\Backups\config-all.tgz' -Merge
```

Tech Support

Technical support data for the entire UCS Central instance is created and downloaded to the specified file.

```
Get-UcsCentralTechSupport -PathPattern 'C:\${UcsCentral}-techsupp-ucsc.tar' -All -RemoveFromUcsCentral  
-TimeoutSec 600
```

Technical support data for the Operation Manager provider is created and downloaded to the specified file.

```
Get-UcsCentralTechSupport -PathPattern 'C:\${UcsCentral}-techsupp-opmgr.tar' -OperationMgr -RemoveFromUcsCentral  
-TimeoutSec 600
```

Filters

Get all Local Service Profiles with Name containing string 'SJC'.

```
Get-UcsCentralComputeRequirement -Filter 'Name -cmatch SJC' | select UcsCentral, Dn, Name
```

Get all Roles that have the fault privilege.

```
Get-UcsCentralRole -Filter 'Priv -ccontains fault' | select UcsCentral, Dn, Name
```

Get all Roles that have the fault or operations privilege.

```
Get-UcsCentralRole -Filter 'Priv -canybit fault,operations' | select UcsCentral, Dn, Name
```

Get all Roles that have the fault and operations privilege.

```
Get-UcsCentralRole -Filter 'Priv -callbits fault,operations' | select UcsCentral, Dn, Name
```

Get a list of faults generated between 4/18/2012 9:00 and 4/19/2012 9:30.

```
Get-UcsCentralFault -Filter 'Created -cbw "4/18/2012 9:00","4/19/2012 9:30"' | select UcsCentral, Cause, Dn,  
Created
```

Get Local Service Profiles with Name equals 'SJC'.

```
Get-UcsCentralComputeRequirement -Filter 'Name -ceq SJC' | select UcsCentral, Dn, Name
```

Get all Local Service Profiles with Name equals 'SJC/sjc/SJC' etc.

```
Get-UcsCentralComputeRequirement -Filter 'Name -ieq sjc' | select UcsCentral, Dn, Name
```

Get all Local Service Profiles with Name beginning with string 'SJC/sjc/SJC' etc.

```
Get-UcsCentralComputeRequirement -Filter 'Name -ilike SJC*' | select UcsCentral, Dn, Name
```


Get all Local Service Profiles with Name except 'SJC/sjc/SjC' etc.

```
Get-UcsCentralComputeRequirement -Filter 'Name -ine SJC' | select UcsCentral, Dn, Name
```

Generic Managed Object Queries

Get Managed Object of a specific DN.

```
Get-UcsCentralManagedObject -Dn "compute/sys-1010/chassis-2"
```

Get all Managed Objects of a particular class.

```
Get-UcsCentralManagedObject -ClassId faultInst
```

Get DNs of Managed Objects of a particular class.

```
Get-UcsCentralManagedObject -ClassId faultInst -DnList
```

Get names of all Service Profiles from org-root.

```
Get-UcsCentralOrg -Level root | Get-UcsCentralManagedObject -ClassId computeRequirement | select Name
```

Get immediate children of org-root.

```
Get-UcsCentralOrg -Level root | Get-UcsCentralChild
```

Get parent of a Managed Object.

```
Get-UcsCentralOrg -Name Finance | Get-UcsCentralParent
```

Generic Managed Object Cmdlets

Create an Org

```
Add-UcsCentralManagedObject -ClassId orgOrg -PropertyMap @{Dn = "org-root/org-Finance"; Name = "Finance"}
```

Modify description of Managed Object.

```
Get-UcsCentralManagedObject -Dn org-root/org-Finance | Set-UcsCentralManagedObject -PropertyMap @{Descr = "Example generic set cmdlet";}
```

Remove a Managed Object.

```
Get-UcsCentralOrg -Name Finance | Remove-UcsCentralManagedObject
```

CCO Integration

There are two cmdlets related to CCO Image handling:

Get a List of Images from CCO.

```
$images = Get-UcsCentralCcoImageList
```

Select 1.2(1a) images and download the images. Get-UcsCentralCcoImage first checks if the image is available locally. If the image exists and md5sum matches, no download is initiated. If not, the image is downloaded.

```
$images | where { $_.ImageName -like "ucs-central-bundle.1.2.1a.bin" } | Get-UcsCentralCcoImage -Path C:\work\Images
```

Re-running the command should not initiate any downloads, if the previous download was successful.

```
$images | where { $_.ImageName -like "ucs-central-bundle.1.2.1a.bin" } | Get-UcsCentralCcoImage -Path C:\work\Images
```

A proxy can be used if required.

```
$proxy = New-Object System.Net.WebProxy
$proxy.Address = "http:\\<url>:<port>"
$proxy.UseDefaultCredentials = $false
$proxy.Credentials = New-Object System.Net.NetworkCredential("<username>", "<password>")
$images = Get-UcsCentralCcoImageList -Proxy $proxy
```

Export to XML

Export the configuration of a Managed Object.

```
Export-UcsCentralXml -Dn org-root/org-Finance -Hierarchy -LiteralPath C:\cmd.xml
```

Export the xml of a Managed Object into a file.

```
Get-UcsCentralServiceProfile -Name sp_name | Export-UcsCentralMoXml | Out-File c:\mo.xml
```

Import from XML

Import the configuration from the XML file.

```
Import-UcsCentralXml -LiteralPath C:\cmd.xml
```

Import xml of a Managed Object and convert it into objects.

```
Import-UcsCentralMoXml -LiteralPath c:\mo.xml
```

KVM

Start a KVM session for service profile.

```
Get-UcsCentralComputeRequirement -Name sp_name | Start-UcsCentralKvmSession
```

Start a KVM session for blade 2.

```
Get-UcsCentralChassis -Id 2 | Get-UcsCentralBlade -SlotId 2 | Start-UcsCentralKvmSession
```

Start a KVM session for RackUnit 1 with custom frame Title.

```
Get-UcsCentralRackUnit -Id 1 | Start-UcsCentralKvmSession -FrameTitle "KVM session window for RackUnit"
```

Registered UCS Domains

Get a list of Registered UCS Domains

```
Get-UcsCentralUcsDomain
```

Cmdlet Meta Information

Get Meta information about all Managed Object mapped cmdlets.

```
Get-UcsCentralCmdletMeta
```

Get Meta information about ComputeRequirement mapped cmdlets.

```
Get-UcsCentralCmdletMeta -ClassId computeRequirement
```

View the hierarchy of the ComputeRequirement class.

```
Get-UcsCentralCmdletMeta -Noun UcsCentralComputeRequirement -Tree
```

Get Meta information for the ComputeRequirement noun.

```
Get-UcsCentralCmdletMeta -Noun UcsCentralComputeRequirement
Get-UcsCentralCmdletMeta -Noun ComputeRequirement
```

See the Managed Object information for ComputeRequirement.

```
Get-UcsCentralCmdletMeta -ClassId computeRequirement | Select -ExpandProperty MoMeta
```

See the Managed Object property information for ComputeRequirement.

```
Get-UcsCentralCmdletMeta -ClassId computeRequirement | Select -ExpandProperty MoMeta | Select -ExpandProperty PropertyMeta
```

Add Cmdlet –ModifyPresent Flag

The ModifyPresent option ensures that the add-cmdlets modify the MO, if it already exists, instead of returning an error.

Compare UCS Central Managed Object

The Compare-UcsCentralManagedObject cmdlet compares the managed objects from same or different UcsCentral domains and provides the ManagedObject Difference object generated by the cmdlet.



Note For running this cmdlet, connection handle is not required.

```
$refObj = Get-UcsCentralServiceProfile -Name TestSP -UcsCentral UcsCentral-Ucs-Bidwell-Dev-131a
$diffObj= Get-UcsCentralServiceProfile -Name TestSP -UcsCentral UcsCentral-Ucs-Dev-Auburn
$compareObj = Compare-UcsCentralManagedObject -ReferenceObject $refObj -DifferenceObject $diffObj
```

Synchronize UCS Central Managed Object

The Sync-UcsCentralManagedObject cmdlet uses the ManagedObject Difference object to synchronize the managed objects on a same or different UcsCentral and make the configuration same throughout UcsCentral.

```
Sync-UcsCentralManagedObject -Difference $compareObj -UcsCentral UcsCentral-Dev-Auburn -Force
```

Launch UCS Central Domain Session

The Start-UcsCentralDomainSession cmdlet launches the UCS Manager GUI sessions registered with UcsCentral. You can provide a specific IP address of the UCS Manager to launch a particular UCS Manager GUI session..

```
Start-UcsCentralDomainSession [-Ip <string>] [-UcsCentral <UcsCentralHandle[]>] [<CommonParameters>]
```

Get UCS Central Server

The Get-UcsCentralServer cmdlet returns all the servers regardless of the form factor.

```
Get-UcsCentralServer [-UcsCentral <UcsCentralHandle[]>] [-Xml] [<CommonParameters>]
```

8 Related Cisco UCS Central Documentation and Documentation Feedback

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

- [Release Bundle Contents for Cisco UCS Central Software, Release 1.3](#)
- [Cisco UCS Central XML API Programming Guide](#)
- [Cisco UCS Central Configuration Guides](#)

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

9 Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the *What's New in Cisco Product Documentation* as an RSS feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service. Cisco currently supports RSS Version 2.0.

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. 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.