

# Cisco UCS PowerTool Command Reference for Cisco UCS Manager

## Connection Management

### Connect to Cisco UCS Manager with UCS PowerTool

Cisco UCS® PowerTool cmdlets can connect and communicate with multiple Cisco Unified Computing System™ (Cisco UCS) domains in parallel. Connect to a single Cisco UCS domain, get the current session status, and disconnect by the following cmdlets:

```
connect-ucs <ucsm-ip>
get-ucsstatus -ucs <ucsm-name>
disconnect-ucs -ucs <ucsm-name>
```

By default, multiple concurrent sessions to more than one Cisco UCS domain are not supported. This setting can be overridden for any individual Microsoft PowerShell session using the `Set-UcsPowerToolConfiguration` cmdlet.

```
set-ucspowerToolConfiguration -SupportMultipleDefaultUcs $true
connect-ucs <ucsm-ip1>
connect-ucs <ucsm-ip2>
get-ucsstatus
disconnect-ucs
```

Connect to multiple Cisco UCS domains using the same login credentials:

```
$cred = Get-Credential
$servers = @(("ucsm-ip1", "ucsm-ip2", "ucsm-ip3")
connect-ucs $servers -Credential $cred
```

### Store or Retrieve Cisco UCS Credentials in an Exported XML File

Connect to one or more Cisco UCS domains:

```
connect-ucs <ucsm-ip1>
connect-ucs <ucsm-ip2>
```

Store credentials in a file; the stored credentials are encrypted with a user-specified key:

```
export-ucspSession -LiteralPath 'C:\Work\Tabs.xml'
disconnect-ucs
```

Initiate login from credentials stored in a file:

```
connect-ucs -LiteralPath 'C:\Work\Tabs.xml'
```

Log into an additional system and add the credentials to the file:

```
connect-ucs <ucsm-ip3>
```

```
export-ucspSession -Path 'C:\Work\Tabs.xml' -Merge
```

### Launch Cisco UCS Manager GUI Sessions

Launch a Cisco UCS Manager GUI from a previously connected Cisco UCS session:

```
start-ucsguiSession -ucs <ucsm-name>
```

Launch a Cisco UCS Manager GUI to a Cisco UCS domain that does not have a previous Cisco UCS PowerTool session:

```
start-ucsguiSession -Name <ucsm-ip1>
```

### Launch Cisco UCS Server KVM GUI Sessions

Launch a KVM GUI session to a specific Cisco UCS blade server:

```
get-ucsblade -Chassis 1 -SlotId 1 | start-ucskvmSession
```

Launch a KVM GUI session to a specific Cisco UCS rack-mount server:

```
get-ucsrackunit -Id 1 | start-ucskvmSession
```

Launch a KVM GUI session to a specific Cisco UCS service profile:

```
get-ucsserviceProfile -Name testsp -Org root | start-ucskvmSession
```

## For More Information

### Cisco UCS general user community:

<http://communities.cisco.com/ucs>

### Integrations developed by Cisco for the Cisco UCS user community:

<http://communities.cisco.com/ucsintegrations>

### Cisco UCS Platform Emulator (UCSPE):

<http://communities.cisco.com/ucspe>

### Cisco® Developer Network (DevNET):

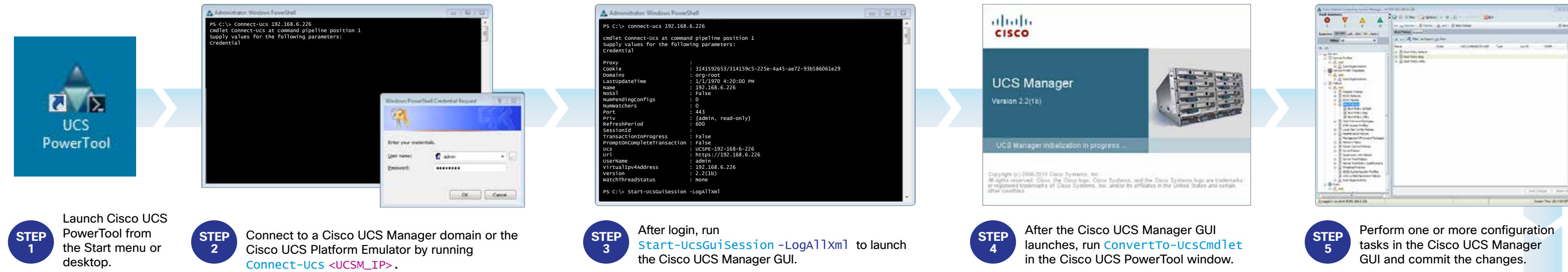
<http://developer.cisco.com>

© 2014 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information. 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. Intel, the Intel logo, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation in the U.S. and/or other countries. (11118) LE-42001-00-08/14

## How to Use the Cisco UCS PowerTool ConvertTo-UcsCmdlet

`ConvertTo-UcsCmdlet` provides the fastest and easiest way to get started working with Cisco UCS PowerTool. This cmdlet converts configuration actions performed with the Cisco UCS Manager GUI into the associated Cisco UCS PowerTool cmdlets, XML, or Python operations.

`ConvertTo-UcsCmdlet` will reduce the learning time for those new to Cisco UCS and Cisco UCS PowerTool, increase script accuracy, and help you quickly implement automation in this environment.



The following are examples of `ConvertTo-UcsCmdlet` usage:

Get the XML requests along with generated cmdlets:

```
convertto-ucscmdlet -verbose
```

Generate cmdlets for objects passed via pipelined input:

```
get-ucsserviceProfile -Name testsp -Hierarchy | ConvertTo-UcsCmdlet
```

Generate cmdlets for actions in the specified GUI log:

```
convertto-ucscmdlet -GuiLog -LiteralPath 'C:\Work\centrale_4711.log'
```

```
convertto-ucscmdlet -GuiLog -Path 'C:\Work\centrale_47*.log?'
```

Generate cmdlets for the specified XML request:

```
convertto-ucscmdlet -xml -Request '<!scClone dn="org-root/1s-sp1"
intargetOrg="org-root" inservName="sp2" inHierarchical="false"></!scClone>'
```

Generate cmdlets for the specified XML requests in file:

```
convertto-ucscmdlet -xml -LiteralPath 'C:\Work\config.xml'
```

Generate cmdlets from a Cisco UCS backup:

```
convertto-ucscmdlet -ucsBackup -LiteralPath 'C:\Work\config-all-backup.xml'
-outputPath 'C:\Work\output.ps1'
```

Note: The Cisco UCS Manager GUI considers a few XML snippets as secure and does not log them, so the `ConvertTo-UcsCmdlet` does not find the logs to perform the translation.

## Transaction Support

You can optimize performance and increase your efficiency with Cisco UCS PowerTool transaction capabilities. Cisco UCS PowerTool transactions allow you to buffer multiple operations and then optimize the request and send it to Cisco UCS Manager as a single operation. Transactions in Cisco UCS are committed in their entirety. If any issues are found within a particular transaction, it will be discarded with no partial commits being applied to the Cisco UCS database. The operations then run in seconds rather than minutes. Elements of a transaction include:

`Start-UcsTransaction` initiates a transaction request. After a transaction is initiated, commands are placed in a queue until the transaction is completed or canceled. Only `Add-*`, `Set-*`, or `Remove-*` cmdlets can be included in a transaction.

`Complete-UcsTransaction` completes a pending transaction. When this cmdlet is issued, Cisco UCS PowerTool optimizes the tasks being performed and sends them as a single operation to Cisco UCS Manager.

`Undo-UcsTransaction` cancels the entire pending transaction. If an operation is canceled, no operation will be passed to Cisco UCS Manager.

`Get-UcsTransactionImpact` estimates the impact of a pending transaction using the `configEstimateImpact` XML API method and returns a `UcsImpact` object. The system returns a message similar to the one provided by the Cisco UCS Manager GUI with the `UcsImpact` object.

**Sample Completed Transaction: Create 900 VLANs**

```
Start-UcsTransaction
$lc = Get-UcsLanCloud
$out = 100..999 | Foreach { $lc | Add-ucsVlan -Name vlan$_ -Id $_ }
Get-UcsPSSession
```

```
Cookie : 3141592653/314159c5-225e-4a45-ae72-93b586061e29
LastUpdateTime : 1/1/1970 4:20:00 PM
Name : 1.2.3.4
NumPendingConfigs : 900
```

```
...
Complete-UcsTransaction
```

**Sample Transaction Impact**

```
Start-UcsTransaction
$sp = Add-UcsServiceProfile -Name sp_name
$eth0 = $sp | Add-ucsvnic -Name eth0 -IdentPoolName empty_pool
$eth0 | Add-ucsvnicInterface -Name primary -DefaultNet true
Get-UcsTransactionImpact
```

This operation will estimate the impact, but does not commit the transaction. In this case, the `UcsImpact` object will be returned. At this point, the transaction can be committed with `Complete-UcsTransaction` or canceled with `Undo-UcsTransaction`.

## Cisco UCS PowerTool Resources

**Options for getting inline help for Cisco UCS PowerTool directly in Microsoft PowerShell**

```
get-help <ucs-cmdlet>
get-help <ucs-cmdlet> -Full
get-help <ucs-cmdlet> -Examples
get-help <ucs-cmdlet> -Detailed
```

**Obtain Cisco UCS PowerTool Object Metadata for Cisco UCS PowerTool Cmdlets**

Cisco UCS PowerTool embeds all the metadata for the Cisco UCS XML data model and relationships in the module itself. Cisco UCS PowerTool uses this metadata as the basis for the definition of every Microsoft PowerShell object returned and passed within every Cisco UCS PowerTool cmdlet. This metadata includes data types, the Cisco UCS version in which an object or property was introduced, and Cisco UCS privileges needed to modify every property and object for all cmdlets in Cisco UCS PowerTool. It provides the verbs defined for each noun, the associated class-name-to-noun mapping, the parent objects that can be used as pipelined input, and the child objects that can be used as pipelined output.

**Sample Uses**

Display the class-name-to-noun mapping, Cisco UCS version in which the object was introduced, valid objects for pipeline input and output, and Cisco UCS privileges needed for the queried Cisco UCS PowerTool cmdlet:

```
Get-UcsCmdletMeta -Noun <cmdlet-noun>
```

Display associated child objects and the relationships to the queried Cisco UCS PowerTool cmdlet:

```
Get-UcsCmdletMeta -Noun <cmdlet-noun> -Tree
```

Display any of the child objects and the relationship to the queried Cisco UCS PowerTool cmdlet:

```
Get-UcsCmdletMeta -Noun <cmdlet-noun> | select -ExpandProperty MoMeta
```

Display the Cisco UCS privileges needed by a user to modify the selected object using Cisco UCS PowerTool:

```
Get-UcsCmdletMeta -Noun <cmdlet-noun> | select -ExpandProperty MoMeta | select -ExpandProperty AccessPrivilege
```

Display the minimum Cisco UCS Manager version where the property or object <cmdlet-noun> was introduced and list the access privileges needed to manipulate the given object via Cisco UCS PowerTool:

```
Get-UcsCmdletMeta -Noun <cmdlet-noun> | select -ExpandProperty MoMeta | select -ExpandProperty PropertyMeta
```

## Compare and Synchronize Cisco UCS Managed Objects

`Compare-UcsManagedObject` compares any similar Cisco UCS PowerTool objects to one other. Output is provided in the form of `diff` objects, which contain one or more items that differ between the two objects being compared. The `diff` object output includes a side indicator with an arrow pointing either left or right signifying each item that is present in one object but not in the other of the two objects being compared. It also can perform a property comparison between two objects, which enables automation of compliance checking.

`Sync-UcsManagedObject` takes the `diff` object output from `Compare-UcsManagedObject` as input to synchronize any found differences with one or more Cisco UCS domains. A `-whatIf` option is provided to display the actions that the cmdlet would perform if it were run against a Cisco UCS domain without applying the changes.

### Sample Uses

**Compare a Collection of Objects Created Across Two Cisco UCS Domains**

Use `Compare-UcsManagedObject` to see the VLAN differences between two Cisco UCS domains. The `diff` object output can then be provided as input to `Sync-UcsManagedObject` to synchronize the changes across one or more of the Cisco UCS domains.

```
$vlansucsm1 = Get-UcsLanCloud -ucs <ucsmip1> | Get-UcsVlan -Limitscope
$vlansucsm2 = Get-UcsLanCloud -ucs <ucsmip2> | Get-UcsVlan -Limitscope
$diff = Compare-UcsManagedObject ($vlansucsm2) ($vlansucsm1)
Sync-UcsManagedObject -ucs <ucsmip2> ($diff)
```

**Copy a Collection of Objects from One Organizational Unit (Org) to Another**

Use the org translation capability provided with `Compare-UcsManagedObject` to see the differences between the two objects being compared. The `diff` object output can then be provided as input to `Sync-UcsManagedObject` to synchronize the changes from one org to another.

```
$srcorg = Get-UcsOrg -Name A -Limitscope
$dstorg = Get-UcsOrg -Name B -Limitscope
$dstsp = Get-UcsServiceProfile -Org $dstorg -Name xyz -Limitscope
$srcsp = Get-UcsServiceProfile -Org $srcorg -Name abc -Limitscope
$diff = Compare-UcsManagedObject ($dstsp) ($srcsp) -XlateOrg org-root/org-B
Sync-UcsManagedObject ($diff) -Force
```

**Copy a Collection of Objects and Remap Values by using a Translation Map**

First, create a translation map to provide property values that need to be translated from one value to another. Then use the translation map along with `Compare-UcsManagedObject` to see the differences between two objects being compared. The `diff` object output can then be provided as input to `Sync-UcsManagedObject` to synchronize the translated objects with the desired destination.

```
$xlatebn = @{ }
$xlatebn['org-root/org-A/1s-abc'] = 'org-root/org-B/1s-xyz'
$srcorg = Get-UcsOrg -Name A -Limitscope
$dstorg = Get-UcsOrg -Name B -Limitscope
$dstsp = Get-UcsServiceProfile -Org $dstorg -Name xyz -Limitscope
$srcsp = Get-UcsServiceProfile -Org $srcorg -Name abc -Limitscope
$diff = Compare-UcsManagedObject ($dstsp) ($srcsp) -XlateMap $xlatebn
Sync-UcsManagedObject ($diff) -Force
```

## Backup, Restore, Upload, and Download Image Firmware

### Backup

**Full-state system backup** creates a snapshot of the entire system and places it into a binary file. The file generated from this backup can be used to perform a full restoration of the system, in the event of a disaster, using the same or a different fabric interconnect. You cannot import this file using the Cisco UCS Manager GUI.

```
Backup-Ucs -Type full-state -PathPattern 'C:\Backups\${ucs}-${yyyy}${mm}${dd}-${HH}${mm}-full-state.tar.gz'
```

**Config-all backup (config-logical)** creates an XML file that includes all system and logical configuration settings, which can be used to import the configuration settings to the original or a different Cisco UCS domain. This backup cannot be used for a full-state system restore operation, and it does not include passwords for locally authenticated users.

```
Backup-Ucs -Type config-all -PathPattern 'C:\Backups\${ucs}-${yyyy}${mm}${dd}-${HH}${mm}-config-all.xml'
```

**Logical backup (config-system)** creates an XML file that includes all logical configuration settings such as service profiles, VLANs, VSANs, pools, and policies. System backup (`config-system`) creates an XML file that includes all system configuration settings such as usernames, roles, and locales. Both a logical and system backup can be imported to the original or a different Cisco UCS domain. Neither can be used for a full system state restoration.

**Delete a backup operation from Cisco UCS Manager**

```
Get-UcsMgmtBackup | Remove-UcsMgmtBackup
```

### Restore (Import)

Restore is available through the `Import-ucsBackup` function. It can be used with `config-all`, `config-logical`, and `config-system` configuration files (XML), but not with `full-state` system backup. An import can be performed while the system is running. When an import is performed, current configuration information is either merged or replaced with the information in the backup file, one object at a time.

### Examples

Replace all configuration information from a `config-all` backup:

```
Import-UcsBackup -LiteralPath 'C:\Backups\config-all.xml'
```

Merge the data in the `config-all.xml` backup with the current Cisco UCS Manager database:

```
Import-UcsBackup -LiteralPath 'C:\Backups\config-all.xml' -Merge
```

### Image Firmware

Cisco UCS PowerTool has two cmdlets to connect to query and download firmware images from Cisco Connect Online. The first, `Get-UcsCcoImageList`, gathers a list of Cisco UCS firmware available for download. The second, `Get-UcsCcoImage`, from Cisco Connect Online the firmware specified.

### Get Firmware List

Get a list of available images from the Cisco website:

```
$ccoCred = Get-Credential
$images = Get-UcsCcoImageList -Credential $ccoCred
```

### Download Firmware

Download the Cisco UCS 2.2(1b) image from the Cisco website (after checking whether the file is available locally):

```
$images | where { $_.ImageName -like "ucs-k9-bundle*2.2.1b*" } | Get-UcsCcoImage -Path 'C:\Work\Images'
```

### Upload Firmware

Upload an image to the connected Cisco UCS domains in the `$DefaultUcs` connection handle:

```
Send-UcsFirmware -LiteralPath 'C:\Work\Images\ucs-k9-bundle-b-series.2.2.1b.B.bin'
```

Upload firmware to a Cisco UCS domain and watch the transfer to verify that sure it completes within the specified time frame:

```
Send-UcsFirmware -LiteralPath 'C:\Work\Images\ucs-k9-bundle-b-series.2.2.1b.B.bin' | Watch-Ucs -Property TransferState -SuccessValue downloaded -PollSec 30 -TimeoutSec 600
```

Cisco UCS  
PowerTool  
Command  
Reference

