



Examples

This chapters contains the following sections:

- [PowerTool Cmdlet Generation](#), on page 3
- [UCS Desired State Configuration \(DSC\)](#), on page 5
- [Get UCS Server](#), on page 13
- [Org](#), on page 13
- [Faults](#), on page 13
- [Get Cmdlet -Hierarchy Flag](#), on page 13
- [Get Cmdlet -LimitScope Flag](#), on page 14
- [Transaction Support](#), on page 14
- [Creating and Deleting VLANs](#), on page 14
- [MAC Pools and Blocks](#), on page 15
- [Server Pools](#), on page 16
- [UUID Suffix Pools and Blocks](#), on page 16
- [WWNN Pools and Blocks](#), on page 16
- [WWPN Pools and Blocks](#), on page 17
- [IQN Suffix Pools and Blocks](#), on page 17
- [Port Roles](#), on page 18
- [Port Channel](#), on page 18
- [Assigning VLANs](#), on page 19
- [Blade Power and Temperature Statistics](#), on page 19
- [Configuration Backup](#), on page 19
- [Import Configuration](#), on page 20
- [Managed Object Synchronization](#), on page 20
- [Monitoring UCS Managed Object Transitions](#), on page 21
- [Technical Support](#), on page 21
- [Service Profile](#), on page 22
- [Service Profile Components](#), on page 22
- [Service Profile Association](#), on page 23
- [Filters](#), on page 24
- [iSCSI Boot](#), on page 25
- [vNIC Template](#), on page 26
- [vHBA Template](#), on page 26
- [Boot Policy](#), on page 27

- [Adapter Policy](#), on page 27
- [BIOS Policy](#), on page 28
- [Host Firmware Package](#), on page 28
- [IPMI Access Profile](#), on page 29
- [Management Firmware Package](#), on page 29
- [Power Control Policy](#), on page 30
- [Server Pool Policy Qualifications](#), on page 30
- [Dynamic vNIC Connection Policy](#), on page 31
- [Network Control Policy](#), on page 31
- [Privileges](#), on page 31
- [User Roles](#), on page 31
- [Locales](#), on page 32
- [User Accounts](#), on page 32
- [Remote Authentication - RADIUS](#), on page 33
- [Remote Authentication - TACACS](#), on page 33
- [Remote Authentication - LDAP](#), on page 33
- [RADIUS Provider](#), on page 34
- [TACACS Provider](#), on page 34
- [LDAP Provider](#), on page 34
- [Authentication Domains](#), on page 34
- [Communication Services](#), on page 35
- [Communication Services - Telnet](#), on page 35
- [Communication Services - CIM XML](#), on page 35
- [Communication Services - SNMP](#), on page 35
- [Communication Services - HTTP](#), on page 36
- [Communication Services - HTTPS](#), on page 37
- [Generic Managed Object Queries](#), on page 38
- [Generic Managed Object Cmdlets](#), on page 39
- [Generic Cmdlet -XmlTag](#), on page 39
- [Upload Firmware](#), on page 40
- [Export to XML](#), on page 40
- [Import from XML](#), on page 40
- [KVM](#), on page 40
- [Launch the UCS Manager Java web UI](#), on page 41
- [Launching the Cisco UCS Manager HTML GUI](#), on page 42
- [UCS Statistics](#), on page 42
- [Configure Scalability Port in UCS 6324 Fabric Interconnect](#), on page 42
- [Transaction Impact](#), on page 43
- [Cmdlet Meta Information](#), on page 43
- [Compare-UcsManagedObject - Dn Translation](#), on page 44
- [Compare-UcsManagedObject - GetPropertyDiff\(\)](#), on page 44
- [Add Cmdlet -ModifyPresent Flag](#), on page 45
- [Capability Catalog Update](#), on page 45
- [Server Operations](#), on page 46
- [32 Parameter Set Limitation](#), on page 47

PowerTool Cmdlet Generation

Generate cmdlets for the specified actions in UCS Manager web UI, using the following cmdlet:

```
ConvertTo-UcsCmdlet
```

Get xml request and generate cmdlets, using the following cmdlet:

```
ConvertTo-UcsCmdlet -Verbose
```

Generate cmdlets for action in the specified web UI log, using the following cmdlets:

```
ConvertTo-UcsCmdlet -GuiLog -LiteralPath 'C:\Work\centrale_7128.log.1'
ConvertTo-UcsCmdlet -GuiLog -Path 'C:\Work\centrale_71*.log.?'
```

Generate cmdlets for the specified xml request, using the following cmdlet:

```
ConvertTo-UcsCmdlet -Xml -Request '<lsClone dn="org-root/ls-spl" inTargetOrg="org-root"
inServerName="sp2" inHierarchical="false"></lsClone>'
```

Generate cmdlets for the specified xml requests in file, using the following cmdlet:

```
ConvertTo-UcsCmdlet -Xml -LiteralPath 'C:\Work\config.xml'
```

Generate cmdlets for the specified MO

From release 1.2(1), you can pipe a manage object to the ConvertTo-UcsCmdlet, and get the cmdlets required to create the managed object.

```
Get-UcsServiceProfile -Name sp1 | ConvertTo-UcsCmdlet
Get-UcsServiceProfile -Name sp1 -Hierarchy | ConvertTo-UcsCmdlet
```

PowerTool DSC Configuration Generation

ConvertTo-UcsDSCConfig cmdlet reads the UCS Manager GUI logs and generates corresponding DSC configuration file. This functionality is similar to ConvertTo-UcsCmdlet cmdlets. The generated DSC configuration file uses the custom UCS DSC UcsManagedObject and UcsScript resources.

Generate DSC configuration for the specified actions in UCS GUI.

```
ConvertTo-UcsDSCConfig
```



Note The ConvertTo-UcsDSCConfig cmdlet is supported on UCS Manager Java GUI only.

Generating DSC Configuration from UCS Manager GUI Actions

You can generate DSC configuration for the actions performed on the UCS Manager GUI using the ConvertTo-UcsDSCConfig cmdlet.

ConvertTo-UcsDSCConfig cmdlet reads the UCS Manager GUI logs and generates corresponding DSC configuration file. This cmdlet is similar to ConvertTo-UcsCmdlet generates cmdlet. The generated DSC configuration file uses the UcsManagedObject and UcsScript resources, which is part of the UCS DSC module.

For more information on launching UCS Manager GUI, see [Generating Cmdlets From UCS Manager GUI Actions](#).

After the UCS Manager web UI is launched using PowerShell window with the Cisco.UcsManager module loaded, run the ConvertTo-UcsDSCConfig cmdlet.

Before you begin

Generating Cmdlets From UCS Manager GUI Actions

You can generate cmdlets for the actions performed on the UCS Manager GUI using the ConvertTo-UcsCmdlet cmdlet. 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 do the translation.

To log the xml snippets of all the user actions in the GUI, launch the UCS Manager GUI by using one of the following ways:

- Using Start-UcsGuiSession -LogAllXml cmdlet
- Manually launch the UCS Manager GUI by performing the following steps:

-
- Step 1** Save the launch link in .jnlp file format. For example, https://<ip_or_hostname>/ucsm/ucsm.jnlp.
- Step 2** Right-click the file and open the file with Notepad.
- Step 3** Add the following line after the other property definitions:
- For Java versions earlier than Java 7 Update 45, add <property name=" log.show.encrypted" value=" true" />
 - For Java 7 Update 45 and later versions, add <property name=" jnlp.ucsm.log.show.encrypted" value=" true" />
- Step 4** Save and close the file.
- Step 5** Right-click the file and open with Java™ Web Start Launcher.
-

After the UCS Manager GUI is launched, from a PowerShell window with the Cisco UCS Manager PowerTool module loaded, run the ConvertTo-UcsCmdlet cmdlet.

Generating Cmdlets from HTML 5 GUI

In HTML GUI one can start XML recording before doing any operation, and stop the recording to download the log, once the operation is done.

-
- Step 1** Launch UCS Manager using HTML
- Step 2** To enable logging:
- Use shortcut keys Ctrl + Alt+ q

- A link (Record XML) comes up on the top right corner of the UI. Click on the link.
Perform operation in the GUI.

Step 3 Click on Stop XML Recording link to stop logging.

Step 4 Enter a log file name in the pop-up.

Step 5 Click on OK button, file gets downloaded in local system.

Step 6 Launch UCS Manager PowerTool and run the **ConvertTo-UcsCmdlet -xml -LiteralPath 'C:\Work\Ucsm.log'** cmdlet

UCS Desired State Configuration (DSC)

Desired State Configuration (DSC) is a new approach for configuring local and remote machines. You can use a UCS DSC resources to configure multiple UCS domains in a datacenter from a centralized root server. PowerTool module Cisco.UCS.DesiredStateConfiguration contains all the custom UCS DSC resources.

```
Get-Module Cisco.UCS.DesiredStateConfiguration -ListAvailable
Get-DscResource | where{$_.Module -ilike 'Cisco*' -and $_.Name -ilike 'ucs*'} | Select Name
```

UCS DSC UcsManagedObject Resource

The UcsManagedObject resource in UCS DSC module provides a mechanism to configure a UCS Manager MO by specifying the details of the MO on multiple UCS Managers using DSC framework.

Syntax

```
UcsManagedObject [string] #ResourceName
{
  Dn = [string]
  Identifier = [string]
  UcsConnectionString = [string]
  UcsCredentials = [PSCredential]
  [ Action = [string] { Add | Set } ]
  [ ClassId = [string] ]
  [ DependsOn = [string[]] ]
  [ Ensure = [string] { Absent | Present } ]
  [ ModifyPresent = [bool] ]
  [ PropertyMap = [string] ]
  [ WebProxyCredentials = [PSCredential] ]
}
```

Property	Description
Dn	Specifies the Dn of managed object.
Identifier	Specifies the unique id for the DSC resource.

Property	Description
UcsConnectionString	Specifies the connection string for UCS Manager. Syntax : Name=<ipAddress> [`nNoSsl=<bool>] [`nPort=<ushort>] [`nProxyAddress=<proxyAddress>] [`nUseProxyDefaultCredentials=<bool>]
UcsCredentials	Indicates the credentials required to access UCS Manager.
Action	Specifies the action you want to perform on managed object. Set this property to Add to add managed object. Set it to Set to modify an existing managed object.
ClassId	Specifies the class id of managed object.
DependsOn	Indicates that the configuration of another resource must run before this resource is configured. For example, if the ID of the resource configuration script block you want to run first is ResourceName and its type is ResourceType, then the syntax for using this property is: DependsOn = "[ResourceType]ResourceName"
Ensure	Indicates if managed object exists. Set this property to Absent. When set to Absent, the resource removes the corresponding MO. Set it to Present to ensure that the managed object does exist. The default is set to Present.
ModifyPresent	Indicates if managed object already exists and the Action is set Add. You can then modify the existing objects.
PropertyMap	Specifies the properties of managed object as keyValue pairs. Syntax: `<key1>=<value1> `<key2>=<value2>
WebProxyCredentials	Indicates the credentials for web proxy.

Example

The following example shows how to use the UcsManagedObject resource to ensure Organization CiscoTest is present in the Dn org-root.

Configuration UcsManagedObjectDemo

```

{
param(
[Parameter(Mandatory=$true)]
[PsCredential] $ucsCredential,

[Parameter(Mandatory=$true)]
[string] $ucsConnectionString
)
Node "localhost"
{
UcsManagedObject AddManagedObject
{
Ensure = "Present"
ModifyPresent = $true
ClassId= "orgOrg"
Dn = "org-root/org-CiscoTest"
PropertyMap= "Descr = test org for UCS DSC `nName = CiscoTest"
UcsCredentials = $ucsCredential
UcsConnectionString = $ucsConnectionString
Identifier = "1"
}
}
}
}

```

UCS DSC UcsSyncMoWithReference Resource

The UcsSyncMoWithReference resource in Cisco UCS Manager PowerTool DSC provides a mechanism to sync configuration from a reference UCS Manager domain. You can specify multiple UCS Manager domains to sync from a reference UCS Manager domain.

Syntax

```

UcsSyncMoWithReference [string] #ResourceName
{
Identifier = [string]
RefUcsConnectionString = [string]
RefUcsCredentials = [PSCredential]
UcsConnectionString = [string]
UcsCredentials = [PSCredential]
[ ClassId = [string] ]
[ DeleteNotPresent = [bool] ]
[ DependsOn = [string[]] ]
[ Dn = [string] ]
[ Ensure = [string] { Absent | Present } ]
[ Hierarchy = [bool] ]
[ WebProxyCredentials = [PSCredential] ]
}

```

Property	Description
Identifier	Specifies the unique id for the DSC resource.

Property	Description
RefUcsConnectionString	Specifies the connection string for reference UCS Manager. Syntax Name=<ipAddress> [`nNoSsl=<bool>][`nPort=<ushort>] [`nProxyAddress=<proxyAddress>] [`nUseProxyDefaultCredentials=<bool>]
RefUcsCredentials	Indicates the credentials required to access reference UCS Manager
UcsConnectionString	Specifies the connection string for target UCS Manager on which sync operation is performed. Syntax: Name=<ipAddress> [`nNoSsl=<bool>][`nPort=<ushort>] [`nProxyAddress=<proxyAddress>] [`nUseProxyDefaultCredentials=<bool>]
UcsCredentials	Indicates the credentials required to access target UCS Manager on which sync operation is performed.
ClassId	Specifies the class id of managed object.If specified 'Dn' and 'Hierarchy' properties are ignored.
DeleteNotPresent	If specified, any missing MOs in reference UCS is deleted.
DependsOn	Indicates the configuration of another resource must run before this resource is configured. For example, if the ID of the resource configuration script block that you want to run first is ResourceName and its type is ResourceType, the syntax for using this property is: DependsOn = "[ResourceType]ResourceName"
Dn	Specifies the Dn of managed object. Hierarchy property is used in combination with Dn property. Dn property is ignored if ClassId property is specified. So, either ClassId or Dn in combination with Hierarchy can be specified at a time.
Ensure	Indicates if SyncMoWithReference operation is performed or not. Set it to Present to perform SyncMoWithReference operation. The default is Present.
Hierarchy	Indicates if all child MOs of specified Dn is synchronized or not. Works when Dn is specified.

Property	Description
WebProxyCredentials	Indicates the credentials for web proxy.

Example

The following example shows how to use the UcsSyncMoWithReference resource to sync MO having "sys/ldap-ext" and all its child MOs.

```
Configuration SyncMoWithReferenceResourceDemo
{
  param(
    [Parameter(Mandatory=$true)]
    [PsCredential] $ucsCredential,

    [Parameter(Mandatory=$true)]
    [PsCredential] $RefUcsCredential
  )
  Import-DSCResource -ModuleName Cisco.Ucs.DesiredStateConfiguration
  Node "localhost"
  {
    UcsSyncMoWithReference "sync1"
    {
      UcsCredentials = $ucsCredential
      UcsConnectionString = "Name=10.65.183.5"
      RefUcsCredentials = $RefUcsCredential
      RefUcsConnectionString = "Name=10.105.214.231"
      Ensure="Present"
      Identifier ="1"
      DeleteNotPresent=$true
      Hierarchy=$true
      Dn="sys/ldap-ext"
    }
  }
}
```

UCS DSC UcsSyncFromBackup Resource

The UcsSyncFromBackup resource in UCS Manager PowerTool DSC provides a mechanism to apply settings to one or more UCS Manager domains from an UCS Manager Backup file.

Syntax

```
UcsSyncFromBackup [string] #ResourceName
{
  Identifier = [string]
  LiteralPath = [string]
  UcsConnectionString = [string]
  UcsCredentials = [PSCredential]
  [ DependsOn = [string[]] ]
  [ Ensure = [string] { Absent | Present } ]
  [ Merge = [bool] ]
```

```
[ WebProxyCredentials = [PSCredential] ]
}
```

Property	Description
Identifier	Specifies the unique id for the DSC resource.
UcsConnectionString	Specifies the connection string for UCS Manager. Syntax: Name=<ipAddress> [`nNoSsl=<bool>] [`nPort=<ushort>] [`nProxyAddress=<proxyAddress>] [`nUseProxyDefaultCredentials=<bool>]
UcsCredentials	Indicates the credentials required to access UCS Manager
DependsOn	Indicates the configuration of another resource must run before this resource is configured. For example, if the ID of the resource configuration script block that you want to run first is ResourceName and its type is ResourceType, the syntax for using this property is: DependsOn = "[ResourceType]ResourceName"
Ensure	Indicates if Sync from Backup is performed or not. The default is Present .
Merge	Indicates if the information in the backup configuration file is compared with the existing configuration information. If there are conflicts, it overwrites the information on the Cisco UCS domain with the information in the backup configuration file.
WebProxyCredentials	Indicates the credentials for web proxy.

Example

The following example shows how to use the UcsSyncFromBackup resource to sync from backup file UcsConfigSystem.

Configuration UcsSyncFromBackupDemo

```
{
param(
[Parameter(Mandatory=$true)]
[PsCredential] $ucsCredential
)
Import-DSCResource -ModuleName Cisco.Ucs.DesiredStateConfiguration

$user = "ru44\administrator"
$password = "<password>" | ConvertTo-SecureString -AsPlainText -Force
```

```

$credential = New-Object System.Management.Automation.PSCredential($user,$password)

Node "localhost"
{
File BackupFileInstance
{
SourcePath = "\\sharedServer\Configs\UcsConfigSystem.xml"
DestinationPath = "C:\BackupFiles\UcsConfigSystem.xml"
Credential = $credential
}

UcsSyncFromBackup SyncFromBackupInstance
{
Ensure = "Present"
LiteralPath = "C:\BackupFiles\UcsConfigSystem.xml"
UcsCredentials = $ucsCredential
UcsConnectionString = "Name=10.65.183.5"
Merge = $true
DependsOn = "[File]BackupFileInstance"
Identifier = "1"
}
}
}

```

UCS DSC UcsScript Resource

The UcsScript resource in UCS Manager PowerTool DSC provides a mechanism to execute UCS Manager PowerTool cmdlets.

Syntax

```

UcsScript [string] #ResourceName
{
Dn = [string]
Identifier = [string]
Script = [string]
UcsConnectionString = [string]
UcsCredentials = [PSCredential]
[ Action = [string] { Add | Set } ]
[ DependsOn = [string[]] ]
[ Ensure = [string] { Absent | Present } ]
[ ModifyPresent = [bool] ]
[ WebProxyCredentials = [PSCredential] ]
}

```

Properties

Property	Description
Dn	Specifies Dn of a managed object.
Identifier	Specifies the unique id for the DSC resource.
Script	Specifies set of PowerTool cmdlets. Use `n as new cmdlet prefix.

Property	Description
UcsConnectionString	Specifies the connection string for UCS Manager. Syntax Name=<ipAddress> [`nNoSsl=<bool>][`nPort=<ushort>] [`nProxyAddress=<proxyAddress>] [`nUseProxyDefaultCredentials=<bool>]
UcsCredentials	Indicates the credentials required to access UCS Manager.
Action	Specifies the action you want to perform on managed object. Set this property to Add to add managed object. Set it to Set to modify an existing managed object
DependsOn	Indicates the configuration of another resource must run before this resource is configured. For example, if the ID of the resource configuration script block that you want to run first is ResourceName and its type is ResourceType, the syntax for using this property is: DependsOn = "[ResourceType]ResourceName"
Ensure	Indicates if Script will execute or not. The default is Present.
ModifyPresent	Indicates if managed object already exists and that Action is set to Add then perform modify the existing objects.
WebProxyCredentials	Indicates the credentials for web proxy.

Example

```

Configuration UcsScriptResourceDemo
{
  param(
    [Parameter(Mandatory=$true)]
    [PsCredential] $ucsCredential,
    [Parameter(Mandatory=$true)]
    [string] $ucsConnectionString
  )
  Import-DSCResource -ModuleName Cisco.Ucs.DesiredStateConfiguration
  Node "localhost"
  {
    UcsScript ResourceInstance
    {
      Ensure = "Present"
      Dn = "org-root/ls-sp_3"
      Script = "Get-UcsOrg -Level root | Get-UcsServiceProfile -Name 'sp3' -LimitScope |
Rename-UcsServiceProfile -NewName 'sp_3' "
      UcsCredentials = $ucsCredential
      UcsConnectionString = $ucsConnectionString
    }
  }
}

```

```
Identifier ="2"  
}  
}  
}
```

Get UCS Server

From Release 1.4.1, use the new cmdlet `Get-UcsServer` to get all the servers regardless of the form factor.

Org

Get a list of orgs across Cisco UCS domains, in the Default UCS list, using the following cmdlet:

```
Get-UcsOrg | select Ucs, Name, Dn
```

Get a handle to the root level Org, using the following cmdlet:

```
Get-UcsOrg -Level root
```

Add 5 orgs to UCS, using the following cmdlet:

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

Faults

Retrieve faults, group them by severity, using the following cmdlet:

```
Get-UcsFault | group Severity
```

Retrieve critical faults, using the following cmdlet:

```
Get-UcsFault -Severity critical | select Ucs, Dn, Cause
```

Acknowledge all unacknowledged faults, using the following cmdlet:

```
Get-UcsFault | ? {$_.Ack -eq "no" } | Acknowledge-UcsFault
```

Get Cmdlet -Hierarchy Flag

Get Managed Object including all children, using the following cmdlet:

```
Get-UcsServiceProfile -Name sp_name -Hierarchy
```

Get Cmdlet -LimitScope Flag

Get service profile at the root level without descending into org root children, using the following cmdlet:

```
Get-UcsServiceProfile -Name sp_name -LimitScope
```

Get service profile from org Finance without descending into org Finance children, using the following cmdlet:

```
Get-UcsOrg -Name Finance | Get-UcsServiceProfile -Name sp_name -LimitScope
```

Get VLAN from LanCloud, using the following cmdlet:

```
Get-UcsLanCloud | Get-UcsVlan -LimitScope
```

Transaction Support

Start a transaction, using the following cmdlet:

```
Start-UcsTransaction
```

Perform an operation.

....

End a transaction, using the following cmdlet:

```
Complete-UcsTransaction
```

Undo a transaction, using the following cmdlet:

```
Undo-UcsTransaction
```

Creating and Deleting VLANs

VLANs in Cisco UCS domains are referred to by name and VLAN definitions can be created under the following four nodes in the MIT:

Node	Description
LanCloud	This is a global VLAN and is applicable to both FIs.
FabricLanCloud	This is a Fabric Specific VLAN and is applicable to either Fabric A or Fabric B.
ApplianceCloud	This is a global VLAN and is applicable to both FIs.
FabricApplianceCloud	This is a fabric specific VLAN applicable to either Fabric A or Fabric B, used during configuration of Appliance Ports.

Create a VLAN under the Global LAN Cloud, using the following cmdlet:

```
Get-UcsLanCloud | Add-UcsVlan -Name lan_cloud_vlan -Id 500
```

Create a VLAN under Fabric A LAN Cloud, using the following cmdlet:

```
Get-UcsFiLanCloud -Id A | Add-UcsVlan -Name fi_a_vlan -Id 500
```

Create a VLAN under Fabric B LAN Cloud, using the following cmdlet:

```
Get-UcsFiLanCloud -Id B | Add-UcsVlan -Name fi_b_vlan -Id 500
```

Create a VLAN under the Global Appliance Cloud, using the following cmdlet:

```
Get-UcsApplianceCloud | Add-UcsVlan -Name appliance_vlan -Id 500
```

Create a VLAN under the Fabric A Appliance Cloud, using the following cmdlet:

```
Get-UcsFabricApplianceCloud -Id A | Add-UcsVlan -Name fi_a_appliance_vlan -Id 500
```

Create a VLAN under the Fabric B Appliance Cloud, using the following cmdlet:

```
Get-UcsFabricApplianceCloud -Id B | Add-UcsVlan -Name fi_b_appliance_vlan -Id 500
```

Import a list of VLANs from a csv file and create the VLANs under the LAN cloud. (This example creates the csv file as well.)

Create VLANs on 1 device, using the following cmdlet:

```
$("Name,Id";foreach ($vlan in 501..550) { "vlan${vlan},${vlan}" }) > C:\work\Demo\vlans.csv
$lc=(Get-UcsLanCloud)
$lc | Get-UcsVlan | select Ucs, Name, Id
Start-UcsTransaction
import-csv C:\work\Demo\vlans.csv | % {$lc | Add-UcsVlan -Name $_.Name -Id $_.Id }
Complete-UcsTransaction
$lc | Get-UcsVlan | select Ucs, Name, Id
```

Remove the added VLANs, using the following cmdlet:

```
$lc | Get-UcsVlan | ? {$_.Id -ge 501 -and $_.Id -le 550 } | Remove-UcsVlan -Force
```

MAC Pools and Blocks

Add a MAC Block to the default MAC Pool, using the following cmdlet:

```
Get-UcsMacPool -Ucs <handle or name> default | Add-UcsMacMemberBlock 00:25:B5:00:A0:00
00:25:B5:00:A0:08
```

Check for any clashes in MAC Pool assignments across all Cisco UCS domains in default list, using the following cmdlet:

```
Get-UcsMacPoolAddr | group Id | where {$_.Count -ne 1 } | select -ExpandProperty Group |
select Ucs, Id, Assigned, AssignedToDn
```

Server Pools

Create a server pool, using the following cmdlet:

```
$server_pool = Add-UcsServerPool -Name server_pool
```

Add Blade 1/4 to the server pool, using the following cmdlet:

```
$server_pool | Add-UcsComputePooledSlot -ChassisId 1 -SlotId 4
```

Add Rack 1 to the server pool, using the following cmdlet:

```
$server_pool | Add-UcsComputePooledRackUnit -Id 1
```

Remove Server pool, using the following cmdlet:

```
$server_pool | Remove-UcsServerPool
```

UUID Suffix Pools and Blocks

Create a UUID Suffix pool, using the following cmdlet:

```
$uuid_pool = Add-UcsUuidSuffixPool -Name uuid_pool -Prefix 3864EB9A-89A2-11DF
```

Add a block of UUID Suffixes to the suffix pool, using the following cmdlet:

```
$uuid_pool | Add-UcsUuidSuffixBlock -From 0000-000000000001 -To 0000-00000000002C
```

Remove a UUID Suffix pool, using the following cmdlet:

```
$uuid_pool | Remove-UcsUuidSuffixPool
```

WWNN Pools and Blocks

Get all WWNN pool in UCS, using the following cmdlet:

```
Get-UcsWwnPool -Purpose node-wnn-assignment
```

Create a WWNN pool, using the following cmdlet:

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

Add a WWN block to the WWNN pool, using the following cmdlet:

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

Add a WWNN initiator to the WWNN pool, using the following cmdlet:

```
$wwnn_pool | Add-UcsWwnInitiator -Id 20:00:00:25:B5:00:00:2C -Name wwnn_initiator
```


Remove a WWNN initiator, using the following cmdlet:

```
$wwnn_pool | Get-UcsWwnInitiator -Id 20:00:00:25:B5:00:00:2C | Remove-UcsWwnInitiator
```

Remove a WWNN pool, using the following cmdlet:

```
$wwnn_pool | Remove-UcsWwnPool
```

WWPN Pools and Blocks

Get all WWPN pool in UCS, using the following cmdlet:

```
Get-UcsWwnPool -Purpose port-wwn-assignment
```

Create a WWPN pool, using the following cmdlet:

```
$wwpn_pool = Add-UcsWwnPool -Name wwpn_pool -Purpose port-wwn-assignment
```

Add a WWN block to the WWPN pool, using the following cmdlet:

```
$wwpn_pool | Add-UcsWwnMemberBlock -From 20:00:00:24:B5:00:00:00 -To 20:00:00:24:B5:00:00:09
```

Add a WWPN initiator to the WWPN pool, using the following cmdlet:

```
$wwpn_pool | Add-UcsWwnInitiator -Id 20:00:00:25:B5:00:00:2D -Name wwpn_initiator
```

Set descr for WWPN initiator, using the following cmdlet:

```
$wwpn_pool | Get-UcsWwnInitiator -Id 20:00:00:25:B5:00:00:2D | Set-UcsWwnInitiator -Descr "WWPN initiator modified"
```

Remove a WWPN pool, using the following cmdlet:

```
$wwpn_pool | Remove-UcsWwnPool
```

IQN Suffix Pools and Blocks

Get IQN pool in UCS, using the following cmdlet:

```
Get-UcsIqnPoolPool -Name iqnSuffixPool
```

Create IQN pool, using the following cmdlet:

```
$iqn_pool = Get-UcsOrg -Level root | Add-UcsIqnPoolPool -Name iqn_pool -Prefix I
```

Create IQN pool block, using the following cmdlet:

```
$iqn_pool_block =  
$iqn_pool | Add-UcsIqnPoolBlock -Suffix B -From 0 -To 10
```

Remove IQN pool block, using the following cmdlet:

```
$iqn_pool_block | Remove-UcsIqnPoolBlock
```

Remove IQN pool, using the following cmdlet:

```
$iqn_pool | Remove-UcsIqnPoolPool
```

Port Roles

Make Fabric A's Slot 1 (Fixed Ports Slot) Port 19 a server port, using the following cmdlet:

```
Get-UcsFabricServerCloud -Id A | Add-UcsServerPort -PortId 19 -SlotId 1
```

Unconfigure Fabric A's Slot 1 (Fixed Ports Slot) Port 19 from being a server port, using the following cmdlet:

```
Get-UcsFabricServerCloud -Id A | Get-UcsServerPort -PortId 19 -SlotId 1 | Remove-UcsServerPort -Force
```

Make Fabric A's Slot 1 (Fixed Ports Slot) Port 15 an appliance port, using the following cmdlet:

```
Get-UcsFabricApplianceCloud -Id A | Add-UcsAppliancePort -PortId 15 -SlotId 1
```

Unconfigure Fabric A's Slot 1 (Fixed Ports Slot) Port 15 from being an appliance port, using the following cmdlet:

```
Get-UcsFabricApplianceCloud -Id A | Get-UcsAppliancePort -PortId 15 -SlotId 1 | Remove-UcsAppliancePort -Force
```

Make Fabric A's Slot 1 (Fixed Ports Slot) Port 16 an uplink port, using the following cmdlet:

```
Get-UcsFiLanCloud -Id A | Add-UcsUplinkPort -PortId 16 -SlotId 1
```

Unconfigure Fabric A's Slot 1 (Fixed Ports Slot) Port 16 from being an uplink port, using the following cmdlet:

```
Get-UcsFiLanCloud -Id A | Get-UcsUplinkPort -PortId 16 -SlotId 1 | Remove-UcsUplinkPort -Force
```

Port Channel

Create an Appliance Port Channel on Fabric A with ports 19 & 20, using the following cmdlet:

```
$ap_pc = Get-UcsFabricApplianceCloud -Id A | Add-UcsAppliancePortChannel -PortId 55
$ap_pc | Add-UcsAppliancePortChannelMember -SlotId 1 -PortId 19
$ap_pc | Add-UcsAppliancePortChannelMember -SlotId 1 -PortId 20
```

Add Port Channel to the Appliance VLAN, using the following cmdlet:

```
Get-UcsApplianceCloud | Get-UcsVlan -Name ApplianceVlan | Add-UcsVlanMemberPortChannel -SwitchId A -PortId $ap_pc.PortId
```

Remove Port Channel from Appliance VLAN, using the following cmdlet:

```
Get-UcsApplianceCloud | Get-UcsVlan -Name ApplianceVlan | Get-UcsVlanMemberPortChannel
-SwitchId A -PortId 55 | Remove-UcsVlanMemberPortChannel -Force
```

Remove Appliance Port Channel, using the following cmdlet:

```
Get-UcsFabricApplianceCloud -id A | Get-UcsAppliancePortChannel -PortId 55 |
Remove-UcsAppliancePortChannel -Force
```

Assigning VLANs

Add Appliance port A/1/15 to an Appliance VLAN, using the following cmdlet:

```
Get-UcsApplianceCloud | Get-UcsVlan -name ApplianceVlan | Add-UcsVlanMemberPort -SwitchId
A -SlotId 1 -PortId 15
```

Remove Appliance port A/1/15 from Appliance VLAN, using the following cmdlet:

```
Get-UcsApplianceCloud | Get-UcsVlan -name ApplianceVlan | Get-UcsVlanMemberPort
-SwitchId A -SlotId 1 -PortId 15 | Remove-UcsVlanMemberPort -Force
```

Blade Power and Temperature Statistics

View Power Statistics of all blades, using the following cmdlet:

```
Get-UcsBlade | Get-UcsComputeBoard | Get-UcsComputeMbPowerStats | Out-GridView
```

View Temperature Statistics of all blades, using the following cmdlet:

```
Get-UcsBlade | Get-UcsComputeBoard | Get-UcsComputeMbTempStats | Out-GridView
```

Configuration Backup

Remove any previously stored backups in UCS.

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

The PathPattern can be auto-filled, allowing the cmdlet to be used with multiple Cisco UCS domains. Create and download full-state system backup of UCS. 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 original fabric interconnect, or recreate the configuration on a different fabric interconnect. You cannot use this file for an import.

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

Create and download logical backup of UCS. 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 original fabric interconnect or to a different fabric interconnect. You cannot use this file for a system restore.

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

Create and download system backup of UCS. 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 original fabric interconnect or to a different fabric interconnect. You cannot use this file for a system restore.

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

Create and download config-all backup of UCS. 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 original fabric interconnect or to a different fabric interconnect. You cannot use this file for a system restore. This file 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'
```

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

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 domain with the information in the import configuration file.

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

Managed Object Synchronization

Sync a set of MOs from SYSA to SYSB, using the following cmdlet:

```
Sync-UcsManagedobject -Ucs SYSB (Compare-UcsManagedObject (Get-UcsOrg -ucs SYSB) (Get-UcsOrg
-ucs SYSA)) -whatif
Sync-UcsManagedobject -Ucs SYSB (Compare-UcsManagedObject (Get-UcsOrg -ucs SYSB) (Get-UcsOrg
-ucs SYSA)) -Force
```

Sync a set of MOs from SYSA to all systems in the default list, using the following cmdlet:

```
Get-UcsPSSession | % {Sync-UcsManagedobject -Ucs $_ (Compare-UcsManagedObject (Get-UcsOrg
-ucs $_) (Get-UcsOrg -ucs SYSA)) -Force}
```

Monitoring UCS Managed Object Transitions

Watch Cisco UCS domains for all events for 60 seconds.

```
Watch-Ucs -TimeoutSec 60
```

Watch Cisco UCS domains for any changes in faults for 60 seconds.

```
Watch-Ucs -TimeoutSec 60 -ClassId FaultInst
```

Watch UCS for a particular field in MO to attain a success value.

```
Send-UcsFirmware -LiteralPath C:\work\Images\ucs-k9-bundle-b-series.1.4.2b.B.bin |
Watch-Ucs -Property TransferState -SuccessValue downloaded -PollSec 30 -TimeoutSec 600
```

Technical Support

Technical support data for the entire UCS Manager instance will be created and downloaded to the specified file.

```
Get-UcsTechSupport -PathPattern 'C:\${ucs}-techsupp-ucsm.tar' -UcsManager -RemoveFromUcs
-TimeoutSec 600
```

Technical support data for the UCS Manager management services(excluding fabric interconnects) will be created and downloaded to the specified file.

```
Get-UcsTechSupport -PathPattern 'C:\${ucs}-techsupp-ucsmgmt.tar' -UcsMgmt -RemoveFromUcs
-TimeoutSec 600
```

Technical support data for Chassis id 1 and Cmc id 1 will be created and downloaded to specified file.

```
Get-UcsTechSupport -PathPattern 'C:\${ucs}-techsupp-chassis.tar' -RemoveFromUcs -TimeoutSec
600 -ChassisId 1 -CmcId 1
```

Technical support data for Chassis id 1 and Iom id 1 will be created and downloaded to specified file.

```
Get-UcsTechSupport -PathPattern 'C:\${ucs}-techsupp-iom.tar' -RemoveFromUcs -TimeoutSec 600
-ChassisId 1 -IomId 1
```

Technical support data for RackServer id 1 and RackAdapter id 1 will be created and downloaded to specified file.

```
Get-UcsTechSupport -PathPattern 'C:\${ucs}-techsupp-rack.tar' -RemoveFromUcs -TimeoutSec
600 -RackServerId 1 -RackAdapterId 1
```

Technical support data for FEX id 1 will be created and downloaded to specified file.

```
Get-UcsTechSupport -PathPattern 'C:\${ucs}-techsupp-fex.tar' -RemoveFromUcs -TimeoutSec 600
-FexId 1
```

Service Profile

Get all service profile instances in UCS.

```
Get-UcsServiceProfile -Type instance
```

Get all service profile updating templates in UCS.

```
Get-UcsServiceProfile -Type updating-template
```

Get all service profile initial templates in UCS.

```
Get-UcsServiceProfile -Type initial-template
```

Add a new service profile `sp_name` from service profile template `sp_template`

```
Add-UcsServiceProfile -SrcTemplName sp_template -Name sp_name
```

Add a service profile.

```
Add-UcsServiceProfile -Name sp_name -BootPolicyName boot_policy -BiosProfileName bios_policy
-HostFwPolicyName 1.4-3i -MgmtFwPolicyName 1.4-3i -MaintPolicyName maint_policy
-MgmtAccessPolicyName ipmi_policy -PowerPolicyName power_policy -ScrubPolicyName scrub_policy

-SolPolicyName sol_policy -StatsPolicyName stats_policy -AgentPolicyName agent_policy
-DynamicConPolicyName vNIC_policy -ExtIPState static -IdentPoolName UUID_pool
-LocalDiskPolicyName
disk_policy -Uuid "00000000-0000-0000-0000-000000000008" -UsrLbl "serviceprofile"
```

Get power state of a service profile.

```
Get-UcsServiceProfile -Name sp_name | Get-UcsServerPower
```

Bind service profile to a template.

```
Get-UcsServiceProfile -Name sp_name | Set-UcsServiceProfile -SrcTemplName sp_template
```

Remove a service profile.

```
Get-UcsServiceProfile -Name sp_name | Remove-UcsServiceProfile
```

Service Profile Components

Create a service profile, using the following cmdlet:

```
$sp = Add-UcsServiceProfile -Name sp_name
```

Create a vNIC with reference to QoS Policy, using the following cmdlet:

```
$eth0 = $sp | Add-UcsVnic -Name eth0 -QosPolicyName qos_policy
```

Add a VLAN for vNIC, make it Native VLAN, using the following cmdlet:

```
$eth0 | Add-UcsVnicInterface -Name fi_a_vlan -DefaultNet true
```

Add a VLAN for vNIC, using the following cmdlet:

```
$eth0 | Add-UcsVnicInterface -Name fi_b_vlan
```

Create a vNIC and instantiate from template, using the following cmdlet:

```
$sp | Add-UcsVnic -Name eth1 -NwTemplName vnic_template
```

Create a vHBA, using the following cmdlet:

```
$fc0 = $sp | Add-UcsVhba -Name fc0 -IdentPoolName wwpn_pool
```

Add a VSAN for vHBA, using the following cmdlet:

```
$fc0 | Set-UcsVhbaInterface -Name fi_b_vsan
```

Add a WWNN pool, using the following cmdlet:

```
$sp | Add-UcsVnicFcNode -IdentPoolName node_default
```

Service Profile Association

Associate a service profile to a blade, using the following cmdlet:

```
Get-UcsServiceProfile sp_name -LimitScope | Associate-UcsServiceProfile -Blade (Get-UcsBlade -Chassis 1 -SlotId 1)
```

Associate service profile to a rack, using the following cmdlet:

```
Get-UcsServiceProfile sp_name -LimitScope | Associate-UcsServiceProfile -RackUnit (Get-UcsRackUnit -ServerId 1)
```

Associate service profile to a server pool, using the following cmdlet:

```
Get-UcsServiceProfile sp_name -LimitScope | Associate-UcsServiceProfile -ServerPoolName FileServerPool
```

Associate a service profile to a server pool along with server pool qualification policy, using the following cmdlet:

```
Get-UcsServiceProfile sp_name -LimitScope | Associate-UcsServiceProfile -ServerPoolName file_server_pool -ServerPoolQualificationPolicyName file_server_pool
```

Disassociate a service profile, using the following cmdlet:

```
Get-UcsServiceProfile sp_name -LimitScope | Disassociate-UcsServiceProfile
```

Create a copy of a service profile, using the following cmdlet:

```
Get-UcsServiceProfile -Name sp_name -LimitScope | Copy-UcsServiceProfile -NewName copy_sp_name
```

Rename a service profile, using the following cmdlet:

```
Get-UcsServiceProfile -Name sp_name | Rename-UcsServiceProfile -NewName rename_sp_name
```

Filters

Get all Service Profile Templates, using the following cmdlet:

```
Get-UcsServiceProfile -Filter 'Type -clike *-template' | select Ucs,Dn,Name
```

Get all Service Profiles with Name containing string 'SJC', using the following cmdlet:

```
Get-UcsServiceProfile -Filter 'Name -cmatch SJC' | select Ucs, Dn, Name
```

Get all Service Profiles with Name beginning with string 'SJC', using the following cmdlet:

```
Get-UcsServiceProfile -Filter 'Name -clike SJC' | select Ucs, Dn, Name
```

Get all VLANs with Id between 8 and 50, using the following cmdlet:

```
Get-UcsVlan -Filter 'Id -cbw 8,50' | select Ucs,Dn, Name, Id
```

Get all Roles that have the fault privilege, using the following cmdlet:

```
Get-UcsRole -Filter 'Priv -ccontains fault' | select Ucs, Dn, Name
```

Get all Roles that have the fault or operations privilege, using the following cmdlet:

```
Get-UcsRole -Filter 'Priv -canybit fault,operations' | select Ucs, Dn, Name
```

Get all Roles that have the fault and operations privilege, using the following cmdlet:

```
Get-UcsRole -Filter 'Priv -callbits fault,operations' | select Ucs, Dn, Name
```

Get a list of blades/rack units with temperature greater than 45, using the following cmdlet:

```
Get-UcsProcessorEnvStats -Filter 'Temperature -cgt 45' | Get-UcsParent | Get-UcsParent |
Get-UcsParent | select Ucs, Dn
```

Get a list of faults generated between 4/18/2012 9:00 and 4/19/2012 9:30, using the following cmdlet:

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

Get Service Profiles with Name equals 'SJC', using the following cmdlet:

```
Get-UcsServiceProfile -Filter 'Name -ceq SJC' | select Ucs, Dn, Name
```

Get all Service Profiles with Name equals 'SJC/sjc/SjC' and so on, using the following cmdlet:

```
Get-UcsServiceProfile -Filter 'Name -ieq sjc' | select Ucs, Dn, Name
```

Get all Service Profiles with Name beginning with string 'SJC/sjc/SjC' and so on, using the following cmdlet:

```
Get-UcsServiceProfile -Filter 'Name -ilike SJC*' | select Ucs, Dn, Name
```


Get all Service Profiles with Name except 'SJC/sjc/SjC' and so on, using the following cmdlet:

```
Get-UcsServiceProfile -Filter 'Name -ine SJC' | select Ucs, Dn, Name
```

iSCSI Boot

Start Ucs Transaction.

```
Start-UcsTransaction
```

Create a service profile.

```
$sp = Add-UcsServiceProfile -Type instance -Name iscsiboot
```

Add a static IP (not related to iSCSI boot).

```
$staticIp = Add-UcsVnicIPv4StaticAddr -ServiceProfile $sp -Addr 10.65.224.161 -DefGw 10.65.224.1 -Subnet 255.255.255.0
```

Create the required vNIC and add VLAN.

```
$vnic = Add-UcsVnic -ServiceProfile $sp -Name enic1 -SwitchId A -Addr 00:25:B5:07:80:00
$vlan605 = Add-UcsVnicInterface -Vnic $vnic -Name vlan605 -DefaultNet yes
```

Create iSCSI vNIC and map it to the vNIC created above.

Add iSCSI Initiator Parameters - VLAN and IP address.

```
$enic = Add-UcsVnicIScsi -ServiceProfile $sp -Name iscsienic1 -InitiatorName iqn.1995-05.com.broadcom.iscsiboot -VnicName enic1
$vlan = Add-UcsVnicVlan -VlanName vlan605 -VnicIScsi $enic
$ipv4if = Add-UcsVnicIPv4If -VnicVlan $vlan
$ipv4iscsi = Add-UcsVnicIPv4IscsiAddrp -VnicIPv4If $ipv4if -Addr 10.65.224.157
```

Add target parameters.

```
$primaryTarget = Add-UcsVnicIScsiStaticTargetIf -VnicVlanp $vlan -IpAddress 10.65.224.16 -Name iqn.1992-08.com.netapp:sn.135037408 -Priority 1
$primaryLun = Add-UcsVnicLunp -VnicIScsiStaticTargetIf $primaryTarget -Id 2
```

Create a specific boot policy.

```
$bootPolicy = Add-UcsBootDefinition -ServiceProfile $sp
```

If installation is required, create a LsbootVirtualMedia.

```
$vmedia = Add-UcsLsbootVirtualMedia -BootDefinition $bootPolicy -Access read-only -Order 1
```

Add iSCSI enic in the boot path.

```
$iscsiBoot = Add-UcsLsbootIScsi -BootDefinition $bootPolicy -Order 2
$iscsiBootImagePath = Add-UcsLsbootIScsiImagePath -LsbootIscsi $iscsiBoot -Type primary -ISCSIVnicName iscsienic1
```

Complete Ucs Transaction.

```
Complete-UcsTransaction | Out-null
```

vNIC Template

Create an Initial vNIC template, using the following cmdlet:

```
$vnic_init_temp = Add-UcsVnicTemplate -Name vnic_init_temp -TemplType initial-template
-SwitchId A
```

Create an Updating vNIC template, using the following cmdlet:

```
$vnic_update_temp = Add-UcsVnicTemplate -Name vnic_update_temp -TemplType updating-template
-SwitchId B -Target adaptor
```

Add a VLAN to an Initial vNIC template, using the following cmdlet:

```
$vnic_init_temp | Add-UcsVnicInterface -Name fi_a_vlan
```

Add a VLAN to an Initial vNIC template and make it Native VLAN, using the following cmdlet:

```
$vnic_init_temp | Add-UcsVnicInterface -Name lan_cloud_vlan -DefaultNet true
```

Set MAC Pool, Network Control policy and QoS policy for Initial vNIC template, using the following cmdlet:

```
$vnic_init_temp | Set-UcsVnicTemplate -IdentPoolName mac_pool -NwCtrlPolicyName network_policy
-QosPolicyName qos_policy
```

Remove an Initial vNIC template, using the following cmdlet:

```
$vnic_init_temp | Remove-UcsVnicTemplate
```

vHBA Template

Create an Initial vHBA template, using the following cmdlet:

```
$vhba_init_temp = Add-UcsVhbaTemplate -Name vhba_init_temp -TemplType initial-template
-SwitchId A
```

Create an Updating vHBA template, using the following cmdlet:

```
$vhba_update_temp = Add-UcsVhbaTemplate -Name vhba_update_temp -TemplType updating-template
-SwitchId B
```

Add a VSAN to Updating vHBA template, using the following cmdlet:

```
$vhba_update_temp | Add-UcsVhbaInterface -Name fi_b_vsan
```

Set WWN Pool, QoS policy, Pin Group and Stats policy for Updating vHBA template, using the following cmdlet:

```
$vhba_update_temp | Set-UcsVhbaTemplate -IdentPoolName wwpn_pool -QosPolicyName qos_policy
```

```
-PinToGroupName san_pin_group -StatsPolicyName threshold_policy
```

Remove an Updating vHBA template, using the following cmdlet:

```
$vhba_update_temp | Remove-UcsVhbtemplate
```

Boot Policy

Create a Boot policy and enable Reboot on boot order change and enforce vNIC/vHBA/iSCSI name, using the following cmdlet:

```
$boot_policy = Get-UcsOrg -Name root | Add-UcsBootPolicy -Name boot_policy -EnforceVnicName  
yes -RebootOnUpdate yes
```

Add a Floppy, using the following cmdlet:

```
$boot_policy | Add-UcsLsbootVirtualMedia -Order 3 -Access read-write
```

Add a CD-ROM, using the following cmdlet:

```
$boot_policy | Add-UcsLsbootVirtualMedia -Order 2 -Access read-only
```

Add a Local Disk, using the following cmdlet:

```
$boot_storage = $boot_policy | Add-UcsLsbootStorage -Order 1 $boot_storage |  
Add-UcsLsbootLocalStorage
```

Add a SAN boot, using the following cmdlet:

```
$boot_storage | Add-UcsLsbootSanImage -VnicName fc0 -Type primary | Add-UcsLsbootSanImagePath  
-Type primary -Lun 1 -Wwn 20:00:00:25:B5:00:00:00
```

Add a LAN boot, using the following cmdlet:

```
$boot_policy | Add-UcsLsbootLan -Order 4 | Add-UcsLsbootLanImagePath -VnicName eth0 -Type  
primary
```

Remove a Boot policy, using the following cmdlet:

```
$boot_policy | Remove-UcsBootPolicy
```

Adapter Policy

Add a custom Eth Adapter policy, that disables receive checksum offload.

```
$eth_adap_policy = Add-UcsEthAdapterPolicy -Name eth_adap_policy -Descr "Custom Adapter  
Policy"  
$eth_adap_policy | Set-UcsEthAdapterOffloadProfile -TcpRxChecksum disabled
```

Add a FC Adapter policy.

```
$fc_adap_policy = Add-UcsFcAdapterPolicy -Name fc_adap_policy -Descr "Fibre Channel Adapter  
Policy"
```

Enable FCP error recovery for Fibre Channel Adapter policy.

```
$fc_adap_policy | Set-UcsAdaptorFcErrorRecoveryProfile -FcpErrorRecovery enabled
```

Add an iSCSI Adapter policy.

```
$iscsi_adap_policy = Add-UcsIScsiAdapterPolicy -Name iscsi_policy
```

Enable TCP timestamp, HBA mode and Boot to target for iSCSI Adapter policy.

```
$iscsi_adap_policy | Set-UcsIScsiAdapterPolicyProperties -BootToTarget yes -TcpTimeStamp yes -HbaMode yes
```

BIOS Policy

Create a Bios policy and enable reboot on Bios setting change. using the following cmdlet:

```
$bios_policy = Add-UcsBiosPolicy -Name bios_policy -RebootOnUpdate yes
```

Modify USB system idle power optimizing setting to high-performance. using the following cmdlet:

```
$bios_policy | Set-UcsBiosVfUSBSystemIdlePowerOptimizingSetting -VpUSBIdlePowerOptimizing high-performance
```

Enable Virtualization technology. using the following cmdlet:

```
$bios_policy | Set-UcsBiosVfIntelVirtualizationTechnology -VpIntelVirtualizationTechnology enabled
```

Enable quiet boot for Bios policy. using the following cmdlet:

```
$bios_policy | Set-UcsBiosVfQuietBoot -VpQuietBoot enabled
```

Resume Ac on power loss to last-state. using the following cmdlet:

```
$bios_policy | Set-UcsBiosVfResumeOnACPowerLoss -VpResumeOnACPowerLoss last-state
```

Disable boot option retry. using the following cmdlet:

```
$bios_policy | Set-UcsBiosVfBootOptionRetry -VpBootOptionRetry disabled
```

Disable console redirection. using the following cmdlet:

```
$bios_policy | Set-UcsBiosVfConsoleRedirection -VpConsoleRedirection disabled
```

Remove a Bios policy. using the following cmdlet:

```
$bios_policy | Remove-UcsBiosPolicy
```

Host Firmware Package

Create a Host Firmware package and set IgnoreCompCheck to No, using the following cmdlet:

```
$host_firm_pack = Add-UcsFirmwareComputeHostPack -Name host_firm_pack -IgnoreCompCheck no
```

Add a Host Firmware pack item, using the following cmdlet:

```
$host_firm_pack | Add-UcsFirmwarePackItem -Type adaptor -HwModel N20-AC0002 -HwVendor "Cisco Systems Inc" -Version '1.4(1i)'
```

Set version of Host Firmware pack item, using the following cmdlet:

```
$host_firm_pack | Get-UcsFirmwarePackItem -HwModel N20-AC0002 | Set-UcsFirmwarePackItem -Version '2.0(1t)'
```

Remove Host Firmware package, using the following cmdlet:

```
$host_firm_pack | Remove-UcsFirmwareComputeHostPack
```

IPMI Access Profile

Create an IPMI Access profile, using the following cmdlet:

```
$ipmi_profile= Get-UcsOrg -Name root | Add-UcsIpmiAccessProfile -Name ipmi_profile
```

Add an IPMI user with Administrator's role, using the following cmdlet:

```
$ipmi_profile | Add-UcsAaaEpUser -Name ipmiUser -Priv admin
```

Modify role for IPMI user, using the following cmdlet:

```
$ipmi_profile | Get-UcsAaaEpUser -Name ipmiUser | Set-UcsAaaEpUser -Priv readonly
```

Remove an IPMI Access profile, using the following cmdlet:

```
$ipmi_profile | Remove-UcsIpmiAccessProfile
```

Management Firmware Package

Create a Management Firmware package and set IgnoreCompCheck to No, using the following cmdlet:

```
$mgmt_firm_pack = Add-UcsFirmwareComputeMgmtPack -Name mgmt_firm_pack -IgnoreCompCheck no
```

Add a Management Firmware pack item, using the following cmdlet:

```
$mgmt_firm_pack | Add-UcsFirmwarePackItem -Type blade-controller -HwModel "N20-B6620-1" -HwVendor "Cisco Systems Inc" -Version '1.4(1i)'
```

Set version of Management Firmware pack item, using the following cmdlet:

```
$mgmt_firm_pack | Get-UcsFirmwarePackItem -HwModel N20-B6620-1 | Set-UcsFirmwarePackItem -Version '2.0(1t)'
```

Remove Management Firmware package, using the following cmdlet:

```
$mgmt_firm_pack | Remove-UcsFirmwareComputeMgmtPack
```

Power Control Policy

Create a Power Control policy. Priority is ranked on a scale of 1-10, where 1 indicates the highest priority and 10 indicates lowest priority. The default priority is 5.

```
$power_policy = get-UcsOrg -Level root | Add-UcsPowerPolicy -Name power_policy -Prio 6
```

Create a Power Control policy with power capping 'no-cap'.

```
$power_nocap = get-UcsOrg -Name root | Add-UcsPowerPolicy -Name power_nocap -Prio no-cap
```

Remove Power Control policy.

```
$power_policy | Remove-UcsPowerPolicy
```

Server Pool Policy Qualifications

Create a Server Pool policy qualification, using the following cmdlet:

```
$server_pool_qual = Add-UcsServerPoolQualification -Name server_pool_qual
```

Create an Adaptor qualification, using the following cmdlet:

```
$server_pool_qual | Add-UcsAdaptorQualification
```

Create a Memory qualification policy with memory clock speed of 1067Mhz and 16 memory units, using the following cmdlet:

```
$server_pool_qual | Add-UcsMemoryQualification -Clock 1067 -Units 16
```

Create a CPU/Cores qualification policy with Pentium_4 processor architecture, using the following cmdlet:

```
$server_pool_qual | Add-UcsCpuQualification -Arch Pentium_4
```

Create a Diskless Storage qualification policy for servers without a local disk (SAN only configuration), using the following cmdlet:

```
$server_pool_qual | Add-UcsStorageQualification -Diskless yes
```

Create a Rack qualification, using the following cmdlet:

```
$server_pool_qual | Add-UcsRackQualification -MaxId 1 -MinId 1[1]
```

Remove a Server Pool policy qualification, using the following cmdlet:

```
$server_pool_qual | Remove-UcsServerPoolQualification
```

Dynamic vNIC Connection Policy

Create a Dynamic vNIC connection policy `dy_vnic_conn` with 54 dynamic vNICs and protection enabled for failover mode.

```
$dy_vnic_conn = Add-UcsDynamicVnicConnPolicy -Name dy_vnic_conn -AdaptorProfileName Linux
-DynamicEth 54 -Protection protected
```

Remove a Dynamic vNIC connection policy.

```
$dy_vnic_conn | Remove-UcsDynamicVnicConnPolicy
```

Network Control Policy

Create a Network Control policy `network_policy` with CDP enabled and VIF configured to change the operational state of a vNIC to down when uplink connectivity is lost on the fabric interconnect.

```
$network_policy = Get-UcsOrg -Level root | Add-UcsNetworkControlPolicy -Name network_policy
-Cdp enabled -UplinkFailAction link-down
```

Set Mac security for Network Control policy to allow forged MAC addresses, using the following cmdlet:

```
$network_policy | Set-UcsPortSecurityConfig -Forge allow
```

Set Mac security for Network Control policy so that after the first packet has been sent to the fabric interconnect, all other packets must use the same MAC address or they will be silently rejected by the fabric interconnect. This enables port security for the associated vNIC.

```
$network_policy | Set-UcsPortSecurityConfig -Forge deny
```

Remove a Network Control policy, using the following cmdlet:

```
$network_policy | Remove-UcsNetworkControlPolicy
```

Privileges

List out all privileges on the UCS, using the following cmdlet:

```
Get-UcsPrivilege
```

User Roles

Add a user role “`test_role`” with admin privileges, using the following cmdlet:

```
Add-UcsRole -Name user_role -Priv admin
```

Change privileges for a user role to allow read-and-write access to fabric interconnect infrastructure, network security operations and read access to the rest of the system, using the following cmdlet:

```
Get-UcsRole -Name user_role | Set-UcsRole -Priv ls-network
```

Set multiple privileges using Set-UcsRole, using the following cmdlet:

```
Get-UcsRole -Name multi_priv_role | Set-UcsRole -Priv "ls-network", "ls-qos"
```

Get all user roles in UCS, using the following cmdlet:

```
Get-UcsRole
```

Get a user role by name, using the following cmdlet:

```
Get-UcsRole -Name multi_priv_role
```

Remove a user role, using the following cmdlet:

```
Get-UcsRole -Name multi_priv_role | Remove-UcsRole
```

Locales

Add a Locale, using the following cmdlet:

```
Add-UcsLocale -Name asia_pacific -Descr "Locale for Asia Pacific users"
```

Get all locales, using the following cmdlet:

```
Get-UcsLocale
```

Add an org to a locale, using the following cmdlet:

```
Get-UcsLocale -Name asia_pacific | Add-UcsAaaOrg -Name org_asia_pacific -OrgDn org-root/org-Finance
```

Remove a locale, using the following cmdlet:

```
Get-UcsLocale -Name asia_pacific | Remove-UcsLocale
```

User Accounts

Add a local user, using the following cmdlet:

```
$user = Add-UcsLocalUser -Name jdoe -Pwd Passw0rdJdoe
```

Edit a local user, using the following cmdlet:

```
$user | Set-UcsLocalUser -FirstName John -Lastname Doe
```

Add to a user, using the following cmdlet:

```
$user | Add-UcsUserRole -Name user_role
```

Remove a local user, using the following cmdlet:


```
Get-UcsLocalUser -Name jdoe | Remove-UcsLocalUser
```

Remote Authentication - RADIUS

Set global configuration for RADIUS authentication, using the following cmdlet:

```
Set-UcsRadiusGlobalConfig -Descr "RADIUS authentication configuration" -Timeout 20 -Retries 3 -Force
```

Create a RADIUS server instance with server key “test1234” and maximum 2 retries, using the following cmdlet:

```
Add-UcsRadiusProvider -Name "192.168.23.84" -Key test1234 -Retries 2
```

Set RADIUS as default authentication, using the following cmdlet:

```
Set-UcsDefaultAuth -Realm radius
```

Remote Authentication - TACACS

Set global configuration for TACACS authentication, using the following cmdlet:

```
Set-UcsTacacsGlobalConfig -Descr "TACACS authentication configuration" -Timeout 20 -Retries 3
```

Add a TACACS Provider, using the following cmdlet:

```
Add-UcsTacacsProvider -Name "192.168.23.84" -Key test1234
```

Set TACACS as default authentication, using the following cmdlet:

```
Get-UcsNativeAuth | Set-UcsNativeAuth -DefLogin tacacs
```

Remote Authentication - LDAP

Set global configuration for LDAP authentication, using the following cmdlet:

```
Set-UcsLdapGlobalConfig -Descr 'LDAP authentication configuration' -Timeout 20 -Retries 3 -Force
```

Add a LDAP Provider, using the following cmdlet:

```
add-UcsLdapProvider -Attribute 'CiscoAVPair' -Basedn 'CN=Users,DC=qasamLab,DC=com' -FilterValue 'cn=$userid' -Key 'Bbv03515' -Name '10.193.23.84' -Rootdn 'CN=Administrator,CN=Users,DC=qasamLab,DC=com'
```

Set LDAP as default authentication, using the following cmdlet:

```
Get-UcsNativeAuth | Set-UcsNativeAuth -DefLogin ldap
```

RADIUS Provider

Create a RADIUS server instance with server key "test1234" and maximum 2 retries.

```
Add-UcsRadiusProvider -Name "192.168.23.84" -Key test1234 -Retries 2
```

Add a RADIUS provider group and set it as the default remote authentication.

```
Get-UcsRadiusGlobalConfig | Add-UcsProviderGroup -Name radiusprovidergroup1
Get-UcsProviderGroup -Name radiusprovidergroup1 | Add-UcsProviderReference -Name
"192.168.23.84"
Get-UcsNativeAuth | Set-UcsNativeAuth -DefLogin radius
Get-UcsDefaultAuth | Set-UcsDefaultAuth -ProviderGroup radiusprovidergroup1
```

TACACS Provider

Add a TACACS provider, using the following cmdlet:

```
Add-UcsTacacsProvider -Name "192.168.23.84" -Key test1234
```

Add a TACACS provider group and set it as the default remote authentication, using the following cmdlet:

```
Get-UcsTacacsGlobalConfig | Add-UcsProviderGroup -Name tacacsprovidergroup1
Get-UcsProviderGroup -Name tacacsprovidergroup1 | Add-UcsProviderReference -Name
"192.168.23.84"
Get-UcsNativeAuth | Set-UcsNativeAuth -DefLogin tacacs
Get-UcsDefaultAuth | Set-UcsDefaultAuth -ProviderGroup tacacsprovidergroup
```

LDAP Provider

Add an LDAP provider, using the following cmdlet:

```
add-UcsLdapProvider -Attribute 'CiscoAVPair' -Basedn
'CN=Users,DC=qasamLab,DC=com' -FilterValue 'cn=$userid' -Key 'Bbv03515'
-Name '192.168.23.84' -Rootdn 'CN=Administrator,CN=Users,DC=qasamLab,DC=com'
```

Add an LDAP provider group and set it as the default remote authentication, using the following cmdlet:

```
Get-UcsLdapGlobalConfig | Add-UcsProviderGroup -Name ldapprovidergroup1
Get-UcsProviderGroup -Name ldapprovidergroup1 | Add-UcsProviderReference -Name "192.168.23.84"
Get-UcsNativeAuth | Set-UcsNativeAuth -DefLogin ldap
Get-UcsDefaultAuth | Set-UcsDefaultAuth -ProviderGroup ldapprovidergroup1
```

Authentication Domains

Authentication Domains configure simultaneous support for different authentication methods (local, TACACS+, RADIUS, and LDAP/Active Directory) and provider groups.

Configure a TACAS Provider Group with a TACACS Provider.

```
$tp = Add-UcsTacacsProvider -Name "192.168.23.84" -Key test1234
$tpg = Get-UcsTacacsGlobalConfig | Add-UcsProviderGroup -Name tacacs_pg $tpg |
Add-UcsProviderReference -Name $tp.Name
```

Create an Authentication Domain and add a reference to the TACACS Provider group.

```
$ad = Add-UcsAuthDomain -Name adtacacs
$ad | Get-UcsAuthDomainDefaultAuth | Set-UcsAuthDomainDefaultAuth -Realm
tacacs-ProviderGroup tacacs_pg
```

Now if a user logs in from the console, web UI or XML API with the user name being "ucs-adtacacs\user" the TACACS configuration created above will be used for authentication.

Communication Services

Get UCS Web Session Limits, which define the maximum number of concurrent web sessions (both web UI and xml) permitted access to the system at any one time, using the following cmdlet:

```
Get-UcsWebSessionLimit
```

Set Web Session Limit for the user to 30 and an overall session limit of 255, using the following cmdlet:

```
Set-UcsWebSessionLimit -SessionsPerUser 30 -TotalSessions 255
```

Communication Services - Telnet

Get UCS telnet configuration.

```
Get-UcsTelnet
```

Allow telnet connections.

```
Set-UcsTelnet -AdminState enabled -Descr "Telnet configuration for UCS"
```

Communication Services - CIM XML

Get UCS CIM XML configuration, using the following cmdlet:

```
Get-UcsCimXml
```

Enable the CIM XML service, using the following cmdlet:

```
Set-UcsCimXml -AdminState enabled
```

Communication Services - SNMP

Get UCS SNMP configuration, using the following cmdlet:

```
Get-UcsSnmp
```

Enable SNMP with community string being “public”, system contact person being “CiscoSystems” and location of the host being “Bangalore”, using the following cmdlet:

```
Set-UcsSnmp -Descr "SNMP config for UCS" -AdminState enabled -SysContact CiscoSystems
-SysLocation Bangalore -Community public
```

Get a UCS SNMP user, using the following cmdlet:

```
Get-UcsSnmpUser
```

Add a UCS SNMP user, using the following cmdlet:

```
Add-UcsSnmpUser -Name joe -Auth md5 -Privpwd Joe@Cisco -Pwd Joe@Cisco -UseAes true
```

Set a UCS SNMP user, using the following cmdlet:

```
Get-UcsSnmpUser -Name joe | Set-UcsSnmpUser -Auth sha -UseAes false
```

Remove a UCS SNMP user, using the following cmdlet:

```
Get-UcsSnmpUser -Name joe | Remove-UcsSnmpUser
```

Get a UCS SNMP trap, using the following cmdlet:

```
Get-UcsSnmpTrap
```

Add a UCS SNMP trap, using the following cmdlet:

```
Add-UcsSnmpTrap -Hostname 168.65.120.32 -Community public -NotificationType traps -Port 162
-V3Privilege noauth -Version v3
```

Set UCS SNMP configuration, using the following cmdlet:

```
Get-UcsSnmpTrap -Hostname 168.65.120.32 | Set-UcsSnmpTrap -Community public -NotificationType
informs -Port 162 -V3Privilege auth -Version v1
```

Remove UCS SNMP configuration, using the following cmdlet:

```
Get-UcsSnmpTrap -Hostname 168.65.120.32 | Remove-UcsSnmpTrap
```

Communication Services - HTTP

Get UCS http configuration, using the following cmdlet:

```
Get-UcsHttp
```

Set UCS http configuration to enable http and enable http to https redirection, using the following cmdlet:

```
Set-UcsHttp -AdminState enabled -RedirectState enabled
```



```
SIB3DQEBAQUAA4IBDwAwggEKAoIBAQC4eSjyX6J/I1ZSswSFxu+NmEW0BE0IOEEX/
zPMJ/yxh/SJKsgybicPAr0SRzgDKRrHEIoIsMSMxiGTFpErMgF4tkT32HNUeL1b5M
N+e/lcx3M7oqGfDWUOMBfVp9qMCTkpn7cPAnOEoYaCx4J79XQJ6RyX1+uI1qAiCh
tz1jPwnTvtzNGTAcP/opZYwtJ0f5iY6ERNQ8WKJke56oulzUhcq40y3oKX/ilGfkI
IG8GT26Yv6a+KPKdRDSO+q+GZSqmIcghETPYThCt3CWDO7AYxRyQtNnGDzN1OEEd
YaCQhcbz0d8qfognpsWIMARzgyC2HWAN9suZ0zO3NGrFKkeg6ep7AgMBAAGjggKv
MIICQzAfBgNVHREBAf8EFTATggtzYXZidS10cGkwMYcEckF4JTAdBgNVHQ4EFgQU
ns86LcentpqeJmT8140jcfYt2DQwHwYDVR0jBBgwFoAURt5dr7Fdbv3VU6VJiHPT
DbOK//gwggEVBgNVHR8EggEMMIIBCDCCAQSgggEAOIH9hoG8bGRhcDovLy9DTj1V
Q1NNLENOPWJscilzYW0tcWEtYWFMScxDTj1DRFAsQ049UHvibG1jJTIwS2V5JTIw
U2VydmljZXMsQ049U2VydmljZXMsQ049Q29uZmlndXJhdGlvbixEQz11Y3NtLERD
PXhlc2FtLWxhYixEQz1pbj9jZXJ0aWZpY2F0ZVJldm9jYXRpb25MaXNOPE2Jhc2U/
b2JqZWNOQ2xhc3M9Y1JMRGlzdHJpYnV0aW9uUG9pbnsSGPGhdHA6Ly9ibHItc2Ft
LXFhLWFhYTEudWNzbS5xYXNhbnN1S1sYWLuaW4vQ2VydEVucm9sbC9VQ1NNLmNybDCC
AS0GCCsGAQUFBwEBBIBHhCCARswgawGCCsGAQUFBzAChOGfbGRhcDovLy9DTj1V
Q1NNLENOPUFJQSxDTj1QdWJsaWMM1MjBLZXk1MjBTZXJ2aWN1cyxDTj1TZXJ2aWN1
cyxDTj1Db25maWdlcmF0aW9uLERDPXVjc20sREM9cWFzYW0tbGFjLERDPWluP2NB
Q2VydGlmawNhdGU/YmFzZT9vYmplY3RDbGFzc21jZXJ0aWZpY2F0aW9uQXV0aG9y
aXR5MGoGCCsGAQUFBzACh15odHRwOi8vYmxyLXNhbnN1S1sYSL1hYWExLnVjc20ucWFz
YW0tbGFjLmLuL0NlcnRfbnJvbGwvYmxyLXNhbnN1S1sYSL1hYWExLnVjc20ucWFzYW0t
bGFjLmLuL1VDU00uY3J0MA0GCSqGSIb3DQEBAQUAA4IBAQ01hNPFBrDqfu9hrI1E
o6Y9GghHNZ4cxwPlhz0U9w4iskWNVHlw7Ijdf7U+WUvulGWcylN73i2r2sOeQqy3
Isx/2dKS4n3YX7x1hYpMubPCCL1fHIPqQwh9dd1HyKftxqMd6/jQJyhLNOX5yz4h
HpORf14xGGWYsv1Jjqqr2jREbV3kE/uOq0NNi+2efWS0YHq9SESKqulcXgM15LyC
ZKQYo1UseboYK90XgLC2yww+75UcgynLZRxbgAPstNeqPTWh12kWoogrO4zkpo18Y
Vz2yB2BA6/ugCbtJuIw352HzZHU9FM4Y7R0r9k75CNjA9wScu56hX2rfIFUwnSMT
gWvg -----END CERTIFICATE----- "
```

Set the keyring for https.

```
Get-UcsHttps | Set-UcsHttps -KeyRing keyring1024 -AdminState enabled
```

Access UCS Manager through https should now give no "untrusted connection" message.

Generic Managed Object Queries

Get Managed Object of a specific DN, using the following cmdlet:

```
Get-UcsManagedObject -Dn "sys/chassis-1"
```

Get all Managed Objects of a particular class, using the following cmdlet:

```
Get-UcsManagedObject -ClassId faultInst
```

Get DN's of Managed Objects of a particular class, using the following cmdlet:

```
Get-UcsManagedObject -ClassId faultInst -DnList
```

Get names of all Service Profiles from org-root, using the following cmdlet:

```
Get-UcsOrg -Level root | Get-UcsManagedObject -ClassId lsServer | Select Name
```

Get immediate children of org-root, using the following cmdlet:

```
Get-UcsOrg -Level root | Get-UcsChild
```

Get parent of a Managed Object, using the following cmdlet:

```
Get-UcsOrg -Name Finance | Get-UcsParent
```

Generic Managed Object Cmdlets

Create a VLAN using parent object.

```
$propMap = @{Name = "lan_cloud_vlan"; Id = 500} Get-UcsLanCloud | Add-UcsManagedObject
-ClassId FabricVlan -PropertyMap $propMap
```

Create a VLAN using parent object, modify if already existing.

```
$propMap = @{Name = "lan_cloud_vlan"; Id = 500}
Add-UcsManagedObject -ClassId FabricVlan -PropertyMap $propMap -Parent (Get-UcsLanCloud)
-ModifyPresent
```

Create a VLAN using DN.

```
$propMap = @{Dn = "fabric/lan/net-lan_cloud_vlan"; Name = "lan_cloud_vlan"; Id = 500}
Add-UcsManagedObject -ClassId FabricVlan -PropertyMap $propMap
```

Modify a VLAN using Managed Object.

```
$vlan = Get-UcsVlan -Name 'lan_cloud_vlan' $propMap = @{DefaultNet = "yes"; Id = 501; Sharing
= "primary"}
Set-UcsManagedObject -PropertyMap $propMap -ManagedObject $vlan
```

Modify a VLAN using DN.

```
$propMap = @{Dn = "fabric/lan/net-lan_cloud_vlan"; DefaultNet = "yes"; Id = 501; Sharing =
"primary"}
Set-UcsManagedObject -PropertyMap $propMap -ClassId FabricVlan
```

Remove a Managed Object.

```
Get-UcsOrg -Name Finance | Remove-UcsManagedObject
```

Generic Cmdlet -XmlTag

The `XmlTag` parameter enables us to work with unknown Managed Objects.

Create a multicast policy.

```
Add-UcsManagedObject -XmlTag fabricMulticastPolicy -PropertyMap
@{Dn="org-root/mc-policy-multicastpolicy";
Name="multicastpolicy"; PolicyOwner="local"; SnoopingState="enabled";
QuerierState="disabled"};
```

Set snooping state to disabled for multicast policy.

```
Set-UcsManagedObject -XmlTag fabricMulticastPolicy -PropertyMap @{Dn =
"org-root/mc-policy-multicastpolicy"; SnoopingState="disabled"};
```

Upload Firmware

Upload an image to the Default Ucs system.

```
Send-UcsFirmware -LiteralPath C:\work\Images\ucs-k9-bundle-b-series.2.0.1q.B.bin
```

Export to XML

Export the configuration of a Managed Object.

This cmdlet exports the configuration of the managed object and the entire hierarchy.

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

Export the xml of a Managed Object into a file.

```
Get-UcsServiceProfile -Name sp_name | Export-UcsMoXml | Out-File c:\mo.xml
```

Import from XML

Import the configuration from the XML file, using the following cmdlet:

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

Import xml of a Managed Object and convert it into objects, using the following cmdlet:

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

KVM

Start KVM session for service profile, blade, and a rack server.

Syntax

```
Start-UcsKvmSession -ServiceProfile <LsServer> [-FrameTitle <string>] [-IPv4Addresses] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Start-UcsKvmSession -Blade <ComputeBlade> [-FrameTitle <string>] [-IPv4Addresses] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Start-UcsKvmSession -RackUnit <ComputeRackUnit> [-FrameTitle <string>] [-IPv4Addresses]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```


To launch KVM session for a blade, service profile or rack unit, enter the following

```
Get-UcsBlade -SlotId 4 | Start-UcsKvmSession
Get-UcsServiceProfile -Name testSP | Start-UcsKvmSession
Get-UcsRackUnit | Start-UcsKvmSession
```

If there are multiple in-band and out-of-band IP addresses configured in the management interface, you are prompted to select the IP address from which the KVM should be launched.

Example

```
Get-UcsBlade -SlotId 4 | Start-UcsKvmSession
```

KVM Launch

Choose an IP address:

```
[1] 1.10.65.183.39 [2] 2.10.65.183.43 [3] 3.2001::105 [4] 4.10.65.183.14 [?] Help
(default is "1"): ?
1 - Ucs:10.65.183.8, IPAddress:10.65.183.39, IpSource:out-band, AccessType:Equipment,
EqDn:sys/chassis-1/blade-4, SpDn:org-root/ls-ssp99
2 - Ucs:10.65.183.8, IPAddress:10.65.183.43, IpSource:in-band, AccessType:Equipment,
EqDn:sys/chassis-1/blade-4, SpDn:org-root/ls-ssp99
3 - Ucs:10.65.183.8, IPAddress:2001::105, IpSource:in-band, AccessType:Equipment,
EqDn:sys/chassis-1/blade-4, SpDn:org-root/ls-ssp99
4 - Ucs:10.65.183.8, IPAddress:10.65.183.14, IpSource:out-band, AccessType:ServiceProfile,
EqDn:sys/chassis-1/blade-4, SpDn:org-root/ls-ssp99
[1] 1.10.65.183.39 [2] 2.10.65.183.43 [3] 3.2001::105 [4] 4.10.65.183.14 [?] Help
(default is "1"):2
```

Launch the UCS Manager Java web UI

Connect to UCS Manager and launch the UCS Manager web UI, using the following cmdlet:

```
Start-UcsGuiSession
```

Enable secure Logging.

Some XML transactions are treated as secure and the UCS Manager web UI does not log them. The LogAllXml flag enables secure logging

```
Start-UcsGuiSession -LogAllXml
```

Launch the UCS Manager web UI using the Get-UcsStatus and Start-UcsGuiSession cmdlets.

```
Get-UcsStatus | Start-UcsGuiSession
```

Launch the UCS Manager web UI without Connecting to UCS Manager, using the following cmdlet:

```
Start-UcsGuiSession -Name <IP Address or Hostname of UCSM>
```

Store the Credentials in a Variable and Pass it to a cmdlet

```
$password = "<Password>" | ConvertTo-SecureString -AsPlainText -Force
```

```
$cred = New-Object System.Management.Automation.PSCredential("UserName", $password)
Start-UcsGuiSession -Name <IP Address or Hostname of UCSM> -Credential $cred
```

Launching the Cisco UCS Manager HTML GUI

New switch parameter added to the Start-UcsGuiSession to launch the HTML GUI of UCS Manager. By default, Start-UcsGuiSession cmdlet launches the Java GUI.

From Cisco UCS Manager 3.1(2) release, automatic login to HTML GUI is allowed. The Start-UcsGuiSession cmdlet is enhanced for automatic login. It also supports the context-based login. For example, if you want to launch the HTML GUI for a particular entity, such as service profile, policies or pools, and so on you can pass the required MO to the **Start-UcsGuiSession** cmdlet. This automatically launches the page of the passed managed object.



Note From Cisco UCS Manager 3.1(2) release, automatic sign is enabled using token for HTML GUI.

Syntax

The following syntax is for automatic login to HTML GUI:

```
Start-UcsGuiSession -HTML [-ManagedObject <ManagedObject>] [-Ucs <UcsHandle[]>]
[<CommonParameters>]
```

Example

```
Get-UcsBlade -SlotId 1 | Start-UcsGuiSession -HTML
```

UCS Statistics

Get current Ucs statistics for Chassis Id 1 and Slot Id 1, using the following cmdlet:

```
Get-UcsBlade -ChassisId 1 -SlotId 1 | Get-UcsStatistics -Current
```

Get Ucs statistics history for Chassis Id 1 and Slot Id 1, using the following cmdlet:

```
Get-UcsBlade -ChassisId 1 -SlotId 1 | Get-UcsStatistics -History
```

Clear Ucs statistics, using the following cmdlet:

```
Get-UcsManagedObject -Dn sys/chassis-1/blade-1/board/temp-stats | Clear-UcsStatistics
```

Configure Scalability Port in UCS 6324 Fabric Interconnect

Configure the breakout port 1/5/1 in UCS 6324 Fabric Interconnect B as a server port

```
$mo = Add-UcsManagedObject -XmlTag fabricSubGroup -PropertyMap
@{dn="fabric/server/sw-B/slot-1-aggr-port-5";aggrPortId="5";slotId="1"}
```

```
$mo | Add-UcsManagedObject -XmlTag fabricDceSwSrvEp -PropertyMap
@{rn="slot-1-port-1";portId="1";slotId="1"}
```

Configure the breakout port 1/5/1 in UCS 6324 Fabric Interconnect B as an FCoE storage port

```
$mo = Add-UcsManagedObject -modifyPresent -XmlTag fabricSubGroup -PropertyMap
@{dn="fabric/fc-estc/B/slot-1-aggr-port-5";aggrPortId="5";slotId="1"}
$mo | Add-UcsManagedObject -XmlTag fabricFcoeEstcEp -PropertyMap
@{rn="phys-fcoe-slot-1-port-1";portId="1";slotId="1"}
```

Configure breakout port 1/5/1 in UCS 6324 Fabric Interconnect B appliance port

```
$mo = Add-UcsManagedObject -modifyPresent -XmlTag fabricSubGroup -PropertyMap
@{dn="fabric/eth-estc/B/slot-1-aggr-port-5";aggrPortId="5";slotId="1"}
$mo | Add-UcsManagedObject -XmlTag fabricEthEstcEp -PropertyMap
@{rn="phys-eth-slot-1-port-1";portId="1";slotId="1"}
```

Transaction Impact

Get-UcsTransactionImpact cmdlet estimates the impact of a pending transaction. The cmdlet uses the ConfigEstimateImpact method and returns a UcsImpact object. A message that is similar to the message provided by UCS Manager web UI is provided as part of the UcsImpact object.

Start a transaction.

```
Start-UcsTransaction
```

Create a service profile.

```
$sp = Add-UcsServiceProfile -Name sp_name
```

Create a vNIC.

```
$eth0 = $sp | Add-UcsVnic -Name eth0 -IdentPoolName empty_pool
```

Add a VLAN for vNIC, make it Native VLAN.

```
$eth0 | Add-UcsVnicInterface -Name primary -DefaultNet true
```

Get Transaction Impact.

```
Get-UcsTransactionImpact
```

A UcsImpact object is returned, which indicates a config-failure for the service profile that would be created, and etc.

Cmdlet Meta Information

Get Meta information about all Managed Object mapped cmdlets.

```
Get-UcsCmdletMeta
```

Get Meta information about LsServer mapped cmdlets.

```
Get-UcsCmdletMeta -ClassId LsServer
```

View the hierarchy of the ServiceProfile (LsServer) class.

```
Get-UcsCmdletMeta -Noun UcsServiceProfile -Tree
```

Get Meta information for the UcsServiceProfile noun.

```
Get-UcsCmdletMeta -Noun UcsServiceProfile
```

See the Managed Object information for LsServer.

```
Get-UcsCmdletMeta -ClassId LsServer | Select -ExpandProperty MoMeta
```

See the Managed Object property information for LsServer.

```
Get-UcsCmdletMeta -ClassId LsServer | Select -ExpandProperty MoMeta | Select -ExpandProperty
PropertyMeta
```

Compare-UcsManagedObject - Dn Translation

Create a service profile under org A. Assume that orgs A & B are in place already.

```
$org = Get-UcsOrg -Name A -LimitScope
$destOrg = Get-UcsOrg -Name B -LimitScope
$sp = Add-UcsServiceProfile -Org $org -Name abc
```

Create a translation map with DNs of entities that needs to be translated.

```
$xlateDn = @{ }
$xlateDn['org-root/org-A/ls-abc'] = 'org-root/org-B/ls-xyz'
```

Combine the translation map with Compare-UcsMo to see the changes required.

```
Compare-UcsManagedObject (Get-UcsServiceProfile -Org $destOrg -Name xyz -LimitScope)
(Get-UcsServiceProfile -Org $org -Name abc -LimitScope) -XlateMap $xlateDn
```

Combine the translation org with Compare to see the changes required.

```
Compare-UcsManagedObject (Get-UcsServiceProfile -Org $destOrg -Name xyz -LimitScope)
(Get-UcsServiceProfile -Org $org -Name abc -LimitScope) -XlateOrg org-root/org-B
```

Sync a service profile from org A to org B while renaming it.

```
Sync-UcsManagedObject (Compare-UcsManagedObject (Get-UcsServiceProfile -Org $destOrg -Name
xyz -LimitScope)
(Get-UcsServiceProfile -Org $org -Name abc -LimitScope) -XlateMap $xlateDn) -Force | select
Dn
```

Compare-UcsManagedObject - GetPropertyDiff()

Use GetPropertyDiff() function on output of Compare- UcsManagedObject to see the difference in properties.

```
$sp1 = Get-UcsServiceProfile -Dn org-root/ls-abc
$sp2 = $sp1 | Set-UcsServiceProfile -Descr 'GetPropertyDiff Example' -Force
$diff = Compare-UcsManagedObject $sp1 $sp2
```

Display all the properties having difference. If \$diff is an array of objects, then GetPropertyDiff works on \$diff[<index>]

```
$diff.GetPropertyDiff()
```

For a specific property \$diff.

```
GetPropertyDiff('descr')
```

Include all operational properties of MOs in comparison

```
Compare-UcsManagedObject $sp1 $sp2 -IncludeOperational
```

Add Cmdlet -ModifyPresent Flag

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

Create a csv file with Name, Id pairs.

```
$("Name,Id"; foreach ($vlan in 501..510) { "vlan${vlan}, ${vlan}" }) | Out-File c:\vlans.csv
```

Import the Name, Vlan pairs from the file and create those vlans.

```
$lc = Get-UcsLanCloud
Start-UcsTransaction
Import-Csv C:\vlans.csv | % { $lc | Add-UcsVlan -ModifyPresent -Name $_.Name -Id $_.Id }
Complete-UcsTransaction
```

Edit the csv file to edit the ids or add new vlans. Re-running the same Add-UcsVlan snippet above results in an error, if existing VLANs created again (with or without changes). Invoking Add-UcsVlan with the ModifyPresent option, addresses this by modifying the VLANs instead, if they already exist.

```
$lc = Get-UcsLanCloud
Start-UcsTransaction
Import-Csv C:\vlans.csv | % { $lc | Add-UcsVlan -ModifyPresent -Name $_.Name -Id $_.Id }
Complete-UcsTransaction
```

Capability Catalog Update

The capability catalog is a set of tunable parameters, strings, and rules. Cisco UCS uses the catalog to update the display and ability to configure the components, such as newly qualified DIMMs and disk drives for servers.

To update the capability catalog from a local file source you can use the following cmdlet:

```
Update-UcsCatalogue -LiteralPath C:\Work\ucs-catalog.2.2.3a.T.bin
```

Server Operations

Added the following new simplified cmdlets to perform server operations:

Action Description	Cmdlets in PowerTool Release 1.4.1 or earlier	Cmdlet in PowerTool Release 1.5.1 and later
Acknowledge UCS Server	Get-UcsChassis -Id 1 Get-UcsBlade -SlotId 1 Set-UcsBlade -AdminPower "policy" -Lc "rediscover" -PolicyOwner "local"	Get-UcsServer where { \$_.Dn -eq "sys/chassis-1/blade-1" } Confirm-UcsServer
Decommission UCS Server	Get-UcsChassis -Id 1 Get-UcsBlade -SlotId 1 Set-UcsBlade -AdminPower "policy" -Lc "decommission" -PolicyOwner "local"	Get-UcsServer where { \$_.Dn -eq "sys/chassis-1/blade-1" } Disable-UcsServer
Hard Reset UCS Server	Get-UcsOrg -Level root Get-UcsServiceProfile -Name "testSP" -LimitScope Get-UcsServerPower Set-UcsServerPower -State "hard-reset-immediate"	Get-UcsServiceProfile -name testSP Reset-UcsServer
Booting UCS Server	Get-UcsOrg -Level root Get-UcsServiceProfile -Name "testSP" -LimitScope Get-UcsServerPower Set-UcsServerPower -State "admin-up"	Get-UcsServiceProfile -name testSP Start-UcsServer
Shutting down UCS Server	Get-UcsOrg -Level root Get-UcsServiceProfile -Name "testSP" -LimitScope Get-UcsServerPower Set-UcsServerPower -State "soft-shut-down"	Get-UcsServiceProfile -name testSP Stop-UcsServer
Power Cycling UCS Server	Get-UcsOrg -Level root Get-UcsServiceProfile -Name "testSP" -LimitScope Get-UcsServerPower Set-UcsServerPower -State "cycle-immediate"	Get-UcsServiceProfile -name testSP Restart-UcsServer

Action Description	Cmdlets in PowerTool Release 1.4.1 or earlier	Cmdlet in PowerTool Release 1.5.1 and later
Resetting CMOS for a UCS Server	Get-UcsOrg -Level root Get-UcsServiceProfile -Name "testSP" -LimitScope Get-UcsServerPower Set-UcsServerPower -State "cmos-reset-immediate"	Get-UcsServiceProfile -name testSP Reset-UcsServerCmos
Resetting BMC for a UCS Server	Get-UcsOrg -Level root Get-UcsServiceProfile -Name "testSP" -LimitScope Get-UcsServerPower Set-UcsServerPower -State "bmc-reset-immediate"	Get-UcsServiceProfile -name testSP Reset-UcsServerBmc
Turn On Locator LED for a UCS Server	Get-UcsChassis -Id 1 Get-UcsBlade -SlotId 1 Get-UcsLocatorLed Set-UcsLocatorLed -AdminState "on" -BoardType "single" -Id 1	Get-UcsServer where { \$_.Dn -eq "sys/chassis-1/blade-1" } Enable-UcsLocatorLed
Turn Off Locator LED for a UCS Server	Get-UcsChassis -Id 1 Get-UcsBlade -SlotId 1 Get-UcsLocatorLed Set-UcsLocatorLed -AdminState "off" -BoardType "single" -Id 1	Get-UcsServer where { \$_.Dn -eq "sys/chassis-1/blade-1" } Disable-UcsLocatorLed

32 Parameter Set Limitation

According to the Microsoft PowerShell framework, cmdlets cannot have more than 32 Parameter Sets. If the number of Parameter Sets for a cmdlet is more than 32, the cmdlet may not be able to identify the Parameters or Parameter Sets correctly, and may behave abnormally.

The following PowerTool cmdlets are affected by this limitation:

- Get-UcsEquipmentFruVariant
- Get-UcsEquipmentManufacturingDef
- Get-UcsEquipmentPhysicalDef
- Get-UcsEquipmentPicture
- Get-UcsEquipmentServiceDef
- Get-UcsEquipmentSlotArrayRef
- Get-UcsFirmwareUpgradeConstraint

After a logical grouping of Parameter Sets is made, each cmdlet is split into two cmdlets. For each existing cmdlet, storage-related Parameter Sets are removed from the original cmdlet and made into a new cmdlet. This provides the following additional storage-related PowerTool cmdlets:

- Get-UcsEquipmentFruVariantStorage
- Get-UcsEquipmentManufacturingDefStorage
- Get-UcsEquipmentPhysicalDefStorage
- Get-UcsEquipmentPictureStorage
- Get-UcsEquipmentServiceDefStorage
- Get-UcsEquipmentSlotArrayRefStorage
- Get-UcsFirmwareUpgradeConstraintStorage

Because storage-specific Parameter Sets now have their own cmdlets, backward compatibility may break while using the original cmdlets for storage-specific Parameter Sets.

Cisco recommends that you use the new storage cmdlets for the specified storage-related Parameter Sets.

The following sections provide detailed syntax for these cmdlets.

Get-UcsEquipmentFruVariant

This cmdlet is used to get information about "EquipmentFruVariant" type of managed object. This managed object is used to establish the mapping between the FRU variant and the PID.

```
Get-UcsEquipmentFruVariant [-Type <string>] [-Description <string>] [-Dn <string>] [-Pid <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentFruVariant -AdaptorFruCapProvider <AdaptorFruCapProvider> [-Type <string>] [-Description <string>] [-Dn <string>] [-Pid <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentFruVariant -DiagSrvCapProvider <DiagSrvCapProvider> [-Type <string>] [-Description <string>] [-Dn <string>] [-Pid <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentFruVariant -EquipmentBaseBoardCapProvider <EquipmentBaseBoardCapProvider> [-Type <string>] [-Description <string>] [-Dn <string>] [-Pid <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentFruVariant -EquipmentBladeBiosCapProvider <EquipmentBladeBiosCapProvider> [-Type <string>] [-Description <string>] [-Dn <string>] [-Pid <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentFruVariant -EquipmentBladeCapProvider <EquipmentBladeCapProvider> [-Type <string>] [-Description <string>] [-Dn <string>] [-Pid <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentFruVariant -EquipmentCatalogCapProvider <EquipmentCatalogCapProvider> [-Type <string>] [-Description <string>] [-Dn <string>] [-Pid <string>] [-Sacl {addchild |
```



```
[ -Pid <string>] [ -Sacl {addchild | cascade | del | mod | none}] [ -Hierarchy] [ -Filter <string>] [ -XtraProperty <hashtable>] [ -Ucs <UcsHandle[]>] [ -Xml] [ <CommonParameters>]
```

```
Get-UcsEquipmentFruVariantStorage -EquipmentStorageNvmeSwitchCapProvider <EquipmentStorageNvmeSwitchCapProvider> [ -Type <string>] [ -Description <string>] [ -Dn <string>] [ -Pid <string>] [ -Sacl {addchild | cascade | del | mod | none}] [ -Hierarchy] [ -Filter <string>] [ -XtraProperty <hashtable>] [ -Ucs <UcsHandle[]>] [ -Xml] [ <CommonParameters>]
```

```
Get-UcsEquipmentFruVariantStorage -EquipmentStorageSasExpanderCapProvider <EquipmentStorageSasExpanderCapProvider> [ -Type <string>] [ -Description <string>] [ -Dn <string>] [ -Pid <string>] [ -Sacl {addchild | cascade | del | mod | none}] [ -Hierarchy] [ -Filter <string>] [ -XtraProperty <hashtable>] [ -Ucs <UcsHandle[]>] [ -Xml] [ <CommonParameters>]
```

Get-UcsEquipmentManufacturingDef

This cmdlet is used to get information about "EquipmentManufacturingDef" type of managed object. This managed object is used to store manufacturing-related properties such as PID and SKU.

```
Get-UcsEquipmentManufacturingDef [ -Caption <string>] [ -Clei <string>] [ -Descr <string>] [ -Description <string>] [ -Dn <string>] [ -FruMajorType <string>] [ -FruMinorType <string>] [ -IsSecFwUpdate {false | no | true | yes}] [ -Name <string>] [ -OemName <string>] [ -OemPartNumber <string>] [ -PartNumber <string>] [ -Pid <string>] [ -PolicyLevel <uint32>] [ -PolicyOwner {local | pending-policy | policy}] [ -Sacl {addchild | cascade | del | mod | none}] [ -Series <string>] [ -Sku <string>] [ -VendorEquipmentType <string>] [ -Vid <string>] [ -Hierarchy] [ -Filter <string>] [ -XtraProperty <hashtable>] [ -Ucs <UcsHandle[]>] [ -Xml] [ <CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDef -AdaptorFruCapProvider <AdaptorFruCapProvider> [ -Caption <string>] [ -Clei <string>] [ -Descr <string>] [ -Description <string>] [ -Dn <string>] [ -FruMajorType <string>] [ -FruMinorType <string>] [ -IsSecFwUpdate {false | no | true | yes}] [ -Name <string>] [ -OemName <string>] [ -OemPartNumber <string>] [ -PartNumber <string>] [ -Pid <string>] [ -PolicyLevel <uint32>] [ -PolicyOwner {local | pending-policy | policy}] [ -Sacl {addchild | cascade | del | mod | none}] [ -Series <string>] [ -Sku <string>] [ -VendorEquipmentType <string>] [ -Vid <string>] [ -Hierarchy] [ -Filter <string>] [ -XtraProperty <hashtable>] [ -Ucs <UcsHandle[]>] [ -Xml] [ <CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDef -DiagSrvCapProvider <DiagSrvCapProvider> [ -Caption <string>] [ -Clei <string>] [ -Descr <string>] [ -Description <string>] [ -Dn <string>] [ -FruMajorType <string>] [ -FruMinorType <string>] [ -IsSecFwUpdate {false | no | true | yes}] [ -Name <string>] [ -OemName <string>] [ -OemPartNumber <string>] [ -PartNumber <string>] [ -Pid <string>] [ -PolicyLevel <uint32>] [ -PolicyOwner {local | pending-policy | policy}] [ -Sacl {addchild | cascade | del | mod | none}] [ -Series <string>] [ -Sku <string>] [ -VendorEquipmentType <string>] [ -Vid <string>] [ -Hierarchy] [ -Filter <string>] [ -XtraProperty <hashtable>] [ -Ucs <UcsHandle[]>] [ -Xml] [ <CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDef -EquipmentBaseBoardCapProvider <EquipmentBaseBoardCapProvider> [ -Caption <string>] [ -Clei <string>] [ -Descr <string>] [ -Description <string>] [ -Dn <string>] [ -FruMajorType <string>] [ -FruMinorType <string>] [ -IsSecFwUpdate {false | no | true | yes}] [ -Name <string>] [ -OemName <string>] [ -OemPartNumber <string>] [ -PartNumber <string>] [ -Pid <string>] [ -PolicyLevel <uint32>] [ -PolicyOwner {local | pending-policy | policy}] [ -Sacl {addchild | cascade | del | mod | none}] [ -Series <string>] [ -Sku <string>] [ -VendorEquipmentType <string>] [ -Vid <string>] [ -Hierarchy] [ -Filter <string>] [ -XtraProperty <hashtable>] [ -Ucs <UcsHandle[]>] [ -Xml] [ <CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDef -EquipmentBladeBiosCapProvider <EquipmentBladeBiosCapProvider> [ -Caption <string>] [ -Clei <string>] [ -Descr <string>] [ -Description <string>] [ -Dn <string>] [ -FruMajorType <string>] [ -FruMinorType <string>] [ -IsSecFwUpdate {false | no | true | yes}] [ -Name <string>] [ -OemName <string>] [ -OemPartNumber <string>] [ -PartNumber <string>] [ -Pid <string>] [ -PolicyLevel <uint32>] [ -PolicyOwner {local | pending-policy | policy}] [ -Sacl {addchild | cascade | del | mod | none}] [ -Series <string>] [ -Sku <string>] [ -VendorEquipmentType <string>] [ -Vid <string>] [ -Hierarchy] [ -Filter <string>] [ -XtraProperty <hashtable>] [ -Ucs <UcsHandle[]>] [ -Xml] [ <CommonParameters>]
```



```
Get-UcsEquipmentManufacturingDef -EquipmentTpmCapProvider <EquipmentTpmCapProvider>
[-Caption <string>] [-Clei <string>] [-Descr <string>] [-Description <string>] [-Dn <string>]
[-FruMajorType <string>] [-FruMinorType <string>] [-IsSecFwUpdate {false | no | true |
yes}] [-Name <string>] [-OemName <string>] [-OemPartNumber <string>] [-PartNumber <string>]
[-Pid <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy | policy}]
[-Sacl {addchild | cascade | del | mod | none}] [-Series <string>] [-Sku <string>]
[-VendorEquipmentType <string>] [-Vid <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

Get-UcsEquipmentManufacturingDefStorage

```
Get-UcsEquipmentManufacturingDefStorage [-Caption <string>] [-Clei <string>] [-Descr <string>]
[-Description <string>] [-Dn <string>] [-FruMajorType <string>] [-FruMinorType <string>]
[-IsSecFwUpdate {false | no | true | yes}] [-Name <string>] [-OemName <string>]
[-OemPartNumber <string>] [-PartNumber <string>] [-Pid <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-Sacl {addchild | cascade | del | mod |
none}] [-Series <string>] [-Sku <string>] [-VendorEquipmentType <string>] [-Vid <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
<CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDefStorage -EquipmentLocalDiskCapProvider
<EquipmentLocalDiskCapProvider> [-Caption <string>] [-Clei <string>] [-Descr <string>]
[-Description <string>] [-Dn <string>] [-FruMajorType <string>] [-FruMinorType <string>]
[-IsSecFwUpdate {false | no | true | yes}] [-Name <string>] [-OemName <string>]
[-OemPartNumber <string>] [-PartNumber <string>] [-Pid <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-Sacl {addchild | cascade | del | mod |
none}] [-Series <string>] [-Sku <string>] [-VendorEquipmentType <string>] [-Vid <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
<CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDefStorage -EquipmentLocalDiskControllerCapProvider
<EquipmentLocalDiskControllerCapProvider> [-Caption <string>] [-Clei <string>] [-Descr
<string>] [-Description <string>] [-Dn <string>] [-FruMajorType <string>] [-FruMinorType
<string>] [-IsSecFwUpdate {false | no | true | yes}] [-Name <string>] [-OemName <string>]
[-OemPartNumber <string>] [-PartNumber <string>] [-Pid <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-Sacl {addchild | cascade | del | mod |
none}] [-Series <string>] [-Sku <string>] [-VendorEquipmentType <string>] [-Vid <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
<CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDefStorage -EquipmentMemoryUnitCapProvider
<EquipmentMemoryUnitCapProvider> [-Caption <string>] [-Clei <string>] [-Descr <string>]
[-Description <string>] [-Dn <string>] [-FruMajorType <string>] [-FruMinorType <string>]
[-IsSecFwUpdate {false | no | true | yes}] [-Name <string>] [-OemName <string>]
[-OemPartNumber <string>] [-PartNumber <string>] [-Pid <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-Sacl {addchild | cascade | del | mod |
none}] [-Series <string>] [-Sku <string>] [-VendorEquipmentType <string>] [-Vid <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
<CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDefStorage -EquipmentMiniStorageCapProvider
<EquipmentMiniStorageCapProvider> [-Caption <string>] [-Clei <string>] [-Descr <string>]
[-Description <string>] [-Dn <string>] [-FruMajorType <string>] [-FruMinorType <string>]
[-IsSecFwUpdate {false | no | true | yes}] [-Name <string>] [-OemName <string>]
[-OemPartNumber <string>] [-PartNumber <string>] [-Pid <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-Sacl {addchild | cascade | del | mod |
none}] [-Series <string>] [-Sku <string>] [-VendorEquipmentType <string>] [-Vid <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
<CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDefStorage -EquipmentStorageEncCapProvider
<EquipmentStorageEncCapProvider> [-Caption <string>] [-Clei <string>] [-Descr <string>]
```

```
[-Description <string>] [-Dn <string>] [-FruMajorType <string>] [-FruMinorType <string>]
[-IsSecFwUpdate {false | no | true | yes}] [-Name <string>] [-OemName <string>]
[-OemPartNumber <string>] [-PartNumber <string>] [-Pid <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-Sacl {addchild | cascade | del | mod |
none}] [-Series <string>] [-Sku <string>] [-VendorEquipmentType <string>] [-Vid <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDefStorage -EquipmentStorageNvmeSwitchCapProvider
<EquipmentStorageNvmeSwitchCapProvider> [-Caption <string>] [-Clei <string>] [-Descr <string>]
[-Description <string>] [-Dn <string>] [-FruMajorType <string>] [-FruMinorType <string>]
[-IsSecFwUpdate {false | no | true | yes}] [-Name <string>] [-OemName <string>]
[-OemPartNumber <string>] [-PartNumber <string>] [-Pid <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-Sacl {addchild | cascade | del | mod |
none}] [-Series <string>] [-Sku <string>] [-VendorEquipmentType <string>] [-Vid <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]
```

```
Get-UcsEquipmentManufacturingDefStorage -EquipmentStorageSasExpanderCapProvider
<EquipmentStorageSasExpanderCapProvider> [-Caption <string>] [-Clei <string>] [-Descr
<string>] [-Description <string>] [-Dn <string>] [-FruMajorType <string>] [-FruMinorType
<string>] [-IsSecFwUpdate {false | no | true | yes}] [-Name <string>] [-OemName <string>]
[-OemPartNumber <string>] [-PartNumber <string>] [-Pid <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-Sacl {addchild | cascade | del | mod |
none}] [-Series <string>] [-Sku <string>] [-VendorEquipmentType <string>] [-Vid <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]
```

Get-UcsEquipmentPhysicalDef

This cmdlet is used to get information about "EquipmentPhysicalDef" type of managed object. This managed object is used to store physical properties such as maximum temperature and removal conditions.

```
Get-UcsEquipmentPhysicalDef [-Depth <string>] [-Descr <string>] [-Dn <string>] [-Height
<string>] [-MaximumPower <string>] [-MaximumTemperature <string>] [-MinimumPower <string>]
[-MinimumTemperature <string>] [-Name <string>] [-NominalPower <string>] [-NominalTemperature
<string>] [-OperatingVoltages <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local |
pending-policy | policy}] [-Sacl {addchild | cascade | del | mod | none}] [-Weight <string>]
[-Width <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPhysicalDef -AdaptorFruCapProvider <AdaptorFruCapProvider> [-Depth
<string>] [-Descr <string>] [-Dn <string>] [-Height <string>] [-MaximumPower <string>]
[-MaximumTemperature <string>] [-MinimumPower <string>] [-MinimumTemperature <string>]
[-Name <string>] [-NominalPower <string>] [-NominalTemperature <string>] [-OperatingVoltages
<string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy | policy}] [-Sacl
{addchild | cascade | del | mod | none}] [-Weight <string>] [-Width <string>] [-Hierarchy]
[-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]
```

```
Get-UcsEquipmentPhysicalDef -DiagSrvCapProvider <DiagSrvCapProvider> [-Depth <string>]
[-Descr <string>] [-Dn <string>] [-Height <string>] [-MaximumPower <string>]
[-MaximumTemperature <string>] [-MinimumPower <string>] [-MinimumTemperature <string>]
[-Name <string>] [-NominalPower <string>] [-NominalTemperature <string>] [-OperatingVoltages
<string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy | policy}] [-Sacl
{addchild | cascade | del | mod | none}] [-Weight <string>] [-Width <string>] [-Hierarchy]
[-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]
```

```
Get-UcsEquipmentPhysicalDef -EquipmentBaseBoardCapProvider <EquipmentBaseBoardCapProvider>
[-Depth <string>] [-Descr <string>] [-Dn <string>] [-Height <string>] [-MaximumPower
<string>] [-MaximumTemperature <string>] [-MinimumPower <string>] [-MinimumTemperature
<string>] [-Name <string>] [-NominalPower <string>] [-NominalTemperature <string>]
```



```
[-Width <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPhysicalDefStorage -EquipmentStorageNvmeSwitchCapProvider
<EquipmentStorageNvmeSwitchCapProvider> [-Depth <string>] [-Descr <string>] [-Dn <string>]
[-Height <string>] [-MaximumPower <string>] [-MaximumTemperature <string>] [-MinimumPower
<string>] [-MinimumTemperature <string>] [-Name <string>] [-NominalPower <string>]
[-NominalTemperature <string>] [-OperatingVoltages <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-Sacl {addchild | cascade | del | mod |
none}] [-Weight <string>] [-Width <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPhysicalDefStorage -EquipmentStorageSasExpanderCapProvider
<EquipmentStorageSasExpanderCapProvider> [-Depth <string>] [-Descr <string>] [-Dn <string>]
[-Height <string>] [-MaximumPower <string>] [-MaximumTemperature <string>] [-MinimumPower
<string>] [-MinimumTemperature <string>] [-Name <string>] [-NominalPower <string>]
[-NominalTemperature <string>] [-OperatingVoltages <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-Sacl {addchild | cascade | del | mod |
none}] [-Weight <string>] [-Width <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

Get-UcsEquipmentPicture

This cmdlet is used to get information about "EquipmentPicture" type of managed object. This managed object is used to store image file detail, type and layout in Cisco UCS Manager.

```
Get-UcsEquipmentPicture [-Type {back | bottom | front | front-bottom-scaled | front-top-scaled
| left | right | top | top-scaled | unknown}] [-Dn <string>] [-FileName <string>] [-Sacl
{addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -AdaptorFruCapProvider <AdaptorFruCapProvider> [-Type {back |
bottom | front | front-bottom-scaled | front-top-scaled | left | right | top | top-scaled
| unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade | del | mod |
none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -DiagSrvCapProvider <DiagSrvCapProvider> [-Type {back | bottom
| front | front-bottom-scaled | front-top-scaled | left | right | top | top-scaled |
unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade | del | mod |
none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -EquipmentBaseBoardCapProvider <EquipmentBaseBoardCapProvider>
[-Type {back | bottom | front | front-bottom-scaled | front-top-scaled | left | right |
top | top-scaled | unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -EquipmentBladeBiosCapProvider <EquipmentBladeBiosCapProvider>
[-Type {back | bottom | front | front-bottom-scaled | front-top-scaled | left | right |
top | top-scaled | unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -EquipmentBladeCapProvider <EquipmentBladeCapProvider> [-Type
{back | bottom | front | front-bottom-scaled | front-top-scaled | left | right | top |
top-scaled | unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade | del
| mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -EquipmentCatalogCapProvider <EquipmentCatalogCapProvider>
[-Type {back | bottom | front | front-bottom-scaled | front-top-scaled | left | right | top
```



```
Get-UcsEquipmentPicture -EquipmentMgmtExtCapProvider <EquipmentMgmtExtCapProvider>
[-Type {back | bottom | front | front-bottom-scaled | front-top-scaled | left | right | top
| top-scaled | unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade |
del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -EquipmentProcessorUnitCapProvider
<EquipmentProcessorUnitCapProvider> [-Type {back | bottom | front | front-bottom-scaled |
front-top-scaled | left | right | top | top-scaled | unknown}] [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -PsuCapProvider <EquipmentPsuCapProvider> [-Type {back | bottom
| front | front-bottom-scaled | front-top-scaled | left | right | top | top-scaled |
unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade | del | mod |
none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -RackUnitCapProvider <EquipmentRackUnitCapProvider> [-Type {back
| bottom | front | front-bottom-scaled | front-top-scaled | left | right | top | top-scaled
| unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade | del | mod |
none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -EquipmentSecurityUnitCapProvider
<EquipmentSecurityUnitCapProvider> [-Type {back | bottom | front | front-bottom-scaled |
front-top-scaled | left | right | top | top-scaled | unknown}] [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -EquipmentServerUnitCapProvider <EquipmentServerUnitCapProvider>
[-Type {back | bottom | front | front-bottom-scaled | front-top-scaled | left | right |
top | top-scaled | unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -EquipmentSwitchCapProvider <EquipmentSwitchCapProvider> [-Type
{back | bottom | front | front-bottom-scaled | front-top-scaled | left | right | top |
top-scaled | unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade | del
| mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -EquipmentSwitchIOCardCapProvider
<EquipmentSwitchIOCardCapProvider> [-Type {back | bottom | front | front-bottom-scaled |
front-top-scaled | left | right | top | top-scaled | unknown}] [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPicture -EquipmentTpmCapProvider <EquipmentTpmCapProvider> [-Type {back
| bottom | front | front-bottom-scaled | front-top-scaled | left | right | top | top-scaled
| unknown}] [-Dn <string>] [-FileName <string>] [-Sacl {addchild | cascade | del | mod |
none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]
```

Get-UcsEquipmentPictureStorage

```
Get-UcsEquipmentPictureStorage [-Type {back | bottom | front | front-bottom-scaled |
front-top-scaled | left | right | top | top-scaled | unknown}] [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentPictureStorage -EquipmentLocalDiskCapProvider
<EquipmentLocalDiskCapProvider> [-Type {back | bottom | front | front-bottom-scaled |
```

```

front-top-scaled | left | right | top | top-scaled | unknown}} [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}} [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentPictureStorage -EquipmentLocalDiskControllerCapProvider
<EquipmentLocalDiskControllerCapProvider> [-Type {back | bottom | front | front-bottom-scaled
| front-top-scaled | left | right | top | top-scaled | unknown}} [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}} [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentPictureStorage -EquipmentMemoryUnitCapProvider
<EquipmentMemoryUnitCapProvider> [-Type {back | bottom | front | front-bottom-scaled |
front-top-scaled | left | right | top | top-scaled | unknown}} [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}} [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentPictureStorage -EquipmentMiniStorageCapProvider
<EquipmentMiniStorageCapProvider> [-Type {back | bottom | front | front-bottom-scaled |
front-top-scaled | left | right | top | top-scaled | unknown}} [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}} [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentPictureStorage -EquipmentStorageEncCapProvider
<EquipmentStorageEncCapProvider> [-Type {back | bottom | front | front-bottom-scaled |
front-top-scaled | left | right | top | top-scaled | unknown}} [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}} [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentPictureStorage -EquipmentStorageNvmeSwitchCapProvider
<EquipmentStorageNvmeSwitchCapProvider> [-Type {back | bottom | front | front-bottom-scaled
| front-top-scaled | left | right | top | top-scaled | unknown}} [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}} [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentPictureStorage -EquipmentStorageSasExpanderCapProvider
<EquipmentStorageSasExpanderCapProvider> [-Type {back | bottom | front | front-bottom-scaled
| front-top-scaled | left | right | top | top-scaled | unknown}} [-Dn <string>] [-FileName
<string>] [-Sacl {addchild | cascade | del | mod | none}} [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

```

Get-UcsEquipmentServiceDef

This cmdlet is used to get information about "EquipmentServiceDef" type of managed object. This managed object is used to store service properties such as removal conditions and slot array descriptor name.

```

Get-UcsEquipmentServiceDef [-CanBeFRUed {false | no | true | yes}] [-Descr <string>] [-Dn
<string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy |
policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable when on or
off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}} [-ServicePhilosophy <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

    Get-UcsEquipmentServiceDef -AdaptorFruCapProvider <AdaptorFruCapProvider> [-CanBeFRUed
{false | no | true | yes}] [-Descr <string>] [-Dn <string>] [-Name <string>] [-PolicyLevel
<uint32>] [-PolicyOwner {local | pending-policy | policy}} [-RemovalConditions {Not
Applicable | Removable when off | Removable when on or off | Unknown}} [-Sacl {addchild |
cascade | del | mod | none}} [-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentServiceDef -DiagSrvCapProvider <DiagSrvCapProvider> [-CanBeFRUed {false
| no | true | yes}] [-Descr <string>] [-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}} [-RemovalConditions {Not Applicable |
Removable when off | Removable when on or off | Unknown}} [-Sacl {addchild | cascade | del

```



```

[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentServiceDef -EquipmentGraphicsCardCapProvider
<EquipmentGraphicsCardCapProvider> [-CanBeFRUed {false | no | true | yes}] [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}] [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}] [-Sacl {addchild | cascade | del | mod | none}] [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]

    Get-UcsEquipmentServiceDef -EquipmentHostIfCapProvider <EquipmentHostIfCapProvider>
[-CanBeFRUed {false | no | true | yes}] [-Descr <string>] [-Dn <string>] [-Name <string>]
[-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy | policy}] [-RemovalConditions
{Not Applicable | Removable when off | Removable when on or off | Unknown}] [-Sacl {addchild
| cascade | del | mod | none}] [-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentServiceDef -EquipmentIOCardCapProvider <EquipmentIOCardCapProvider>
[-CanBeFRUed {false | no | true | yes}] [-Descr <string>] [-Dn <string>] [-Name <string>]
[-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy | policy}] [-RemovalConditions
{Not Applicable | Removable when off | Removable when on or off | Unknown}] [-Sacl {addchild
| cascade | del | mod | none}] [-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentServiceDef -EquipmentIOExpanderCapProvider
<EquipmentIOExpanderCapProvider> [-CanBeFRUed {false | no | true | yes}] [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}] [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}] [-Sacl {addchild | cascade | del | mod | none}] [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]

    Get-UcsEquipmentServiceDef -EquipmentMgmtCapProvider <EquipmentMgmtCapProvider>
[-CanBeFRUed {false | no | true | yes}] [-Descr <string>] [-Dn <string>] [-Name <string>]
[-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy | policy}] [-RemovalConditions
{Not Applicable | Removable when off | Removable when on or off | Unknown}] [-Sacl {addchild
| cascade | del | mod | none}] [-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentServiceDef -EquipmentMgmtExtCapProvider <EquipmentMgmtExtCapProvider>
[-CanBeFRUed {false | no | true | yes}] [-Descr <string>] [-Dn <string>] [-Name <string>]
[-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy | policy}] [-RemovalConditions
{Not Applicable | Removable when off | Removable when on or off | Unknown}] [-Sacl {addchild
| cascade | del | mod | none}] [-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentServiceDef -EquipmentProcessorUnitCapProvider
<EquipmentProcessorUnitCapProvider> [-CanBeFRUed {false | no | true | yes}] [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}] [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}] [-Sacl {addchild | cascade | del | mod | none}] [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]

    Get-UcsEquipmentServiceDef -PsuCapProvider <EquipmentPsuCapProvider> [-CanBeFRUed {false
| no | true | yes}] [-Descr <string>] [-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>]
[-PolicyOwner {local | pending-policy | policy}] [-RemovalConditions {Not Applicable |
Removable when off | Removable when on or off | Unknown}] [-Sacl {addchild | cascade | del
| mod | none}] [-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsEquipmentServiceDef -RackUnitCapProvider <EquipmentRackUnitCapProvider>
[-CanBeFRUed {false | no | true | yes}] [-Descr <string>] [-Dn <string>] [-Name <string>]
[-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy | policy}] [-RemovalConditions

```

```
{Not Applicable | Removable when off | Removable when on or off | Unknown}} [-Sacl {addchild
| cascade | del | mod | none}} [-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentServiceDef -EquipmentSecurityUnitCapProvider
<EquipmentSecurityUnitCapProvider> [-CanBeFRUed {false | no | true | yes}} [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}} [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentServiceDef -EquipmentServerUnitCapProvider
<EquipmentServerUnitCapProvider> [-CanBeFRUed {false | no | true | yes}} [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}} [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentServiceDef -EquipmentSwitchCapProvider <EquipmentSwitchCapProvider>
[-CanBeFRUed {false | no | true | yes}} [-Descr <string>] [-Dn <string>] [-Name <string>]
[-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy | policy}} [-RemovalConditions
{Not Applicable | Removable when off | Removable when on or off | Unknown}} [-Sacl {addchild
| cascade | del | mod | none}} [-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentServiceDef -EquipmentSwitchIOCardCapProvider
<EquipmentSwitchIOCardCapProvider> [-CanBeFRUed {false | no | true | yes}} [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}} [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentServiceDef -EquipmentTpmCapProvider <EquipmentTpmCapProvider> [-CanBeFRUed
{false | no | true | yes}} [-Descr <string>] [-Dn <string>] [-Name <string>] [-PolicyLevel
<uint32>] [-PolicyOwner {local | pending-policy | policy}} [-RemovalConditions {Not
Applicable | Removable when off | Removable when on or off | Unknown}} [-Sacl {addchild |
cascade | del | mod | none}} [-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>]
[-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

Get-UcsEquipmentServiceDefStorage

```
Get-UcsEquipmentServiceDefStorage [-CanBeFRUed {false | no | true | yes}} [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}} [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentServiceDefStorage -EquipmentLocalDiskCapProvider
<EquipmentLocalDiskCapProvider> [-CanBeFRUed {false | no | true | yes}} [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}} [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentServiceDefStorage -EquipmentLocalDiskControllerCapProvider
<EquipmentLocalDiskControllerCapProvider> [-CanBeFRUed {false | no | true | yes}} [-Descr
<string>] [-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local |
pending-policy | policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable
```

```

when on or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}}
[-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsEquipmentServiceDefStorage -EquipmentMemoryUnitCapProvider
<EquipmentMemoryUnitCapProvider> [-CanBeFRUed {false | no | true | yes}} [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}} [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]

Get-UcsEquipmentServiceDefStorage -EquipmentMiniStorageCapProvider
<EquipmentMiniStorageCapProvider> [-CanBeFRUed {false | no | true | yes}} [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}} [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]

Get-UcsEquipmentServiceDefStorage -EquipmentStorageEncCapProvider
<EquipmentStorageEncCapProvider> [-CanBeFRUed {false | no | true | yes}} [-Descr <string>]
[-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable when on
or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}} [-ServicePhilosophy
<string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>]
[-Xml] [<CommonParameters>]

Get-UcsEquipmentServiceDefStorage -EquipmentStorageNvmeSwitchCapProvider
<EquipmentStorageNvmeSwitchCapProvider> [-CanBeFRUed {false | no | true | yes}} [-Descr
<string>] [-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local |
pending-policy | policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable
when on or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}}
[-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsEquipmentServiceDefStorage -EquipmentStorageSasExpanderCapProvider
<EquipmentStorageSasExpanderCapProvider> [-CanBeFRUed {false | no | true | yes}} [-Descr
<string>] [-Dn <string>] [-Name <string>] [-PolicyLevel <uint32>] [-PolicyOwner {local |
pending-policy | policy}} [-RemovalConditions {Not Applicable | Removable when off | Removable
when on or off | Unknown}} [-Sacl {addchild | cascade | del | mod | none}}
[-ServicePhilosophy <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

```

Get-UcsEquipmentSlotArrayRef

The cmdlet is used to get information about "EquipmentSlotArrayRef" type of managed object. This managed object contains reference from a FRU to a Slot Array. Also defines the slot span of the FRU.

```

Get-UcsEquipmentSlotArrayRef [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-Sacl {addchild | cascade | del | mod | none}} [-SlotSpanOrientation {inline
| transverse | unknown}} [-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsEquipmentSlotArrayRef -AdaptorFruCapProvider <AdaptorFruCapProvider> [-Name
<string>] [-Descr <string>] [-Dn <string>] [-NumberOfSlotsSpanned <uint16>] [-PolicyLevel
<uint32>] [-PolicyOwner {local | pending-policy | policy}} [-Sacl {addchild | cascade | del
| mod | none}} [-SlotSpanOrientation {inline | transverse | unknown}} [-TargetDn <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
<CommonParameters>]

Get-UcsEquipmentSlotArrayRef -DiagSrvCapProvider <DiagSrvCapProvider> [-Name <string>]

```



```
del | mod | none}} [-SlotSpanOrientation {inline | transverse | unknown}} [-TargetDn <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]
```

```
Get-UcsEquipmentSlotArrayRef -EquipmentSecurityUnitCapProvider
<EquipmentSecurityUnitCapProvider> [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-Sacl {addchild | cascade | del | mod | none}} [-SlotSpanOrientation {inline
| transverse | unknown}} [-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentSlotArrayRef -EquipmentServerUnitCapProvider
<EquipmentServerUnitCapProvider> [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-Sacl {addchild | cascade | del | mod | none}} [-SlotSpanOrientation {inline
| transverse | unknown}} [-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentSlotArrayRef -EquipmentSwitchCapProvider <EquipmentSwitchCapProvider>
[-Name <string>] [-Descr <string>] [-Dn <string>] [-NumberOfSlotsSpanned <uint16>]
[-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy | policy}} [-Sacl {addchild
| cascade | del | mod | none}} [-SlotSpanOrientation {inline | transverse | unknown}}
[-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentSlotArrayRef -EquipmentSwitchIOCardCapProvider
<EquipmentSwitchIOCardCapProvider> [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-Sacl {addchild | cascade | del | mod | none}} [-SlotSpanOrientation {inline
| transverse | unknown}} [-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentSlotArrayRef -EquipmentTpmCapProvider <EquipmentTpmCapProvider> [-Name
<string>] [-Descr <string>] [-Dn <string>] [-NumberOfSlotsSpanned <uint16>] [-PolicyLevel
<uint32>] [-PolicyOwner {local | pending-policy | policy}} [-Sacl {addchild | cascade |
del | mod | none}} [-SlotSpanOrientation {inline | transverse | unknown}} [-TargetDn <string>]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]
```

Get-UcsEquipmentSlotArrayRefStorage

```
Get-UcsEquipmentSlotArrayRefStorage [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-Sacl {addchild | cascade | del | mod | none}} [-SlotSpanOrientation {inline
| transverse | unknown}} [-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentSlotArrayRefStorage -EquipmentLocalDiskCapProvider
<EquipmentLocalDiskCapProvider> [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}} [-Sacl {addchild | cascade | del | mod | none}} [-SlotSpanOrientation {inline
| transverse | unknown}} [-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

```
Get-UcsEquipmentSlotArrayRefStorage -EquipmentLocalDiskControllerCapProvider
<EquipmentLocalDiskControllerCapProvider> [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local |
pending-policy | policy}} [-Sacl {addchild | cascade | del | mod | none}}
[-SlotSpanOrientation {inline | transverse | unknown}} [-TargetDn <string>] [-Hierarchy]
[-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]
```

```
Get-UcsEquipmentSlotArrayRefStorage -EquipmentMemoryUnitCapProvider
```

```

<EquipmentMemoryUnitCapProvider> [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}] [-Sacl {addchild | cascade | del | mod | none}] [-SlotSpanOrientation {inline
| transverse | unknown}] [-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsEquipmentSlotArrayRefStorage -EquipmentMiniStorageCapProvider
<EquipmentMiniStorageCapProvider> [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}] [-Sacl {addchild | cascade | del | mod | none}] [-SlotSpanOrientation {inline
| transverse | unknown}] [-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsEquipmentSlotArrayRefStorage -EquipmentStorageEncCapProvider
<EquipmentStorageEncCapProvider> [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}] [-Sacl {addchild | cascade | del | mod | none}] [-SlotSpanOrientation {inline
| transverse | unknown}] [-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsEquipmentSlotArrayRefStorage -EquipmentStorageNvmeSwitchCapProvider
<EquipmentStorageNvmeSwitchCapProvider> [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local | pending-policy
| policy}] [-Sacl {addchild | cascade | del | mod | none}] [-SlotSpanOrientation {inline
| transverse | unknown}] [-TargetDn <string>] [-Hierarchy] [-Filter <string>] [-XtraProperty
<hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsEquipmentSlotArrayRefStorage -EquipmentStorageSasExpanderCapProvider
<EquipmentStorageSasExpanderCapProvider> [-Name <string>] [-Descr <string>] [-Dn <string>]
[-NumberOfSlotsSpanned <uint16>] [-PolicyLevel <uint32>] [-PolicyOwner {local |
pending-policy | policy}] [-Sacl {addchild | cascade | del | mod | none}]
[-SlotSpanOrientation {inline | transverse | unknown}] [-TargetDn <string>] [-Hierarchy]
[-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

```

Get-UcsFirmwareUpgradeConstraint

This cmdlet is used to get information about "FirmwareUpgradeConstraint" type of managed object. This managed object is used to define a firmware constraint for an upgrade operation.

```

Get-UcsFirmwareUpgradeConstraint [-Dn <string>] [-MinVer <string>] [-Sacl {addchild |
cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -AdaptorFruCapProvider <AdaptorFruCapProvider> [-Dn
<string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy]
[-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -DiagSrvCapProvider <DiagSrvCapProvider> [-Dn <string>]
[-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter
<string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentBaseBoardCapProvider
<EquipmentBaseBoardCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentBladeBiosCapProvider
<EquipmentBladeBiosCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]

```



```

Get-UcsFirmwareUpgradeConstraint -EquipmentBladeCapProvider <EquipmentBladeCapProvider>
[-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentCatalogCapProvider
<EquipmentCatalogCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -ChassisCapProvider <EquipmentChassisCapProvider> [-Dn
<string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy]
[-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentDbgPluginCapProvider
<EquipmentDbgPluginCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -FanModuleCapProvider <EquipmentFanModuleCapProvider>
[-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -FexCapProvider <EquipmentFexCapProvider> [-Dn <string>]
[-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter
<string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentGemCapProvider <EquipmentGemCapProvider>
[-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy]
[-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentGraphicsCardCapProvider
<EquipmentGraphicsCardCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild |
cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentHostIfCapProvider <EquipmentHostIfCapProvider>
[-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentIOCardCapProvider <EquipmentIOCardCapProvider>
[-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentIOExpanderCapProvider
<EquipmentIOExpanderCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentMgmtCapProvider <EquipmentMgmtCapProvider>
[-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy]
[-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentMgmtExtCapProvider
<EquipmentMgmtExtCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]

```

```

Get-UcsFirmwareUpgradeConstraint -EquipmentProcessorUnitCapProvider
<EquipmentProcessorUnitCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild |
cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -PsuCapProvider <EquipmentPsuCapProvider> [-Dn <string>]
[-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy] [-Filter
<string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -RackUnitCapProvider <EquipmentRackUnitCapProvider>
[-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy]
[-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentSecurityUnitCapProvider
<EquipmentSecurityUnitCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild |
cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentServerUnitCapProvider
<EquipmentServerUnitCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentSwitchCapProvider <EquipmentSwitchCapProvider>
[-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}]
[-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentSwitchIOCardCapProvider
<EquipmentSwitchIOCardCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild |
cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraint -EquipmentTpmCapProvider <EquipmentTpmCapProvider>
[-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade | del | mod | none}] [-Hierarchy]
[-Filter <string>] [-XtraProperty <hashtable>] [-Ucs <UcsHandle[]>] [-Xml]
[<CommonParameters>]

```

Get-UcsFirmwareUpgradeConstraintStorage

```

Get-UcsFirmwareUpgradeConstraintStorage [-Dn <string>] [-MinVer <string>] [-Sacl {addchild
| cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraintStorage -EquipmentLocalDiskCapProvider
<EquipmentLocalDiskCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraintStorage -EquipmentLocalDiskControllerCapProvider
<EquipmentLocalDiskControllerCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild
| cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraintStorage -EquipmentMemoryUnitCapProvider
<EquipmentMemoryUnitCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]

Get-UcsFirmwareUpgradeConstraintStorage -EquipmentMiniStorageCapProvider
<EquipmentMiniStorageCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild |
cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]

```

```
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsFirmwareUpgradeConstraintStorage -EquipmentStorageEncCapProvider
<EquipmentStorageEncCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild | cascade
| del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>] [-Ucs
<UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsFirmwareUpgradeConstraintStorage -EquipmentStorageNvmeSwitchCapProvider
<EquipmentStorageNvmeSwitchCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild
| cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]

    Get-UcsFirmwareUpgradeConstraintStorage -EquipmentStorageSasExpanderCapProvider
<EquipmentStorageSasExpanderCapProvider> [-Dn <string>] [-MinVer <string>] [-Sacl {addchild
| cascade | del | mod | none}] [-Hierarchy] [-Filter <string>] [-XtraProperty <hashtable>]
[-Ucs <UcsHandle[]>] [-Xml] [<CommonParameters>]
```

