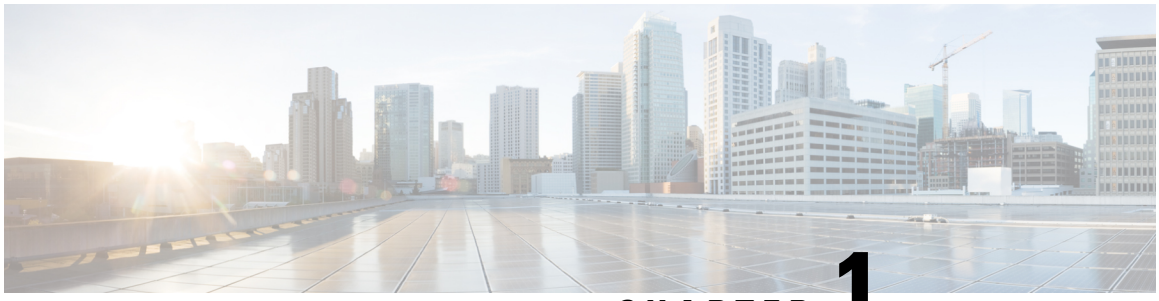
Cisco UCS PowerTool Suite Installation and Configuration Guide, Release 2.x

First Published: 2016-03-01

Last Modified: 2019-01-18

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Preface

This preface includes the following sections:

Audience

This guide is intended primarily for data center administrators with responsibilities and expertise in one or more of the following:

- Server administration
- Storage administration
- Network administration
- Network security

Conventions

This document uses the following conventions:

Conventions	Indication
bold font	Commands an keywords and user-entered text appear in bold font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.

Conventions	Indication
{ x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marts around the string or the string will include the quotation marks.
`courier` font	Terminal sessions and information the system displays appear in `courier` font.
< >	Nonprinting characters, such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!,#	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means reader take a note. Notes contain helpful suggestions or references to material not covered in the manual.

Tip Means the following information will help you solve a problem. The tips information might not be troubleshooting or even an action, buy could be useful information, similar to a Timesaver.

Caution Means reader be careful. In this situation, you might perform an action that could result in equipment damage or loss of data.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Warning IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Related Cisco UCS Documentation

Documentation Roadmaps

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

- Cisco UCS Manager PowerTool Release 2.x User Guide
- Cisco IMC PowerTool, Release 2.x User Guide
- Cisco UCS Central PowerTool, Release 2.x User Guide
- Cisco UCS Documentation Roadmap
- Cisco UCS C-Series Documentation Roadmap
- Cisco UCS Central Configuration Guides

Other Documentation Resources

An ISO file containing all B and C-Series documents is available at the following URL:https://software.cisco.com/download/type.html?mdfid=283853163&flowid=25821 From this page, click **Unified Computing System (UCS) Documentation Roadmap Bundle**.

The ISO file is updated after every major documentation release.

Follow Cisco UCS Docs on Twitter to receive document update notifications.

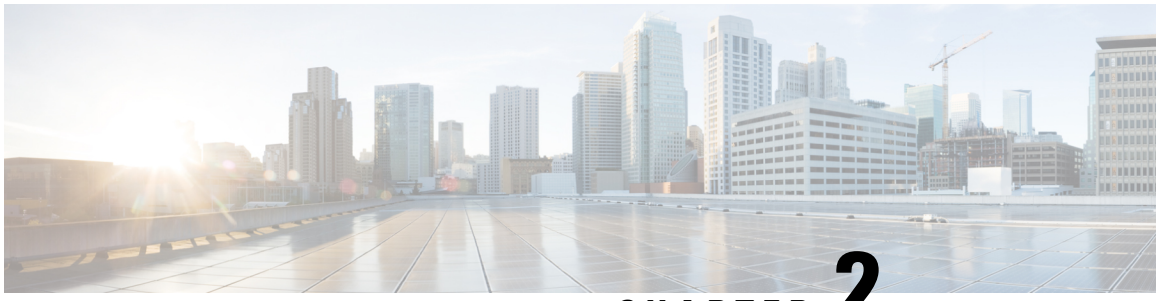
Documentation Feedback

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

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at:http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.



CHAPTER 2

Overview

This chapter consists of the following sections:

About Cisco UCS PowerTool Suite

Cisco UCS PowerTool suite is a PowerShell module that helps automate all aspects of Cisco UCS Manager, Cisco UCS Central, and Cisco IMC. It also helps automate server, network, storage, and hypervisor management. Cisco UCS PowerTool suite enables easy integration with existing IT management processes and tools. The PowerTool cmdlets work on the Cisco UCS Management Information Tree (MIT). These cmdlets can be used to execute read, create, modify, and delete operations on all the UCS Managed Objects (MOs) in the MIT.

System Requirements

Ensure that the system meets the following requirements:

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

Cisco UCS Manager

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

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

- Release 2.1
- Release 2.0

Cisco UCS C-Series Servers

Cisco IMC PowerTool is compatible with the following Cisco UCS C-Series releases:

- Release 4.0
- Release 3.1
- Release 3.0
- Release 2.0
- Release 1.5

Cisco UCS E-Series Servers

Cisco IMC PowerTool is compatible with the following Cisco UCS E-Series releases:

- Release 2.2(1) and higher

Cisco UCS Central

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

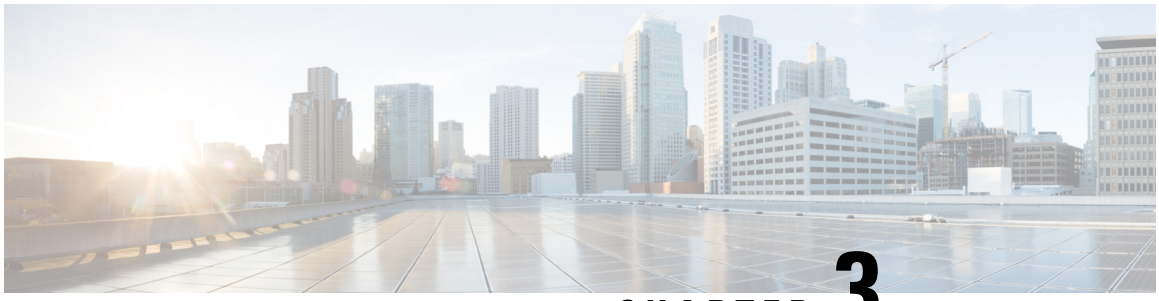
- Release 2.0
- Release 1.5
- Release 1.4
- Release 1.3
- Release 1.2

Installation Requirements

To install or uninstall the Cisco UCS PowerTool Suite on the systems, open a command prompt using Run as Administrator and navigate to the directory where the MSI is located and launch the installer.

Important Upgrade from Release 1.x of Cisco UCS Manager PowerTool, Cisco IMC PowerTool and Cisco UCS Central is not supported. Uninstall 1.x version of PowerTool before installing the Cisco UCS PowerTool Suite.

Close any instances of PowerShell running with the PowerTool module loaded.



CHAPTER 3

Install Cisco UCS PowerTool Suite

This chapter consists of the following sections:

Installing Cisco UCS PowerTool Suite

You can install the Cisco UCS PowerTool suite for all Cisco UCS modules or single module using a unified installer.

Note The default install path for all the modules is `C:\Program Files (x86)\WindowsPowerShell\Modules`. However, you can change the install path using the **Custom Install** option.

Before you begin

Install the PowerTool using admin privileges.

Step 1 On the Cisco.com download site for Cisco UCS Management Partner Ecosystem Software, download the Cisco UCS PowerTool suite.

Step 2 Open a command prompt as Run as Administrator.

Step 3 Navigate to the location where the MSI file is downloaded and enter the MSI name to start the installation.

Note If a PowerShell session is opened when you launch the installer, an error displays stating that the installation or uninstallation of the module is not possible when the Windows PowerShell is running. So, we recommend that you close any PowerShell session running.

Cisco UCS PowerTool Suite - Installer wizard opens.

Step 4 Click **Next**.

Step 5 Review the **EULA** and click **Accept** > **Next**.

Step 6 Select the **Setup Type**.

This can be one of the following:

- **Complete**—Installs Cisco UCS Manager PowerTool, Cisco UCS Central PowerTool, Cisco IMC PowerTool, and Cisco UCS DSC modules
- **Custom**—Allow you to install the modules which you want. Also, you can specify the installation location of the modules

Step 7 Click **Next**.

Step 8 For customized installation, follow these steps:

a) Select the module which you do not want to install, and click **This feature will not be available** from the drop-down list.
b) To change the installation location of the modules, click **Change**.
c) Browse to the location where you want to install and click **OK**.

Step 9 Click **Install**.

If you want to create a desktop shortcut for the modules, click the **Create Desktop Shortcut** option.

Step 10 Click **Finish**.

Upgrade to Cisco UCS PowerTool Suite, Release 2.x

Upgrade from 1.x versions of Cisco UCS Manager PowerTool, Cisco IMC PowerTool, and Cisco UCS Central is not supported. Uninstall 1.x versions of PowerTool before installing the Cisco UCS PowerTool Suite Release 2.x.

When you launch the installer on a system running Cisco UCS PowerTool Suite 2.x version, you are prompted to upgrade the system to Cisco UCS PowerTool Suite latest version. Click **Yes** to upgrade, and then continue with the installation.

For more information about how to install the Cisco UCS PowerTool Suite, see Installing Cisco UCS PowerTool Suite.

Note By default, all the modules are updated to the latest release. Once updated you cannot retain the modules from earlier installation.

Modifying and Repairing Cisco UCS PowerTool Suite Installation

Step 1 Navigate to **Start** > **Control Panel** > **Programs and Features**.

Step 2 Select **Cisco UCS PowerTool Suite** from the list of programs installed.

Step 3 Click **Change**.
The **Cisco UCS PowerTool Suite - Installation Wizard** opens.

Step 4 Click **Next**.

Step 5 To modify the installation, click **Modify**, and follow these steps:

a) Click **Next**.
Custom Setup page appears.

b) Select the module you want to add or remove from the existing installation, and choose the corresponding option from the drop-down list.

Step 6 To repair the installation, click **Repair** > **Next**.

Step 7 Click **Install**.

If you want to create a desktop shortcut for the modules, click the **Create Desktop Shortcut** option.

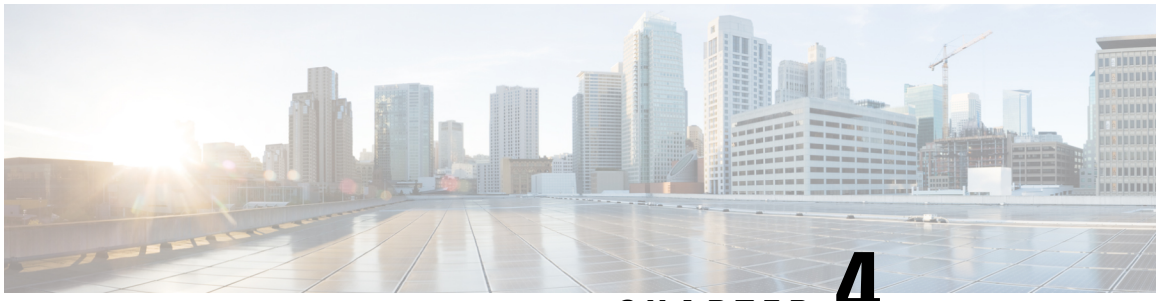
Step 8 Click **Finish**.

Uninstalling Cisco UCS PowerTool Suite

Step 1 Navigate to **Start** > **Control Panel** > **Programs and Features**.

Step 2 Select **Cisco UCS PowerTool Suite** from the list of programs installed.

Step 3 Click **Uninstall**.



CHAPTER 4

Configure Cisco UCS PowerTool Suite

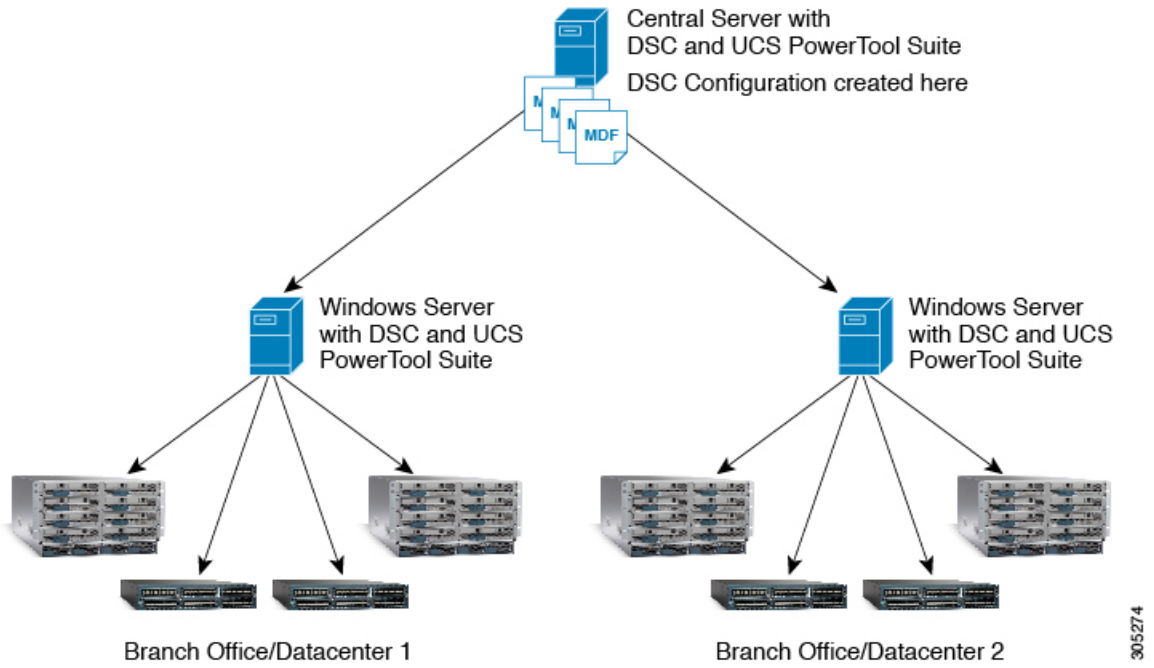
This chapter consists of the following sections:

Cisco UCS Desired State Configuration (DSC) Resource

Desired State Configuration (DSC) is a management platform in Windows PowerShell which enables you to configure, deploy, and manage systems.

Cisco UCS DSC resource enables you to configure Cisco UCS Manager, and Cisco IMC using the Window PowerShell DSC managment platform.

Figure 1: Cisco UCS DSC Solution Architecture



To apply any configuration using Cisco UCS DSC resource, install Cisco UCS PowerTool Suite on Windows server with DSC.

The Cisco UCS DSC architecture is comprised of the following:

- **Central Server**—This server is used to write the UCS DSC configuration scripts
- **Intermediate Server**— Central server deploys the configuration to this server. This server applies the configuration to the Cisco UCS Manager, or Cisco IMC

Cisco UCS Core Module

Cisco UCS core module contains cmdlets for features or actions common to all Cisco UCS modules. A cmdlet replaces the duplicate cmdlets available in the older releases of Cisco UCS Manager, Cisco IMC and Cisco UCS Central PowerTool. The cmdlets in **Retained Cmdlet** column have been retained and it works for all three PowerTool modules.

Retained Cmdlet	Cmdlet Removed in Cisco Central	Cmdlet Removed in Cisco IMC
Get-UcsPowerToolCommunity	Get-UcsCentralPowerToolCommunity	Get-ImcPowerToolCommunity
Get-UcsPowerToolConfiguration	Get-UcsCentralPowerToolConfiguration	Get-ImcPowerToolConfiguration
Set-UcsPowerToolConfiguration	Set-UcsCentralPowerToolConfiguration	Set-ImcPowerToolConfiguration
Get-UcsPSSession	Get-UcsCentralPSSession	Get-ImcPSSession
Export-UcsPSSession	Export-UcsCertralPSSession	Export-ImcPSSession

If you are using cmdlets that are removed from this release in any of the scripts, then use the cmdlets from the **Retained Cmdlet** column instead.

Getting Cisco UCS Software Images

The following cmdlets can be used to get the available driver or firmware images for Cisco UCS Manager, Cisco UCS Central, or Cisco IMC servers from Cisco.com.

- Get-UcsSoftwareImageList—To get all the images available for either UCS Manager, Cisco IMC, or UCS Central
- Get-UcsSoftwareImage—This cmdlet downloads the image to a local file system. This cmdlet takes pipeline input from the Get-UcsSoftwareImageList cmdlet.

Get-UcsSoftwareImageList

Syntax

```
Get-UcsSoftwareImageList -Credential <PSCredential>
[-Category <string>] [-Type <string>] [-Model <string>] [-AllReleases]
[-Proxy <WebProxy>] [-Ucs <BaseHandle[]>] [<CommonParameters>]

Get-UcsSoftwareImageList -Credential <PSCredential>
```

```
-MdfId <uint> -SoftwareId <uint> [-AllReleases] [-Proxy <WebProxy>]
[-Ucs <BaseHandle[]>] [<CommonParameters>]
```

Note *AllReleases* switch parameter is available from UCS PowerTool Suite 2.0(2) and higher. Specifying the *–AllRelease* switch parameter, lists all the available images in Cisco.com for the selected device type.

–Version parameter is valid for UCS PowerTool Suite Release 2.0(1) only.

Get-UcsSoftwareImage

Syntax

```
Get-UcsSoftwareImage -SoftwareImage <ImageDetails> -Path <string> [-EulaAccepted] [-Xml]
[<CommonParameters>]
```

Example

```
Get-UcsSoftwareImageList
Get-UcsSoftwareImageList -Category Default
Get-UcsSoftwareImageList -Category ModularServer
Get-UcsSoftwareImageList -Category ModularServer -Type Firmware
Get-UcsSoftwareImageList -Category ModularServer -Type Drivers
Get-UcsSoftwareImageList  -Type Firmware
Get-UcsSoftwareImageList -Type Drivers
$images= Get-UcsSoftwareImageList
$image[0] | Get-UcsSoftwareImage -Path "C:\Images" -EulaAccepted -verbose
Get-UcsSoftwareImageList  | Get-UcsSoftwareImage -Path "C:\Images" -verbose
```

The following cmdlets which were used for getting software images from Cisco.com are deprecated from this release:

- Get-UcsCcoImageList
- Get-UcsCcoImage

Cisco UCS Hardware and Software Compatibility Tool Integration

Cisco UCS Hardware and Software Compatibility List (HCL) Tool provides interoperability information for UCS components and configurations tested and validated by Cisco, by Cisco partners, or both.

From Release 2.1.1 of UCS PowerTool Suite, custom cmdlets for interacting with the HCL tool are added. With this feature, you can create a Hardware Profile from a UCS blade or rack server and upload it to the HCL tool. Also, you can get the recommended versions of the firmware or driver versions from the HCL tool by selecting the uploaded hardware profile.

Connect to Cisco UCS

Before executing any cmdlet, you should first connect to Cisco UCS using the following:

```
C:\> $cred = Get-Credential
C:\> $ucs = Connect-Ucs -Name <<IP Address>> -Credential $cred
```

Get-UcsOsVendor

The Get-UcsOsVendor cmdlet is used to get the operating system vendor code and name from the HCL tool.

Syntax

```
Get-UcsOsVendor [-Id <string>] [-Proxy <WebProxy>] [<CommonParameters>]
```

Detailed Description

Property	Description	Required
Id	Specifies the OsVendor Id.	False
Proxy	Specifies the web proxy to be used for the communication with HCL api.	False
CommonParameters	This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable.	

Example

```
C:\> $osVendor = Get-UcsOsVendor

C:\> $OSVendor

T_ID ID OSVENDOR
---- -- --------
2    0  Citrix
3    1  Microsoft
6    2  SuSE
4    3  Oracle
5    4  Red Hat
8    5  VMware
7    7  Ubuntu
1    8  CentOS
```

Get-UcsOperatingSystem

The Get-UcsOperatingSystem cmdlet is used to get all the supported operating systems for the selected vendor.

Syntax

```
Parameter Set: Vendor
Get-UcsOperatingSystem -OsVendor <OsVendor> [-Proxy <WebProxy>] [<CommonParameters>]

Parameter Set: Id
Get-UcsOperatingSystem -OsVendorTreeId <string> [-Proxy <WebProxy>] [<CommonParameters>]
```

Detailed Description

Property	Description	Required
OsVendor	Specifies the OsVendor object.	True
OsVendorTreeId	Specifies the tree id of OsVendor.	True

Property	Description	Required
Proxy	Specifies the web proxy to be used for the communication with HCL api.	False
CommonParameters	This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable.	

Example

The following example has Microsoft OS vendor details passed as **`$osVendor[1]`**.

```
C:\> $osVersion = Get-UcsOperatingSystem -OsVendor $osVendor[1]

C:\> $osVersion

T_ID ID  OSVERSION
---- --  ---------
39   141 Windows Server 2008 R2 SP1
41   72  Windows Server 2012
38   338 Windows Server 2008 R2
42   118 Windows Server 2012 R2
40   175 Windows Server 2008 SP2
44   810 Windows Server 2019
43   601 Windows Server 2016
```

New-UcsHardwareProfile

The New-UcsHardwareProfile cmdlet is used to create a hardware profile from the selected server. The hardware profile created using this cmdlet can be uploaded to the HCL tool using the Add-UcsHardwareProfile cmdlet. This cmdlet takes managed object of the type ComputeBlade, ComputeRackUnit, ComputeServerNode, and ServiceProfile as pipeline input. By default, the cmdlet collects the serial number of the server. If you do not want the cmdlet to collect the serial number, use the **-NoSerialNumber** switch parameter.

Syntax

```
New-UcsHardwareProfile [-NamePrefix <string>] [-NoSerialNumber] -ManagedObject <ManagedObject>

-OsVendorCode <string> -OsVersionCode <string> [-Ucs <BaseHandle[]>] [<CommonParameters>]
New-UcsHardwareProfile -Name <string> [-NoPostFix] [-NoSerialNumber] -ManagedObject
<ManagedObject> -OsVendorCode <string> -OsVersionCode <string> [-Ucs <BaseHandle[]>]
[<CommonParameters>]
```

Detailed Description

Property	Description	Required
Name	Specifies the name of the Hardware profile.	True
NamePrefix	Specifies the name prefix of the Hardware profile.	False

Property	Description	Required
Path	Specifies path at which the Hardware Profile JSON file is located.	True
LiteralPath	Specifies path at which the Hardware Profile JSON file is located.	True
NoPostFix	Specifies that the cmdlet does not attach the postfix string to the Hardware Profile name.	False
ManagedObject	Specifies the managed object. You can pass an associated ServiceProfile or blade or RackUnit or ServerNode as the parameter or from pipeline	True
NoSerialNumber	Switch parameter instructs cmdlet not to capture Serial Number inside the Hardware Profile.	False
OsVendorCode	Specifies the operating system's vendor code obtained using Get-UcsOsVendor cmdlet.	True
OsVersionCode	Specifies the operating system code obtained using Get-UcsOperatingSystem cmdlet.	True
Ucs	Specifies the ucs handle or handles	False
CommonParameters	This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable.	

Example

```
C:\> $managedObject = Get-UcsManagedObject -Dn "sys/rack-unit-1"

C:\> $hardwareProfile = New-UcsHardwareProfile -Name "Test-HardwareProfile"
     -OsVendorCode $osVendor[1].T_ID -OsVersionCode $osVersion[1].T_ID -ManagedObject
     $managedObject

C:\> $hardwareProfile | ConvertTo-Json -Depth 100 | Out-File C:\hardwareProfile.json
```

Add-UcsHardwareProfile

The Add-UcsHardwareProfile cmdlet is used to upload the hardware profile to the HCL tool. Specify the Cisco.com credentials and hardware profile objects which were created using the New-UcsHardwareProfile cmdlet. You can save multiple hardware profiles for the account specified. You can also specify the JSON

file as input for uploading a hardware profile to the HCL tool. These hardware profiles can be used later to get the information about the recommended driver and firmware versions for the adaptors.

Syntax

```
Add-UcsHardwareProfile -Credential <PSCredential>
-HardwareProfile <CiscoHardwareProfile>
[-Proxy <WebProxy>] [-Xml] [<CommonParameters>]

Add-UcsHardwareProfile -Credential <pscredential>
-Path <string> [-Proxy <WebProxy>]
[-Xml]  [<CommonParameters>]

Add-UcsHardwareProfile -Credential <pscredential>
-LiteralPath <string> [-Proxy <WebProxy>] [-Xml]
    [<CommonParameters>]
```

Detailed Description

Property	Description	Required
HardwareProfile	Specifies the hardware profile object created using the New-UcsHardwareProfile cmdlet.	True
Proxy	Specifies the web proxy to be used for the communication with HCL api.	False
Credential	Specifies the Cisco.com credential of user.	True
Path	Specifies path at which Hardware Profile JSON file is located	True
LiteralPath	Specifies path at which Hardware Profile JSON file is located.	True
CommonParameters	This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable.	

Example

```
C:\> $result = Add-UcsHardwareProfile -Credential $cecCred -Path C:\hardwareProfile.json

C:\> $result

CreationDate : 2019-06-20T06:37:07+00:00
LastModified : 2019-06-20T06:37:07+00:00
CecId        :
Id           : 9463
Name         : Test-HardwareProfile_0
UcsServer    : Cisco.Ucs.Core.CiscoUcsServer
```

Get-UcsHardwareProfile

The Get-UcsHardwareProfile cmdlet is used to get all the hardware profiles present in the HCL tool for the given cisco.com credential. You can use the optional name or Id parameter to get the specific hardware profile.

Syntax

```
Parameter Set: Default
Get-UcsHardwareProfile -Credential <PSCredential> [-Proxy <WebProxy>] [<CommonParameters>]

Parameter Set: Name
Get-UcsHardwareProfile -Credential <PSCredential> -Name <string> [-Proxy <WebProxy>]
[<CommonParameters>]
Parameter Set: Id
Get-UcsHardwareProfile -Credential <PSCredential> -Id <string> [-Proxy <WebProxy>]
[<CommonParameters>]
```

Detailed Description

Property	Description	Required
Name	Specifies the name of the Hardware profile.	True
Id	Specifies the name of the Hardware Profile.	True
Proxy	Specifies the web proxy to be used for the communication with HCL api.	False
Credential	Specifies the Cisco.com credential of user.	True
CommonParameters	This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable.	

Example

```
C:\> $getHardwareProfile = Get-UcsHardwareProfile -Credential $cecCred -Name
     "Test-HardwareProfile_0"

C:\> $getHardwareProfile

CreationDate : 2019-06-20T06:37:07+00:00
LastModified : 2019-06-20T06:37:07+00:00
CecId        :
Id           : 9463
Name         : Test-HardwareProfile_0
UcsServer    : Cisco.Ucs.Core.CiscoUcsServer
```

Remove-UcsHardwareProfile

The Remove-UcsHardwareProfile cmdlet is used to delete the hardware profile from the HCL tool.

Syntax

```
Remove-UcsHardwareProfile -Credential <PSCredential> -HardwareProfile <CiscoHardwareProfile>
 [-Proxy <WebProxy>] [<CommonParameters>]
Remove-UcsHardwareProfile -Credential <PSCredential> -Id <string> [-Proxy <WebProxy>]
[<CommonParameters>]
Remove-UcsHardwareProfile -Credential <PSCredential> -Name <string> [-Proxy <WebProxy>]
[<CommonParameters>]
```

Detailed Description

Property	Description	Required
Name	Specifies the name of the Hardware profile.	True
Id	Specifies the name of the Hardware Profile.	True
Proxy	Specifies the web proxy to be used for the communication with HCL api.	False
Credential	Specifies the Cisco.com credential of user.	True
CommonParameters	This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable.	

Example

```
C:\> Remove-UcsHardwareProfile -Credential $cecCred -Name "Test-HardwareProfile_0"

Hardware Profile with Name Test-HardwareProfile_0 removed successfully
```

Invoke-UcsHclUtility

The Invoke-UcsHclUtility cmdlet is used to get the recommended driver and firmware versions of the adaptors for the hardware profile available in the HCL tool.

Syntax

```
Invoke-UcsHclUtility -Credential <PSCredential> -HardwareProfile <CiscoHardwareProfile>
[-Proxy <WebProxy>] [-Tree] [-Advisories] [-AdvisoryType <string>] [<CommonParameters>]
Invoke-UcsHclUtility -Credential <PSCredential> -Name <string> [-Proxy <WebProxy>] [-Tree]

[-Advisories] [-AdvisoryType <string>] [<CommonParameters>]
Invoke-UcsHclUtility -Credential <PSCredential> -Id <string> [-Proxy <WebProxy>] [-Tree]
[-Advisories] [-AdvisoryType <string>] [<CommonParameters>]
```

Detailed Description

Property	Description	Required
Name	Specifies the name of the Hardware profile.	True

Property	Description	Required
Advisories	Specifies the switch parameter to show the Advisories published for the components or adaptors that is part of the hardware profile.	False
AdvisoriesType	Specifies the type of advisory to be fetched.	False
Tree	If specified, the cmdlet displays detailed output in a tree view. **Note** If you use -Tree, then you cannot use the output of cmdlet for the pipeline	False
HardwareProfile	Specifies the hardware profile object.	True
Proxy	Specifies the web proxy to be used for the communication with HCL api.	False
Credential	Specifies the Cisco.com credential of user.	True
CommonParameters	This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable.	

Example

```
C:\> $hclResult = Invoke-UcsHclUtility -Credential $cecCred -Name "Test-HardwareProfile_0"

C:\> $hclResult

CompatibilityProfiles Advisories
--------------------- ----------
{810}

 C:\> $hclResult.CompatibilityProfiles


Version            : 4.0(1)
DownloadUrl        :
https://software.cisco.com/download/release.html?mdfid=283862063&flowid=25886&softwareid
=283655681
InstallationDocUrl :
http://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-manager/products
-installation-guides-list.html
DriverIsoDownload  :
https://software.cisco.com/download/home/283862063/type/283853158/release/4.0(1)
HardwareTypes      : Cisco.Ucs.Core.HardwareTypes
Component          : {ST1000NM0045, HUSMR7680BDP301, Micron_5100_MTFDDAV240TCB}
OsVersion          : Windows Server 2019
```

```
OsVendor           : Microsoft
OsVersionId        : 810
OsVendorId         : 1
Notes              : {96 : Requires minimum FW version 4.1.2d}
```

Get-UcsHclAdvisoryTypes

Syntax

To view the advisories for the server model and its components, use the **Get-UcsHclAdvisoryTypes** cmdlet.

```
Get-UcsHclAdvisoryTypes -Credential <pscredential>
[-Proxy <WebProxy>][<CommonParameters>]
```

Property	Description	Required
Credential	Specify the Cisco.com user credential	True
Proxy	Specify the web proxy to be used for communicating with HCL API.	False
CommonParameters	This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable.	

Example

```
C:\> $advisoryResult = Get-UcsHclAdvisoryTypes -Credential $cecCred

C:\> $advisoryResult
EOL Advisory
Field Notice
```