



## 例

---

この章は、次の項で構成されています。

- [Cisco IMC ファームウェアのアクティブ化 \(2 ページ\)](#)
- [ユーザの追加 \(2 ページ\)](#)
- [Cisco IMC Desired State Configuration \(DSC\) \(3 ページ\)](#)
- [Cisco IMC ファームウェアを更新する \(7 ページ\)](#)
- [ブート ドライブのクリア \(7 ページ\)](#)
- [\[NTP 設定の構成 \(Configure NTP Settings\) \] \(7 ページ\)](#)
- [Confirm フラグ \(7 ページ\)](#)
- [SoL の設定 \(8 ページ\)](#)
- [仮想ドライブの作成 \(8 ページ\)](#)
- [ドライブセキュリティの無効化 \(8 ページ\)](#)
- [ドライブセキュリティの有効化 \(9 ページ\)](#)
- [Enable ImcPidCatalog \(9 ページ\)](#)
- [IP ブロッキングの有効化 \(9 ページ\)](#)
- [Export-ImcHardwareInventory \(9 ページ\)](#)
- [Filters \(10 ページ\)](#)
- [Force フラグ \(11 ページ\)](#)
- [アダプタおよびコントローラの情報の取得 \(11 ページ\)](#)
- [Get ImcKmpDownloadStatus \(11 ページ\)](#)
- [Get-ImcKmpUploadStatus \(12 ページ\)](#)
- [HUU ファームウェアの更新 \(12 ページ\)](#)
- [SD カードを通じた HUU ファームウェアの更新 \(12 ページ\)](#)
- [ドライブセキュリティ情報の変更 \(13 ページ\)](#)
- [管理対象オブジェクトの同期 \(13 ページ\)](#)
- [Syslog 設定の変更 \(14 ページ\)](#)
- [新しい証明書署名要求 \(14 ページ\)](#)
- [PowerTool コマンドレットの生成 \(14 ページ\)](#)
- [IMC の証明書の受け取り \(15 ページ\)](#)
- [Receive-ImcKmpEntity \(15 ページ\)](#)

- [Receive-ImcLdapCACertificate](#) (15 ページ)
- [Remove-ImcLdapCACertificate](#) (16 ページ)
- [Reset-ImcEventFilters](#) (16 ページ)
- [Send-ImcBiosProfile](#) (16 ページ)
- [Send-ImcKmpEntity](#) (16 ページ)
- [Send-ImcLdapCACertificate](#) (17 ページ)
- [Send-ImcPidCatalog](#) (17 ページ)
- [サーバの操作](#) (18 ページ)
- [ブート ドライブの設定](#) (18 ページ)
- [ディスク モード \(JBOD から UG、およびその逆\) の変更](#) (18 ページ)
- [ブート順序の設定](#) (19 ページ)
- [BIOS パスワードの設定](#) (19 ページ)
- [Start-ImcOsInstallation](#) (19 ページ)
- [Test-ImcLdapBinding](#) (20 ページ)
- [トランザクション サポート](#) (21 ページ)
- [vMedia の設定](#) (21 ページ)
- [vNIC/アダプタの作成](#) (21 ページ)
- [Cisco UCS コミュニティ](#) (22 ページ)
- [関連する Cisco IMC のマニュアルとマニュアル フィードバック](#) (22 ページ)
- [マニュアルの入手方法およびテクニカル サポート](#) (22 ページ)

## Cisco IMC ファームウェアのアクティブ化

次のコマンドレットを使用して、Cisco IMC ファームウェアをアクティブ化します。

```
Get-ImcFirmwareBootDefinition -Type "blade-controller" |
Get-ImcFirmwareBootUnit | Set-ImcFirmwareBootUnit-AdminState trigger -Image backup
-ResetOnActivate yes -Force
```

## ユーザの追加

```
Get-ImcLocalUser -Id 9 | Set-ImcLocalUser -Name "admin" -pwd "Password" -AccountStatus
"active" -Priv "admin"
```



(注) Clear-ImcLocal は、ステータスを非アクティブに変更し、ユーザやデータの削除は行いません。

## Cisco IMC Desired State Configuration (DSC)

Desired State Configuration (DSC) は、ローカルマシンやリモートマシンを構成するための新しい方式です。IMC DSCのリソースを使用して、集中型ルートサーバからデータセンターに複数のIMCを設定できます。PowerTool モジュール「Cisco.UCS.DesiredStateConfiguration」には、すべてのカスタム IMC DSC リソースが含まれています。

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

DSC リソースは並行して実行でき、Cisco IMC での XML API 接続の最大数は 4 に制限されています。したがって、そのような場合は、各 IMC DSC リソースにさらに DependsOn プロパティを指定します。

## ImcManagedObject リソース

ImcManagedObject リソースは、Cisco.UCS.DesiredStateConfiguration モジュールの一部です。このリソースを使用すると、DSC フレームワークによって複数の Cisco IMC サーバに MO の詳細を指定することで、Cisco IMC 管理対象オブジェクト (MO) を設定できます。

### 構文

```
ImcManagedObject [string] #ResourceName
{
  Dn = [string]
  Identifier = [string]
  ImcConnectionString = [string]
  ImcCredentials = [PSCredential]
  [ Action = [string] { Add | Set } ]
  [ ClassId = [string] ]
  [ DependsOn = [string[]] ]
  [ Ensure = [string] { Absent | Present } ]
  [ PropertyMap = [string] ]
  [ WebProxyCredentials = [PSCredential] ]
}
```

| プロパティ               | 説明   |
|---------------------|--|
| Dn                  | 管理対象オブジェクトの Dn を指定します。   |
| ID                  | DSC リソースの一意の ID を指定します。  |
| ImcConnectionString | IMC サーバの接続文字列を指定します。<br>構文：<br>Name=<ipAddress><br>[`nNoSsl=<bool>][`nPort=<ushort>]<br>[`nProxyAddress=<proxyAddress>]<br>[`nUseProxyDefaultCredentials=<bool>] |

| プロパティ               | 説明  |
|---------------------|---|
| ImcCredentials      | IMC にアクセスするために必要な資格情報を示します。   |
| Action              | 管理対象オブジェクトに対して実行するアクションを指定します。管理対象オブジェクトを追加する場合は、このプロパティを <b>Add</b> に設定します。既存の管理対象オブジェクトを変更するには、 <b>Set</b> に設定します。  |
| ClassId             | 管理対象オブジェクトのクラスIDを指定します。   |
| DependsOn           | 当該リソースを設定する前に、別のリソースの設定を実行する必要があることを示します。たとえば、実行するリソース設定スクリプトブロックの最初のIDが <b>ResourceName</b> 、そのタイプが <b>ResourceType</b> であるとしします。このプロパティを使用するための構文は、次のとおりです。<br><br>DependsOn = "[ResourceType]ResourceName" |
| Ensure              | 管理対象オブジェクトが存在しているかどうかを示します。管理対象オブジェクトが存在していないことを示すには、このプロパティを <b>Absent</b> に設定します。管理対象オブジェクトが存在していることを示すには、 <b>Present</b> に設定します。デフォルトは <b>Present</b> です。  |
| PropertyMap         | 管理対象オブジェクトのプロパティを keyValue ペアとして指定します。<br><br>構文：<br><br>`<key1>=<value1> `<key2>=<value2>  |
| WebProxyCredentials | Web プロキシの資格情報を示します。   |

### 例

以下の例は、ImcManagedObject リソースを使用して管理対象オブジェクトを追加し、Dn として "sys/rack-unit-1/boot-policy/efi-read-only" を指定する方法を示しています。

既存の MO を編集するには、Action="Set" を使用します。

ImcManagedObjectResourceDemo の設定

```
{
```

```

param(
  [Parameter(Mandatory=$true)]
  [PSCredential] $imcCredential,

  [Parameter(Mandatory=$true)]
  [string] $connectionString
)
Import-DSCResource -ModuleName Cisco.Ucs.DesiredStateConfiguration
Node "localhost"
{
  ImcManagedObject ResourceInstance
  {
    Ensure = "Present"
    ClassId= "lsbootEfi"
    Dn = "sys/rack-unit-1/boot-policy/efi-read-only"
    PropertyMap = "Access = read-only `nType = efi `nOrder = 4"
    ImcCredentials = $imcCredential
    ImcConnectionString = $connectionString
    Identifier = "2"
  }
}
}

```

### ImcScript リソース

Cisco.Ucs.DesiredStateConfiguration モジュールの ImcScript リソースは、IMC PowerTool コマンドレットを実行するメカニズムを備えています。

#### 構文

ImcScript [string] #ResourceName

```

{
  Dn = [string]
  Identifier = [string]
  ImcConnectionString = [string]
  ImcCredentials = [PSCredential]
  Script = [string]
  [ Action = [string] { Add | Set } ]
  [ DependsOn = [string[]] ]
  [ Ensure = [string] { Absent | Present } ]
  [ WebProxyCredentials = [PSCredential] ]
}

```

| プロパティ  | 説明   |
|--------|--|
| Dn     | 管理対象オブジェクトの Dn を指定します。                                       |
| ID     | DSC リソースの一意的 ID を指定します。                                      |
| Script | PowerTool コマンドレットのセットを指定します。新しいコマンドレットのプレフィクスとして `n` を使用します。 |

| プロパティ               | 説明  |
|---------------------|---|
| ImcConnectionString | IMC サーバの接続文字列を指定します。<br>構文：<br>Name=<ipAddress><br>[`nNoSsl=<bool>][`nPort=<ushort>]<br>[`nProxyAddress=<proxyAddress>]<br>[`nUseProxyDefaultCredentials=<bool>]  |
| ImcCredentials      | IMC サーバへのアクセスに必要な資格情報を示します。   |
| Action              | 管理対象オブジェクトに対して実行するアクションを指定します。管理対象オブジェクトを追加する場合は、このプロパティを <b>Add</b> に設定します。既存の管理対象オブジェクトを変更するには、 <b>Set</b> に設定します。  |
| DependsOn           | 当該リソースを設定する前に、別のリソースの設定を実行する必要があることを示します。たとえば、最初に実行するリソース設定スクリプトブロックの ID が <b>ResourceName</b> 、そのタイプが <b>ResourceType</b> であるとしします。このプロパティを使用するための構文は、次のとおりです。<br><br>DependsOn = "[ResourceType]ResourceName" |
| Ensure              | スクリプトが実行されているかどうかを示します。デフォルトは <b>Present</b> です。  |
| WebProxyCredentials | Web プロキシの資格情報を示します。   |
| WebProxyCredentials | Web プロキシの資格情報を示します。   |

### 構文

```

Configuration ImcScriptResourceDemo
{
    param (
        [Parameter(Mandatory=$true)]
        [PsCredential] $imcCredential,

        [Parameter(Mandatory=$true)]
        [string] $connectionString
    )
    Import-DSCResource -ModuleName Cisco.Ucs.DesiredStateConfiguration

    Node "localhost"
    {
        ImcScript ResourceInstance
    }
}

```

```
{
Ensure = "Present"
Dn = "sys/svc-ext/snmp-svc/snmpv3-user-9"
Script= "Clear-ImcSnmpUser -id 2 -force
`n Add-ImcSnmpUser -Id 9 -Name 'testuser'
-Auth MD5 -AuthPwd password1 -Privacy AES
-PrivacyPwd password2 -SecurityLevel authpriv
`n Clear-ImcSnmpUser -id 2 -force "
ImcCredentials = $imcCredential
ImcConnectionString = $connectionString
Identifier = "2"

} }
}
```

## Cisco IMC ファームウェアを更新する

次のコマンドレットを使用して、ユーザ資格情報を作成します。

```
$user = "<username>"
$password = "<password>"
$cred = New-Object System.Management.Automation.PSCredential($user,$password)
```

次のコマンドレットを使用して、Cisco IMC ファームウェアを更新します。

```
Get-ImcFirmwareUpdatable -Type blade-controller | Set-ImcFirmwareUpdatable -AdminState
trigger -Type blade-controller -Protocol ftp -RemoteServer "10.65.183.111" -RemotePath
"/UcseBin/UCSE_CIMC_2.3.1.bin"-RemoteCredential $cred-Force
```

## ブート ドライブのクリア

ブート ドライブをクリアするには、次のコマンドレットを使用します。

```
Get-ImcStorageController | Set-ImcStorageController -AdminAction "clear-boot-drive"
-Force
```

## [NTP 設定の構成 (Configure NTP Settings)]

次のコマンドレットを使用して、NTP の設定項目を設定します。

```
Get-ImcNtpServer | Set-ImcNtpServer -NtpEnable "yes" -NtpServer1 1.1.1.1 -Force
```

## Confirm フラグ

PowerTool コマンドレットで「-」付き Confirm スイッチパラメータを指定すると、変更の確認を要求されます。コマンドレットは、Windows PowerShell 環境外部のシステムに適用された変

更の確認を求める要求を送信します。たとえば、SNMPユーザをクリアするコマンドレットを実行した場合、コマンドレットはアクションを完了するかどうかユーザに確認を求めます。

### Syntax

```
Get-ImcSnmUser -Name snmpuser | Clear-ImcSnmUser -Confirm
Confirm
Are you sure you want to perform this action?
Performing the operation "Clear-ImcSnmUser" on target "Clear".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
```

## SoL の設定

次のコマンドレットを使用して、SoL を設定します。

```
Get-ImcSolif -Dn "sys/rack-unit-1/sol-if" | Set-ImcSolif -AdminState "enable" -Speed
"57600" -Force
```

## 仮想ドライブの作成

未使用の物理ドライブを使用して、仮想ドライブを作成します

```
Get-ImcStorageVirtualDriveCreatorUsingUnusedPhysicalDrive |
Set-ImcStorageVirtualDriveCreatorUsingUnusedPhysicalDrive
-AdminState trigger -size "400 MB" -DriveGroup "[2]" -RaidLevel 0 -VirtualDriveName
"vd_111" -Force
```

仮想ドライブ グループを使用して仮想ドライブを作成する

```
Get-ImcStorageVirtualDriveCreatorUsingVirtualDriveGroup |
Set-ImcStorageVirtualDriveCreatorUsingVirtualDriveGroup
-AdminState trigger -VirtualDriveName "vd_New"-SharedVirtualDriveId "3" -Size "100 MB"
-Force
```

複数のドライブから仮想ドライブを作成します

```
Get-ImcStorageController |
Set-ImcStorageVirtualDriveCreatorUsingUnusedPhysicalDrive
-AdminState trigger -DriveGroup "[1,2]" -RaidLevel 1 -Size "285148 MB" -VirtualDriveName
"RAID1_12" -WritePolicy "Always Write Back" -Force
```

## ドライブセキュリティの無効化

ディスク上の現在の状態に応じてコントローラ ロック キーを無効にします。



(注) ドライブセキュリティを無効にすると、すべての安全なドライブ上のデータを使用できなくなります。



```
Get-ImcStorageController | Disable-ImcDriveSecurity -Force
Get-ImcSelfEncryptStorageController | Disable-ImcDriveSecurity -Force
```

## ドライブセキュリティの有効化

ディスク上の現在の状態に応じてコントローラ ロック キーを有効にします。

```
Get-ImcStorageController | Enable-ImcDriveSecurity -KeyId "myKey123" -SecurityKey
"myPass123" -Force
```

## Enable ImcPidCatalog

IMC サーバにアップロードされた PID カタログを有効にします。

### Syntax

```
Get-ImcPidCatalog | Enable-ImcPidCatalog -Force
```

## IP ブロッキングの有効化

次のコマンドレットを使用して、IP ブロッキングを有効にします。

```
Get-ImcIpBlocking | Set-ImcIpBlocking -Enable "yes"
```

## Export-ImcHardwareInventory

**Export-ImcHardwareInventory** コマンドレットは、リモートの場所に、システムのハードウェア インベントリをエクスポートします。リモートサーバの IP/ホスト名、プロトコル、パスとファイル名、ユーザ名とパスワードなど、プロファイルの詳細を指定することもできます。

### 構文

```
Export-ImcHardwareInventory -Chassis <EquipmentChassis> -Hostname <string> [-Proto
<string>] [-Pwd <string>]
-RemoteFile <string> [-User <string>] [-XtraProperty <Hashtable>] [-Force]
[<CommonParameters>]
```

```
Export-ImcHardwareInventory -TopSystem <TopSystem> -Hostname <string> [-Proto <string>]
[-Pwd <string>] -RemoteFile <string> [-User <string>] [-XtraProperty <Hashtable>] [-Force]
[<CommonParameters>]
```

## 例

```
Get-ImcTopSystem | Export-ImcHardwareInventory -Hostname "10.10.10.10" -Proto scp -User
root
-Pwd <password> -RemoteFile "/root/test/InventoryExportReport.txt" -Force
```

## Filters

#SysdebugMEpLog 管理対象オブジェクトを取得。Typeは「SEL」または「Syslog」と等しい。

```
Get-ImcRackUnit | Get-ImcMgmtController | Get-ImcSysdebugMEpLog -Filter '(type -ilike
SEL) -or (Type -clike Syslog)'
```

#SysdebugMEpLog 管理対象オブジェクトを取得。Typeは「SEL」または「Syslog」と等しく、Idは「0」と等しい。

```
Get-ImcRackUnit | Get-ImcMgmtController | Get-ImcSysdebugMEpLog -Filter '(type -ilike
SEL) -or (Type -clike Syslog)' -Id 0 -Type SEL
```

#ローカルユーザを取得。名前は「admin」（大文字小文字を区別）とすることができる。

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'Name -clike admin'
```

#ユーザを取得。名前は「test\*」とすることができる（\*正規表現または大文字小文字の区別をサポート）。

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'Name -clike test*'
```

#ローカルユーザを取得。AccountStatusはinactiveと等しくない。

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'AccountStatus -cne inactive'
```

#ローカルユーザを取得。AccountStatusは「inacti」と一致する。

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'AccountStatus -cmatch inacti'
```

#ローカルユーザを取得。AccountStatusは「active」と一致する（activeで始まるか、大文字小文字を区別）。

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'AccountStatus -cmatch ^active'
```

#ローカルユーザを取得。AccountStatusは「active」と一致しない（activeで始まるか、大文字小文字を区別）。

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'AccountStatus -cnotmatch ^active'
```

#ローカルユーザを取得。AccountStatusは「active」でない（activeで始まるか、大文字小文字を区別）。

```
Get-ImcManagedObject -ClassId aaaUser -Filter 'AccountStatus -cnotlike active'
```

## Force フラグ

PowerTool の Set および Remove コマンドレットはすべて確認を要求します。-Force フラグを使用すると、この確認をスキップできます。

### Syntax

```
Get-ImcSnmper -Name snmpuser | Clear-ImcSnmper -Force
```

## アダプタおよびコントローラの情報取得

# PCI アダプタのプロパティ

```
Get-ImcPciEquipSlot -Id "1"
```

# ネットワーク アダプタ情報

```
Get-ImcNetworkAdapterEthIf -Dn "sys/rack-unit-1/network-adapter-L/eth-1"
```

# ストレージ コントローラ情報

```
Get-ImcStorageController -Dn "sys/rack-unit-1/board/storage-SAS-SLOT-4"
```

## Get ImcK mipDownloadStatus

**Get ImcK mipDownloadStatus** コマンドレットでは、ルート CA 証明書、クライアント証明書、およびクライアントの秘密キーのような KMIP エンティティのダウンロードステータスを取得するためのオプションを提供します。

### 構文

```
Get-ImcK mipDownloadStatus [-Type <string>] [-XtraProperty <Hashtable>] [<CommonParameters>]
```

### 例

```
Get-ImcK mipDownloadStatus
Get-ImcK mipDownloadStatus -Type RootCACertificate
Get-ImcK mipDownloadStatus -Type ClientCertificate
Get-ImcK mipDownloadStatus -Type ClientPrivateKey
```

## Get-ImcKmpUploadStatus

**Get-ImcKmpUploadStatus** コマンドレットでは、ルート CA 証明書、クライアント証明書、およびクライアントの秘密キーのような KMIP エンティティのアップロードステータスを取得するためのオプションを提供します。

### 構文

```
Get-ImcKmpUploadStatus [-Type <string>] [-XtraProperty <Hashtable>] [<CommonParameters>]
```

### 例

```
Get-ImcKmpUploadStatus
Get-ImcKmpUploadStatus -Type RootCACertificate
Get-ImcKmpUploadStatus -Type ClientCertificate
Get-ImcKmpUploadStatus -Type ClientPrivateKey
```

## HUU ファームウェアの更新

次の cmdlet を使用して、ユーザ資格情報を作成します。

```
$user = "<username>"
$password = "<password>"
$cred = New-Object System.Management.Automation.PSCredential($user,$password)
```

次の cmdlet を使用して、HUU ファームウェアを更新します。

```
Set-ImcHuuFirmwareUpdater -AdminState trigger -MapType nfs -RemoteIp 10.105.219.83
-RemoteCredential $cred -RemoteShare "/huuIso/ucs-c2x-huu-2.0.3d-1.iso" -StopOnError yes
-TimeOut 60 -UpdateComponent All-VerifyUpdate no -force -Xml
```

## SD カードを通じた HUU ファームウェアの更新

NFS マッピング:

```
Get-ImcStorageFlexUtilVirtualDriveImageMap -VirtualDrive "HUU" |
Set-ImcStorageFlexUtilVirtualDriveImageMap -AdminAction map -Map nfs -RemoteShare
"x.x.x.x:/nfsShareLocation"
-RemoteFile "ucs-c240m5-huu-3.1.3a.iso" -MountOptions "nolock" -Force
```

指定されたマウント場所から HUU のパーティションにマッピングされた画像を更新します。

```
Get-ImcStorageFlexUtilVirtualDrive -PartitionName HUU |
Set-ImcStorageFlexUtilVirtualDrive -AdminAction update-vd -Force
```

以下のクエリを使用して、更新ステータスを見つけることができます。

```
Get-ImcStorageFlexUtilVirtualDrive -PartitionName HUU |
select OperationInProgress, LastOperationStatus, HostAccessible
```



(注) OperationInProgress:値は、Update-Successでなければなりません。

パーティションのホストに表示されるようにする仮想ドライブを有効にするように要求します。

```
Get-ImcStorageFlexUtilVirtualDrive -PartitionName HUU |
Set-ImcStorageFlexUtilVirtualDrive -AdminAction enable-vd -Force
```



(注) HostAccessible:値は、Connectedでなければなりません。

ブート順序を設定するために LUN ID を取得します。

```
Get-ImcStorageFlexUtilVirtualDrive -PartitionName HUU | select LunId
```

LUN ID に基づいて flex util HUU パーティションからのブート順序を設定します。

```
Get-ImcLsbootSd | set-ImcLsbootSd -Lun <lunId selected in above cmdlet>
-Order 1 -State enabled -Subtype flex-util -Force
Get-ImcLsbootDevPrecision | Set-ImcLsbootDevPrecision -RebootOnUpdate yes
```

HUU ファームウェアの更新プロセスを開始します。

```
$user = "testUser"
$password = "testPassword" | ConvertTo-SecureString -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential($user,$password)
Set-ImcHuuFirmwareUpdater -AdminState trigger -MapType nfs -RemoteIp "NA" -RemoteCredential
$cred -RemoteShare "NA"
-StopOnError yes -TimeOut 120 -UpdateComponent All -VerifyUpdate no -BootMedium "microsd"
-Force
```

## ドライブセキュリティ情報の変更

次の cmdlet を使用して、ドライブセキュリティ MO の security key/keyId を更新します。

```
Get-ImcStorageController |Set-ImcDriveSecurity -KeyId "newkey" -KeyManagement local
-SecurityKey "password4321"
-ExistingSecurityKey "myPass123" -Force
```

## 管理対象オブジェクトの同期

# 次の cmdlet を使用して、複数の Cisco IMC に接続するために SupportMultipleDefaultUcs を有効にします。

```
Set-UcsPowerToolConfiguration -SupportMultipleDefaultUcs $true
```

# 次の cmdlet を使用して、資格情報を取得し、変数に保存します。

```

$secpasswd = ConvertTo-SecureString password -AsPlainText -Force
$mycreds = New-Object System.Management.Automation.PSCredential ("admin",$secpasswd)

# 次の cmdlet を使用して、別の Cisco IMC に接続します。

$cimc1 = Connect-Imc xx.xx.xx.xx -Credential $mycreds
$cimc2 = Connect-Imc xx.xx.xx.xx -Credential $mycreds

# 次の cmdlet を使用して、別の Cisco IMC からローカル ユーザを取得します。

$user1 = Get-ImcLocalUser -Imc $cimc1 -Id 1
$user2 = Get-ImcLocalUser -Imc $cimc2 -Id 1

# 次の cmdlet を使用して、MO のセットを Cisco IMC2 から Cisco IMC1 に同期します。

Compare-ImcManagedObject $user1 $user2
Sync-ImcManagedObject (Compare-ImcManagedObject $user1 $user2) -Imc $cimc1

```

## Syslog 設定の変更

次の cmdlet を使用して、syslog の設定を変更します。

```
Get-ImcSyslog | Set-ImcSyslog -LocalSeverity warning -RemoteSeverity debug -Force
```

## 新しい証明書署名要求

新しい証明書を取得するために証明書署名要求 (CSR) を生成します。Cisco IMC に新しい証明書をアップロードして、現在のサーバ証明書と置き換えることができます。VeriSign などの公的な認証局 (CA) またはユーザ独自の認証局によってサーバを認定します。生成される証明書キーの長は 2048 ビットです。

```

New-ImcCertificateSigningRequest -CommonName "CSR2" -CountryCode India -Locality "GG6"
-Organization "cisco" -OrganizationalUnit "Tpidev" -Protocol ftp -State "Haryana"
-RemoteFile "ImcCertificate.txt" -RemoteServer 10.105.219.xx -User administrator -Pwd
*****

```

## PowerTool コマンドレットの生成

ConvertTo-ImcCmdlet:

Cisco IMC の GUI では、XML ログインがサポートされていません。ConvertTo-ImcCmdlet cmdlet を生成するには、Get コマンドレットの出力を使用して、同じオブジェクト階層を複製するコマンドレットを生成します。

指定されている MO のコマンドレットを生成する。

```
Get-ImcBiosSettings -Hierarchy | ConvertTo-ImcCmdlet
```

コマンドレットの出力をファイルに保存する。

```
Get-ImcBiosSettings -Hierarchy | ConvertTo-ImcCmdlet -OutputPath "C:/OutputFile.txt"
```

## IMC の証明書の受け取り

Cisco IMC サーバで現在利用可能な証明書の情報を取得します。

```
Receive-ImcCertificate
```

## Receive-ImcK mipEntity

**Receive-ImcK mipEntity** cmdlet では、ルート CA 証明書、クライアント証明書、およびクライアントの秘密キーのような KMIP エンティティをダウンロードするためのオプションを提供します。

### 構文

```
Receive-ImcK mipEntity -Type <string> [-Protocol <string>] [-Pwd <string>] [-RemoteFile <string>] [-RemoteServer <string>] [-User <string>] [-XtraProperty <Hashtable>] [-Force] [<CommonParameters>]
```

### 例

```
Receive-ImcK mipEntity -Type RootCACertificate -RemoteServer 10.10.10.10 -User root -Pwd <password> -Protocol scp -RemoteFile "/root/test/RootCACertificate.pem" -Force
Receive-ImcK mipEntity -Type ClientCertificate -RemoteServer 10.10.10.10 -User root -Pwd <password> -Protocol scp -RemoteFile "/root/test/ClientCertificate.pem" -Force
Receive-ImcK mipEntity -Type ClientPrivateKey -RemoteServer 10.10.10.10 -User root -Pwd <password> -Protocol scp -RemoteFile "/root/test/ClientPrivateKey.pem" -Force
```

## Receive-ImcLdapCACertificate

LDAP CA 証明書を IMC サーバからリモート サーバにエクスポートします。

### Syntax

```
Get-ImcExportLdapCACertificate | Receive-ImcLdapCACertificate -Protocol scp -RemoteServer "10.10.10.10" -RemoteFile "/root/test/ExportFileLdapCACertificate.crt" -User "user" -Pwd "Password123" -Force
```

## Remove-ImcLdapCACertificate

IMC サーバから LDAP CA 証明書を削除します。

### Syntax

```
Get-ImcLdapCACertificate | Remove-ImcLdapCACertificate -Force
```

## Reset-ImcEventFilters

イベント フィルタをリセットします。

### Syntax

```
Get-ImcEventManager | Reset-ImcEventFilters -Force
Get-ImcRackUnit | Reset-ImcEventFilters -Force
```

## Send-ImcBiosProfile

**Send-ImcBiosProfilecmdlet** は、Cisco IMC に BIOS プロファイルをアップロードします。リモートロケーションの IP/ホスト名、プロトコル、パスとファイル名、ユーザ名とパスワードなど、プロファイルの詳細を指定することができます。

### 構文

```
Send-ImcBiosProfile -BiosProfileManagement <BiosProfileManagement> [-Protocol <string>]
[-Pwd <string>] [-RemoteFile <string>] [-RemoteServer <string>] [-User <string>]
[-XtraProperty <Hashtable>]
[-Force] [<CommonParameters>]
```

### 例

```
Get-ImcBiosProfileManagement | Send-ImcBiosProfile -Protocol scp -User root -Pwd <password>
-RemoteServer 10.10.10.10 -RemoteFile "/root/test/bios_profile_1" -Force
```

## Send-ImcKmipEntity

**Send-ImcKmipEntity cmdlet** は、ルート CA 証明書、クライアント証明書、およびクライアントの秘密キーと同様に、KMIP エンティティをアップロードするオプションを提供します。



## 構文

```
Send-ImcKmpEntity -Type <string> [-Protocol <string>] [-Pwd <string>] [-RemoteFile
<string>]
[-RemoteServer <string>] [-User <string>] [-XtraProperty <Hashtable>] [-Force]
[<CommonParameters>]
```

## 例

```
Send-ImcKmpEntity -Type RootCACertificate -RemoteServer 10.10.10.10 -User root -Pwd
<password>
-Protocol scp -RemoteFile "/root/test/RootCACertificate.pem" -Force

Send-ImcKmpEntity -Type ClientCertificate -RemoteServer 10.10.10.10 -User root -Pwd
<password>
-Protocol scp -RemoteFile "/root/test/ClientCertificate.pem" -Force

Send-ImcKmpEntity -Type ClientPrivateKey -RemoteServer 10.10.10.10 -User root -Pwd
<password>
-Protocol scp -RemoteFile "/root/test/ClientPrivateKey.pem" -Force
```

# Send-ImcLdapCACertificate

IMC サーバのリモートサーバにある LDAP CA 証明書をアップロードします。

## Syntax

```
Get-ImcDownloadLdapCACertificate | Send-ImcLdapCACertificate
-Protocol scp -RemoteServer "10.10.10.10" -RemoteFile "
/root/test/LDAPCACertificate.cer" -User "user" -Pwd
>Password123" -Force
```

# Send-ImcPidCatalog

IMC サーバにリモートサーバにあるファイルの PID カタログをアップロードします。

## Syntax

```
Get-ImcPidCatalog | Send-ImcPidCatalog -Protocol scp
-RemoteServer "10.10.10.10" -RemoteFile
"/root/test/pid-ctlg-2_0_13a18.tar.gz" -User
"user" -Pwd "Password123" -Force

Get-ImcUploadPIDCatalog | Send-ImcPidCatalog -Protocol scp
-RemoteServer "10.10.10.10" -RemoteFile
"/root/test/pid-ctlg-2_0_13a18.tar.gz" -User
"user" -Pwd "Password123" -Force
```

## サーバの操作

次の表では、次のサーバアクションを実行する新規および変更された cmdlet をリストします。

| 操作の説明                | PowerTool リリース 1.3.1 以前の<br>コマンドレット                                      | PowerTool 1.4.1 以降の cmdlet                   |
|----------------------|--|--|
| Power On Server      | Get-ImcRackUnit  <br>Set-ImcRackUnit -AdminPower<br>up                   | Get-ImcRackUnit  <br>Start-ImcServer         |
| Power Off Server     | Get-ImcRackUnit  <br>Set-ImcRackUnit -AdminPower<br>soft-shut-down       | Get-ImcRackUnit  <br>Stop-ImcServer          |
| Power Cycle Server   | Get-ImcRackUnit  <br>Set-ImcRackUnit -AdminPower<br>cycle-immediate      | Get-ImcRackUnit  <br>Restart-ImcServer       |
| Hard Reset Server    | Get-ImcRackUnit  <br>Set-ImcRackUnit -AdminPower<br>hard-reset-immediate | Get-ImcRackUnit  <br>Reset-ImcServer         |
| Turn On Locator LED  | Get-ImcLocatorLed  <br>Set-ImcLocatorLed -AdminState<br>on               | Get-ImcLocatorLed  <br>Enable-ImcLocatorLed  |
| Turn Off Locator LED | Get-ImcLocatorLed  <br>Set-ImcLocatorLed -AdminState<br>off              | Get-ImcLocatorLed  <br>Disable-ImcLocatorLed |

## ブート ドライブの設定

次の cmdlet を使用して、ブート ドライブとして物理ドライブを設定します。

```
Get-ImcStorageLocalDisk -Id 2 | Set-ImcStorageLocalDisk -AdminAction "set-boot-drive"
-Force
```

次の cmdlet を使用して、ブート ドライブとして仮想ドライブを設定します。

```
Get-ImcStorageVirtualDrive -Id 2 | Set-ImcStorageVirtualDrive -AdminAction "set-boot-drive"
-Force
```

## ディスク モード (JBOD から UG、およびその逆) の変更

ディスク モード (JBOD から UG、およびその逆) の変更

```
Get-ImcStorageController | Set-ImcStorageController -AdminAction enable-jbod -Force -Xml
get-ImcStorageLocalDisk -Id 3 | Set-ImcStorageLocalDisk -AdminAction make-jbod -Force
get-ImcStorageLocalDisk -Id 3 | Set-ImcStorageLocalDisk -AdminAction make-unconfigured-good
-Force
```

## ブート順序の設定

次の cmdlet を使用して、ブート順序を設定します。

```
Get-ImcLsbootStorage | Set-ImcLsbootStorage -Order 2 -Force
```

```
Get-ImcLsbootDevPrecision | Add-ImcLsbootHdd -Name "RAID1_12" -Order 1 -State "Enabled"
-Type "LOCALHDD"
Get-ImcLsbootDevPrecision | Add-ImcLsbootVMedia -Name "CIMCDVD" -Order 2 -State "Enabled"
-Type "VMEDIA"
Get-ImcLsbootDevPrecision -Hierarchy | ConvertTo-ImcCmdlet
```

## BIOS パスワードの設定



(注) BIOS パスワード設定機能は、E シリーズ サーバだけに適用されます。

```
Get-ImcBiosPassword | Set-ImcBiosPassword -Password "<password>" -Force
```

## Start-ImcOsInstallation

**Start-ImcOsInstallation** cmdlet は NI SCU オペレーティング システムのインストール プロセスを開始します。



(注) 設定ファイル、応答ファイルなどの作成方法の詳細については、[Cisco UCS C シリーズ サーバ 設定ユーティリティ](#) のマニュアルを参照してください。

### 構文

```
Start-ImcOsInstallation -OsInstallation <OsiStart> [-AnswerFilePassword <string>]
[-AnswerFileShareFile <string>]
[-AnswerFileShareIp <string>] [-AnswerFileSharePath <string>] -AnswerFileShareType
<string> [-AnswerFileUsername
<string>] [-ConfigShareFile <string>] [-ConfigShareIp <string>] [-ConfigSharePassword
<string>]
[-ConfigSharePath <string>] -ConfigShareType <string> [-ConfigShareUsername <string>]
-IsoShare <string> [-IsoShareIp <string>]
-IsoShareType <string> [-Password <string>] [-RemoteShareFile <string>] [-RemoteShareIp
<string>]
```

```
[-RemoteSharePassword <string>] [-RemoteSharePath <string>] -RemoteShareType <string>
[-RemoteShareUsername <string>] [-Timeout
<uint>] [-Username <string>] [-XtraProperty <Hashtable>] [-Force] [<CommonParameters>]
```

```
Start-ImcOsInstallation -OsInstallationController <OsiController> [-AnswerFilePassword
<string>]
[-AnswerFileShareFile <string>] [-AnswerFileShareIp <string>] [-AnswerFileSharePath
<string>] -AnswerFileShareType <string>
[-AnswerFileUsername <string>] [-ConfigShareFile <string>] [-ConfigShareIp <string>]
[-ConfigSharePassword <string>] [-ConfigSharePath <string>] -ConfigShareType <string>
[-ConfigShareUsername <string>] -IsoShare <string>
[-IsoShareIp <string>] -IsoShareType <string> [-Password <string>] [-RemoteShareFile
<string>]
[-RemoteShareIp <string>] [-RemoteSharePassword <string>] [-RemoteSharePath <string>]
-RemoteShareType <string> [-RemoteShareUsername
<string>] [-Timeout <uint>] [-Username <string>] [-XtraProperty <Hashtable>] [-Force]
[<CommonParameters>]
```

## 例

```
Get-ImcOsInstallation | Start-ImcOsInstallation -AnswerFileShareIp 10.10.10.10
-AnswerFileUsername root -AnswerFilePassword <password> -AnswerFileSharePath
"/root/test/osi"
-AnswerFileShareFile "" -AnswerFileShareType scp -ConfigShareIp 10.10.10.10
-ConfigShareUsername root
-ConfigSharePassword <password> -ConfigSharePath "/root/test/osi" -ConfigShareFile
"conf_file1" -ConfigShareType scp
-IsoShareIp 11.11.11.11 -IsoShare "/nfsshare/ucs-cxxx-scu-5.0.1a.iso" -IsoShareType nfs
-Username administrator
-Password <password> -RemoteShareIp 10.10.10.10 -RemoteShareUsername root
-RemoteSharePassword <password>
-RemoteSharePath "/root/test/osi" -RemoteShareFile "" -RemoteShareType scp -Force
```

```
Get-ImcOsInstallationController | Start-ImcOsInstallation -AnswerFileShareIp 10.10.10.10
-AnswerFileUsername root -AnswerFilePassword <password> -AnswerFileSharePath
"/root/test/osi"
-AnswerFileShareFile "" -AnswerFileShareType scp -ConfigShareIp 10.10.10.10
-ConfigShareUsername root
-ConfigSharePassword <password> -ConfigSharePath "/root/test/osi" -ConfigShareFile
"conf_file1"
-ConfigShareType scp -IsoShareIp 11.11.11.11 -IsoShare "/nfsshare/ucs-cxxx-scu-5.0.1a.iso"
-IsoShareType nfs -Username administrator -Password <password> -RemoteShareIp 10.10.10.10
-RemoteShareUsername root -RemoteSharePassword <password> -RemoteSharePath "/root/test/osi"
-RemoteShareFile "" -RemoteShareType scp -Force
```

# Test-ImcLdapBinding

IMC サーバでの LDAP バインドのテスト

## Syntax

```
Get-ImcLdapCACertificate | Test-ImcLdapBinding -User "user"  
-Pwd "Password123" -Force
```

# トランザクション サポート

# 次の cmdlet を使用して、トランザクションを開始します

```
Start-ImcTransaction
```

# 次の cmdlet を使用して、操作を実行します

```
$adapterHostEthIf = Get-ImcadapterUnit | Add-ImcadapterHostEthIf -Name adapterHostEth  
$adapterHostEthIfModify = $adapterHostEthIf | Set-ImcadapterHostEthIf -PxeBoot enabled  
$adapterEthISCSIProfile = $adapterHostEthIfModify | Add-ImcadapterEthISCSIProfile  
-InitiatorName adapterHostEth -InitiatorIPAddress xx.xx.xx.xx -InitiatorSubnetMask  
255.255.255.0 -DhcpISCSI enabled  
$adapterEthISCSIProfile | Remove-ImcadapterEthISCSIProfile  
$adapterHostEthIfModify | Remove-ImcadapterHostEthIf
```

# 次の cmdlet を使用して、トランザクションを終了します

```
Complete-ImcTransaction
```

# 次の cmdlet を使用して、トランザクションを元に戻します。

```
Undo-ImcTransaction
```

## vMedia の設定

次の cmdlet を使用して、vMedia を設定します

```
Get-ImcCommVMedia | Set-ImcCommVMedia -AdminState "enabled" -EncryptionState "enabled"  
-Force
```

## vNIC/アダプタの作成

vNIC/アダプタの作成

```
Get-ImcAdaptorUnit -Id "1" | Add-ImcAdaptorHostEthIf -Name "eth2" -UplinkPort "0"
```

## Cisco UCS コミュニティ

Cisco UCS コミュニティはシスコの製品およびテクノロジーについて議論、情報共有、学習するためのプラットフォームです。Cisco UCS コミュニティ パートナー エコシステムと UCS の統合に関するブログ、ディスカッションフォーラム、およびドキュメントについては、<https://communities.cisco.com/ucsintegrations> にアクセスしてください。

## 関連する Cisco IMC のマニュアルとマニュアル フィードバック

詳細については、次のリンクから関連資料を参照できます。

- [Cisco UCS C-Series Documentation Roadmap](#)
- Cisco UCS C シリーズ サーバの [Cisco IMC XML API プログラマー ガイド](#)
- [Cisco UCS E-Series Documentation Roadmap](#)
- Cisco UCS E シリーズ サーバの [Cisco IMC XML API プログラマー ガイド](#)

## マニュアルの入手方法およびテクニカル サポート

マニュアルの入手、サービス要求の提出、および追加情報の収集については、下記 <http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html> の『*What's New in Cisco Product Documentation*』を参照してください。

『*What's New in Cisco Product Documentation*』は、シスコの新規および改訂版の技術マニュアルの一覧も示し、RSS フィードとして購読できます。また、リーダー アプリケーションを使用してコンテンツをデスクトップに配信することもできます。RSS フィードは無料のサービスです。

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. シスコの商標の一覧については、[www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks) をご覧ください。掲載されている第三者の商標はそれぞれの権利者の財産です。「パートナー」または「partner」という用語は、シスコと他社との間のパートナーシップ関係を意味するものではありません。(1110R)

このマニュアルで使用している IP アドレスは、実際のアドレスを示すものではありません。マニュアル内の例、コマンド出力、および図は、説明のみを目的として使用されています。説明の中に実際のアドレスが使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。