



## **Cisco Integrated Management Controller (IMC) PowerTool ユーザ ガイド、リリース 2.x**

初版：2016年3月1日

最終更新：2019年1月18日

### **シスコシステムズ合同会社**

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>





## 目次

---

第 1 章	はじめに 1
	Cisco IMC PowerTool の概要 1
	管理情報モデル 1
	システム要件 3
	Cisco IMC PowerTool マッピング 4

---

第 2 章	使用する前に 7
	Cisco IMC に接続します 7
	デフォルトの Cisco IMC 8
	複数の Cisco IMC への接続 9
	ファイルに対する資格情報の読み取り/書き込み 9
	SSL の処理 10
	エイリアス 10

---

第 3 章	例 13
	Cisco IMC ファームウェアのアクティブ化 14
	ユーザの追加 14
	Cisco IMC Desired State Configuration (DSC) 15
	ImManagedObject リソース 15
	Cisco IMC ファームウェアを更新する 19
	ブートドライブのクリア 19
	[NTP 設定の構成 (Configure NTP Settings) ] 19
	Confirm フラグ 19
	SoL の設定 20

仮想ドライブの作成	20
ドライブセキュリティの無効化	20
ドライブセキュリティの有効化	21
Enable ImcPidCatalog	21
IP ブロッキングの有効化	21
Export-ImcHardwareInventory	21
Filters	22
Force フラグ	23
アダプタおよびコントローラの情報の取得	23
Get ImcKmpDownloadStatus	23
Get-ImcKmpUploadStatus	24
HUU ファームウェアの更新	24
SD カードを通じた HUU ファームウェアの更新	24
ドライブセキュリティ情報の変更	25
管理対象オブジェクトの同期	25
Syslog 設定の変更	26
新しい証明書署名要求	26
PowerTool コマンドレットの生成	26
IMC の証明書の受け取り	27
Receive-ImcKmpEntity	27
Receive-ImcLdapCACertificate	27
Remove-ImcLdapCACertificate	28
Reset-ImcEventFilters	28
Send-ImcBiosProfile	28
Send-ImcKmpEntity	28
Send-ImcLdapCACertificate	29
Send-ImcPidCatalog	29
サーバの操作	30
ブートドライブの設定	30
ディスク モード (JBOD から UG、およびその逆) の変更	30
ブート順序の設定	31
BIOS パスワードの設定	31

Start-ImcOsInstallation	31
Test-ImcLdapBinding	32
トランザクションサポート	33
vMedia の設定	33
vNIC/アダプタの作成	33
Cisco UCS コミュニティ	34
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	34
マニュアルの入手方法およびテクニカルサポート	34





# 第 1 章

## はじめに

---

この章の内容は、次のとおりです。

- [Cisco IMC PowerTool の概要](#) (1 ページ)
- [管理情報モデル](#) (1 ページ)
- [システム要件](#) (3 ページ)

## Cisco IMC PowerTool の概要

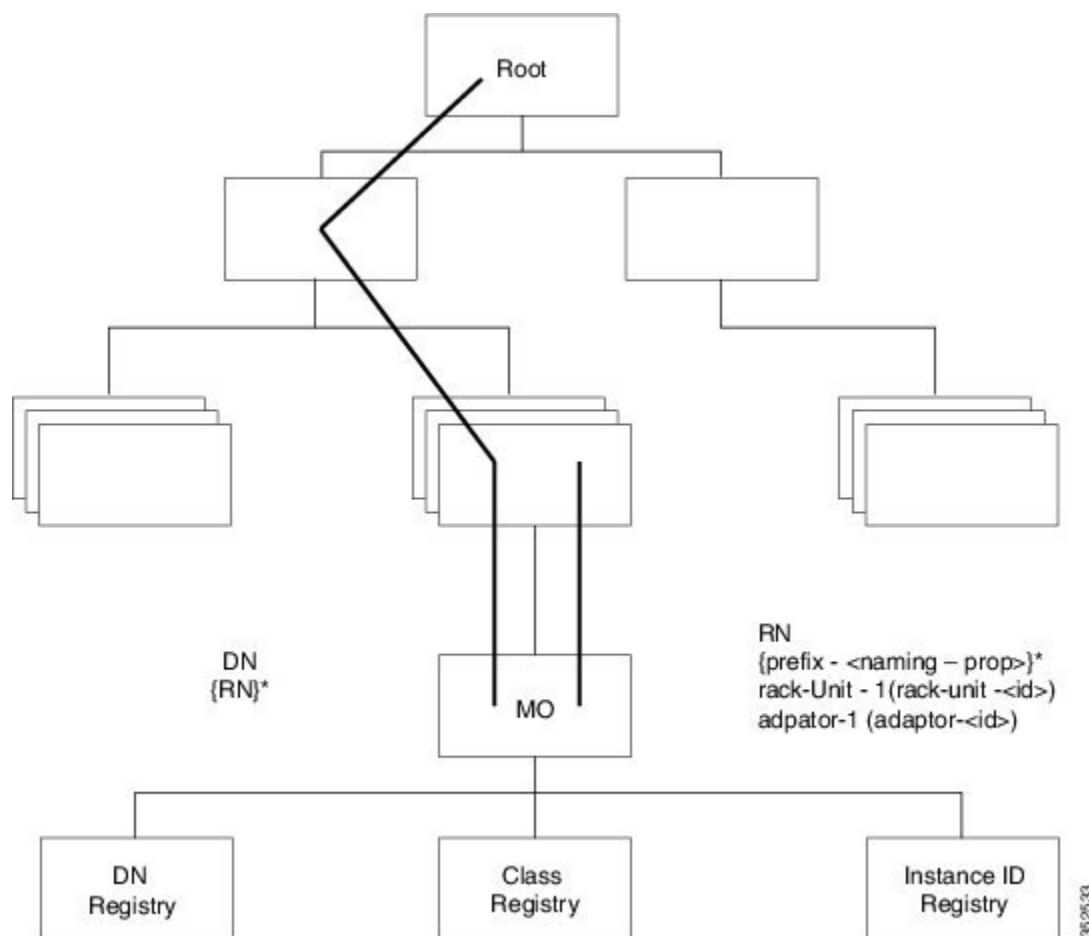
Cisco IMC PowerTool は、XML API を使用して Cisco IMC のさまざまな側面の自動化を支援する PowerShell モジュールです。既存の IT 管理プロセスやツールとの容易な統合を可能にします。

PowerTool コマンドレットは、Cisco IMC 管理情報ツリー (MIT) で機能します。コマンドレットを使用して、ツリーの管理対象オブジェクト (MO) に対するアクションを作成、変更、削除することができます。

## 管理情報モデル

Cisco IMC を構成するすべての物理および論理コンポーネントは、MIT と呼ばれる階層型の管理情報モデル (MIM) で表されます。ツリー内の各ノードは管理対象オブジェクト (MO) を表し、一意の識別名 (DN) で識別されます。

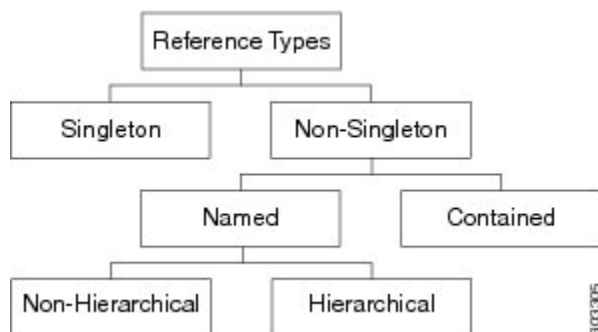
管理情報モデル



### 管理対象オブジェクト

管理対象オブジェクトは、CPU、DIMM、アダプタカード、ファン、電源装置などの Cisco IMC MIT リソースを抽象化したものです。MO は、Cisco IMC MIT で設定または管理される論理または物理エンティティを表します。たとえば、CPU、DIMM、アダプタカード、ファンなどの物理エンティティや、ユーザ、通信サービス（HTTP、SSH など）などの論理エンティティが MO として表されます。

### 管理対象オブジェクト



ツリー内では、各 MO は識別名 (DN) によって識別されます。親コンテキスト内では、相対名 (RN) によって MO を識別できます。DN は、MIT での MO の位置を識別します。DN は、ルートから開始してその MO 自身までのすべての相対名を連結したものです。基本形は、DN = [RN]/[RN]/[RN]/.../[RN] です。

次の例では、DN はモデル内の adapter-1 の完全修飾名を示しています。

```
< dn = "sys/rack-unit-1/adapter-1" />
```

この DN は次の RN から構成されています。

```
topSystem MO: rn="sys" computeRackUnit MO: rn="rack-unit-1" adapterUnit MO: rn="adapter-<id>"
```

RN には、1 つ以上の MO プロパティの値を組み込むことができます。これによって、親コンテキスト内で同タイプの複数の MO を区別することができます。RN の一部を成すプロパティは、Naming プロパティと呼ばれます。

たとえば、1 つのラックユニット MO の下には複数のアダプタ MO があります。アダプタ MO には RN の一部としてアダプタ ID (adapter-[Id]) が含まれているので、ラックユニットのコンテキストで各アダプタ MO を一意に識別できます。

## システム要件

Cisco IMC PowerTool をインストールする前に、システムが次の要件を満たしていることを確認します。

- Windows PowerShell 3.0 以降をインストール
- .NET Framework バージョン 4.5 以降
- Windows PowerShell 4.0 以降 (DSC の場合)

### Cisco UCS C シリーズ サーバ

Cisco IMC PowerTool は、次の Cisco IMC リリースと互換性があります。

- Release 4.0
- リリース 3.1
- リリース 3.0
- リリース 2.0 以降
- リリース 1.5 以降

### Cisco UCS E-Series Servers

Cisco IMC PowerTool は、次の Cisco UCS E シリーズ リリースと互換性があります。

- E シリーズ サーバ向けリリース 2.2(1) 以降

## Methods

システムの管理および監視には、Cisco IMC XML API メソッドが使用されます。次の方式がサポートされています。

- 認証
- aaaLogin—初期ログイン方式
- aaaRefresh—現在の認証 Cookie を更新します
- aaaLogout—現在のセッションを終了し、対応する認証 Cookie を非アクティブ化します
- configResolveDn — DN によってオブジェクトを取得します
- configResolveClass —指定されたクラスのオブジェクトを取得します。
- configResolveChildren—オブジェクトの子オブジェクトを取得します
- configResolveParent—オブジェクトの親オブジェクトを取得します
- configConfMo—単一の管理対象オブジェクトに影響します。たとえば、DN
- eventSubscribe—イベントへの登録に使用されます

## Cisco IMC PowerTool マッピング

ほとんどの Cisco IMC PowerTool コマンドレットは MO の仕様から生成されます。タイプの代わりに名詞が使用されます (EquipmentFan の代わりに Fan など)。さまざまな MO タイプに対して、Get、Add、Set、Remove コマンドレットまたはサブセットが生成されます。すべてのコマンドレットは、XML 要求および応答を画面にダンプする XML パラメータをサポートします。

### コマンドレットを追加します

-指定されたプロパティ値とともに MO ステータス「created」で ConfigConfMo メソッドを使用します。Force パラメータが指定されている場合、確認を求めるプロンプトは表示されません。

### コマンドレットを取得します

-ConfigResolveClass メソッドを使用して MO を取得します。Cisco IMC の XML API はいずれのフィルタもサポートしていません。プロパティパラメータを指定すると、PowerTool は指定されたクラスのインスタンスを収集し、プロパティ値を使用してクライアント側でフィルタリングします。

### Set コマンドレット

-指定されたプロパティ値とともに MO ステータス「modified」で ConfigConfMo メソッドを使用します。Force パラメータが指定されている場合、確認を求めるプロンプトは表示されません。

### コマンドレットを削除します

-MO ステータス「deleted」を指定して ConfigConfMo メソッドを使用します。Force パラメータが指定されている場合、確認を求めるプロンプトは表示されません。

次の表は、特定の verb に指定できるプロパティのリストです。

プロパティ	get	Add	Set
Naming	Yes (位置指定)	Yes (位置指定)	No
Create-Only	Yes	Yes	No
Read-Write	Yes	Yes	Yes
Operational/ Read-Only	はい	No	×

次の表は、対応するコマンドレットのパイプラインに表示されるタイプのリストです。

動詞/タイプ	パイプラインの入力
get	Singleton-None non-singleton-Parent Type
[Add	Singleton-None non-singleton-Parent Type
セット	Naming プロパティがある MO – Naming プロパティがない同一タイプの MO – 同一タイプまたは親タイプ
Remove/Clear	同一タイプ

次の表は、必要な XML 要求を生成するために呼び出されるメソッドを示します。

コマンドレット	方式
Add-Imc Set-Imc	ConfigConfMo
Get-Imc	ConfigResolveClass (クライアント側でフィルタ)
Get-ImcManagedObject -ClassId	ConfigResolveClass
Get-ImcManagedObject -ClassId -Dnlist	ConfigResolveClass (一致する DN を検出するため出力がフィルタリングされます)
Get-ImcManagedObject -Dn	ConfigResolveDn
Connect-Imc	AaaLogin
Disconnect-Imc	AaaLogout

コマンドレット	方式
Background これはコマンドレットではありません。バックグラウンド サービスです。	AaaRefresh
Get-ImcChild	ConfigResolveChildren

Get-ImcCmdletMeta は、MO のタイプ、対応する名詞、サポートされる動詞、および MO のプロパティを検出するために使用されます。プロパティの詳細 (Naming、Read、Write などのタイプ、プロパティが導入された Cisco IMC のバージョンなど) の表示にも使用されます。



## 第 2 章

# 使用する前に

---

この章は、次の項で構成されています。

- [Cisco IMC に接続します \(7 ページ\)](#)
- [デフォルトの Cisco IMC \(8 ページ\)](#)
- [複数の Cisco IMC への接続 \(9 ページ\)](#)
- [ファイルに対する資格情報の読み取り/書き込み \(9 ページ\)](#)
- [SSL の処理 \(10 ページ\)](#)
- [エイリアス \(10 ページ\)](#)

## Cisco IMC に接続します

---

**ステップ 1** デスクトップ ショートカットから IMC PowerTool を起動します。

**ステップ 2** 次のコマンドレットを使用して、Cisco UCS PowerTool でサポートされるすべてのコマンドレット、機能、およびエイリアスを表示します。

```
Get-Command -Module Cisco.Imc
Get-Command -Module Cisco.Imc | group CommandType
Get-Command -Module Cisco.Imc | measure
```

**ステップ 3** 次のコマンドレットを使用して、Cisco IMC に接続します。

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

-NotDefault オプションが指定されていない場合、デフォルトでは、ログイン後、Cisco IMC ハンドルがデフォルトの Cisco IMC リストに追加されます。Cisco IMC で実行されるすべてのコマンドレットで、-Imc パラメータを付けてハンドルを指定できます。

**ステップ 4** 次のコマンドレットを使用して、プロキシを使用して Cisco IMC に接続します。

```
$proxy = New-Object System.Net.WebProxy
$proxy.Address = "http:\\<url>:<port>"
$proxy.UseDefaultCredentials = $false
$proxy.Credentials = New-Object System.Net.NetworkCredential("<user name>", "<password>")
$handle = Connect-Imc <ip or hostname> -Proxy $proxy
```

**ステップ 5** 次のコマンドレットを使用します。

- a) Cisco IMC から統合ステータス情報を取得します。

```
Get-ImcStatus -Imc $handle
```

- b) Cisco IMC のインベントリ サマリーを取得します。

```
Get-ImcRackUnit -Imc $handle
```

- c) 接続を切断します。

```
Disconnect-Imc -Imc $handle
```

## デフォルトの Cisco IMC

ハンドルも名前も指定されていない場合は、Cisco IMC ハンドルが `DefaultImc` サーバリストに追加されます（ただし、`-Imc` パラメータが指定されている場合を除く）。パイプラインの最初のコマンドレットは、デフォルトの Cisco IMC リストで実行されます。

**Cisco IMC に接続します。**

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

**デフォルトの Cisco IMC を取得します。**

```
Get-UcsPsSession
```

**ステータス情報と Cisco IMC のバージョンを取得します。**

```
Get-ImcStatus
```

**Cisco IMC サーバの詳細を取得します。**

```
Get-ImcRackUnit
```

**Cisco IMC で HTTP を有効にします。**

```
Get-ImcHttp | Set-ImcHttp -AdminState enabled
```

**Cisco IMC で HTTP を無効にします。**

```
Get-ImcHttp | Set-ImcHttp -AdminState disabled
```

**Cisco IMC を接続解除します。**

```
Disconnect-Imc
```

## 複数の Cisco IMC への接続

複数のハンドルを指定すると、Cisco IMC PowerTool コマンドレットを複数の Cisco IMC と連携させることができます。

次のコマンドレットを使用して、複数の IMC に接続します。

Cisco IMC に接続しています。

```
$handle1 = Connect-Imc <ip1> -NotDefault
$handle2 = Connect-Imc <ip2> -NotDefault
Get-ImcStatus -Imc $handle1,$handle2
Disconnect-Imc -Imc $handle1,$handle2
```

デフォルトでは、複数の Cisco IMC ハンドルは **DefaultImc** では許可されません。

**Set-UcsPowerToolConfiguration** コマンドレットを使用して、この制限を上書きできます。

```
Get-UcsPowerToolConfiguration
Set-UcsPowerToolConfiguration -SupportMultipleDefaultUcs $true
Connect-Imc <ip1>
Connect-Imc <ip2>
Get-ImcStatus
Disconnect-Imc
```

複数の Cisco IMC に接続しています。

Cisco IMC への接続に使用する資格情報を使用することができます。

```
$user = "<username>"
$password = "<password>" |
ConvertTo-SecureString -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential($user, $password)
$servers = @("<Imc1>", "<Imc2>", "<Imc3>")
Connect-Imc $servers -Credential $cred
```

## ファイルに対する資格情報の読み取り/書き込み

```
Connect-Imc <ip1>
Connect-Imc <ip2>
```

資格情報はファイルに保存できます。保存された資格情報は、指定されたキーで暗号化されます。

```
Export-UcsPsSession -LiteralPath C:\work\labs.xml
Disconnect-Imc
```

ログインはファイルに保存されている資格情報から開始できます。

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

ファイルに保存されている資格情報でログイン中にプロキシを指定します。

```
$proxy = New-Object System.Net.WebProxy
$proxy.Address = "http:\\<url>:<port>"
$proxy.UseDefaultCredentials = $false
$proxy.Credentials = New-Object System.Net.NetworkCredential("<user name>", "<password>")
Connect-Imc -LiteralPath C:\work\lab.xml -Proxy $proxy
```

別のシステムにログインして、ファイルに資格情報を追加します。

```
Connect-Imc <ip3>
Export-UcsPSSession -Path C:\work\lab.xml -Merge
```

## SSL の処理

Cisco IMC に接続するときに、サーバは有効な証明書を認識しません。接続は `InvalidCertificateAction` によって決まります。 `InvalidCertificateAction` は、証明書を無視するようにデフォルトで設定されます。デフォルトでは、Cisco IMC PowerTool は有効な証明書なしで接続を確立するように設定されます。

**Set-UcsPowerToolConfiguration** コマンドレットを使用して、この設定を上書きできます。

```
Get-UcsPowerToolConfiguration
Set-UcsPowerToolConfiguration -InvalidCertificateAction Fail
```

次の表に、証明書の有効性を確認するためのオプションを示します。

オプション	説明
Fail	証明書が無効な場合、コマンドレットは接続を確立しません。
Ignore	コマンドレットは、証明書が無効であることを考慮せずに、接続を確立します。
Default	(Windows のデフォルト) 証明書が有効な場合、コマンドレットは接続を確立します。

## エイリアス

便宜上、いくつかのエイリアスが事前定義されます。すべてのエイリアスの一覧を表示するには、次のコマンドレットを実行します。

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

次の表に、エイリアスと対応するコマンドレットを示します。

Alias	コマンドレット
Clear-ImcKmpLogin	Get-ImcKmpServerLogin   Set-ImcKmpServerLogin -AdminAction clear
Clear-ImcKmpServer	Get-ImcKmpServer   Set-ImcKmpServer -AdminAction clear
Remove-ImcRootCACertificate	Get ImcKmpManagement   Set- ImcKmpManagement -AdminAction delete-root-ca-certificate
Remove-ImcClientCertificate	Get ImcKmpManagement   Set- ImcKmpManagement -AdminAction delete-client-certificate
Remove-ImcClientPrivateKey	Get ImcKmpManagement   Set- ImcKmpManagement -AdminAction delete-client-private-key
Enable ImcBiosProfile	Get-ImcBiosProfile  Set-ImcBiosProfile -AdminAction activate
Remove-ImcBiosProfile	Get ImcBiosProfile  Set ImcBiosProfile - AdminAction delete
Backup-ImcBiosProfile	Get-ImcBiosProfileManagement   Set-ImcBiosProfileManagement -AdminAction backup
Clear-ImcOneTimePrecisionBoot Device	Get-ImcOneTimePrecisionBoot Device   Set ImcOneTimePrecisionBoot Device -AdminAction clear-one-time-boot-device
Reset-ImcStorageController	Get-ImcStorageController   Set-ImcStorageController -AdminAction delete-all-vds-reset-pds
Clear-ImcBootDrive	Get-ImcStorageController   Set-ImcStorageController -AdminAction clear-boot-drive
Clear-ImcForeignConfig	Get-ImcStorageController   Set-ImcStorageController -AdminAction clear-foreign-config
Disable ImcJbod	Get-ImcStorageController   Set-ImcStorageController -AdminAction disable-jbod

Alias	コマンドレット
Enable-ImcJbod	Get-ImcStorageController   Set-ImcStorageController -AdminAction enable-jbod
Get-ImcTtyLog	Get-ImcStorageController   Set-ImcStorageController -AdminAction get-tty-log
Import-ImcForeignConfig	Get-ImcStorageController  Set-ImcStorageController -AdminAction import-foreign-config
Add-ImcMo	Add-ImcManagedObject
Disable-ImcLocatorLed	Set-ImcLocatorLed -AdminState off
Enable-ImcLocatorLed	Set-ImcLocatorLed -AdminState on
Enable-ImcPidCatalog	Set-ImcActivatePIDCatalog -AdminState trigger
Get-ImcMo	Get-ImcManagedObject
Remove-ImcLdapCertificate	Set-ImcLdapCACertificate -AdminAction delete-ca-certificate
Remove-ImcMo	Remove-ImcManagedObject
Reset-ImcServer	Set-ImcRackUnit -AdminPower hard-reset-immediate
Reset-ImcEventFileters	Set-ImcEventManager -AdminAction reset-event-filters
Restart-ImcServer	Set-ImcRackUnit -AdminPower cycle-immediate
Set-ImcMo	Set-ImcManagedObject
Start-ImcServer	Set-ImcRackUnit -AdminPower up
Stop-ImcServer	Set-ImcRackUnit -AdminPower soft-shut-down
Invoke-ImcPowerCharacterization	Set-ImcPowerBudget -AdminAction start-power-char
Reset-ImcPowerProfile	Set-ImcPowerBudget -AdminAction reset-power-profile-default
Test-ImcLdapBinding	Set-ImcLdapCACertificate -AdminAction test-ldap-binding



## 第 3 章

### 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 (14 ページ)
- ユーザの追加 (14 ページ)
- Cisco IMC Desired State Configuration (DSC) (15 ページ)
- Cisco IMC ファームウェアを更新する (19 ページ)
- ブート ドライブのクリア (19 ページ)
- [NTP 設定の構成 (Configure NTP Settings) ] (19 ページ)
- Confirm フラグ (19 ページ)
- SoL の設定 (20 ページ)
- 仮想ドライブの作成 (20 ページ)
- ドライブセキュリティの無効化 (20 ページ)
- ドライブセキュリティの有効化 (21 ページ)
- Enable ImcPidCatalog (21 ページ)
- IP ブロッキングの有効化 (21 ページ)
- Export-ImcHardwareInventory (21 ページ)
- Filters (22 ページ)
- Force フラグ (23 ページ)
- アダプタおよびコントローラの情報の取得 (23 ページ)
- Get ImcKmpDownloadStatus (23 ページ)
- Get-ImcKmpUploadStatus (24 ページ)
- HUU ファームウェアの更新 (24 ページ)
- SD カードを通じた HUU ファームウェアの更新 (24 ページ)
- ドライブセキュリティ情報の変更 (25 ページ)
- 管理対象オブジェクトの同期 (25 ページ)
- Syslog 設定の変更 (26 ページ)
- 新しい証明書署名要求 (26 ページ)
- PowerTool コマンドレットの生成 (26 ページ)
- IMC の証明書の受け取り (27 ページ)
- Receive-ImcKmpEntity (27 ページ)

- [Receive-ImcLdapCACertificate](#) (27 ページ)
- [Remove-ImcLdapCACertificate](#) (28 ページ)
- [Reset-ImcEventFilters](#) (28 ページ)
- [Send-ImcBiosProfile](#) (28 ページ)
- [Send-ImcKmpEntity](#) (28 ページ)
- [Send-ImcLdapCACertificate](#) (29 ページ)
- [Send-ImcPidCatalog](#) (29 ページ)
- [サーバの操作](#) (30 ページ)
- [ブート ドライブの設定](#) (30 ページ)
- [ディスク モード \(JBOD から UG、およびその逆\) の変更](#) (30 ページ)
- [ブート順序の設定](#) (31 ページ)
- [BIOS パスワードの設定](#) (31 ページ)
- [Start-ImcOsInstallation](#) (31 ページ)
- [Test-ImcLdapBinding](#) (32 ページ)
- [トランザクション サポート](#) (33 ページ)
- [vMedia の設定](#) (33 ページ)
- [vNIC/アダプタの作成](#) (33 ページ)
- [Cisco UCS コミュニティ](#) (34 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](#) (34 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](#) (34 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。 構文： Name=<ipAddress> [`nNoSsl=<bool>][`nPort=<ushort>] [`nProxyAddress=<proxyAddress>] [`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。  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。  構文：  `<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。 構文： Name=<ipAddress> [`nNoSsl=<bool>][`nPort=<ushort>] [`nProxyAddress=<proxyAddress>] [`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。  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-ImcSnmpUser -Name snmpuser | Clear-ImcSnmpUser -Confirm
Confirm
Are you sure you want to perform this action?
Performing the operation "Clear-ImcSnmpUser" 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 ロギング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 コマンドレット	PowerTool 1.4.1 以降の cmdlet
Power On Server	Get-ImcRackUnit   Set-ImcRackUnit -AdminPower up	Get-ImcRackUnit   Start-ImcServer
Power Off Server	Get-ImcRackUnit   Set-ImcRackUnit -AdminPower soft-shut-down	Get-ImcRackUnit   Stop-ImcServer
Power Cycle Server	Get-ImcRackUnit   Set-ImcRackUnit -AdminPower cycle-immediate	Get-ImcRackUnit   Restart-ImcServer
Hard Reset Server	Get-ImcRackUnit   Set-ImcRackUnit -AdminPower hard-reset-immediate	Get-ImcRackUnit   Reset-ImcServer
Turn On Locator LED	Get-ImcLocatorLed   Set-ImcLocatorLed -AdminState on	Get-ImcLocatorLed   Enable-ImcLocatorLed
Turn Off Locator LED	Get-ImcLocatorLed   Set-ImcLocatorLed -AdminState off	Get-ImcLocatorLed   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。