



Cisco Integrated Management Controller PowerTool ユーザガイドリ

リース 2.x

1. 概要
2. 管理情報モデル
3. システム要件
4. 例
5. サンプル
6. 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
7. マニュアルの入手方法およびテクニカル サポート

1 概要

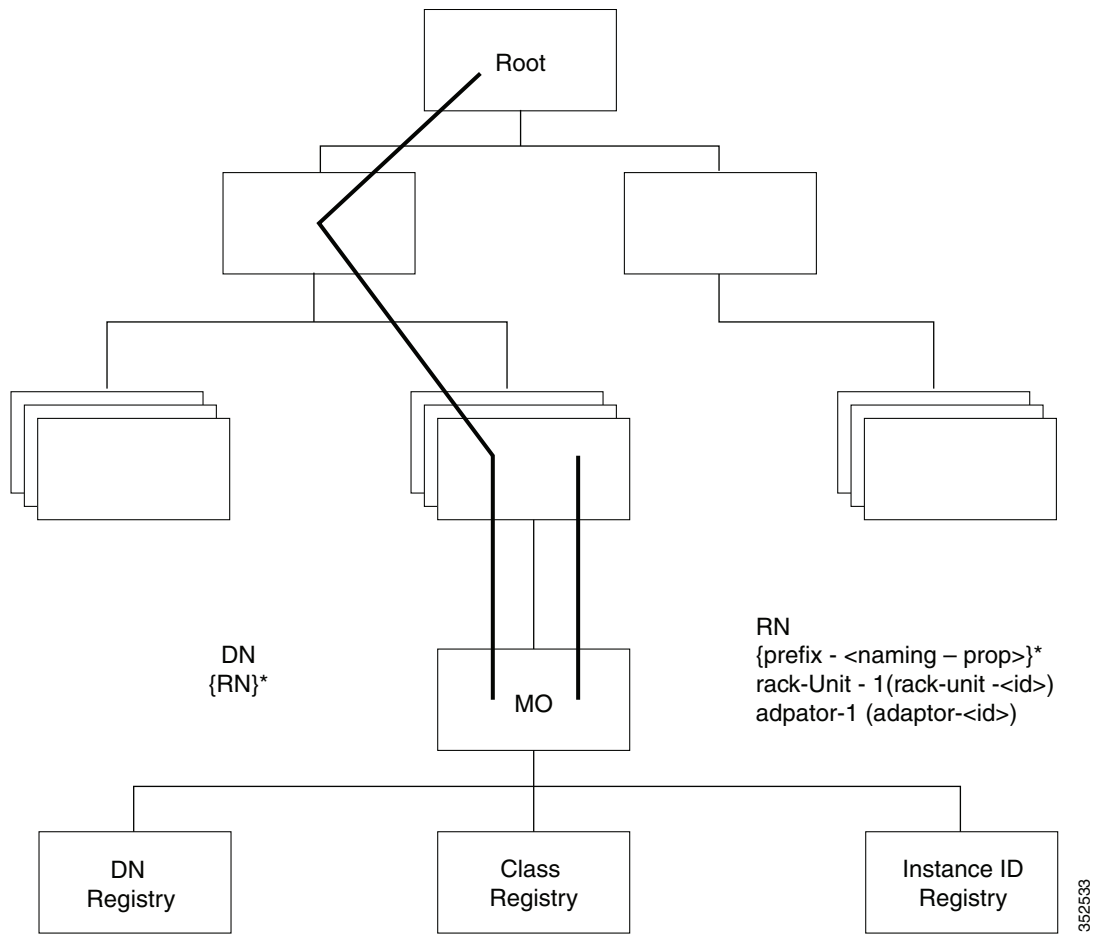
Cisco IMC PowerTool は、XML API を使用して Cisco IMC のさまざまな側面の自動化を支援する PowerShell モジュールです。Cisco IMC PowerTool は既存の IT 管理プロセスやツールとの容易な統合を可能にします。

Cisco IMC PowerTool のコマンドレットは、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。

2 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。[図 1](#)を参照してください。

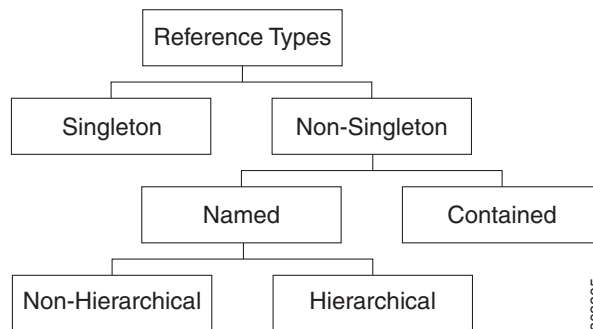
図1 管理情報モデル



管理対象オブジェクト

管理対象オブジェクト (MO) (図2を参照)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。

図2 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 を識別できません。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。

3 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 以降がインストールされている必要があります。
- .NET Framework バージョン 4.5 以降
- Windows PowerShell 4.0 以降 (DSC の場合)

Cisco IMC

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

- C シリーズ サーバ向けリリース 1.5 以降
- 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 によりオブジェクトを取得します。
- configResolveClass: 該当する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: 1 つの管理対象オブジェクト (例: DN) に影響します。
- 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。

Add 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「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。

Get コマンドレット: 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。

Remove コマンドレット: 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
名前付け	Yes(位置指定)	Yes(位置指定)	No
Create-Only	Yes	Yes	No
Read-Write	Yes	Yes	Yes
Operational/ Read-Only	Yes	No	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 – Non-Singleton なし – 親タイプ
Add	Singleton – Non-Singleton なし – 親タイプ
Set	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
Background ¹	AaaRefresh
Get-ImcChild	ConfigResolveChildren

1. 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。

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。また、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。

使用する前に

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 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 1 つの Cisco IMC に接続します。

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



注

-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 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。

1 つの Cisco IMC に接続します。

```
$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
```

同じログイン クレデンシャルを使用して複数の Cisco IMC ドメインに接続します。

```
$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。接続の確立は `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。

	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、証明書が無効かどうかにかかわらず接続を確立します。
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定義されています。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。

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。

エイリアス	コマンドレット
Add-ImcMo	Add-ImcManagedObject
Disable-ImcLocatorLed	Set-ImcLocatorLed -AdminState off
Enable-ImcLocatorLed	Set-ImcLocatorLed -AdminState on
Get-ImcMo	Get-ImcManagedObject
Remove-ImcMo	Remove-ImcManagedObject
Reset-ImcServer	Set-ImcRackUnit -AdminPower hard-reset-immediate
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
Set-ImcPowerBudget -AdminAction start-power-char	Invoke-ImcPowerCharacterization
Set-ImcPowerBudget -AdminAction reset-power-profile-default	Reset-ImcPowerProfile

4 例

以降の例は、コマンドレットの実行方法を示しています。

- [Force フラグ](#)
- [Confirm フラグ](#)
- [Cisco IMC Desired State Configuration \(DSC\)](#)
- [ユーザの追加](#)
- [IP ブロッキングの有効化](#)
- [NTP の設定](#)

- Syslog 設定の変更
- SoL の設定
- ブート順序の設定
- vMedia の設定
- アダプタおよびコントローラの情報の取得
- トランザクション サポート
- 管理対象オブジェクトの同期
- フィルタ
- HUU ファームウェアの更新
- Cisco IMC ファームウェアの更新
- Cisco IMC ファームウェアのアクティブ化
- 仮想ドライブの作成
- ブートドライブの設定
- ブートドライブのクリア
- PowerTool コマンドレットの生成
- BIOS パスワードの設定
- サーバの操作

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

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。

Syntax

```
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] ]
}
```

Properties

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>]
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には、このプロパティを Add 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、 Set 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 ResourceName 、そのタイプが ResourceType 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 Absent 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、 Present に設定します。デフォルトは Present です。

プロパティ	説明
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。

Syntax

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] ]
}

```

Properties

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。
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。
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 Add 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、 Set 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 ResourceName 、そのタイプが ResourceType 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 Present です。
WebProxyCredentials	Web プロキシの資格情報を示します。

Syntax

```

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"
    }
}
```

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
```

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 *****
```

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
```

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 セキュリティ情報の変更

ドライブ セキュリティ MO のセキュリティ キー/キー ID を更新します。

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

ユーザの追加

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



注 `Clear-ImcLocalUser` は、ステータスを非アクティブに変更するだけであり、ユーザやデータの削除は行いません。

IP ブロッキングの有効化

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

NTP の設定

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

Syslog 設定の変更

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

SoL の設定

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

ブート順序の設定

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

vMedia の設定

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

トランザクション サポート

トランザクションを開始する

```
Start-ImcTransaction
```

操作を実行する。

```
$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
```

トランザクションを修了する。

```
Complete-ImcTransaction
```

トランザクションを取り消す。

```
Undo-ImcTransaction
```

管理対象オブジェクトの同期

複数の Cisco IMC に接続するために SupportMultipleDefaultUcs を有効にする。

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

資格情報を取得して変数に格納する。

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

異なる Cisco IMC に接続する。

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

異なる Cisco IMC からローカル ユーザを取得する。

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

Cisco IMC2 の一連の MO を Cisco IMC1 に同期させる。

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

フィルタ

Type が「SEL」または「Syslog」に等しい SysdebugMEpLog 管理対象オブジェクトを取得する。

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

(Type が「SEL」または「Syslog」に等しく)かつ Id が「0」に等しく、Type が「SEL」に等しい SysdebugMEpLog 管理対象オブジェクトを取得する。

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

名前に「admin」などが含まれているローカル ユーザを取得する(大文字と小文字を区別)。

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

名前に「test*」などが含まれているユーザを取得する(* 正規表現をサポート、または大文字と小文字を区別)。

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

AccountStatus が「inactive」以外のローカル ユーザを取得する。

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

AccountStatus が「inacti」と一致するローカル ユーザを取得する。

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

AccountStatus が「active」と一致するローカル ユーザを取得する (active で始まる、または大文字と小文字を区別)。

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

AccountStatus が「active」と一致しないローカル ユーザを取得する (active で始まる、または大文字と小文字を区別)。

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

AccountStatus が「active」ではないローカル ユーザを取得する (active で始まる、または大文字と小文字を区別)。

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

HUU ファームウェアの更新

ユーザ資格情報を作成する

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

HUU ファームウェアを更新する

```
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
```

Cisco IMC ファームウェアの更新

ユーザ資格情報を作成する

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

Cisco IMC ファームウェアを更新する

```
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
```

Cisco IMC ファームウェアのアクティブ化

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

```
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
```

ブートドライブの設定

物理ドライブをブートドライブとして設定する

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

仮想ドライブをブートドライブとして設定する

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

ブートドライブのクリア

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

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 コマンドレットを生成するには、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"
```

BIOS パスワードの設定

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



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。

サーバの操作

サーバを操作するために追加された新しいエイリアスを次に示します。

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 以降のコマンドレット
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

5 サンプル

今回のリリースでは、Cisco IMC PowerTool を使用して開発されたサンプル スクリプトが communities.cisco.com で公開される予定です。

6 Cisco IMC 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。

7 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 IMC XML API Programmer's Guide for Cisco UCS C-Series Servers](#)
- [Cisco UCS E-Series Documentation Roadmap](#)
- [Cisco IMC XML API Programmer's Guide for Cisco UCS E-Series Servers](#)

このマニュアルに関する技術的なフィードバック、または誤りや記載漏れに関する報告は、ucs-docfeedback@external.cisco.com に送信してください。ご協力をよろしくお願いいたします。

8 マニュアルの入手方法およびテクニカル サポート

マニュアルの入手方法、テクニカル サポート、その他の有用な情報について、次の URL で、毎月更新される『*What's New in Cisco Product Documentation*』を参照してください。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されています。
<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

『*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. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (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。

© 2016 Cisco Systems, Inc. All rights reserved.

