

## ユーザ ガイド



### Cisco UCS Manager PowerTool ユーザ ガイド、リリース 1.x

- 1 概要
- 2 管理情報モデル
- 3 システム要件
- 4 インストール
- 5 例
- 6 サンプル
- 7 関連する Cisco UCS マニュアルとマニュアル フィードバック
- 8 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 UCS Manager PowerTool は、サーバ、ネットワーク、ストレージ、ハイパーバイザの管理など、Cisco UCS Manager のあらゆる側面での自動化を支援する PowerShell モジュールです。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。

PowerTool コマンドレットは UCS Manager の管理情報ツリー (MIT) で動作し、ツリーの管理対象オブジェクト (MO) の作成、編集、削除を行います。次の章では、Cisco UCS 管理情報モデル (MIM) および PowerTool コマンドレットとの関係について概要を説明します。

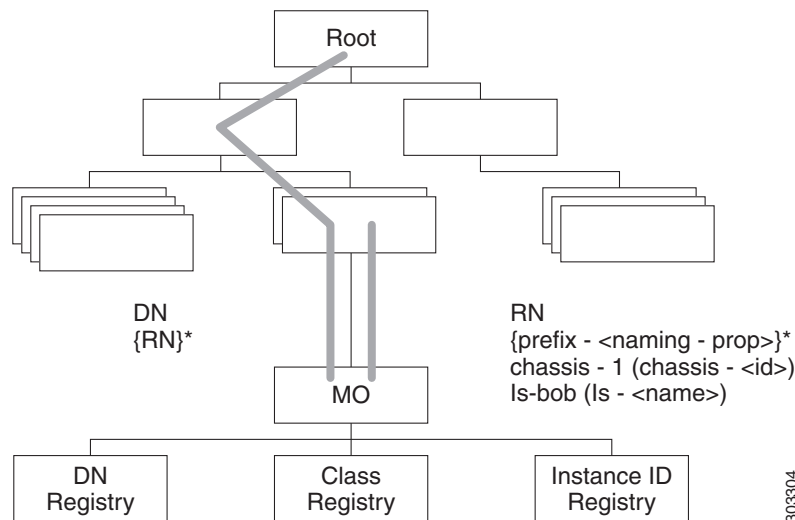
PowerTool での UCS 設定について学習する最も簡単な方法の 1 つは、GUI で行う設定作業について、PowerTool コマンドレットを生成し、[「PowerTool コマンドレットの生成」セクション \(13 ページ\)](#) で説明されている `ConvertTo-UcsCmdlet` を使用して実行していただくことです。

## 2 管理情報モデル

Cisco UCS ドメインを構成するすべての物理および論理コンポーネントは、管理情報ツリー (MIT) と呼ばれる階層的な管理情報モデル (MIM) で表されます。ツリー内の各ノードは、識別名 (DN) で識別される管理対象オブジェクト (MO) を表します。

図 1 は MIM についての説明です。

図 1 管理情報モデル



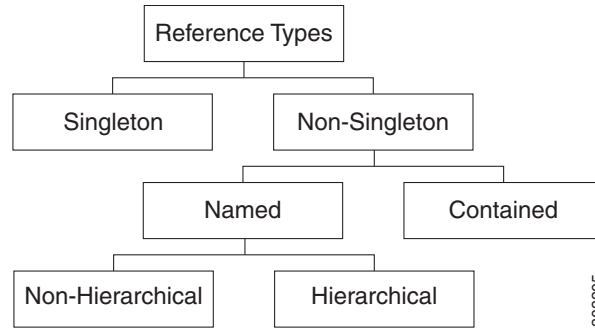
次の図に、シャーシが 3 つの場合のサンプル (一部) MIT を示します。

ツリー (topRoot)	識別名
— sys	sys
— chassis-1	sys/chassis-1
— chassis-2	sys/chassis-2
— chassis-3	sys/chassis-3
— blade-1	sys/chassis-3/blade-1
— adaptor-1	sys/chassis-3/blade-1/adaptor-1
— blade-2	sys/chassis-3/blade-2
— adaptor-1	sys/chassis-3/blade-2/adaptor-1
— adaptor-2	sys/chassis-3/blade-2/adaptor-2

## 管理対象オブジェクト

管理対象オブジェクト (MO) は、ファブリック インターコネクト、シャーシ、ブレード、ラックマウント サーバなどの Cisco UCS ドメイン リソースを抽象化したものです。管理対象オブジェクトは、Cisco UCS MIT で設定される、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、サーバ、シャーシ、I/O カード、およびプロセッサなどの物理エンティティ、リソース プール、ユーザ ロール、サービス プロファイル、およびポリシーなどの論理エンティティが、管理対象オブジェクトとして表されます。

図2 管理対象オブジェクト



それぞれの管理対象オブジェクトは、識別名 (DN) によってツリー内で特定でき、相対名 (RN) によって親のコンテキスト内で特定できます。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。基本的に、Dn = [Rn]/[Rn]/[Rn]/.../[Rn] の形になります。

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

```
< dn = "sys/chassis-5/blade-2/adaptor-1" />
```

上記の DN は次の RN で構成されます。

```
topSystem MO: rn="sys" equipmentChassis MO: rn="chassis-<id>" computeBlade MO: rn ="blade-<slotId>" adaptorUnit MO: rn="adaptor-<id>"
```

相対名 (RN) には、1 つ以上の MO の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、親のコンテキスト内で同じタイプの複数の MO を区別することができます。前述したような 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。

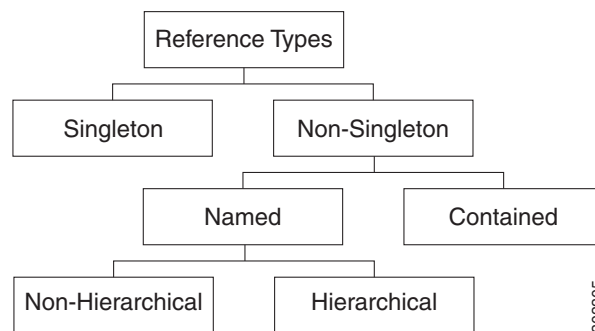
たとえば、複数のブレードの MO が 1 つのシャーシの MO の下にあります。ブレード MO には RN の一部としてブレードの ID 情報 (blade- [ID]) が含まれるので、シャーシのコンテキスト内で各ブレード MO を個別に識別できます。

## 管理対象オブジェクトの参照

管理対象オブジェクトの内容は、Cisco UCS の動作時に参照されます。MO の一部は暗黙的に参照されるか (ログイン時の PreLoginBanner)、別の MO の展開の一部として参照されます (サービス プロファイル MO はテンプレートを参照し、VNIC は多くの VLAN MO を参照します)。

参照の種類は、次のように分類できます。

図3 管理対象オブジェクトの参照



単一 MO タイプは、MIT 全体で一度だけ出現し、通常、暗黙的に参照されます。

非単一 MO タイプは MIT 内で 1 回以上 インスタンス化されることがあります。多くの場合、MO が別の MO を参照するとき、参照は名前で行われます。参照された MO のタイプに応じて、解決は階層的になる可能性があります。たとえば、サービスプロファイル テンプレートは Org の下に定義されているとします。Org には下位 Org が含まれていることがあるため、下位 Org に同じ名前で作成されたサービス プロファイル テンプレートがある場合があります。ここでサービス プロファイル インスタンスがサービス プロファイル テンプレートを (名前) で参照する場合、名前がサービス プロファイル インスタンスの Org からルート Org まで階層的に検索されます。最初に一致したエントリが使用されます。一致するエントリが見つからない場合、「default」という名前が同様の方法で検索され、最初に一致したエントリが使用されます。

参照タイプ	例
単一	ChassisDiscoveryPolicy PreLoginBanner
非単一/名前付き/非階層型	CallHomePolicy
非単一/名前付き/階層型	BiosPolicy BootPolicy
非単一/包含	LsServer (ServiceProfile) の下の BootDefinition VnicEther の下の VnicEtherIf

## 管理対象オブジェクトのプロパティ

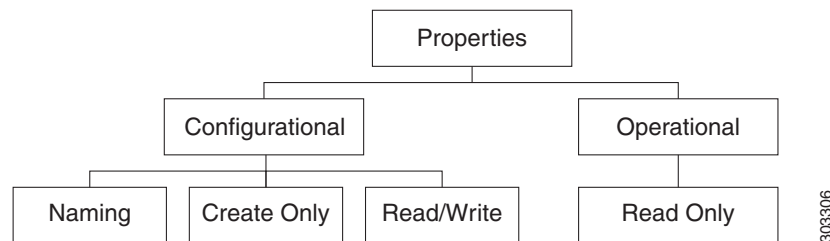
管理対象オブジェクトのプロパティは、コンフィギュレーションまたはオペレーションとして分類されます。

コンフィギュレーション プロパティは次のように分類されます。

- Naming プロパティ: RN の一部になります。MO の作成中に指定する必要があり、後で変更できません。
- Create-Only プロパティ: MO の作成中にのみ指定でき、後で変更できません。プロパティが指定されていない場合、デフォルト値が使用されます。
- Read/Write プロパティ: MO の作成時に指定でき、後で変更もできます。

オペレーション プロパティは MO/システムの現在のステータスを示し、読み取り専用です。

図 4 管理対象オブジェクトのプロパティ



次の表は、さまざまなプロパティ タイプの例を示します。

プロパティタイプ	例
Naming	LsServer (サービス プロファイル MO) 内の名前
Create-Only	LsServer 内のタイプ
Read / Write	LsServer 内の説明
Read-Only	LsServer 内の OperState

## メソッド

Cisco UCS XML API は、システムの管理および監視に使用されます。以下のサポートするメソッドがあります。

- Authentication
- AaaLogin
- AaaRefresh
- AaaLogout
- Configuration
- ConfigConfMo(s)
- LsClone
- LsInstantiate\*
- FaultAckFaults
- Query
- ConfigResolveDn(s)
- ConfigResolveClass(es)
- ConfigResolveChildren
- Event Monitor
- EventSubscribe

クラス クエリー メソッド (ConfigResolveClass、ConfigResolveChildren) ではフィルタが指定でき、一致した特定の MO のセットを返すことができます。

次のフィルタがサポートされています。

- プロパティ フィルタ:

サポートされる フィルタ	定義
allbits	指定したすべての値が複数値プロパティに存在する場合に一致
anybit	指定した値のいずれかが複数値プロパティに存在する場合に一致
bw	プロパティの値が指定した 2 つの値の間にある場合一致
eq	プロパティの値が指定した値と同一の場合一致
ge	プロパティの値が指定した値以上の場合一致
gt	プロパティの値が指定した値より大きい場合一致
le	プロパティの値が指定した値以下の場合一致
lt	プロパティの値が指定した値より小さい場合一致
ne	プロパティの値が指定した値と等しくない場合一致
wcard	プロパティの値が指定したパターンに一致した場合一致

- コンポジット フィルタ (サブ フィルタでの動作)

コンポジット フィルタ	定義
not	サブ フィルタの結果を逆にします
and	すべてのサブ フィルタが true を返した場合 true
or	サブ フィルタのいずれかが true を返した場合 true

## PowerTool マッピング

約 30 の PowerTool コマンドレットのすべては MO 仕様から生成されます。便利な名詞がタイプに使用されています (LsServer の代わりに ServiceProfile など)。コマンドレットの Get, Add, Set, Remove、またはサブセットが、さまざまな MO タイプに対して生成されます。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。ModifyPresent パラメータが指定されている場合、「created, modified」というステータスが代わりに指定されます。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。プロパティ パラメータが指定されている場合、「eq」フィルタの生成に使用されます。複数のプロパティ パラメータが指定されている場合は、複数の「eq」フィルタが「and」フィルタで結合されます。

**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」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。

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	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渡すことができるタイプのリストです。

Verb/タイプ	パイプラインの入力
Get	単一 - 非単一以外 - 親タイプ
Add	単一 - 非単一以外 - 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	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	メソッド
Add-Ucs <sup>1</sup>	ConfigConfMos
Set-Ucs1 <sup>1</sup>	
Get-Ucs <sup>1</sup>	ConfigResolveClass (フィルタ使用)
Get-UcsManagedObject -ClassId	ConfigResolveClass
Get-UcsManagedObject -ClassId -Dnlist	ConfigFindDnsByClassId
Get-UcsManagedObject -Dn	ConfigResolveDns
Connect-Ucs	AaaLogin
Disconnect-Ucs	AaaLogout

Background <sup>1</sup>	AaaRefresh
Copy-UcsServiceProfile	LsClone
Add-UcsServiceProfileFromTemplate	LsInstantiateTemplate, LsInstantiateNTemplate, LsInstantiateNNamedTemplate
Get-UcsChild	ConfigResolveChildren
Acknowledge-UcsFault	FaultAckFaults
Start-UcsKvmSession	AaaGetNComputeAuthTokenByDn
Watch-Ucs	EventSubscribe (First Watcher)
Clear-UcsStatistics	StatsClearInterval
Get-UcsTransactionImpact	ConfigEstimateImpact

1. これはコマンドレットではありません。これはバックグラウンド サービスです。

Get-UcsCmdletMeta は、MO タイプ、対応する名詞、サポートされる Verb、MO のプロパティ、タイプ (Naming、Read/Write など) およびプロパティが導入された UCS Manager のバージョンなどのプロパティの詳細を調べるのに有効なコマンドレットです。

## 3 システム要件

Cisco UCS Manager 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てください。

- Windows PowerShell 2.0 以降がシステムに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 2.0 SP1 (以降)

## Cisco UCS Manager

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

- リリース 3.0
- リリース 2.5
- リリース 2.2
- リリース 2.1
- リリース 2.0

## Windows PowerShell の要件

- Cisco UCS Manager PowerTool は Windows PowerShell 2.0 以降をサポートしています。

## 4 インストール

Cisco UCS Manager PowerTool をインストールする前に、次の作業を実行する必要があります。

- ロードされた PowerTool モジュールで動作中の PowerShell のインスタンスは閉じてください。
- Cisco UCS Manager PowerTool リリース 0.9.1.0 より古いバージョンの Cisco UCS Power Tool はアンインストールしてください。

## インストール

---

**ステップ 1** インストーラをダウンロードし、起動します。

**ステップ 2** (任意)デスクトップにショートカットを追加するには、[Create Shortcut] を選択します。

## 使用する前に

---

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

**ステップ 2** Cisco UCS Manager 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 CiscoUcsPS
Get-Command -Module CiscoUcsPS | group CommandType
Get-Command -Module CiscoUcsPS | measure
```

**ステップ 3** Cisco UCS ドメインに接続します。

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



**注** デフォルトでは、-NotDefault オプションが指定されていない場合は、ログイン後、Cisco UCS ハンドルがデフォルトの Cisco UCS ドメイン リストに追加されます。Cisco UCS ドメインで実行するすべてのコマンドレットには、ハンドルが指定可能な -Ucs パラメータを付けることができます。

---

**ステップ 4** プロキシを使用して Cisco UCS ドメインに接続します。

```
$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-Ucs <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 UCS ドメインの統合ステータス情報を取得します。

```
Get-UcsStatus -Ucs $handle
```

**b.** シャーシ オブジェクトを取得します。

```
Get-UcsChassis -Ucs $handle
```

**c.** サービス プロファイル インスタンスのリストを取得します。

```
Get-UcsServiceProfile -Ucs $handle -Type インスタンス
```

**d.** 接続を切断します。

```
Disconnect-Ucs -Ucs $handle
```

---

## デフォルト UCS

ハンドルまたは名前が指定されていない場合、-Ucs パラメータが指定されていない場合は、Cisco UCS ドメイン ハンドルが DefaultUcs ドメイン リストに追加され、パイプラインの最初のコマンドレットがデフォルト Ucs リストで動作します。

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

**デフォルト UCS リストを取得します。**

```
Get-UcsPSSession
```

**UCS の統合ステータス情報を取得します。**

```
Get-UcsStatus
```



すべてのシャーシオブジェクトのセットを取得します。

```
Get-UcsChassis
```

シャーシ 1 に関連するオブジェクトを取得します。

```
Get-UcsChassis -Id 1
```

シャーシ 1 に関連するブレードのセットを取得します。

```
Get-UcsChassis -Id 1 | Get-UcsBlade
```

FI で HTTP をイネーブルにします。

```
Get-UcsHttp | Set-UcsHttp -AdminState enabled
```

FI で HTTP をディセーブルにします。

```
Get-UcsHttp | Set-UcsHttp -AdminState disabled
```

接続を切断します。

```
Disconnect-Ucs
```

## 複数の UCS を使用した UCS デフォルト リスト

PowerTool コマンドレットは、複数のハンドルを指定することで、複数の Cisco UCS ドメインで使用できます。

**Cisco UCS ドメインに接続します。**

```
$handle1 = Connect-Ucs <ip1> -NotDefault
$handle2 = Connect-Ucs <ip2> -NotDefault
Get-UcsStatus -Ucs $handle1,$handle2
Disconnect-Ucs -Ucs $handle1,$handle2
```

デフォルトでは、複数の Cisco UCS ハンドルは DefaultUcs では許可されません。これは、Set-UcsPowerToolConfiguration コマンドレットを使用して上書きできます。

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

```
Connect-Ucs <ip1>
Connect-Ucs <ip2>
Get-UcsStatus
Disconnect-Ucs
```

同じログイン資格情報を使用して複数の Cisco UCS 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 = @("<ucs1>", "<ucs2>", "<ucs3>")
Connect-Ucs $servers -Credential $cred
```

## 資格情報の読み取り/書き込み

```
Connect-Ucs <ip1>
Connect-Ucs <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-Ucs
```

ログインはファイルに保存された資格情報から開始できます。

```
Connect-Ucs -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-Ucs -LiteralPath C:\work\lab.xml -Proxy $proxy
```

追加システムにログインし、ファイルに資格情報を追加します。

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

## IPv6 のサポート

- IPv6 アドレスを使用して Cisco UCS Manager に接続できます。
- scp, ftp, tftp, ntp, dns などの外部クライアント アプリケーションや、sshd, httpd, snmpd などの外部クライアント サービスに IPv6 アドレスでアクセスできます。

```
Connect-Ucs [2001::0202:*3F*:*E1*:8**9]
```

## SSL の処理

ユーザが Cisco UCS サーバに接続した際に有効な証明書が認識されなかった場合は、接続の確立は InvalidCertificateAction の設定に依存します。デフォルトでは、無視するよう InvalidCertificateAction は設定されています。デフォルトでは、証明書が無効かどうかにかかわらず接続を確立するように PowerTool は設定されています。

Set-UcsPowerToolConfiguration コマンドレットを使用してこれを上書きできます。

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

	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。

## Cisco UCS Central への登録/登録解除

Cisco UCS Central を Cisco UCS ドメインの管理に使用する場合は、そのドメインを登録する必要があります。登録する際、Cisco UCS Central や Cisco UCS Manager によって管理されるポリシーのタイプや、バックアップやファームウェアなどのその他の設定を選択する必要があります。

Cisco UCS Central を使用して Cisco UCS ドメインを登録する前に、次を実行してください。

**ステップ 1** Cisco UCS Manager と Cisco UCS Central を確実に同期させるために、双方で NTP サーバおよび正しいタイムゾーンを設定します。Cisco UCS ドメインと Cisco UCS Central の日時が同期していない場合、登録は失敗する可能性があります。

**ステップ 2** Cisco UCS Central のホスト名または IP アドレスを取得します。

**ステップ 3** Cisco UCS Central を導入したときに設定した共有秘密を取得します。

```
$password = "SharedSecret" | ConvertTo-SecureString -AsPlainText -Force  
Register-UcsCentral -Name 10.10.10.10 -SharedSecret $password
```

**ステップ 4** UCS Central からの登録解除

```
Get-UcsCentral | Unregister-UcsCentral
```

---

## 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。

```
gal | ? { $_.Name -like "*-Ucs*" } | select Name  
Name  
----  
Acknowledge-UcsBlade  
Acknowledge-UcsChassis  
Acknowledge-UcsFault  
Acknowledge-UcsFex  
Acknowledge-UcsRackUnit  
Acknowledge-UcsServerUnit  
Acknowledge-UcsSlot  
Add-UcsMo  
Associate-UcsServiceProfile  
Compare-UcsMo  
Decommission-UcsBlade  
Decommission-UcsChassis  
Decommission-UcsFex  
Decommission-UcsRackUnit  
Decommission-UcsServerUnit  
Disassociate-UcsServiceProfile  
Get-UcsCentral  
Get-UcsMo  
Recommission-UcsRackUnit  
Recommission-UcsBlade  
Recommission-UcsChassis  
Recommission-UcsFex  
Recommission-UcsServerUnit  
Remove-UcsBlade  
Remove-UcsCartridge  
Remove-UcsChassis  
Remove-UcsFex  
Remove-UcsMo  
Remove-UcsRackUnit  
Set-UcsMo  
Sync-UcsMo  
Unregister-UcsCentral
```

## 5 例

ここでは、コマンドレットを実行する例を示します。ここでは、次の例について説明します。

- [PowerTool コマンドレットの生成](#)
- [UCS サーバの取得](#)
- [Org](#)
- [障害](#)
- [Get コマンドレット -Hierarchy フラグ](#)
- [Get コマンドレット -LimitScope フラグ](#)
- [トランザクション サポート](#)
- [VLAN\(作成および削除\)](#)

- MAC プールおよびブロック
- サーバプール
- UUID 接尾辞のプールおよびブロック
- WWNN プールおよびブロック
- WWPN プールおよびブロック
- IQN 接尾辞のプールおよびブロック
- ポートのロール
- ポート チャンネル
- VLAN(割り当て)
- ブレードの電力および温度の統計情報
- コンフィギュレーションのバックアップ
- コンフィギュレーションのインポート
- 管理対象オブジェクトの同期
- UCS 管理対象オブジェクト移行の監視
- テクニカル サポート
- サービス プロファイル
- サービス プロファイルのコンポーネント
- サービス プロファイルのアソシエーション
- フィルタ
- iSCSI ブート
- vNIC テンプレート
- vHBA テンプレート
- ブート ポリシー
- アダプタ ポリシー
- BIOS ポリシー
- ホスト ファームウェア パッケージ
- IPMI アクセス プロファイル
- 管理ファームウェア パッケージ
- 電力制御ポリシー
- サーバプール ポリシー資格情報ダイナミック vNIC 接続ポリシー
- ネットワーク制御ポリシー
- 特権
- ユーザ ロール
- ロケール
- ユーザ アカウント
- リモート認証 - RADIUS
- リモート認証 - TACACS
- リモート認証 - LDAP
- RADIUS プロバイダー
- TACACS プロバイダー
- LDAP プロバイダー
- 認証ドメイン
- コミュニケーション サービス
- 通信サービス - Telnet

- 通信サービス - CIM XML
- 通信サービス - SNMP
- 通信サービス - HTTP
- 通信サービス - HTTPS
- 一般的な管理対象オブジェクトのクエリー
- 一般的な管理オブジェクトのコマンドレット
- 一般的なコマンドレット - XmlTag
- Get/Add/Set コマンドレット中の XtraProperty
- CCO 統合
- アップロード ファームウェア
- XML へのエクスポート
- XML からのインポート
- KVM
- UCSM GUI の起動
- UCS 6324 ファブリック インターコネクトのスケールビリティ ポートの設定
- UCS 統計情報
- トランザクションの影響
- コマンドレットのメタ情報
- Compare-UcsManagedObject - Dn 変換
- Compare-UcsManagedObject - GetPropertyDiff()
- Add コマンドレット -ModifyPresent フラグ
- 機能カタログの更新
- サーバの操作

## PowerTool コマンドレットの生成

**UCS GUI での特定のアクションに対するコマンドレットを生成します。**

```
ConvertTo-UcsCmdlet
```

**生成されたコマンドレットを伴う xml 要求を取得します。**

```
ConvertTo-UcsCmdlet -Verbose
```

**指定 GUI ログ アクション用コマンドレットを生成します。**

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

**指定 xml 要求のコマンドレットを生成します。**

```
ConvertTo-UcsCmdlet -Xml -Request '<lsClone dn="org-root/ls-sp1" inTargetOrg="org-root" inServerName="sp2"
inHierarchical="false"></lsClone>'
```

**ファイル内に指定された xml 要求のコマンドレットを生成します。**

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

**特定の MO のコマンドレットを生成します。**

バージョン 1.2(1)から、管理オブジェクトを ConvertTo-UcsCmdlet にパイプライン処理し、管理対象オブジェクトの作成に必要なコマンドレットを取得できるようになりました。

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

## UCS Manager GUI アクションによるコマンドレットの生成

UCS Manager GUI で実行されたアクションのコマンドレットを、ConvertTo-UcsCmdlet コマンドレットを使用して生成することができます。

Cisco UCS Manager GUI では、安全のためにわずかな XML スニペットが考慮されていて、これはログに記録されません。したがって、ConvertTo-UcsCmdlet では、変換を実行するためのログは見つかりません。

GUI でのあらゆるユーザ アクションの XML スニペットをログに記録するには、次のいずれかの方法で UCSM GUI を起動します。

- Start-UcsGuiSession -LogAllXml コマンドレットを使用する
- 次の手順を実行して手動で UCS Manager GUI を起動する

---

**ステップ 1** 起動リンクを .jnlp ファイル形式で保存します。例: `https://<ip_or_hostname>/ucsm/ucsm.jnlp`

**ステップ 2** ファイルを右クリックし、メモ帳で開きます。

**ステップ 3** その他のプロパティ定義の後に以下の行を追加します。

- Java のバージョンが Java 7 Update 45 よりも前の場合、`<property name="log.show.encrypted" value="true"/>` を追加します。
- Java 7 Update 45 以降のバージョンの場合、`<property name="jnlp.ucsm.log.show.encrypted" value="true"/>` を追加します

**ステップ 4** ファイルを保存して、閉じます。

**ステップ 5** ファイルを右クリックし、Java™ Web Start Launcher で開きます。

---

UCS Manager GUI が起動したら、Cisco UCS モジュールがロードされている PowerShell ウィンドウから ConvertTo-UcsCmdlet コマンドレットを実行します。

## UCS サーバの取得

リリース 1.4.1 からは、新しい Get-UcsServer コマンドレットを使用することで、フォーク ファクタにかかわらずすべてのサーバを取得できるようになりました。

## Org

デフォルト UCS リストに、Cisco UCS ドメイン全体で使用する Org のリストを取得します。

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

ルートレベルの Org にハンドルを取得します。

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

UCS に 5 個の Org を追加します。

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

## 障害

障害を取得し、重大度別にグループ化します。

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

重大な障害を取得します。

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

すべての未確認の障害を確認します。

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

## Get コマンドレット -Hierarchy フラグ

子を含めすべての管理対象オブジェクトを取得します。

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

## Get コマンドレット -LimitScope フラグ

org の子ルートまでたどらずに root レベルでサービス プロファイルを取得します。

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

Org Finance の子までたどらずに Org Finance からサービス プロファイルを取得します。

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

LanCloud から VLAN を取得します。

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

## トランザクション サポート

トランザクションを開始します。

```
Start-UcsTransaction
```

アクションを実行します。

...

トランザクションを終了します。

```
Complete-UcsTransaction
```

トランザクションを取り消します。

```
Undo-UcsTransaction
```

## VLAN (作成および削除)

Cisco UCS ドメインの VLAN は名前参照され、VLAN 定義は、MIT で 4 ノードに作成できます。

	説明
LanCloud	これはグローバル VLAN で、両方の FI に適用できます。
FabricLanCloud	これはファブリック特定の VLAN で、ファブリック A またはファブリック B に適用できます。

	説明
ApplianceCloud	これはグローバル VLAN で、両方の FI に適用できます。
FabricApplianceCloud	これはファブリック A またはファブリック B に適用できるファブリック特定の VLAN で、アプライアンス ポートの設定中に使用されます。

### グローバル LAN クラウドで VLAN を作成します。

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

### ファブリック A LAN クラウドで VLAN を作成します。

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

### ファブリック B LAN クラウドで VLAN を作成します。

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

### グローバル アプライアンス クラウドで VLAN を作成します。

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

### ファブリック A アプライアンス クラウドで VLAN を作成します。

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

### ファブリック B アプライアンス クラウドで VLAN を作成します。

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

### VLAN のリストを CSV ファイルからインポートし、LAN クラウドで VLAN を作成します(この例では CSV ファイルを作成します)。

#### 1つのデバイスで VLAN を作成します。

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

#### 追加された VLAN を削除します。

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

## MAC プールおよびブロック

### デフォルトの MAC プールに MAC ブロックを追加します。

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

### デフォルト リストのすべての Cisco UCS ドメインにわたる MAC プール割り当ての任意の競合をチェックします。

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



## サーバプール

サーバプールを作成します。

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

サーバプールにブレード 1/4 を追加します。

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

サーバプールにラック 1 を追加します。

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

サーバプールを削除します。

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

## UUID 接尾辞のプールおよびブロック

UUID 接尾辞プールを作成します。

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

接尾辞プールに UUID 接尾辞のブロックを追加します。

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

UUID 接尾辞プールを削除します。

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

## WWNN プールおよびブロック

UCS のすべての WWNN プールを取得します。

```
Get-UcsWwnPool -Purpose node-wwn-assignment
```

WWNN プールを作成します。

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

WWNN プールへ WWNN ブロックを追加します。

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

WWNN プールへ WWNN イニシエータを追加します。

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

WWNN イニシエータを削除します。

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

WWNN プールを削除します。

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

## WWPN プールおよびブロック

UCS のすべての WWPN プールを取得します。

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

WWPN プールを作成します。

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

WWPN プールへ WWN ブロックを追加します。

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

WWPN プールへ WWPN イニシエータを追加します。

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

WWPN イニシエータに対する記述子を設定します。

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

WWPN プールを削除します。

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

## IQN 接尾辞のプールおよびブロック

UCS の IQN プールを取得します。

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

IQN のプールを作成します。

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

IQN プールのブロックを作成します。

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

IQN プールのブロックを削除します。

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

IQN プールを削除します。

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

## ポートのロール

ファブリック A のスロット 1(固定ポート スロット)ポート 19 をサーバポートにします。

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

ファブリック A のスロット 1(固定ポート スロット)ポート 19 がサーバポートであることの設定を解除します。

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

ファブリック A のスロット 1(固定ポート スロット)ポート 15 をアプライアンスポートにします。

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

ファブリック A のスロット 1(固定ポート スロット)ポート 15 がアプライアンス ポートであることの設定を解除します。

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

ファブリック A のスロット 1(固定ポート スロット)ポート 16 をアップリンク ポートにします。

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

ファブリック A のスロット 1(固定ポート スロット)ポート 16 がアップリンク ポートであることの設定を解除します。

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

## ポート チャネル

ポート 19 および 20 のあるファブリック A にアプライアンス ポート チャネルを作成します。

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

アプライアンス VLAN にポート チャネルを追加します。

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

アプライアンス VLAN からポート チャネルを削除します。

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

アプライアンス ポート チャネルを削除します。

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

## VLAN(割り当て)

アプライアンス VLAN にアプライアンスのポート A/1/15 を追加します。

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

アプライアンス VLAN からアプライアンスのポート A/1/15 を削除します。

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

## ブレードの電力および温度の統計情報

すべてのブレードの電力統計情報を表示します。

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

すべてのブレードの温度の統計情報を表示します。

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

## コンフィギュレーションのバックアップ

### UCSの前に保存したバックアップを削除します。

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

PathPattern は自動入力され、コマンドレットを複数の Cisco UCS ドメインで使うことができます。UCS の完全な状態のシステム バックアップを作成し、ダウンロードします。これにより、システム全体のスナップショットを含むバイナリ ファイルを作成します。このバックアップにより生成されたファイルを使用して、ディザスタ リカバリ時にシステムを復元できます。このファイルにより、元のファブリック インターコネクト上で設定を復元または再構築できます。また、別のファブリック インターコネクト上で設定を再現することもできます。このファイルは、インポートには使用できません。

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

UCS の論理バックアップを作成し、ダウンロードします。これにより、サービス プロファイル、VLAN、VSAN、プール、ポリシーなどのすべての論理設定が含まれる XML ファイルが作成されます。このバックアップにより生成されたファイルを使用して、これらの設定を元のファブリック インターコネクトまたは別のファブリック インターコネクトにインポートできます。このファイルは、システムの復元には使用できません。

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

UCS のシステム バックアップを作成し、ダウンロードします。これにより、ユーザ名、ロール、ロケールなどのすべてのシステム設定を含む XML ファイルが作成されます。このバックアップにより生成されたファイルを使用して、これらの設定を元のファブリック インターコネクトまたは別のファブリック インターコネクトにインポートできます。このファイルは、システムの復元には使用できません。

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

UCS のすべてのコンフィギュレーションのバックアップを作成し、ダウンロードします。これによって、すべてのシステムおよび論理設定を含む XML ファイルが作成されます。このバックアップにより生成されたファイルを使用して、これらの設定を元のファブリック インターコネクトまたは別のファブリック インターコネクトにインポートできます。このファイルは、システムの復元には使用できません。このファイルには、ローカル認証されたユーザのパスワードは含まれません。

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

## コンフィギュレーションのインポート

インポート機能は、すべてのコンフィギュレーション ファイル、システム コンフィギュレーション ファイル、および論理コンフィギュレーション ファイルで使用できます。インポートは、システムがアップ状態で、稼働中に実行できます。

すべてのコンフィギュレーション xml (すべてのシステムおよび論理設定を含む XML ファイル) をインポートします。現在のコンフィギュレーション情報が、インポートされたコンフィギュレーション ファイルの情報で一度に 1 つのオブジェクトについて置き換えられます。

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

すべてのコンフィギュレーション xml をインポートします。インポートされたコンフィギュレーション ファイルの情報が既存の設定情報と比較されます。矛盾が存在する場合、インポートされたコンフィギュレーション ファイルの情報で Cisco UCS ドメインの情報が上書きされます。

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

## 管理対象オブジェクトの同期

### 一連の MO を SYSA から SYSB に同期します。

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

### 一連の MO を SYSA からデフォルト リストのすべてのシステムに同期します。

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

## UCS 管理対象オブジェクト移行の監視

Cisco UCS ドメインのすべてのイベントを 60 秒間監視します。

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

Cisco UCS ドメインのエラーになったすべての変更を 60 秒間監視します。

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

MO の特定のフィールドが成功値になるまで UCS を監視します。

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

## テクニカル サポート

全体の UCSM インスタンスのテクニカル サポート データが指定のファイルに作成およびダウンロードされます。

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

UCSM 管理サービス(ファブリック インターコネクトを除く)のテクニカル サポート データが指定のファイルに作成およびダウンロードされます。

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

シャーシ ID 1 および CIMC ID 1 のテクニカル サポート データが指定のファイルに作成およびダウンロードされます。

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

シャーシ ID 1 および IOM ID 1 のテクニカル サポート データが指定のファイルに作成およびダウンロードされます。

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

ラック サーバ ID 1 およびラック アダプタ ID 1 のテクニカル サポート データが指定のファイルに作成およびダウンロードされます。

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

FEX ID 1 のテクニカル サポート データが指定のファイルに作成およびダウンロードされます。

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

## サービス プロファイル

UCS のすべてのサービス プロファイル インスタンスを取得します。

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

UCS のすべてのサービス プロファイルのアップデート テンプレートを取得します。

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

UCS のサービス プロファイルの初期テンプレートを取得します。

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

サービス プロファイル テンプレートの sp\_template から新しいサービス プロファイルの sp\_name を追加します。

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

サービス プロファイルを追加します。

```
Add-UcsServiceProfile -Name sp_name -BootPolicyName boot_policy -BiosProfileName bios_policy -HostFwPolicyName 1.4-3i -MgmtFwPolicyName 1.4-3i -MaintPolicyName maint_policy -MgmtAccessPolicyName ipmi_policy -PowerPolicyName power_policy -ScrubPolicyName scrub_policy -SolPolicyName sol_policy -StatsPolicyName stats_policy -AgentPolicyName agent_policy -DynamicConPolicyName vNIC_policy -ExtIPState static -IdentPoolName UUID_pool -LocalDiskPolicyName disk_policy -Uuid "00000000-0000-0000-0000-000000000008" -UsrLbl "serviceprofile"
```

サービス プロファイルの電源状態を取得します。

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

サービス プロファイルをテンプレートにバインドします。

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

サービス プロファイルを削除します。

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

## サービス プロファイルのコンポーネント

サービス プロファイルを作成します。

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

QoS ポリシーの参照付きで vNIC を作成します。

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

vNIC に VLAN を追加し、これをネイティブ VLAN にします。

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

vNIC に VLAN を追加します。

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

vNIC を作成し、テンプレートからインスタンス化します。

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

vHBA を作成します。

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

vHBA に VSAN を追加します。

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

WWNN プールを追加します。

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

## サービス プロファイルのアソシエーション

ブレードにサービス プロファイルに関連付けます。

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

ラックにサービス プロファイルに関連付けます。

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

サーバプールにサービス プロファイルに関連付けます。

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

サーバプール資格ポリシーのあるサーバプールにサービスプロファイルに関連付けます。

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

サービスプロファイルの関連付けを解除します。

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

サービスプロファイルのコピーを作成します。

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

サービスプロファイルの名前を変更します。

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

## フィルタ

すべてのサービスプロファイルテンプレートを取得します。

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

名前に文字列「SJC」が含まれるすべてのサービスプロファイルを取得します。

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

名前が文字列「SJC」で始まるすべてのサービスプロファイルを取得します。

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

8~50のIDを持つすべてのVLANを取得します。

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

障害権限があるすべてのロールを取得します。

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

障害または操作権限があるすべてのロールを取得します。

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

障害および操作権限があるすべてのロールを取得します。

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

45より高い温度のブレード/ラックユニットのリストを取得します。

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

2012/4/18 9:00~2012/4/19 9:30の間に生成された障害の一覧を取得します。

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

名前が「SJC」と等しいサービスプロファイルを取得します。

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

名前が「SJC/sjc/SjC」などと等しいすべてのサービスプロファイルを取得します。

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

名前が文字列「SJC/sjc/SjC」などで始まるすべてのサービスプロファイルを取得します。

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

名前が「SJC/sjc/SjC」などでないすべてのサービスプロファイルを取得します。

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

## iSCSI ブート

UCS のトランザクションを開始します。

```
Start-UcsTransaction
```

サービス プロファイルを作成します。

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

スタティック IP アドレス (iSCSI ブートに関連していない) を追加します。

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

必須の vNIC を作成し、VLAN を追加します。

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

iSCSI の vNIC を作成し、上記で作成された vNIC にマッピングします。

iSCSI イニシエータ パラメータの VLAN と IP アドレスを追加します。

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

ターゲット パラメータを追加します。

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

特定のブート ポリシーを作成します。

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

インストールが必要な場合は、LsbootVirtualMedia を作成します。

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

ブート パスに iSCSI enic を追加します。

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

UCS のトランザクションを完了します。

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

## vNIC テンプレート

初期 vNIC テンプレートを作成します。

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

アップデート vNIC テンプレートを作成します。

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

初期 vNIC テンプレートに VLAN を追加します。

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



**VLAN を初期 vNIC テンプレートに追加し、これをネイティブ VLAN に変更します。**

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

**初期 vNIC テンプレートに対して MAC プール、ネットワーク制御ポリシーと QoS ポリシーを設定します。**

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

**初期 vNIC テンプレートを削除します。**

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

## vHBA テンプレート

**初期 vHBA テンプレートを作成します。**

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

**アップデート vHBA テンプレートを作成します。**

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

**アップデート vHBA テンプレートに VSAN を追加します。**

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

**アップデート vHBA テンプレートに対して WWN プール、QoS ポリシー、ピン グループ、および統計ポリシーを設定します。**

```
$vhba_update_temp | Set-UcsVhbaTemplate -IdentPoolName wwn_pool -QosPolicyName qos_policy -PinToGroupName san_pin_group -StatsPolicyName threshold_policy
```

**アップデート vHBA テンプレートを削除します。**

```
$vhba_update_temp | Remove-UcsVhbatemplate
```

## ブート ポリシー

**ブート ポリシーを作成し、ブート 順序の変更で Reboot をイネーブルにして、vNIC/vHBA/iSCSI 名にします。**

```
$boot_policy = Add-UcsBootPolicy -Name boot_policy -EnforceVnicName yes -RebootOnUpdate yes
```

**フロッピーを追加します。**

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

**CD-ROM を追加します。**

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

**ローカル ディスクを追加します。**

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

**SAN ブートを追加します。**

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

**LAN ブートを追加します。**

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

**ブート ポリシーを削除します。**

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

## アダプタ ポリシー

**Receive Checksum Offload を無効にするカスタム イーサネット アダプタ ポリシーを追加します。**

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

**FC アダプタ ポリシーを追加します。**

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

**ファイバチャネルアダプタ ポリシーの FCP エラー回復を有効にします。**

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

**iSCSI アダプタ ポリシーを追加します。**

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

**iSCSI アダプタ ポリシーに対して、TCP タイムスタンプ、HBA モード、および対象へのブートをイネーブルにします。**

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

## BIOS ポリシー

**BIOS ポリシーを作成し、BIOS 設定変更でリブートをイネーブルにします。**

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

**USB システムのアイドル時の電力最適化設定を high-performance に変更します。**

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

**仮想化テクノロジーをイネーブルにします。**

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

**BIOS ポリシーの Quiet Boot をイネーブルにします。**

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

**電力損失時の AC のレジュームを last-state にします。**

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

**ブート オプションの再試行をディセーブルにします。**

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

**コンソールリダイレクションをディセーブルにします。**

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

**BIOS ポリシーを削除します。**

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

## ホスト ファームウェア パッケージ

**ホスト ファームウェア パッケージを作成して、IgnoreCompCheck を No に設定します。**

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

**ホスト ファームウェア パック項目を追加します。**

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

ホストファームウェアパック項目のバージョンを設定します。

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

ホストファームウェアパッケージを削除します。

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

## IPMI アクセス プロファイル

IPMI アクセス プロファイルを作成します。

```
$ipmi_profile = Add-UcsIpmiAccessProfile -Name ipmi_profile
```

管理者の役割を持つ IPMI ユーザを追加します。

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

IPMI ユーザのロールを変更します。

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

IPMI アクセス プロファイルを削除します。

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

## 管理ファームウェア パッケージ

管理ファームウェアパッケージを作成して、IgnoreCompCheck を No に設定します。

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

管理ファームウェアパック項目を追加します。

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

管理ファームウェアパック項目のバージョンを設定します。

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

管理ファームウェアパッケージを削除します。

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

## 電力制御ポリシー

電力制御ポリシーを作成します。優先順位は1～10の段階にランク付けされ、1が優先順位最高、10が優先順位最低を表します。デフォルトのプライオリティは5です。

```
$power_policy = Add-UcsPowerPolicy -Name power_policy -Prio 6
```

電力の制限が「非制限」の電力制御ポリシーを作成します。

```
$power_nocap = Add-UcsPowerPolicy -Name power_nocap -Prio no-cap
```

電力制御ポリシーを削除します。

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

## サーバプールポリシー資格情報

サーバプールポリシー資格情報を作成します。

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

アダプタ資格情報を作成します。

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

1067Mhzのメモリのクロック速度で16メモリユニットを持つメモリ資格情報ポリシーを作成します。

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

Pentium\_4プロセッサアーキテクチャのCPU/コア資格情報ポリシーを作成します。

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

ローカルディスクを持たないサーバ(SANのみの構成)のディスクレスストレージ資格情報ポリシーを作成します。

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

ラック資格情報を作成します。

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

サーバプールポリシー資格情報を削除します。

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

## ダイナミックvNIC接続ポリシー

54のダイナミックvNICがありフェールオーバーモードの保護がイネーブルのダイナミックvNIC接続ポリシーdy\_vnic\_connを作成します。

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

ダイナミックvNIC接続ポリシーを削除します。

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

## ネットワーク制御ポリシー

CDPがイネーブルで、アップリンク接続がファブリックインターコネクで失われるときにvNICの動作状態を下げるように変更するように設定されたVIFを持つネットワーク制御ポリシーnetwork\_policyを作成します。

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

偽のMACアドレスを許可するネットワーク制御ポリシーのMacセキュリティを設定します。

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

ネットワーク制御ポリシーのMacのセキュリティを設定します。これによって、最初のパケットをファブリックインターコネクトに送信後は、他のすべてのパケットが同じMACアドレスを使用しなければならないか、ファブリックインターコネクトによって暗黙的に拒否されます。これは、関連付けられたvNICのポートセキュリティをイネーブルにします。

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

ネットワーク制御ポリシーを削除します。

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

## 特権

UCS のすべての権限をリストします。

```
Get-UcsPrivilege
```

## ユーザ ロール

admin 権限を持つユーザ ロール「test\_role」を追加します。

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

ファブリック インターコネクトのインフラストラクチャへの読み取り/書き込みアクセス、ネットワーク セキュリティ動作、およびシステムのその他の部分への読み取りアクセスを許可するためにユーザ ロールの権限を変更します。

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

Set-UcsRole を使用して複数の権限を設定します。

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

UCS のすべてのユーザ ロールを取得します。

```
Get-UcsRole
```

名前別にユーザ ロールを取得します。

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

ユーザ ロールを削除します。

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

## ロケール

ロケールを追加します。

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

すべてのロケールを取得します。

```
Get-UcsLocale
```

ロケールに Org を追加します。

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

ロケールを削除します。

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

## ユーザ アカウント

ローカル ユーザを追加します。

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

ローカル ユーザを編集します。

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

ユーザに追加します。

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

ローカルユーザを削除します。

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

## リモート認証 - RADIUS

RADIUS 認証用のグローバル設定を設定します。

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

サーバキーが「test1234」で最大リトライ回数が2回の RADIUS サーバのインスタンスを作成します。

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

デフォルトの認証方式として RADIUS を設定します。

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

## リモート認証 - TACACS

TACACS 認証用のグローバル設定を設定します。

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

TACACS プロバイダーを追加します。

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

デフォルトの認証方式として TACACS を設定します。

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

## リモート認証 - LDAP

LDAP 認証用のグローバル設定を設定します。

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

LDAP プロバイダーを追加します。

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

デフォルトの認証方式として LDAP を設定します。

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

## RADIUS プロバイダー

サーバキーが「test1234」で最大リトライ回数が2回の RADIUS サーバのインスタンスを作成します。

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

RADIUS プロバイダーグループを追加して、デフォルトのリモート認証方式として設定します。

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

## TACACS プロバイダー

TACACS プロバイダーを追加します。

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

TACACS プロバイダー グループを追加して、デフォルトのリモート認証方式として設定します。

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

## LDAP プロバイダー

LDAP プロバイダーを追加します。

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

LDAP プロバイダー グループを追加して、デフォルトのリモート認証方式として設定します。

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

## 認証ドメイン

認証ドメインは、異なる認証方式(ローカル、TACACS+、RADIUS、および LDAP/Active Directory)とプロバイダーのグループへの同時サポートを設定します。

TACACS プロバイダーを持つ TACAS プロバイダー グループを設定します。

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

認証ドメインを作成し、TACACS Provider のグループへの参照を追加します。

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

これで、ユーザ名「ucs-adtacacs\user」でコンソール、GUI または XML API からユーザがログインした場合に、上記で作成した TACACS 設定が認証に使用されるようになります。

## コミュニケーション サービス

一度にシステムにアクセスすることを許可された並列 Web セッション(GUI および xml 両方)の最大数を定義する UCS Web セッション制限を取得します。

```
Get-UcsWebSessionLimit
```

ユーザの Web セッション制限を 30 に、全体のセッション制限を 255 に設定します。

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

## 通信サービス - Telnet

UCS telnet の設定を取得します。

```
Get-UcsTelnet
```

Telnet 接続を許可します。

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

## 通信サービス - CIM XML

UCS CIM XML の設定を取得します。

```
Get-UcsCimXml
```

CIM XML サービスをイネーブルにします。

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

## 通信サービス - SNMP

UCS SNMP 設定を取得します。

```
Get-UcsSnmp
```

コミュニティストリングが「public」、システム担当者が「CiscoSystems」、ホストの場所が「Bangalore」の SNMP をイネーブルにします。

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

UCS の SNMP ユーザを取得します。

```
Get-UcsSnmpUser
```

UCS の SNMP ユーザを追加します。

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

UCS の SNMP ユーザを設定します。

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

UCS の SNMP ユーザを削除します。

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

UCS SNMP トラップを取得します。

```
Get-UcsSnmpTrap
```

UCS SNMP トラップを追加します。

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

UCS SNMP コンフィギュレーションを設定します。

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

UCS の SNMP コンフィギュレーションを削除します。

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



## 通信サービス - HTTP

UCS http の設定を取得します。

```
Get-UcsHttp
```

http をイネーブルにし、http から https へのリダイレクションをイネーブルにするための UCS http コンフィギュレーションを設定します。

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

## 通信サービス - HTTPS

UCS https コンフィギュレーションを取得します。

```
Get-UcsHttps
```

1024 ビットのキー サイズのキー リングを作成します。

```
Add-UcsKeyring -Name keyring1024 -Modulus mod1024
```

必要なサブジェクト名(マシンのホスト名)を引き渡す証明書要求を作成します。

```
Get-UcsKeyRing -Name keyring2048 | Add-UcsCertRequest -SubjName savbu-pti01
```

生成された証明書要求に対する証明書を取得し、クライアント マシンにインストールしておきます。「certmgr.msc」を実行して証明書を確認します。

トラスト ポイントを追加します。

```
Add-UcsTrustPoint -Name TPkeyring1024
```

TP の証明書チェーンを設定します。

```
Get-UcsTrustPoint -Name TPkeyring1024 | Set-UcsTrustPoint -CertChain "
-----BEGIN CERTIFICATE----- MIIBoDCCA4igAwIBAgIQMjE/6XYi/a9CU8PPgr20ZDANBgkqhkiG9w0BAQUFADBU
MRiWEAYKCZImiZPyLQGGRYCaW4xGTAXBgoJkiaJk/IsZAEZFglxYXNhbS1sYWlX
FDASBgoJkiaJk/IsZAEZFgR1Y3NtMQ0wCwYDVQQDEwRVQ1NNM4XDTEwMDcxNjEy
MzYzZGCMf8FQy4SpLgeDXAn8DbobDdKbcH7txYRUMPCrmtYeEjV1QhfMPulhAs5B
cDChAg0wN7InoGexsNQVhdAQPpY7S18h0iml/GiR9XWbhcfaanbxDXUBepOve07UU
6kDnVwxGh9uQrgAgrI5oPatbbiE4zUjU1D2WYjZQ3UH+UGOP+Ub3OcaL+OhteHQh
dQWt/EuAprJeUp4jVjZwiaTbC8URAedMy8DjzP3WsbxMS+CHtF/Tz/dHBT+Z3ptK
syomrXro2/Kv0HWl9o921ryXHnz133sSdmFJ//LVbvZLqD2PM2UzZuX/+4C5S+44
Hghlv1uiNQ3yRdcCAwEAAaOCaWwggFoMAsGA1UdDwQEAwIBhjAPBgNVHRMBAf8E
BTADAQH/MB0GA1UdDgQWBBERG311HsVlu/dVTpUmIc9MNs4r/+DCCARUGA1UdHwSC
AQwwggEIMIIBKCCAQCggf2GgbxsZGFwOi8vL0NOPVVDU00sQ049YmxYLNhbS1x
YS1hYWExLENOPUNEUCxDtj1QdWJsaWMLMjBLZXk1MjBTZXJ2aWNLcyxDTj1TZXJ2
aWNLcyxDTj1Db25maWd1cmF0aw9uLERDPXVjc20sREM9cWFzYW0tbGFiLERDPWlu
P2N1cnRpb2ZlYXRlUmV2b2NhdGlvbkxpc3Q/YmFzZT9vYmplY3RDbGFzc21jUkxE
aXN0cmliidXRpb25Qb2ludIY8aHR0cDovL2JscilzYW0tcWEtYWVhMS51Y3NtLnFh
c2FtLWxhYi5pbj9DZSJ0RW5yb2xsL1VDU00uY3JsMBAGCSsGAQQBjzcvAQQDAGEA
MA0GCSqGSIb3DQEBBQUAA4TBAQBeuIZYIeI07ZhXalPjCs/YeBdR+S7+i0GKDYJq
nLtyWAua8YMyJQ57vJFB0I5MbEmHPt2JaKmFGRSYTMfLH417Z7vQUspkaw5h1kwwk
zQ4/VQusHEasioazFhbfSDPVzA9IRd71TNHGP5ruVoaThQJouavcnYSp5FFeOCM
xQcFUtGTkl/1XHoRv8R0Wjv24YXLpPxC+7DwMtmKLS00MGP8su9+nf4OrLGB2M1
0cVhfAqwl1MoVtfg6uzkI6xcss3xI1y7tuFOBZ60CkBVd+1C7ZhYe212RN75Uo6Z
jL77g422uodkMO5TSqj6pbI/wJmIQMs45NDit0M90x7TpvZ -----END CERTIFICATE-----"
```

キー リングに対する TP および証明書を設定します。

```
Get-UcsKeyRing keyring1024 | Set-UcsKeyRing -Tp TPkeyring1024 -Cert "
```

```
-----BEGIN CERTIFICATE----- MIIFnzCCBIegAwIBAgIKRy4WzAAAAAADTANBgkqhkiG9w0BAQUFADBUMRIwEAYK
CZImiZPyLQBGGRYCaW4xGTAXBgoJkiaJk/IsZAEZFglxYXNhbS1sYWlxdASBgoJ
kiaJk/IsZAEZFgRlY3NtMQ0wCwYDVQQDEwRlY3NtMQ0wCwYDVQJ1NNMBA4XDTEwMTEzZmZyOVoX
DTEyMTIwMTEzNDYwVowFjEUMBIGA1UEAxMLc2F2YnUtdHBpMDEwGCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQC4eSjyX6J/IIzSWSFXu+NmEW0BE0IOEEX/
zpmJ/yxh/SJKsgybicPAr0SRzgdKRhEIoIsMSMxigTFpErMgF4tkT32HNuElb5M
N+e/lcx3M7ogQfDUOMBFPV9qMCTkpn7cPAnOEOYaCx4J79XQJ6RyX1+uI1qAiCh
tz1jPwnTvpzNGTaccp/opZYwtJ0f5iY6ERNQ8WKJke56oulzUhcq40y3oKX/i1Gfki
IG8GT26Yv6a+KPKdRDS0+q+GZSsqmIcghETPYThCt3CWD07AYxRyQtNnGDzN10Ed
YaCQhcbz0d8qfoggpnsWIMARzgyC2HWAN9suZ0zO3NGrFKkeg6ep7AgMBAAGjggKv
MIICQzAfBgNVHREBAf8EFTATggtzYXZidS10cGkwMYcEckF4JTAAdBgNVHQ4EFgQU
ns86LcentpgeJmT8140jcfYt2DQwHwYDVR0jBBgwFoAURt5dr7Fdbv3VU6VJiHPT
DbOK//gwggEVBgNVHR8EggEMMIIBCDCCAQSGggEAOIH9h0G8bGRhcDovLy9DTj1V
Q1NNLENOFPWJscilzYW0tcWEtYWfhMSxDTj1DRFAsQ049UHVibG1jJTlW5J2V5JTIw
U2VydmljZXMzQ049U2VydmljZXMzQ049Q29uZmlndXJhdGlvbixEQz11Y3NtLERD
PXFhc2FtLWxhYixEQz11pbj9jZXJ0aWZpY2F0ZVJldm9jYXRpb25MaXN0P2Jhc2U/
b2JqZWN0Q2xhc3M9Y1JMRGlzdHJpYnV0aW9uUG9pbnsSGPGh0dHA6Ly9ibHItc2Ft
LXFhLWFhYTEudWNzbS5xYXNhbS1sYWludW4vQ2VydEVucm9sbC9VQ1NNLmNybDCC
ASOGCCsGAQUFBwEBBIIbHzCCARswgawGCCsGAQUFBzAChogGfbGRhcDovLy9DTj1V
Q1NNLENOFUFJQSxDTj1QdWJsaW1mJmJBTZlZ2aW1lcycxDTj1TZlZ2aW1lcycxDTj1
Db25maWd1cmF0aW9uLERDPXVjc20sREM9cWFzYW0tbGFiLERDPW1uP2NB
Q2VydG1maW5hdGU/YmFzZT9vYmp1Y3RDbGFzc2ljZXJ0aWZpY2F0aW9uQXV0aG9y
aXR5MGoGCCsGAQUFBzACh15odHRwOi8vYmxyLXNhbS1xYS1hYWExLnVjc20ucWFz
YW0tbGFiLm1uL0NlcnRFbnJvbGwvYmxyLXNhbS1xYS1hYWExLnVjc20ucWFzYW0t
bGFiLm1uL1VDU00uY3J0MAOGCSqGSIb3DQEBAQUAA4IBAQBO1hNPBrDqfu9hrIIE
o6Y9GghHNZ4cxwPlhz0U9w4iskWNVHlw7Ijdf7U+WUvulGWcylN73i2r2sOeQqy3
Isx/2dKS4n3YX7x1hYpMubPCL1fHIPqQwh9dd1HyKFtxqMd6/jQJyhLNOX5yz4h
HpORf14xGGWysv1JjqqR2jREbV3kE/uOq0NNi+2efWS0YHq9SESKqu1cXgM15LyC
ZKQYolUseboYK90XgLC2yww+75UcgynLZRxbAPstNeqPTWh12kWoGrO4zkpo18Y
Vz2yB2BA6/ugCbtJuIw352HzZHU9FM4Y7R0r9k75CNjA9wScu56hX2rfIFUwnSMT
gWvg -----END CERTIFICATE----- "
```

## HTTPS 用のキー リングを設定します。

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

これで、https を介した UCSM へのアクセスでは「untrusted connection」メッセージは表示されません。

## 一般的な管理対象オブジェクトのクエリー

### 特定の DN の管理対象オブジェクトを取得します。

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

### 特定のクラスのすべての管理対象オブジェクトを取得します。

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

### 特定のクラスの管理対象オブジェクトの DN を取得します。

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

### すべてのサービス プロファイルの名前を Org ルートから取得します。

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

### Org ルート直下の子を取得します。

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

### 管理対象オブジェクトの親を取得します。

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

## 一般的な管理オブジェクトのコマンドレット

親オブジェクトを使用して VLAN を作成します。

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

親オブジェクトを使用して VLAN を作成し、すでに存在している場合は変更します。

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

DN を使用して VLAN を作成します。

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

管理対象オブジェクトを使用して VLAN を変更します。

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

DN を使用して VLAN を変更します。

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

管理対象オブジェクトを削除します。

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

## 一般的なコマンドレット - XmlTag

XmlTag パラメータによって、未知の管理対象オブジェクトを使用することができます。

マルチキャスト ポリシーを作成します。

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

マルチキャスト ポリシーを無効にするスヌーピング状態を設定します。

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

## Get/Add/Set コマンドレット中の XtraProperty

XtraProperty パラメータによって、未知の管理対象オブジェクトのプロパティも Get/Add/Set コマンドレットで使用できることが保障されます。

追加のプロパティ ExtIPPoolName を持つサービス プロファイルを作成します。

```
Add-UcsServiceProfile -Name sp_name -XtraProperty @{ExtIPPoolName = "ext-mgmt"};
```

ExtIPPoolName が ext-mgmt であるすべてのサービス プロファイルを取得します。

```
Get-UcsServiceProfile -XtraProperty @{ExtIPPoolName = "ext-mgmt"};
```

## CCO 統合

CCO イメージの処理に関連するコマンドレットには、次の 2 つがあります。

**CCO からイメージのリストを取得します。**

```
$images = Get-UcsCcoImageList
```

**2.0(1x) イメージを選択し、このイメージをダウンロードします。イメージがローカルに使用可能な場合は、Get-UcsCcoImage によって最初に確認されます。イメージが存在し md5sum との一致がある場合、ダウンロードは開始されません。そうでない場合は、イメージがダウンロードされます。**

```
$images | where { $_.ImageName -like "ucs-k9-bundle*2.0.1*" } | Get-UcsCcoImage -Path C:\work\Images
```

**以前のダウンロードが成功している場合、コマンドを再実行しても開始されるダウンロードはありません。**

```
$images | where { $_.ImageName -like "ucs-k9-bundle*2.0.1*" } | Get-UcsCcoImage -Path C:\work\Images
```

**プロキシは必要に応じて使用できます。**

```
$proxy = New-Object System.Net.WebProxy
$proxy.Address = "http://<url>:<port>"
$proxy.UseDefaultCredentials = $false
$proxy.Credentials = New-Object System.Net.NetworkCredential("<username>", "<password>")
$images = Get-UcsCcoImageList -Proxy $proxy
```

## アップロード ファームウェア

**デフォルトの UCS システムにイメージをアップロードします。**

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

## XML へのエクスポート

**管理対象オブジェクトのコンフィギュレーションをエクスポートします。**

このコマンドレットは管理対象オブジェクトと階層全体のコンフィギュレーションをエクスポートします。

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

**管理対象オブジェクトの xml をファイルにエクスポートします。**

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

## XML からのインポート

**XML ファイルからコンフィギュレーションをインポートします。**

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

**管理対象オブジェクトの xml をインポートし、オブジェクトに変換します。**

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

## KVM

サービスプロファイルの KVM セッションを開始し、KVM ウィンドウのカスタマイズされたタイトルを追加します。

```
Get-UcsServiceProfile -Name sp_name -LimitScope | Start-UcsKvmSession -FrameTitle "Custom Frame Title & | 1 2 3"
```

ブレード 1 の KVM セッションを開始します。

```
Start-UcsKvmSession -Blade (Get-UcsBlade -SlotId 1 -ChassisId 1)
```

ラックユニット 1 の KVM セッションを開始します。

```
Start-UcsKvmSession -RackUnit (Get-UcsRackUnit -Id 1)
```

## UCSM GUI の起動

UCSM に接続し、UCSM GUI を起動します。

```
Start-UcsGuiSession
```

セキュア ロギングをイネーブルにします。

一部の XML トランザクションは安全なものとして処理されるため、UCSM GUI によってログに記録されません。LogAllXml フラグによってセキュア ロギングがイネーブルにされます。

```
Start-UcsGuiSession -LogAllXml
```

Get-UcsStatus および Start-UcsGuiSession コマンドレットを使用して UCSM GUI を起動します。

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

UCSM に接続せずに UCSM GUI を起動します。

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

クレデンシャルを変数に格納し、コマンドレットに渡します。

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

## UCS 統計情報

シャーシ ID 1 およびスロット ID 1 の現在の UCS 統計情報を取得します。

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

シャーシ ID 1 およびスロット ID 1 の UCS 統計情報の履歴を取得します。

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

UCS 統計情報をクリアします。

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

## UCS 6324 ファブリック インターコネクトのスケラビリティ ポートの設定

UCS 6324 ファブリック インターコネクト B のブレイクアウト ポート 1/5/1 をサーバポートとして設定します。

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

**UCS 6324 ファブリック インターコネクト B のブレイクアウト ポート 1/5/1 を FCoE ストレージ ポートとして設定します。**

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

**UCS 6324 ファブリック インターコネクト B のブレイクアウト ポート 1/5/1 をアプライアンス ポートとして設定します。**

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

## トランザクションの影響

**Get-UcsTransactionImpact** コマンドレットによって、保留状態のトランザクションの影響を推定します。このコマンドレットでは、**ConfigEstimateImpact** メソッドを使用し、**UcsImpact** オブジェクトを返します。UCS Manager GUI によって配信されたメッセージのようなメッセージが **UcsImpact** オブジェクトの一部として提供されます。

トランザクションを開始します。

```
Start-UcsTransaction
```

サービス プロファイルを作成します。

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

vNIC を作成します。

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

vNIC に VLAN を追加し、これをネイティブ VLAN にします。

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

トランザクションの影響を取得します。

```
Get-UcsTransactionImpact
```

ここで、作成されたサービス プロファイルなどのコンフィギュレーションの失敗を示す **UcsImpact** オブジェクトが返されたことを確認できます。

## コマンドレットのメタ情報

コマンドレットにマッピングされているすべての管理オブジェクトに関するメタ情報を取得します。

```
Get-UcsCmdletMeta
```

コマンドレットにマッピングされている **LsServer** に関するメタ情報を取得します。

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

**ServiceProfile(LsServer)** クラスの階層を表示します。

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

**UcsServiceProfile** 名詞のメタ情報を取得します。

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

LsServer の管理対象オブジェクト情報を参照します。

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

LsServer の管理対象オブジェクトのプロパティ情報を参照します。

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

## Compare-UcsManagedObject - Dn 変換

Org A の下にサービス プロファイルを作成します。Org A と B はすでに設定されているものとします。

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

変換する必要があるエンティティの DN で変換マップを作成します。

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

必要な変更を参照する Compare-UcsMo で変換マップを結合します。

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

必要な変更を参照する Compare で変換 Org を結合します。

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

ファイルの名前を変更する際に、Org A から B へサービス プロファイルを同期します。

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

## Compare-UcsManagedObject - GetPropertyDiff()

プロパティ間の相違を確認するために、Compare-UcsManagedObject の出力で GetPropertyDiff() 関数を使用します。

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

違いを持つすべてのプロパティを表示します。\$diff がオブジェクトの配列の場合、GetPropertyDiff は \$diff [<index>] で動作します。

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

特定のプロパティ \$diff 用。

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

比較に MO のすべてのオペレーション プロパティを含みます。

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

## Add コマンドレット -ModifyPresent フラグ

ModifyPresent オプションによって、MO がすでにある場合に Add コマンドレットがエラーを返す代わりに MO を変更できるようにできます。

名前と ID のペアを含む CSV ファイルを作成します。

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

ファイルから名前と VLAN ペアをインポートし、それらの VLAN を作成します。

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

ID を編集したり、新しい VLAN を追加するには CSV ファイルを編集します。既存の VLAN を再び作成した場合は、上記の同じ **Add-UcsVlan** スニペットを再実行してもエラーになります(変更の有無にかかわらず)。すでに VLAN が存在する場合は、代わりに、**ModifyPresent** オプションを使って **Add-UcsVlan** を起動し、この VLAN を編集することでこのエラーを解決します。

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

## 機能カタログの更新

機能カタログは調整可能なパラメータ、文字列、およびルールのセットです。Cisco UCS では、カタログを使用して、サーバの新しく承認された DIMM やディスク ドライブなどのコンポーネントの表示と設定可能性を更新します。

ローカルのファイル ソースから機能カタログを更新する場合は、次のコマンドレットを使用できます。

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

## 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簡易コマンドレットを次に示します。

操作の説明	PowerTool リリース 1.4.1 以前の コマンドレット	PowerTool リリース 1.5.1 以降の コマンドレット
UCS サーバの確認応答	Get-UcsChassis -Id 1   Get-UcsBlade -SlotId 1   Set-UcsBlade -AdminPower "policy" -Lc "rediscover" -PolicyOwner "local"	Get-UcsServer   where { \$_.Dn -eq "sys/chassis-1/blade-1"}   Confirm-UcsServer
UCS サーバの稼働中止	Get-UcsChassis -Id 1   Get-UcsBlade -SlotId 1   Set-UcsBlade -AdminPower "policy" -Lc "decommission" -PolicyOwner "local"	Get-UcsServer   where { \$_.Dn -eq "sys/chassis-1/blade-1"}   Disable-UcsServer
UCS サーバのハード リセット	Get-UcsOrg -Level root   Get-UcsServiceProfile -Name "testSP" -LimitScope   Get-UcsServerPower   Set-UcsServerPower -State "hard-reset-immediate"	Get-UcsServiceProfile -name testSP   Reset-UcsServer
UCS サーバのブート	Get-UcsOrg -Level root   Get-UcsServiceProfile -Name "testSP" -LimitScope   Get-UcsServerPower   Set-UcsServerPower -State "admin-up"	Get-UcsServiceProfile -name testSP   Start-UcsServer
UCS サーバのシャットダウン	Get-UcsOrg -Level root   Get-UcsServiceProfile -Name "testSP" -LimitScope   Get-UcsServerPower   Set-UcsServerPower -State "soft-shut-down"	Get-UcsServiceProfile -name testSP   Stop-UcsServer



操作の説明	PowerTool リリース 1.4.1 以前のコマンドレット	PowerTool リリース 1.5.1 以降のコマンドレット
UCS サーバの電源再投入	Get-UcsOrg -Level root   Get-UcsServiceProfile -Name "testSP" -LimitScope   Get-UcsServerPower   Set-UcsServerPower -State "cycle-immediate"	Get-UcsServiceProfile -name testSP   Restart-UcsServer
UCS サーバの CMOS のリセット	Get-UcsOrg -Level root   Get-UcsServiceProfile -Name "testSP" -LimitScope   Get-UcsServerPower   Set-UcsServerPower -State "cmos-reset-immediate"	Get-UcsServiceProfile -name testSP   Reset-UcsServerCmos
UCS サーバの BMC のリセット	Get-UcsOrg -Level root   Get-UcsServiceProfile -Name "testSP" -LimitScope   Get-UcsServerPower   Set-UcsServerPower -State "bmc-reset-immediate"	Get-UcsServiceProfile -name testSP   Reset-UcsServerBmc
UCS サーバのロケータ LED の点灯	Get-UcsChassis -Id 1   Get-UcsBlade -SlotId 1   Get-UcsLocatorLed   Set-UcsLocatorLed -AdminState "on" -BoardType "single" -Id 1	Get-UcsServer   where { \$_.Dn -eq "sys/chassis-1/blade-1"}   Enable-UcsLocatorLed
UCS サーバのロケータ LED の消灯	Get-UcsChassis -Id 1   Get-UcsBlade -SlotId 1   Get-UcsLocatorLed   Set-UcsLocatorLed -AdminState "off" -BoardType "single" -Id 1	Get-UcsServer   where { \$_.Dn -eq "sys/chassis-1/blade-1"}   Disable-UcsLocatorLed

## 6 サンプル

インストールに含まれている次のようなサンプルがあります。

### すべてのグローバルポリシーをフェッチします

```
$id=(Get-UcsPowerToolConfiguration).InstallDir
& "$id\Samples\Get-UcsGlobalPolicy.ps1" bgl-abcd18
```

## 7 関連する Cisco UCS 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 Documentation Roadmap](#)
- [Release Bundle Contents for Cisco UCS Software, Release 2.1](#)

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](mailto: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 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がデスクトップに直接配信されるように設定することもできます。RSS フィードは無料のサービスです。シスコは現在、RSS バージョン 2.0 をサポートしています。

---

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](http://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。

© 2015 Cisco Systems, Inc. All rights reserved