



APIC CLI の使用

- [オブジェクト モデル CLI へのアクセス, 1 ページ](#)
- [管理対象オブジェクトの表示, 2 ページ](#)
- [管理情報ツリーでの移動, 2 ページ](#)
- [設定の入力, 3 ページ](#)
- [設定ウィザードの使用, 4 ページ](#)
- [設定テンプレートの作成, 6 ページ](#)
- [コマンドのカスタマイズ, 8 ページ](#)

オブジェクト モデル CLI へのアクセス



(注) Cisco APIC リリース 1.0 から リリース 1.2 までは、オブジェクト モデル CLI が、SSH を使用して APIC にログインする時に表示されるデフォルトの CLI でした。Cisco APIC リリース 1.2 以降、デフォルトの CLI は、NX-OS スタイルの CLI です。

手順

-
- ステップ 1** セキュア シェル (SSH) クライアントから、`username@ip-address` で APIC への SSH 接続を開きます。初期設定時に設定した管理者のログイン名とアウトオブバンド管理 IP アドレスを使用します。たとえば、`admin@192.168.10.1` などがこれに該当します。
- ステップ 2** プロンプトが表示されたら、管理者パスワードを入力します。
- ステップ 3** コマンドラインプロンプトで、`bash` を入力します。
-

例

この例では、最初の CLI プロンプトからオブジェクト モデル CLI に到達する方法を示しています。

```
apicl# bash
admin@apicl:~>
```

管理対象オブジェクトの表示

管理情報ツリー (MIT) 内にある特定のコンテキストの管理対象オブジェクト (MO) について要約を表示するには、次のように、**cat summary** コマンドを使用します。



(注) **less** と **more** コマンドを使用すると、1つの画面に同時に MO ファイルを表示できます。

```
admin@apicl:common> cat summary
name           : common
description    :
tags           : uni/tn-common
ownerkey       :
ownertag       :
alias          :
monitoring-policy :
epg-address-pool :

security-domains:
name  description
-----
common
```

管理情報ツリーでの移動

管理情報ツリー (MIT) には、次のようなさまざまなスコープが含まれています。

- aaa
- auditlog
- コントローラ
- eventlog
- fabric-policies
- 障害
- faults-history
- ファームウェア
- health
- health-history
- import-export

- l4-l7-inventory
- l4-l7-packages
- local-user
- pod
- schedulers
- security-domains
- switch
- テナント
- trafficmap
- version
- vm-inventory
- vm-policies

これらのスコープ間をすばやく移動するには、次のコマンドを使用します。

- **scope** : コンテキストのディレクトリにジャンプします。
- **show** : コンテキストの要約を表示します。
- **where** : コンテキストの管理情報ツリー (MIT) ディレクトリパスを表示します。

これらのコマンドの詳細については、次を参照してください。 [コマンドリファレンス](#)

MO ブラウザ ユーティリティ

APIC CLI には管理対象オブジェクト (MO) ブラウザ ユーティリティが含まれており、vi に類似したインターフェイスで MO を表示して編集することができます。MO ブラウザの詳細については、「[mobrowser](#)」の項を参照してください。

設定の入力

moconfig、**moset** および **modelete** コマンドを使用して、設定を作成できます。

moconfig コマンドは指定された名前で作成した新しいコンテキストを作成し、**moset** は既存の MO にプロパティを設定します。**modelete** コマンドは、指定された名前とスコープ (通常はサブスコープ) を削除します。

デフォルトの設定を上書きするには、**mocreate** コマンドで追加のプロパティを指定します。コンテキストのデフォルト設定を上書きする場合も、**mocreate** コマンドを使って追加のプロパティを指定できます。詳細については、「[mocreate](#)」の項を参照してください。



(注) また、APIC の GUI、REST API、Python API を使用して設定を入力することもできます。これらのツールの詳細については、『*APIC Getting Started Guide*』および『*APIC Python API and SDK*』を参照してください。

相違を表示するコマンド

moconfig diff コマンドは、設定バッファに存在する未保存の変更を要約します。また、**moconfig commit** コマンドを使用すると、MO に新しいプロパティを適用することができます。

設定ウィザードの使用

ウィザードは設定の作成プロセスを簡易化します。特定のコンテキスト（テナントなど）でウィザードを実行すると、特定のコンテキスト内（テナント、プライベートネットワークなど）で完全な設定を作成できます。

ウィザードの起動

ウィザードを起動するには **.wiz** ファイルを実行します。たとえば、テナントコンテキストの場合は、**./tenant.wiz** Bash コマンドを使ってウィザードを実行できます。

ウィザードのオプション

ウィザードはコマンド補完機能を備えています。**?** を入力して、使用可能なオプションをリストします。

```
description      : MyCompany BD
network          : ?
  default        network
  inb            network
  overlay-1     network
network         : inb
```

例

次の例は、テナントウィザードの全出力を示しています。

```
admin@apic1:tenants> ./tenant.wiz

tenant
-----
name           : MyCompany
alias          : MyCompany_tenant
description    : This is MyCompany
monitoring-policy : default

private-network
-----
name           : MyCompany_net
description    : MyCompany Network
```

```

bgp-timers      : default
ospf-timers     : default
monitoring-policy : default

bridge-domain
-----
name           : MyCompany_domain
description    : MyCompany BD
network       : ?
  default      : network
  inb          : network
  overlay-1    : network
network       : inb

Do you want to create another private-network (y/n): n

Do you want to view the corresponding commands? (y/n): y
-----

mcreate MyCompany
cd MyCompany
moset alias MyCompany_tenant
moset description This is MyCompany
moset monitoring-policy default
cd /aci/tenants/MyCompany/networking
cd /aci/tenants/MyCompany/networking/private-networks
mcreate MyCompany_net
cd MyCompany_net
moset description MyCompany Network
moset bgp-timers default
moset ospf-timers default
moset monitoring-policy default
cd /aci/tenants/MyCompany/networking/bridge-domains
mcreate MyCompany_domain
cd MyCompany_domain
moset description MyCompany BD
moset network inb
cd /aci/tenants/MyCompany/networking/private-networks/MyCompany_net
cd /aci/tenants/MyCompany/networking
cd /aci/tenants/MyCompany

Do you want to commit changes? (y/n): y

Committing all the mos...
Committed mo tenants/MyCompany
Committed mo tenants/MyCompany/networking/private-networks/MyCompany_net
Committed mo tenants/MyCompany/networking/bridge-domains/MyCompany_domain
done
admin@apic1:tenants>

```

プロパティのスキップ

Ctrl+N コマンドを使用すると、ウィザード内でオプションをスキップできます。



(注)

ウィザードは欠落しているプロパティを動的に追跡します。プロパティをスキップした場合は、後で適切なウィザードを実行して設定を完了することができます。たとえば、テナントウィザードを実行している場合は、プライベートネットワーク コンテキスト内のプロパティをスキップすることができます。

```

admin@apic1:tenants> ./tenant.wiz
<output truncated>
private-network
-----
name           : Company_net

```

```
description      : s...skipping
bgp-timers       : s...skipping
ospf-timers      : s...skipping
monitoring-policy : s...skipping
```

```
bridge-domain
-----
```

```
name              : default
```

```
<output truncated>
```

後でプライベートネットワーク ウィザードを実行して、設定を完了できます。

```
admin@apic1:networking> ls
bridge-domains          external-routed-networks  fv-tenant-common  fv-tenant-mgmt
private-network.wiz    protocol-policies
external-bridged-networks  fv-tenant-MyCompany      fv-tenant-infra    fv-tenant-test
private-networks
admin@apic1:networking> ./private-network.wiz
```

設定テンプレートの作成

設定テンプレートを使用すると、オーケストレーションツール、シェル スクリプト、その他のツールを使用して適用できる、再利用可能なネットワーク設定を作成できます。ここでは、APIC CLI を使用して設定テンプレートを作成する方法を説明します。

moconfig コマンドによるテンプレートの作成

moconfig コマンドは、設定テンプレートの作成プロセスを簡易化します。GUI、CLI、または API を使って設定を作成する場合は、**moconfig running** コマンドを使用して、特定のコンテキスト内の設定結果を表示できます。

たとえば、GUI を使用して、次のプロパティを含むテナント設定を作成することができます。

- 名前
- Alias
- 説明
- タグ
- Monitoring Policy
- Security Domains

GUI で設定を入力した後、その新しい APIC コンテキストで **moconfig** コマンドを使用すると、設定を構成しているコマンドが表示されます。たとえば、新しいテナント **MyCompany** を作成した場合は、次のような設定コマンドが表示されます。

```
admin@apic1:tenants> ls
common  infra  mgmt  MyCompany  tenant.wiz
admin@apic1:tenants> cd MyCompany/
admin@apic1:MyCompany> moconfig running
cd /aci/viewfs/tenants
mcreate MyCompany
cd MyCompany
moset description 'My Company Network'
```

```

moset alias Home
moset monitoring-policy default
moconfig commit
cd networking
cd private-networks
mcreate local_net
cd local_net
moset description 'Local network'
moset bgp-timers default
moset ospf-timers default
moset monitoring-policy default
moconfig commit
cd ..
cd ..
cd bridge-domains
mcreate BD1
cd BD1
moset description 'Bridge domain 1'
moset custom-mac-address 00:22:BD:F8:19:FF
moset arp-flooding no
moset unicast-routing yes
moset network overlay-1
moconfig commit
cd ..
cd ..
cd ..
cd ..
admin@apic1:MyCompany>

```

moconfig running コマンドの詳しい使用方法については、「[moconfig コマンド](#)」の項を参照してください。 .

設定ウィザードによるテンプレートの作成

設定ウィザードを実行している場合は、対応するコマンドオプションを使用して、ウィザードで作成された設定を要約できます。この設定は、変更して他のノードまたはデバイスに複製することが可能です。

次の例は、設定ウィザードのコマンド出力の表示方法を示しています。



(注) コマンド出力は短縮されています。

```

admin@apic1:tenants> ./tenant.wiz
<Output truncated>

Do you want to create another private-network (y/n): n

Do you want to view the corresponding commands? (y/n): y
-----
mcreate MyCompany
cd MyCompany
moset alias Home
moset description My Company Network
moset monitoring-policy default
cd /aci/tenants/MyCompany/networking
cd /aci/tenants/MyCompany/networking/private-networks
mcreate local_net
cd local_net
moset description Local network
moset bgp-timers default
moset ospf-timers default
moset monitoring-policy default
cd /aci/tenants/MyCompany/networking/bridge-domains
mcreate BD1

```

```

cd BD1
moset description Bridge domain 1
moset network overlay-1
cd /aci/tenants/MyCompany/networking/private-networks/local_net
cd /aci/tenants/MyCompany/networking
cd /aci/tenants/MyCompany

```

<Output truncated>

ウィザードの詳細な使用方法については、「[設定ウィザードの使用](#)」を参照してください。

コマンドのカスタマイズ

APIC CLI では、`/etc/scopedefs` ディレクトリ内の YAML (`.yml`) ファイルを使用して、Bash インターフェイスの Linux コマンドを拡張することができます。YAML コンフィギュレーション ファイルでは、実行する Linux コマンドや各スコープで使用可能なオプションを指定します。

YAML ファイルを使用して、新しいコマンドを作成したり、既存の Linux コマンドを拡張することができます。また、固有の `.yml` ファイルを MIT 内のユーザのスコープに配置することで、ユーザ用のカスタム インターフェイスを定義できます。

YAML を使用して次のコマンドをカスタマイズすることが可能です。

- **show** : Cisco IOS や NX-OS に類似した形式で APIC の設定を表示します。詳細については、「[show](#)」の項を参照してください。
- **create** : 特定のスコープ内でウィザードを実行します。ウィザードは MIT 内に関連するオブジェクトを作成します。詳細については、「[create](#)」の項を参照してください。
- **where** : コンテキストのディレクトリ (`tenant` や `14-17-services` など) を表示します。詳細については、「[where](#)」の項を参照してください。
- **scope** : コンテキストのディレクトリ (`tenant` や `14-17-services` など) にジャンプします。詳細については、「[scope](#)」の項を参照してください。
- **attach** : 指定されたファブリック ノードへの SSH セッションを開きます。詳細については、「[attach](#)」の項を参照してください。

YAML コマンド定義の例

controller コマンド

次の例は、**controller** コマンドの出力を示しています。

```

admin@apic1:aci> controller

operational-cluster-size                : 3
differences-between-local-time-and-unified-cluster-time : 0
administrative-cluster-size             : 3

controllers:
id  name  ip          cluster-admin-state  cluster-operational-  health-state  up-time
      system-current-time                                state

```



```

-----
1  apic1 10.0.0.1 in-service      available      fully-fit      62:02:38:00.000
   2014-05-
      01T21:40:46.120+00:00
2  apic2 10.0.0.2 in-service      available      fully-fit      62:02:38:00.000
   2014-05-
      01T21:40:46.211+00:00
3  apic3 10.0.0.3 in-service      available      fully-fit      62:02:38:00.000
   2014-05-
      01T21:40:46.263+00:00

```

次の例は、**controller** コマンドの YAML 定義を示しています。

```

- controller:
  help: 'Controller Node'
  type: alias
  dirFormat: '/aci/system/controllers/'
  fileType: 'summary'
  sub:
    - name: id
      label: id
      type: arg
      modelclass: fabric.Node
      modelprop: id
      classfilter: 'fabric.Node.role == "1"'
      dirFormat: '/aci/system/controllers/?(id)s'
      fileType: 'summary'
      help: 'controller'

```

tenant コマンド

次の例は、**tenant** コマンドの出力を示しています。

```

admin@apic1:~> show tenant infra bridge-domains default
# Executing command: cat /aci/tenants/infra/networking/bridge-domains/default/mo

# bridge-domain

# Naming properties (DO NOT EDIT):
# name : default

# Configurable Properties:
description :
custom-mac-address : 00:22:BD:F8:19:FF
l2-unknown-unicast : hardware-proxy
arp-flooding : no
unicast-routing : yes
ownerkey :
ownertag :
network : overlay-1
igmp-snoop-policy :
end-point-retention-policy :
l3-out :
external-route :
route-profile :
monitoring-policy :

```

次の例は、**tenant** コマンドの YAML 定義の抜粋を示しています。

```

- tenant:
  help: 'Tenant'
  type: alias
  dirFormat: '/aci/tenants/'
  fileType: 'summary'
  name: tenant
  sub:
    - name: name
      label: name

```

```

type: arg
modelclass: fv.Tenant
modelprop: name
dirFormat: '/aci/tenants/%(name)s'
fileType: 'summary'
help: Tenant name
sub:
  - name: bridge-domains
    label: bridge-domains
    type: keyword
    dirFormat: '/aci/tenants/%(name)s/networking/bridge-domains/'
    fileType: 'summary'
    help: "All Bridge-domains"
    sub:
      - name: bd
        label: bridge-domain-name
        type: arg
        modelclass: fv.BD
        modelprop: name
        dirFormat: '/aci/tenants/%(name)s/networking/bridge-domains/%(b\d)s'
        fileType: 'mo'
        help: Bridge domain name
  - name: application-profiles
    label: application-profiles
    type: keyword
    dirFormat: '/aci/tenants/%(name)s/application-profiles/'
    fileType: 'summary'
    help: "All application profiles"
    sub:
      - name: ap
        label: application-profile-name
        type: arg
        modelclass: fv.Ap
        modelprop: name
        dirFormat: '/aci/tenants/%(name)s/application-profiles/%(ap)s'
        fileType: 'mo'
        help: Application profile name
  - name: private-networks
    label: private-networks
    type: keyword
    dirFormat: '/aci/tenants/%(name)s/networking/private-networks/'
    fileType: 'summary'
    help: "All private networks"
    sub:
      - name: pn
        label: private-network-name
        type: arg
        modelclass: fv.Ctx
        modelprop: name
        dirFormat: '/aci/tenants/%(name)s/networking/private-networks/%(pn)s'
        fileType: 'mo'
        help: Private network name
        type: arg
        modelclass: fv.Ctx
        modelprop: name
        dirFormat: '/aci/tenants/%(name)s/networking/private-networks/%(pn)s'
        fileType: 'mo'
        help: Private network name
        (...)

```

YAML ファイルの形式

ファイル形式

次のキーワードを使用して、カスタム コマンドを使用する .yaml ファイルを定義できます。

- **help** : コマンド、引数、キーワードの機能を定義するヘルプ文字列。例 : **help: 'Displays faults for the current path.'**
- **type** : 次のコマンドアクションを 1 つ指定します。
 - **alias** : 標準の UNIX **alias** コマンドに類似。MIT のディレクトリを参照します。
 - **command** : UNIX コマンド (**cat** や **version** など) を実行します。
 - **showcmd** : 設定コマンドの表示オプション (**firmware list** など) を実行します。
- **dirFormat** : スコープのディレクトリ形式を指定します。たとえば、`aci/fabric/inventory/pod-1/node-%(id)s` は、各ノードのサブディレクトリを指定します。



(注) `%(<arg>s` によって、**dirFormat** および **cmdFormat** 文字列の引数を指定します。

- **fileType** : ファイルタイプを指定します。 **summary** または **mo** を指定できます。
- **cmdFormat** : 実行するコマンドを指定します (例 : **cmdFormat: 'eventlog'**)。特定のスコープでコマンドを実行するように指定できます。
- 次のオプションは、コマンドの引数とキーワードです。
 - **sub** : サブスコープを定義します。 **alias** コマンドにのみ適用されます。
 - **name** : 引数またはキーワードの名前。
 - **label** : 引数またはキーワードのラベルを定義します。
 - **type** : サブコマンドのパラメータタイプ。 **arg** は引数を指定し、 **keyword** はキーワードを指定します。
 -
- 次のオートコンプリート用オプションを使用できます。
 - **classfilter** : クラスフィルタを定義します。たとえば、 **classfilter: 'fabric.Node.role == "1"** と指定した場合、結果にはロール値が 1 の MO だけが含まれます。
 - **fill** : スコープの子ディレクトリを表示するには、「**fill: auto**」と入力します。 **alias** コマンドにのみ適用されます。
 - **modelclass** : 結果のオートコンプリートに使用するスコープを定義します。
 - **modelprop** : 結果のオートコンプリートに使用するプロパティ (**name** や **id** など) を定義します。

