



コマンドリファレンス

この章では、次の CLI コマンドについて説明します。

- [コマンドのヘルプ, 2 ページ](#)
- [attach, 3 ページ](#)
- [auditlog, 3 ページ](#)
- [create, 4 ページ](#)
- [コントローラ, 5 ページ](#)
- [診断, 5 ページ](#)
- [eraseconfig, 6 ページ](#)
- [eventlog, 7 ページ](#)
- [障害, 8 ページ](#)
- [ファームウェア, 9 ページ](#)
- [health, 11 ページ](#)
- [loglevel, 12 ページ](#)
- [man, 13 ページ](#)
- [mobrowser, 13 ページ](#)
- [moconfig, 14 ページ](#)
- [mocreate, 15 ページ](#)
- [modelete, 16 ページ](#)
- [mofind, 16 ページ](#)
- [moprint, 17 ページ](#)
- [moquery, 18 ページ](#)
- [moset, 20 ページ](#)

- [mostats, 21 ページ](#)
- [password, 23 ページ](#)
- [reload, 23 ページ](#)
- [scope, 24 ページ](#)
- [show, 24 ページ](#)
- [svcping, 26 ページ](#)
- [techsupport, 26 ページ](#)
- [trafficmap, 28 ページ](#)
- [troubleshoot eptoe session \(IP および MAC\) , 28 ページ](#)
- [troubleshoot epext session EP-to-External-IP and External-IP-to-EP, 29 ページ](#)
- [troubleshoot eptoe session <session name>, 30 ページ](#)
- [troubleshoot eptoe session <session name> atomiccounter, 31 ページ](#)
- [troubleshoot eptoe session <session name> traceroute, 32 ページ](#)
- [troubleshoot eptoe session <session name> traceroute protocol, 32 ページ](#)
- [troubleshoot eptoe session <session name> traceroute protocol tcp dst port, 33 ページ](#)
- [show troubleshoot eptoe, 33 ページ](#)
- [show troubleshoot eptoe session <session name>, 33 ページ](#)
- [version, 34 ページ](#)
- [値は次のとおりです。 , 35 ページ](#)

コマンドのヘルプ

次のツールを使用して CLI コマンドのヘルプを表示できます。

- コマンド名 `-help` : コマンドの要約を表示します。

```
admin@apic1:aci> controller -h
Usage: controller [TARGETNODE_ID] [commission|decommission]

Display controller info. Commission or Decommission controllers.

Options:
  -h --help
```

- `man` コマンド名 : コマンドに関する Linux 形式の `man` ページ (マニュアルページ) を表示します。

```
admin@apic1:aci> man controller
```

attach

attach コマンドは、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。

attach apic1

attach leaf1

attach spine1

例

次の例は、**attach** コマンドを使用して leaf1 ノードに接続する方法を示しています。

```
admin@apic1:aci> attach leaf1
# Executing command: ssh leaf1
Warning: Permanently added 'leaf1,10.0.75.31' (RSA) to the list of known hosts.
admin@leaf1's password:
admin@leaf1:~>
```

auditlog

監査ログには、ログイン時刻やログアウト時刻など、監査情報が含まれています。特定のノード、モジュール、インターフェイスの監査の要約を表示するには、**auditlog** コマンドを使用します。

auditlog[*auditlog-id*]

構文の説明

auditlog-id

表示する監査ログの番号を指定します。

例

次の例は、**auditlog** コマンドの使用例を示しています。

```
admin@apic1:Solar> pwd
/home/admin/aci/tenants/Solar
admin@apic1:Solar> auditlog 4294967305
ID                : 4294967305
Description       : Tenant Solar created
Affected Object   : uni/tn-Solar
Time Stamp        : 2014-07-21T20:00:25.518+00:00
Cause             : transition
Code              : E4206326
Severity          : info
Change Set        : name:Solar
Action Performed  : creation
Action Trigger    : config
Transaction ID    : 14411518807585652035
User              : admin
```

create

create コマンドは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。

create scope

例

次の例は、**create** コマンドの使用方法を示しています。

```
admin@apic1:~> create tenant
# Executing command: 'cd /aci/tenants; ./tenant.wiz'

Create Tenant:
-----
Name          : Cisco
Description   : Cisco Systems
Monitoring Policy:

Security Domains:
-----
Name          :
skipping...

Create new network:
-----
Name          :
skipping...

Do you want to view the corresponding commands? (Yes/No): Yes
-----
mcreate Cisco
pushd .
cd Tenant-Test
moset description "Cisco Systems"

pushd .
cd security-domains
popd

pushd .
cd networking

pushd .
cd private-networks
popd
popd
popd
-----

Do you want to commit changes? (Yes/No): Yes
Adding mo tenants/Cisco
All requests processed successfully!
```

create YAML ファイルの **tenant** セクションは、次のように定義されます。

```
- tenant:
  help: 'Tenant'
  type: alias
  dirFormat: '/aci/tenants/'
  fileType: 'summary'
  createFile: tenant.wiz
  name: tenant
```



(注) YAML (.yaml) ファイルの形式の詳細については、「[コマンドのカスタマイズ](#)」を参照してください。

コントローラ

コントローラの情報を表示したり、ノードをコミッションまたはデコミッションするには、**controller** コマンドを使用します。

controller [*controller-id*] [**commission** | **decommission**]

構文の説明

commission	ノードをコミッション（作成）します。
decommission	指定されたノードをデコミッションします。
<i>controller-id</i>	コントローラの ID。

例

次の例は、**controller** コマンドの使用方法を示しています。

```
admin@apic1:> controller 1 decommission
```

診断

機器の診断テストを表示するには、**diagnostics** コマンドを使用します。

diagnostics *node-id*

構文の説明

<i>node-id</i>	ターゲットのノード ID またはノード名。一連のノード ID やノード名のリストを指定できません。
----------------	---

例

次の例は、**diagnostics** コマンドの使用方法を示しています。

```
admin@apic1:aci> diagnostics 1
Dn                               Group           Model           Subject Class  Test Set
-----
topology/pod-1/node-            internal-conn   N9K-C9396PX    eqptSupC      mgmtp-lb
```

```

19/sys/diag/grptests-
eqptSupC-model-[N9K-
C9396PX]-grp-internal-
conn
topology/pod-1/node-      cpu          N9K-C93128TX  eqptSupC      cpu-cache
19/sys/diag/grptests-
eqptSupC-model-[N9K-
C93128TX]-grp-cpu
topology/pod-1/node-      sys-mem      N9K-C93128TX  eqptSupC      bios-mem,mem-health
19/sys/diag/grptests-
eqptSupC-model-[N9K-
C93128TX]-grp-sys-mem
topology/pod-1/node-      peripherals   Nagano         eqptSupC      act2-acc,cons-dev,fpga-
19/sys/diag/grptests-      reg-chk,ge-
eqptSupC-model-          eeprom,nvram-
[Nagano]-grp-            cksum,obfl-acc,spi-
peripherals              cksum,ssd-acc,usb-bus
topology/pod-1/node-      fex          NXS8-4532     eqptLC        extch-fp,extch-
19/sys/diag/grptests-      hp,extch-sprom
eqptLC-model-[NXS8-
4532]-grp-fex

admin@apic1:aci>

```

eraseconfig

最初の設定情報以外の APIC の設定を消去して、APIC をリブートするには、**eraseconfig** コマンドを使用します。



(注) このコマンドを実行すると、APIC がリブートします。



(注) このコマンドは、APIC Release 1.2(2) 以降のリリースで削除されます。**acidiag touch** コマンドを使用した後でリブートして設定を消去します。『*Cisco APIC Troubleshooting Guide*』の **acidiag** コマンドのドキュメントを参照してください。

eraseconfig [setup]

構文の説明

セッティングアップ	最初の設定情報を消去します。リブート後、最初の APIC 設定ダイアログがコンソールに表示されます。
-----------	--

例

次の例は、**eraseconfig** コマンドの使用方を示しています。

```
admin@apic1:~> eraseconfig
```

eventlog

特定のノード、モジュール、インターフェイスのイベントの要約を表示するには、**eventlog** コマンドを使用します。

eventlog controller *node-id*

eventlog switch *node-id*

eventlog switch interface *interface-name* *node-id*

eventlog switch module *module-id* *node-id*

eventlog switch module *module-id* **port** *port-number* *node-id*

構文の説明

コントローラ	コントローラのイベント ログを表示します。
switch	スイッチのイベント ログを表示します。
<i>node-id</i>	ターゲットのノード ID またはノード名。一連のノード ID やノード名のリストを指定できます。
interface	インターフェイスの ID またはインターフェイスの範囲を指定します。
<i>interface-name</i>	インターフェイスの ID または範囲。
module	モジュールを指定します。
<i>module-id</i>	モジュール ID。

例

次の例は、**eventlog** コマンドの使用方法を示しています。

```
admin@apic1: /> eventlog switch 101 interface eth1/1
```

障害

特定のノード、モジュール、ポート、インターフェイスの障害の要約を表示するには、**faults** コマンドを使用します。

```
faults switch node-id {ack|detail|history| interface interface-name| module module-id port port-number|unack} fault-code
```

```
faults controller controller-id {ack|detail|history| unack} fault-code
```

構文の説明

コントローラ	コントローラの健全性ログを表示します。
<i>controller-id</i>	コントローラを指定します。
switch	スイッチの健全性ログを表示します。
<i>node-id</i>	ターゲットのノードIDまたはノード名。一連のノードIDやノード名のリストを指定できます。
interface	インターフェイスのIDまたはインターフェイスの範囲を指定します。
<i>interface-name</i>	インターフェイスのIDまたは範囲。
module	モジュールを指定します。
<i>module-id</i>	モジュールID。
detail	障害の詳細を表示します。
ack	確認応答された障害を表示します。
unack	確認応答されていない障害を表示します。
history	履歴を表示します。
port	ポート範囲を指定します。
<i>port-number</i>	ポート番号。
<i>fault-code</i>	障害コードを指定します。

例

次の例は、**faults** コマンドの使用方法を示しています。

```
admin@apic1:faults> faults controller 1 detail
```

ファームウェア

ファブリック コントローラ ノードのリポジトリ内のファームウェア イメージを管理するには、**firmware** コマンドを使用します。



(注)

このコマンドは、ローカル コントローラのソフトウェアのアップグレード用です。クラスタ内のファブリック コントローラ ノードのファームウェアをアップグレードするには、ポリシーベースのファームウェア アップグレードを使用します。

firmware add *image-name*

firmware delete *image-name*

firmware upgrade status

firmware upgrade status node *node-id*

firmware upgrade catalog *image-name*

firmware upgrade controller *image-name*

firmware upgrade switch node *node-id image-name*

構文の説明

add	リポジトリにファームウェア イメージを追加します。FTP、SCP、HTTP、または任意のユーザ認証方式を使用して、ファームウェアをダウンロードできます。
delete	リポジトリからファームウェア イメージを削除します。
<i>image-name</i>	イメージ ファイルの名前。
リスト	ファームウェア リポジトリ内のファームウェア イメージを一覧表示します。
upgrade	スイッチまたはローカル APIC のファームウェアをアップグレードします。

コントローラ	コントローラへのローカルイメージのインストールを指定します。
status	ファームウェアの更新状態を表示します。
node-id	ターゲットのノード ID またはノード名。同時に1つのノードにのみファームウェアをインストールできます。 (注) APIC の場合、ファームウェアはクラスタ内のすべてのAPICにインストールされます。
switch	スイッチへのイメージのインストールを指定します。
catalog	イメージカタログ内のイメージをアップグレードします。

例

次の例は、**firmware** コマンドの使用方法を示しています。

```
admin@apic1:~> firmware list
Name                               Type           Major-Version  Minor-Version  Size(Bytes)  Download-Date
-----
ifabric-k9-catalog-1.0.0-566.bin  catalog        1.0            (0.566)        7461         2014-01-
28T11:17:36.054+00:00
admin@apic1:~> firmware add ifabric-k9-simsw-1.0.0-559.bin
Firmware Image ifabric-k9-simsw-1.0.0-559.bin is added to the repository

admin@apic1:~> firmware list
Name                               Type           Major-Version  Minor-Version  Size(Bytes)  Download-Date
-----
ifabric-k9-catalog-1.0.0-566.bin  catalog        1.0            (0.566)        7461         2014-01-
28T11:17:36.054+00:00
ifabric-k9-simsw-1.0.0-559.bin    switch         1.0            (0.559)        854412177    2014-01-

admin@apic1:~> firmware upgrade switch node 17 ifabric-k9-simsw-1.0.0-559.bin
Firmware Installation on Switch Scheduled
To check the upgrade status, use 'firmware upgrade status -t <node-id>'
admin@apic1:~>
admin@apic1:~> firmware upgrade status node 17
Firmware Upgrade Status:
Upgrade-Status  Status  Desired-Version  Install-Stage  Start-Date  End-Date
-----
inprogress      simsw-1.0(0.559)  InstallNotStarted  2014-01-
28T11:26:38.313+00:00

28T10:59:37.746+00:00.
admin@apic1:~> firmware upgrade status
Node-Id  Role  Upgrade-Status
-----
3        controller  notscheduled
17       leaf       completeok
20       spine     notscheduled
1        controller  notscheduled
```

```
2      controller  notscheduled
19     spine      notscheduled
18     leaf       notscheduled
```

health

ノード、モジュール、インターフェイス、ポートの健全性に関する要約を表示するには、**health** コマンドを使用します。

health switch *node-id* {**ack**|**detail**|**history**| **interface** *interface-name*| **module** *module-id* **port** *port-number*| **unack**}

health controller *controller-id* {**ack**|**detail**|**history**| **unack**}

構文の説明

コントローラ	コントローラの障害を表示します。
switch	スイッチの障害を表示します。
<i>node-id</i>	ターゲットのノードIDまたはノード名。一連のノードIDやノード名のリストを指定できます。
interface	1つのインターフェイスまたはインターフェイスの範囲を指定します。
<i>interface-name</i>	インターフェイスの名前または範囲。
module	IDによって1つ以上のモジュールを指定します。
<i>module-id</i>	モジュール名。
port	1つのポートまたはポート範囲を指定します。
<i>port_id</i>	ポートの番号または範囲。
history	履歴を表示します。

例

次の例は、**health** コマンドの使用方法を示しています。

```
admin@apic1:admin> health switch 101 interface eth1/1
Current Score  Previous Score  Timestamp
-----
95             96             2014-07-
21T15:25:24.092+00:00

Total : 1
```

loglevel

APIC のロギング設定を表示するには、**loglevel** コマンドを使用します。

loglevel get node *node-name* **dme** *dme-name*

loglevel set node *node-name* **dme** *dme-name* **topic** *topic-name* **severity** *severity-level*

構文の説明

get	ノードのサービス ログ レベルを返します。
セット	ノードのサービス ログ レベルを設定します。
ノード	ノードを指定します。
<i>node-name</i>	ノード名。
dme	ノードで実行しているサービスプロセスを識別します。
<i>dme-name</i>	サービスプロセス (DME) の名前。使用可能な DME はノードによって異なり、次のような DME があります。 <ul style="list-style-type: none"> • ac • appliancedirector • bootmgr • dbgr • eventmgr • nginx • observer • policymgr • scripthandler • topomgr • vmmmgr
topic	ロギング サブシステムを指定します。
<i>topic-name</i>	ロギング サブシステム。
severity	ロギングの重要度を指定します。

severity-level

ロギングの重要度。次の値を設定できます。

- CRIT : クリティカルエラー
 - ERROR : 重大なエラー
 - WARN : 警告
 - INFO : 情報エラー
 - DBG4 : デバッグ レベル 4
 - DBG3 : デバッグ レベル 3
 - DBG2 : デバッグ レベル 2
-

例

次の例は、**loglevel** コマンドの使用方法を示しています。

```
admin@apic1:pod-1> loglevel get node spine1 dme dbgrelem
logDefault : DBG4
```

man

コマンドの **man** (マニュアル) ページを表示するには、**man** コマンドを使用します。

man *command-name*

構文の説明

command-name コマンド名。

例

次の例は、**man** コマンドの使用方法を示しています。

```
admin@apic1> man trafficmap
```

mobrowser

管理対象オブジェクト (MO) ブラウザを起動するには、**mobrowser** コマンドを使用します。

mobrowser [*scope*]

構文の説明

<i>scope</i>	MIT内のスコープを指定します (aaa、access など)。
--------------	----------------------------------

例

次の例は、**mobrowser** コマンドの使用方法を示しています。

```
admin@apic1:> mobrowser
```

moconfig

設定バッファに格納されている設定をコミットまたは破棄するには、**moconfig** コマンドを使用します。

moconfig{commit|discard|diff|running}

構文の説明

commit	設定バッファに格納されている設定をコミットします。
discard	設定バッファに格納されている設定を破棄します。
diff	アクティブな設定と設定バッファとの相違について要約を表示します。
実行	特定のコンテキストの設定を作成するために使用されたCLIコマンドを表示します。このオプションは、テンプレート設定の作成プロセスを簡易化します。設定テンプレートの詳細については、「 設定テンプレートの作成 」を参照してください。

例

次の例は、**moconfig** コマンドの使用方法を示しています。

```
admin@apic1:local-users> moconfig diff
--- ./mario/mo 2013-10-01 21:17:06.000000000 -0700
+++ ./mario/mo.buffer 2013-10-01 21:17:53.000000000 -0700
@@ -2,8 +2,8 @@
 local-user          :
-----
 login-id            : george
 -first-name         :
 -last-name          :
 +first-name         : George
 +last-name          : Washington
 phone               :
 email               :
 description         :
```

```
admin@apic1:local-users> moconfig commit
Commit Successful
```

```
admin@apic1:local-users> moconfig diff
admin@apic1:local-users>
admin@apic1:aci > cd tenants/
admin@apic1:tenants> moconfig running
cd /aci/viewfw/tenants
cd networking
mocreate fv-tenant-common
moconfig commit
mocreate fv-tenant-test
moconfig commit
mocreate fv-tenant-mgmt
moconfig commit
cd external-routed-networks
mocreate l3ext-out-x
moconfig commit
mocreate l3-outside-x
moconfig commit
cd l3-outside-x
cd logical-node-profiles
mocreate nodex
cd nodex
moset tag yellow-green
moconfig commit
```

mocreate

管理対象オブジェクト (MO) を作成するには、**mocreate** コマンドを使用します。



(注) スコープを指定しない場合は、現在のコンテキストで MO が作成されます。

mocreate [*context*] *name* *property-name* *property-value*

構文の説明

<i>context</i>	MO のコンテキスト。
<i>name</i>	(任意) MO の名前。
<i>property-name</i>	(任意) MO のプロパティを指定します。
<i>property-value</i>	(任意) プロパティの値を指定します。

例

次の例は、**mocreate** コマンドを使用して、ユーザを表す MO を作成する方法を示しています。

```
admin@apic1:node-associations> mocreate LS-all/
admin@apic1:node-associations> moconfig commit
Committed mo
'fabric/policies/fabric-policy-associations/leaf/node/LNP/node-associations/LS-all'
All mos committed successfully.
admin@apic1:node-associations> ls
LS-all
```

デフォルトの設定を上書きするには、次の例のように、**mocreate** コマンドで追加のプロパティを指定します。

```
admin@apic1:private-networks> pwd
/aci/tenants/common/networking/private-networks
admin@apic1:private-networks> mocreate Private1 monitoring-policy Monitor1
```

modelete

管理対象オブジェクト (MO) を削除するには、**modelete** コマンドを使用します。



(注) このコマンドは、通常、下位レベルの範囲を削除する場合に使用します。

modelete *mo-name*

構文の説明

<i>mo-name</i>	MO を格納しているディレクトリの名前。
----------------	----------------------

例

```
admin@apic1:node-associations> modelete LS-all/
```

mofind

管理情報ツリー (MIT) 内の特定の MO を検索するには、**mofind** コマンドを使用します。

mofind *scope class package.class mo-value*

構文の説明

class	クラス引数。MO クラスを返すように指定します。
<i>package</i>	MO のパッケージ名。
<i>class</i>	MO のクラス名。
<i>mo-value</i>	MO の名前。

例

次の例は、**mofind** コマンドの使用方法を示しています。

```
admin@apic1:aci> mofind . class fv.Tenant /.aci/viewfs/tenants/t14/mo
/.aci/viewfs/tenants/infra/mo
/.aci/viewfs/tenants/common/mo
```



```
/.aci/viewfs/tenants/Solar/mo
/.aci/viewfs/tenants/mgmt/mo
admin@apic1:aci> mofind . class aaa.User /.aci/mitfs/uni/userext/user-admin/mo
/.aci/viewfs/admin/aaa/security-management/local-users/admin/mo
```

moprint

管理対象オブジェクトや管理対象オブジェクトバッファ ファイルの出力形式を指定するには、**moprint** コマンドを使用します。



(注) このコマンドを使用すると出力を標準化できるので、自動化する場合に役立ちます。

moprint{**exclude-help**|**include-help**} {**json**|**pretty**|**xml**}

構文の説明

exclude-help	プロパティの説明を除外した出力を指定します。
include-help	プロパティの説明を含む出力を指定します。
json	JSON 出力を指定します。
pretty	表形式による XML 出力を指定します。
xml	XML 出力を指定します。

例

次の例は、**moprint** コマンドを使用して、MOのプロパティを表示する JSON 出力を指定する方法を示しています。

```
admin@apic1:local-users> moprint json
admin@apic1:local-users> cat ./mario/mo
{
  "aaaUser": {
    "attributes": {
      "aaaUserclearPwdHistory": {
        "value": "no"
      },
      "aaaUseremail": {
        "value": ""
      },
      "aaaUserlastName": {
        "value": "Washington"
      },
      "aaaUserphone": {
        "value": ""
      },
      "aaaUserdescr": {
        "value": ""
      },
      "aaaUserexpiration": {
```

```

    "value": "never"
  },
  "aaaUserexpires": {
    "value": "no"
  },
  },
  "aaaUserencPwd": {
    "value": ""
  },
  },
  "aaaUseraccountStatus": {
    "value": "active"
  },
  },
  "aaaUsername": {
    "value": "george"
  },
  },
  "aaaUserfirstName": {
    "value": "George"
  },
  },
  "aaaUserpwdLifeTime": {
    "value": "no-password-expiration"
  },
  },
  "aaaUserpwd": {
    "value": ""
  }
}
}
}

```

moquery

管理対象オブジェクト（MO）に関するクエリーを実行するには、**moquery** コマンドを使用します。

```
moquery {--help| --host host-id| --port portname| --dn dn| --class classname| --filter property| --attrs attributes|
--output output| --user username| --options options}
```

構文の説明

--help または -h	APIC ホストを指定します。
--host または -i	APIC ホストを指定します。
<i>host-id</i>	APIC のホスト名または IP アドレス。
--port または -p	REST インターフェ이스のポートを指定します。
<i>portname</i>	REST インターフェ이스のポート番号。
--dn または -d	管理対象オブジェクト（MO）の識別名（DN）を指定します。
<i>dn</i>	MO の DN。
--class または -c	クエリーのクラス名を指定します。
<i>classname</i>	クラスを指定します。複数のクラスをカンマで区切って入力することもできます。
--filter または -f	MO をフィルタするプロパティを指定します。
<i>Property</i>	MO をフィルタするプロパティ。

--Attrs または -a	クエリーが表示する属性を指定します。
属性	表示する属性のタイプ。 config (設定の属性) または all を選択できます。 config を選択すると、設定可能な属性のみが表示されます。テーブルの出力形式を指定しない限り、デフォルトは all です。
--output または -o	クエリーの出力形式を指定します。
<i>output</i>	クエリーの出力形式。 json 、 XML 、 ブロック 、または テーブル を選択できます。
--user または -u	ユーザ名を指定します。
<i>username</i>	ユーザ名。
--options または -x	クエリーのオプションを指定します。
<i>options</i>	無効にするクエリーオプション。詳細については、使用上のガイドラインを参照してください。

使用上のガイドライン

--options または (**-X**) を使用して、REST API でサポートされているクエリー オプションを指定できます。次のようなコマンド構文を使用して、コマンドに複数のオプションステートメントを追加できます。

```
-x [OPTIONS [OPTIONS ...]][-x [OPTIONS [OPTIONS ...]]]
```

次に例を示します。

```
moquery -c firmwareCtrlrFwStatusCont -x query-target=subtree
target-subtree-class=firmwareCtrlrRunning
```

例

次の例は、**moquery** コマンドの使用方法を示しています。

```
admin@apic1:~> moquery --dn unallocencap-[uni/infra]
Total Objects shown: 1

# stp.UnAllocEncapCont
infraPKey      : uni/infra
allocSize     : 0
childAction   :
descr        :
dn            : unallocencap-[uni/infra]
lastAssigned  : 8192
lcOwn        : local
modTs        : 2014-07-26T16:46:27.176+00:00
name         :
ownerKey      :
ownerTag     :
rn           : unallocencap-[uni/infra]
```

```
size          : 0
status       :
```

moset

管理対象オブジェクト（MO）のプロパティを設定するには、**moset** コマンドを使用します。

```
moset { property-name property-value [add | remove ] }
```

構文の説明

<i>property-name</i>	プロパティ名。
<i>property-value</i>	プロパティ値
add	管理対象オブジェクトにプロパティを追加します。
remove	管理対象オブジェクトからプロパティを削除します。

例

次の例は、**moset** コマンドを使用して、管理対象オブジェクトのプロパティを設定する方法を示しています。

```
admin@apic0:local-users> cat george/mo
# aaa.User
local-user          :
-----
login-id            : george
first-name          :
last-name           :
phone               :
email               :
description         :
account-status      : active
account-expires     : no
expiration-date     : never
clear-password-history : no
encrypted-password  :
password            :
password-life-time  : no-password-expiration
admin@apic0:local-users> moset first-name George last-name Washington
admin@apic0:local-users> cat mario/mo.buffer
# aaa.User
local-user          :
-----
login-id            : george
first-name          : George
last-name           : Washington
phone               :
email               :
description         :
account-status      : active
account-expires     : no
expiration-date     : never
clear-password-history : no
encrypted-password  :
password            :
```

```
password-life-time      : no-password-expiration
admin@ifc0:local-users>
```

mostats

MO の統計情報を表示するには、**mostats** コマンドを使用します。

mostats [*stats-class*] [**sampling-interval** *interval*] [**location** *location-name*] [**counter** *counter-name*] [**values** *values-name*] [**from** *date-from*] [**to** *date-to*] [**thresholded** *thresholded-flags*] [**output-to** *outputname*]

構文の説明

<i>stats-class</i>	統計情報のタイプ。現在のスコープで使用可能な統計情報のリストを表示するには、Tab オートコンプリート機能を使用します。
sampling-interval	統計情報のサンプリング間隔を指定します。
間隔	サンプリング間隔。次の値を選択できます。 <ul style="list-style-type: none"> • 5 分 • 15 分 • 1h • 1d • 1w • 1mo • 1qtr • 1year <p>デフォルト値は 5 分です。</p>
場所	統計情報の表示対象の場所を指定します。
<i>location-name</i>	統計情報の表示対象の場所。history または current を選択できます。
counter	表示する特定のカウンタを指定します。このキーワードを省略した場合は、すべてのカウンタが表示されます。
<i>counter-name</i>	カウンタ名。カウンタ名を指定しない場合は、すべてのカウンタの値が表示されます。 <p>オートコンプリート機能を使用すると、使用可能なカウンタを一覧表示できます。</p>
値	表示する特定の値を指定します。

<i>values-name</i>	表示する値のタイプ。オートコンプリート機能を使用すると、使用可能な値を一覧表示できます。 (注) 統計値は、指定したカウンタおよび場所に応じて異なります。
from	統計情報の開始日時を指定します。このキーワードは履歴統計情報に使用されます。
<i>date-from</i>	クエリの開始日。
to	統計の終了日時を指定します。このキーワードは履歴統計情報に使用されます。
<i>date-to</i>	クエリの終了日。
thresholded	しきい値を超えている履歴統計を指定します。
<i>thresholded-flags</i>	しきい値のフラグ値。
output-to	特定の出力タイプを指定します。
<i>output-name</i>	出力のタイプ。次の値を選択できます。 • table • グラフ

例

次の例は、**mostats** コマンドの使用方法を示しています。

```
admin@apic0:leafport-17> mostats ingress-byte-counters location history
Counters:
  flood (bytes) : periodic value
  multicastRate (bytes-per-second) : average value
  multicast (bytes) : periodic value
  unicastRate (bytes-per-second) : average value
  unicast (bytes) : periodic value
```

Time Interval	flood	multicastRate	multicast	unicastRate	unicast
2013-10-23 13:40:10 + 300sec	1692622494	6038011	1811403699	5959938	1787981697
2013-10-23 13:45:10 + 290sec	1701770043	5896513	1709988944	6350713	1841707150
2013-10-23 13:50:00 + 300sec	1875699742	6327240	1898172394	5204047	1561214263
2013-10-23 13:55:00 + 300sec	1991025635	6407343	1922203057	5961950	1788585183
2013-10-23 14:00:00 + 310sec	2020555778	6857403	2125795303	7152710	2217340307
2013-10-23 14:05:10 + 290sec	1884001802	6545303	1898138103	5878862	1704870238
2013-10-23 14:10:00 + 310sec	2037567241	5880848	1823063295	6927670	2147577849
2013-10-23 14:15:10 + 300sec	1651084097	6128338	1838501627	5696007	1708802494

```
2013-10-23 14:20:10 + 300sec 2119253728 5719718 1715322961 5606184 1681939173
2013-10-23 14:25:10 + 300sec 1824918785 6553074 1965922597 6167935 1850380704
2013-10-23 14:30:10 + 300sec 1794072506 6508516 1952555134 6745063 2023519193
2013-10-23 14:35:10 + 290sec 2305467846 6493923 1883237807 6693507 1941117370
```

password

APIC のパスワードを変更するには、**password** コマンドを使用します。

password

例

次の例は、**password** コマンドの使用方法を示しています。

```
admin@apic1:aci> passwd
Changing password for user admin.
(current) password:
New password:
Retype new password:
Password for user admin is changed successfully.
admin@apic1:aci>
```

reload

特定のノードまたはモジュールをリロードするには、**reload** コマンドを使用します。



(注)

ノードを指定しない場合は、現在のコンテキストのノードがリロードされます。

reload {**controller**| **switch**} *node-id*

構文の説明

コントローラ	コントローラをリロードします。
switch	スイッチをリロードします。
<i>node-id</i>	ターゲットのノードIDまたはノード名。一連のノードIDやノード名のリストを指定できます。

例

次の例は、**reload** コマンドの使用方法を示しています。

```
admin@apic1:aci> reload switch 118
```

scope

スコープのディレクトリにジャンプするには、**scope** コマンドを使用します。



(注) **where** コマンドはコンテキストの MIT ディレクトリを表示しますが、**scope** はディレクトリを開きます。

scope *scope-name*

構文の説明

scope-name スコープ名 (aaa、access-policies など)。

例

次の例は、**scope** コマンドの使用方法を示しています。

```
admin@apic1:~> pwd
/home/admin
admin@apic1:~/> scope tenant
Changing directory to /.aci/tenants/
admin@apic1:tenants> pwd
/aci/tenants
```

show

show コマンドは、Cisco IOS や NX-OS に類似した形式で APIC の設定を表示します。このコマンドは、Linux の **alias** コマンドに似ています。

show *context*

構文の説明

コンテキスト コンテキスト名 (aaa、access-policies など)。

Contexts

次の例は、標準の **show** コマンド オプションを示しています。

```
admin@apic1:~> show <Esc><Esc>
aaa                aaa
access             Fabric Access Policies
auditlog           Show auditlog on current path
bgp                Show BGP information
cdp                Show Cisco Discovery Protocol information
controller         Controller Node
cores              cores
```



```

eventlog                Show eventlog on current path
external-data-collectors external-data-collectors
fabric                  Fabric Details
faults                  Show faults current path
fex                     Show fex information
firmware                Show firmware
health                  Show health on current path
historical-record-policy historic-record-policies
import-export           Import/Export
interface               Show interface status and information
interface-policies      interface-policies
ip                      Display IP information
isis                    Display IS-IS status and configuration
l4-l7                   L4-L7 Sevices Details
lldp                    Show information about lldp
module                  Show module information
schedulers              schedulers
switch                  Switch Node
tenant                  Tenant
trafficmap              Show trafficmap
version                 Show version
vmware                  VMware vCenter/vShield Controllers
vpc                     Show vpc information

```

show コマンドのカスタマイズ

シンプルな YAML (.yaml) 設定を使用して show コマンドをカスタマイズできます。たとえば、/etc/scopedefs ディレクトリの .yaml ファイルを参照してください。

/home/username/scopedefs/ ディレクトリに .yaml ファイルを作成して、show カスタム コマンドを定義できます。また、/home/username/scopedefs/.ignore.yaml ファイルに特定の show スコープを追加することによって、そのスコープを無視できます。

次の例に示す cmdFormat 値のように、特定のスコープで実行する show カスタム コマンドを定義することもできます。

```

vmware :
  type: alias
  help: "VMware vCenter/vShield Controllers"
  name: vmware
  label: vmware
  sub:
    - name: controllers
      label: controllers
      type: keyword
      cmdFormat: "find /aci/vm-networking/inventory/VMware/vmm-domains/ -name controllers
    -exec echo {';' -exec echo {';' -exec cat '{}/summary' {';'"
      help: "Status of all Controllers"
    - name: domain
      label: domain
      type: keyword
      help: "Domain"

```



(注) YAML (.yaml) ファイルの形式の詳細については、「[コマンドのカスタマイズ](#)」を参照してください。

例

次の例は、**show** コマンドを使用して、ローカル ユーザを表示する方法を示しています。

```

admin@apic1:~> show aaa local-users
# Executing command: cat /aci/admin/aaa/security-management/local-users/summary

local-users:

```

```
login-id  first-name  last-name  email  phone
-----  -
admin
```

次の抜粋は、**show** コマンドの **aaa** スコープの YAML 定義を示しています。

```
- aaa:
  name: aaa
  help: 'aaa'
  type: alias
  dirFormat: ' '
  sub:
    - name: local-users
      label: local-users
      type: keyword
      dirFormat: '/aci/admin/aaa/security-management/local-users/'
      fileType: 'summary'
      help: 'local users'
```

svcping

サービス デバイスの管理インターフェイスを ping するには、**svcping** コマンドを使用します。



(注)

このコマンドは管理情報ツリー (MIT) 内でサポートされます。aci ファイル システム内ではサポートされません。

svcping path

構文の説明

path mit ファイル システム内のサービス デバイス (CDev) のパス。

techsupport

トラブルシューティング情報を表示するには、**techsupport** コマンドを使用します。

```
techsupport all { [status] | [remotename fname] }
```

```
techsupport controllers [status]
```

```
techsupport controllers remotename fname
```

```
techsupport db svc svcname [delete]
```

```
techsupport local
```

```
techsupport remote { list | name } [fname] {delete | [ {host remoteport protocol username password  
remotepath } ] }
```

```
techsupport switch nodeid { [status] | [remotename fname] }
```

構文の説明

all	ACI ファブリック内の全ノードのテクニカル サポート情報を表示します。
controllers	ファブリック コントローラの障害を表示します。
db	データベース情報のスナップショットを収集します。
delete	テクニカル サポート ファイルを削除します。
<i>fname</i>	リモート接続先の名前。
ホスト	リモート ホストの名前。
リスト	すべてのリモート接続先のリスト。
local	テクニカル サポート情報をローカルに収集します。
name	リモート接続先を指定します。
<i>node-id</i>	ターゲットのノードIDまたはノード名。一連のノードIDやノード名のリストを指定できます。
リモート	テクニカルサポート情報のリモート宛先を一覧表示、追加、または削除します。
remotename	リモート接続先の名前。
<i>remotepath</i>	リモート接続先のパス。
<i>remoteport</i>	リモート ポート番号。
<i>password</i>	リモート接続先のパスワード。
<i>protocol</i>	リモート接続先のプロトコル。
status	テクニカル サポートの出力のステータス。
svc	サービスを指定します。
<i>svcname</i>	サービス名
switch	スイッチの障害を表示します。
<i>username</i>	リモート接続先のユーザ名。

Techsupport コマンドは、ACIファブリックまたはノードの現在の状態に関する情報を含むファイルをエクスポートします。この情報は、シスコサポートに非常に有益で、問題の原因を特定するために必要な情報を提供します。指定したリモート接続先にファイルがエクスポートされます。

Cisco APIC リリース 1.1 以降、3つのファイルがこのコマンドで作成され、エクスポートされます。

- `filename.tar.gz` : 設定ファイル、障害、イベント、デバッグカウンタ、およびその他のシステム情報が含まれます。
- `filename_db.tar.gz` : シヤードおよびレプリカが 1 ずつあるノードから集めたデータベース (.db ファイル) が含まれます。
- `filename_logs.tar.gz` : ノードから集められたすべてのログが含まれます。スイッチ ノードの場合は、NX-OS の `techsupport` データはこのファイルに含まれています。

例

次の例は、Cisco APIC リリース 1.1 以前のリリースでの `techsupport` コマンドの使用方法を示します。

```
admin@apic1:~> techsupport switch 101
Triggering techsupport for Switch 101 using policy supNode101
Triggered on demand tech support successfully for node 101, will be available at:
/data/techsupport on the controller.
Use 'status' option with your command to check techsupport status
```

trafficmap

2 つのノード間のトラフィックの要約を表示するには、`trafficmap` コマンドを使用します。

controller srcnode *source-node-id* **destnode** *dest-node-id*

構文の説明

srcnode	ノード名を指定します。
<i>source-node-id</i>	送信元ノードの名前。
destnode	宛先ノードを指定します。
<i>dest-node-id</i>	宛先ノードの名前。

例

次の例は、`trafficmap` コマンドの使用方法を示しています。

```
admin@apic1:> trafficmap srcnode 102 destnode 112
```

troubleshoot eptoep session (IP および MAC)

IP トラブルシューティングセッションを作成するには、`troubleshoot eptoep session <session_name> srcip <src_ip> tenant <src_tenant> app <src_app> epg <src_epg> destip <dest_ip> tenant <dest_tenant> app <dest_app> epg <dest_epg>` コマンドを使用します。

MAC トラブルシューティングセッションを作成するには、**troubleshoot eptoe session** `<session_name> srcmac <src_mac> tenant <src_tenant> app <src_app> epg <src_epg> destmac <dest_mac> tenant <dest_tenant> app <dest_app> epg <dest_epg>` コマンドを使用します。

セッションが作成されると、次の設定オプションが使用できます。

- atomiccounter start
- atomiccounter stop
- traceroute start
- traceroute stop
- traceroute protocol <prot> dstport <dst_port>
- report [<format>]
- delete
- description <descr>
- latestminutes <num_min>
- starttime <start_time> endtime <end_time>
- monitor destination tenant <tenant_name> application <appln> epg <epg_name> ip_addr <ip_address> srcipprefix <ip_prefix> [(flowid <flow_id>)]
- monitor stop
- scheduler <scheduler-name>
- scheduler delete

例

次の例は、IP の **troubleshoot eptoe session** の作成方法を示します。

```
admin@apic1: /> troubleshoot eptoe session <session_name> srcip <src_ip> tenant <src_tenant>
  app <src_app> epg <src_epg> destip <dest_ip> tenant <dest_tenant> app <dest_app> epg
  <dest_epg>
```

次の例は、MAC の **troubleshoot eptoe session** の作成方法を示します。

```
admin@apic1: /> troubleshoot eptoe session <session_name> srcmac <src_mac> tenant <src_tenant>
  app <src_app> epg <src_epg> destmac <dest_mac> tenant <dest_tenant> app <dest_app> epg
  <dest_epg>
```

troubleshoot epext session EP-to-External-IP and External-IP-to-EP

EP ツー外部 IP アドレス トラブルシューティングセッションを作成するには、**troubleshoot epext session** `<session_name> srcip <src_ip> tenant <src_tenant> app <src_app> epg <src_epg> destextip <dest_ip>` コマンドを使用します。

外部 IP アドレス ツー EP トラブルシューティングセッションを作成するには、**troubleshoot ept session <session_name> srcextip <src_ip> destip <dest_ip> tenant <dest_tenant> app <dest_app> epg <dest_epg>** コマンドを使用します。

セッションが作成されると、次の設定オプションが使用できます。

- atomiccounter start
- atomiccounter stop
- traceroute start
- traceroute stop
- traceroute protocol <prot> dstport <dst_port>
- report [<format>]
- delete
- description <descr>
- latestminutes <num_min>
- starttime <start_time> endtime <end_time>
- monitor destination tenant <tenant_name> application <appln> epg <epg_name> ip_addr <ip_address> srcipprefix <ip_prefix> [(flowid <flow_id>)]
- monitor stop
- scheduler <scheduler-name>
- scheduler delete

例

次の例は、外部 IP の **troubleshoot ept session** の作成方法を示します。

```
admin@apic1:/> troubleshoot ept session <session_name> srcextip <src_ip> destip <dest_ip>
tenant <dest_tenant> app <dest_app> epg <dest_epg>
```

troubleshoot eptoe session <session name>

トラブルシューティングセッションをスケジュールするには、**schedule troubleshoot eptoe session <session name>** オプションコマンドを使用します。

構文の説明

atomiccounter	送信元と宛先のエンドポイント間のアトミック カウンタを設定します。
delete	このトラブルシューティングセッションを削除します。

説明	このトラブルシューティングセッションのテキスト説明。
latestminutes	現在時刻から分単位でタイム ウィンドウを入力します。
モニタ	送信元および宛先インターフェイスにまたがるモニタセッションを設定します。
レポート	トラブルシューティング レポートを生成します。
scheduler	このセッションのスケジュールを設定します。
srcip	送信元エンドポイント IP を設定します。
srcmac	送信元エンドポイント MAC を設定します。
starttime	問題が発生した時刻。
traceroute	2つのエンドポイント間の traceroute セッションを設定します。

例

次の例は、**troubleshoot eptoe session <session name>** コマンドの使用方法を示しています。

```
admin@apic1:/> troubleshoot eptoe session <session name>report
```

troubleshoot eptoe session <session name> atomiccounter

新しいエンドポイント (ep) ツー エンドポイントアトミック カウンタ セッションを設定するには、**troubleshoot eptoe session newSession atomiccounter** オプションコマンドを使用します。

構文の説明

start	atomiccounter セッションを開始します。
stop	atomiccounter セッションを停止します。

例

次の例は、**troubleshoot eptoe session <session name> atomiccounter** コマンドの使用方法を示しています。

```
admin@apic1:/> troubleshoot eptoe session <session name> atomiccounter start
```

troubleshoot eptoe session <session name> traceroute

新しいエンドポイント (ep) ツーエンドポイント traceroute セッションを設定するには、**troubleshoot eptoe session <session name> traceroute** オプションコマンドを使用します。

構文の説明

protocol	traceroute プロトコルを設定します。
start	traceroute ポリシーを開始します。
stop	traceroute ポリシーを停止します。

例

次の例は、**troubleshoot eptoe session <session name> traceroute** コマンドの使用方法を示しています。

```
admin@apic1:/> troubleshoot eptoe session <session name> traceroute start
```

troubleshoot eptoe session <session name> traceroute protocol

新しいエンドポイント (ep) ツーエンドポイント traceroute プロトコルセッションを設定するには、**troubleshoot eptoe session <session name> traceroute protocol** オプションコマンドを使用します。

構文の説明

<prot>	IP プロトコル (tcp udp icmp) を指定します。
---------------------	---------------------------------

例

次の例は、**troubleshoot eptoe session <session name> traceroute protocol** コマンドの使用方法を示しています。

```
admin@apic1:/> troubleshoot eptoe session <session name> traceroute protocol icmp
```


troubleshoot eptoe session <session name> traceroute protocol tcp dst port

新しいエンドポイント (ep) ツーエンドポイント traceroute プロトコルセッションを設定するには、**troubleshoot eptoe session <session name> traceroute protocol tcp** オプションコマンドを使用します。

構文の説明

<dstport>	traceroute で使用される宛先 L4 ポートを指定します。
------------------------	-----------------------------------

例

次の例は、**troubleshoot eptoe session <session name> traceroute protocol** コマンドの使用方法を示しています。

```
admin@apic1: /> troubleshoot eptoe session <session name> traceroute protocol tcp dstport 80
```

show troubleshoot eptoe

エンドポイント接続にエンドポイント (ep) を表示するには、**show troubleshoot eptoe** オプションコマンドを使用します。

構文の説明

session	セッション名情報を表示します。
sessions	すべてのセッション名を表示します。

例

次の例は、**show troubleshoot eptoe** コマンドの使用方法を示しています。

```
admin@apic1: /> show troubleshoot eptoe
```

show troubleshoot eptoe session <session name>

エンドポイント (ep) ツー MAC セッションを表示するには、**show troubleshoot eptoe session <session name>** オプションコマンドを使用します。

構文の説明

atomiccounter	アトミック カウンタを表示します。
----------------------	-------------------

監査	監査情報を表示します。
contracts	担当者情報を表示します。
deployments	導入の変更を表示します。
events	イベントを表示します。
障害	障害を表示します。
モニタ	モニタ ステータスを表示します。
レポート	レポートを表示します。
統計情報	統計情報を表示します。
トポロジ	トポロジを表示します。
traceroute	Traceroute の結果を表示します。

例

次の例は、**show troubleshoot eptoe session <session name>** コマンドの使用方法を示しています。

```
admin@apic1:/> show troubleshoot eptoe session <session name>
```

version

ノードの現在のソフトウェア バージョンを表示するには、**version** コマンドを使用します。



(注)

ノードを指定しない場合は、設定されている全ノードの現在のソフトウェア バージョンが表示されます。

version {controller| switch} [node-id]

構文の説明

コントローラ	コントローラのバージョンを表示します。
switch	スイッチのバージョンを表示します。
node-id	ターゲットのノードIDまたはノード名。一連のノードIDやノード名のリストを指定できます。

■ 値は次のとおりです。