Cisco APIC オブジェクト モデル コマンドライン インターフェイス ユーザ ガイド

最終更新：2015 年 12 月 08 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system.All rights reserved.Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号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、ネットワーク トポロジ図、およびその他の図は、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および電話番号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。

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit.（http://www.openssl.org/）

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

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: 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)

FSF および GNU に関する一般的な質問は、gnu@gnu.org までお送りください。別の方法で FSF に連絡することもできます。リンク切れやその他の修正、または提案については、bug-bash@gnu.org までご連絡ください。本記事の翻訳の調整および提出については、翻訳 README をご覧ください。

Copyright © 2007, 2009, 2011 Free Software Foundation, Inc. 本許可告知および著作権表示を残す限り、媒体を問わず、使用料を支払うことなく、世界中で本記事全体を複製および配布することを許可する。最終更新日:2011/06/28 02:44:32

© 2014-2015 Cisco Systems, Inc. All rights reserved.

目 次

はじめに vii

対象読者 vii

表記法 vii

関連資料 ix

マニュアルに関するフィードバック xi

コマンドライン インターフェイスの概要 1

Application Policy Infrastructure Controller について 1

設定オプション 1

管理対象オブジェクトの理解 2

ファイル システムの理解 2

GNU Bash シェルの理解 3

Bash 拡張機能 4

ネットワークの命名規則 4

インターフェイス名 4

ネットワーク アドレスの命名 4

コマンドの補完 4

コマンド履歴 5

コマンドのヘルプ 5

マウント ポイント 5

aci マウント ポイント 6

mit マウント ポイント 6

debug マウント ポイント 6

ロールベース アクセス コントロール 6

権限とセキュリティの適用 6

ユーザ管理 7

APIC CLI の使用 9

オブジェクト モデル CLI へのアクセス 9
管理対象オブジェクトの表示 10
管理情報ツリーでの移動 10
MO ブラウザ ユーティリティ 11
設定の入力 11
相違を表示するコマンド 12
設定ウィザードの使用 12
プロパティのスキップ 13
設定テンプレートの作成 14
moconfig コマンドによるテンプレートの作成 14
設定ウィザードによるテンプレートの作成 15
コマンドのカスタマイズ 16
YAML コマンド定義の例 16
YAML ファイルの形式 18
コマンド リファレンス 21
コマンドのヘルプ 22
attach 23
auditlog 23
create 24
controller 25
diagnostics 25
eraseconfig 26
eventlog 27
faults 28
firmware 29
health 31
loglevel 32
man 33
mobrowser 33
moconfig 34
mocreate 35
modelete 36
mofind 36
moprint 37

moquery 38
moset 40
mostats 41
password 43
reload 43
scope 44
show 44
svcping 46
techsupport 46
trafficmap 48
troubleshoot eptoep session（IP および MAC） 49
troubleshoot epext session EP-to-External-IP and External-IP-to-EP 50
troubleshoot eptoep session <session name> 50
troubleshoot eptoep session <session name> atomiccounter 51
troubleshoot eptoep session <session name> traceroute 52
troubleshoot eptoep session <session name> traceroute protocol 52
troubleshoot eptoep session <session name> traceroute protocol tcp dst port 53
show troubleshoot eptoep 53
show troubleshoot eptoep session <session name> 54
version 54
where 55

はじめに

この前書きは、次の項で構成されています。

- 対象読者，vii ページ
- 表記法，vii ページ
- 関連資料，ix ページ
- マニュアルに関するフィードバック，xi ページ

対象読者

このガイドは、アプリケーション セントリック インフラストラクチャ ファブリック の設定および維持に携わるネットワーク管理者およびシステム管理者を対象としています。

表記法

コマンドの説明には、次のような表記法が使用されます。

表記法	説明
bold	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
italic	イタリック体の文字は、ユーザが値を入力する引数です。
[x]	省略可能な要素（キーワードまたは引数）は、角カッコで囲んで示しています。
[x | y]	いずれか 1 つを選択できる省略可能なキーワードや引数は、角カッコで囲み、縦棒で区切って示しています。

表記法	説明
{x | y}	必ずいずれか1つを選択しなければならない必須キーワードや引数は、波カッコで囲み、縦棒で区切って示しています。
[x {y | z}]	角カッコまたは波カッコが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角カッコ内の波カッコと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。
`variable`	ユーザが値を入力する変数であることを表します。イタリック体が使用できない場合に使用されます。
string	引用符を付けない一組の文字。string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。

例では、次の表記法を使用しています。

表記法	説明
`screen フォント`	スイッチが表示する端末セッションおよび情報は、screen フォントで示しています。
`太字の screen フォント`	ユーザが入力しなければならない情報は、太字の screen フォントで示しています。
イタリック体の *screen* フォント	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
< >	パスワードのように出力されない文字は、山カッコ（<>）で囲んで示しています。
[ ]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符（!）またはポンド記号（#）がある場合には、コメント行であることを示します。

このマニュアルでは、次の表記法を使用しています。

（注） 「注釈」です。役立つ情報やこのマニュアルに記載されていない参照資料を紹介しています。

注意 「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

警告 安全上の重要事項

「危険」の意味です。人身事故を予防するための注意事項が記述されています。機器の取り扱い作業を行うときは、電気回路の危険性に注意し、一般的な事故防止対策に留意してください。各警告の最後に記載されているステートメント番号を基に、装置に付属の安全についての警告を参照してください。

これらの注意事項を保存しておいてください。

関連資料

アプリケーション セントリック インフラストラクチャのマニュアル セットには、次の URL の Cisco.com から入手可能な次のドキュメントが含まれます。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

Web ベースのマニュアル

- 『*Cisco APIC Management Information Model Reference*』
- 『*Cisco APIC Online Help Reference*』
- 『*Cisco APIC Python SDK Reference*』
- 『*Cisco ACI Compatibility Tool*』
- 『*Cisco ACI MIB Support List*』

ダウンロード可能なドキュメント

- ナレッジ ベースの記事（KB 記事）は、次の URL から入手できます。 http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html
- 『*Cisco Application Centric Infrastructure Controller Release Notes*』
- 『*Cisco Application Centric Infrastructure Fundamentals Guide*』
- 『*Cisco APIC Getting Started Guide*』
- 『*Cisco ACI Basic Configuration Guide*』
- 『*Cisco ACI Virtualization Guide*』
- 『*Cisco APIC REST API User Guide*』

- 『*Cisco APIC Object Model Command Line Interface User Guide*』
- 『*Cisco APIC NX-OS Style Command-Line Interface Configuration Guide*』
- 『*Cisco APIC Faults, Events, and System Messages Management Guide*』
- 『*Cisco ACI System Messages Reference Guide*』
- 『*Cisco APIC Layer 4 to Layer 7 Services Deployment Guide*』
- 『*Cisco APIC Layer 4 to Layer 7 Device Package Development Guide*』
- 『*Cisco APIC Layer 4 to Layer 7 Device Package Test Guide*』
- 『*Cisco ACI Firmware Management Guide*』
- 『*Cisco ACI Troubleshooting Guide*』
- 『*Cisco APIC NX-OS Style CLI Command Reference*』
- 『*Cisco ACI Switch Command Reference, NX-OS Release 11.0*』
- 『*Verified Scalability Guide for Cisco ACI*』
- 『*Cisco ACI MIB Quick Reference*』
- 『*Cisco Nexus CLI to Cisco APIC Mapping Guide*』
- 『*Application Centric Infrastructure Fabric Hardware Installation Guide*』
- 『*Cisco NX-OS Release Notes for Cisco Nexus 9000 Series ACI-Mode Switches*』
- 『*Nexus 9000 Series ACI Mode Licensing Guide*』
- 『*Cisco Nexus 9332PQ ACI-Mode Switch Hardware Installation Guide*』
- 『*Cisco Nexus 9336PQ ACI-Mode Switch Hardware Installation Guide*』
- 『*Cisco Nexus 9372PX and 9372PX-E ACI-Mode Switch Hardware Installation Guide*』
- 『*Cisco Nexus 9372TX ACI-Mode Switch Hardware Installation Guide*』
- 『*Cisco Nexus 9396PX ACI-Mode Switch Hardware Installation Guide*』
- 『*Cisco Nexus 9396TX ACI-Mode Switch Hardware Installation Guide*』
- 『*Cisco Nexus 93128TX ACI-Mode Switch Hardware Installation Guide*』
- 『*Cisco Nexus 9504 NX-OS Mode Switch Hardware Installation Guide*』
- 『*Cisco Nexus 9508 ACI-Mode Switch Hardware Installation Guide*』
- 『*Cisco Nexus 9516 ACI* モード スイッチ ハードウェア設置ガイド』

シスコ アプリケーション セントリック インフラストラクチャ（ACI）シミュレータのマニュアル

次のシスコ ACI シミュレータのマニュアルは、次の URL から入手できます。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-centric-infrastructure-simulator/tsd-products-support-series-home.html

- 『*Cisco ACI Simulator Release Notes*』
- 『*Cisco ACI Simulator Installation Guide*』
- 『*Cisco ACI Simulator Getting Started Guide*』

Cisco Nexus 9000 シリーズ スイッチのマニュアル

Cisco Nexus 9000 シリーズ スイッチのマニュアルは、次の URL で入手できます。http://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/tsd-products-support-series-home.html

Cisco Application Virtual Switch のマニュアル

Cisco Application Virtual Switch（AVS）のマニュアルは、次の URL で入手できます。http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/tsd-products-support-series-home.html

シスコ アプリケーション セントリック インフラストラクチャ（ACI）と OpenStack の統合に関するマニュアル

Cisco ACI と OpenStack の統合に関するマニュアルは、次の URL から入手できます。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、apic-docfeedback@cisco.com までご連絡ください。ご協力をよろしくお願いいたします。

第 1 章

コマンドライン インターフェイスの概要

- Application Policy Infrastructure Controller について，1 ページ
- 設定オプション，1 ページ
- 管理対象オブジェクトの理解，2 ページ
- ファイル システムの理解，2 ページ

Application Policy Infrastructure Controller について

このガイドでは、Application Policy Infrastructure Controller（APIC）のコマンドライン インターフェイス（CLI）の使用方法について説明します。CLI は、標準の Bash コマンド言語インタプリタ シェルおよび APIC 用の一連のカスタム コマンドから構成されています。

API のクラス、メソッド、タイプの詳細については、Web ベースのアプリケーションである『*Cisco APIC Management Information Model Reference*』を参照してください。Application Policy Infrastructure Controller の機能と運用については、提供されているホワイト ペーパーおよび『*Cisco Application Centric Infrastructure Fundamentals*』を参照してください。

設定オプション

Cisco Application Policy Infrastructure Controller（APIC）は、次の設定オプションを備えています。

- オブジェクト モデル CLI を使用した直接設定：CLI オブジェクトの拡張機能を Bash シェルに使用して、管理対象オブジェクト（MO）および管理情報ツリー（MIT）を直接操作できます。このマニュアルでは、オブジェクト モデル CLI を使用した直接設定について説明します。
- NX-OS スタイルの CLI：Cisco APIC リリース 1.2 以降では、構成には NX-OS スタイルの CLI コマンドを使用できます。

（注） このドキュメントでは、APIC NX-OS スタイル の CLI インターフェイスに関する情報を記載していません。詳細については、『*Cisco APIC NX-OS Style Command-Line Interface Configuration Guide*』を参照してください。

- シェル スクリプト：Bash シェルを使用し、シェル スクリプトによって一部のタスクを自動化できます。Bash の詳細については、「GNU Bash シェルの理解」を参照してください。
- Python API：さらに広範な自動化が可能になります。Python API の詳細については、『*Cisco APIC Python SDK Reference*』 を参照してください。

（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。オブジェクト モデル CLI は、最初の CLI プロンプトでの **Bash** コマンドの入力に使用できます。

管理対象オブジェクトの理解

APIC システムの設定とステータスは、管理対象オブジェクト（MO）の集合としてモデル化されます。MO は、一連の設定とプロパティを含む物理または論理エンティティの抽象的表現です。たとえば、サーバ、シャーシ、I/O カード、プロセッサなどは、MO として表現される物理エンティティです。また、リソース プール、ユーザ ロール、サービス プロファイル、ポリシーなどは、MO として表現される論理エンティティです。

実行時、すべての MO は管理情報ツリーと呼ばれるツリー構造に編成されます。これによって、システム内のすべての MO に構造化された一貫性のある方法でアクセスできるようになります。

ファイル システムの理解

管理情報ツリー（MIT）を構成する MO は階層的に編成されていて、APIC の管理に使用できます。各 MO は Linux ディレクトリとしてモデル化されます。このディレクトリには、すべての子 MO がサブディレクトリとして含まれ、すべてのプロパティを含む mo ファイルも格納されます。

以下にファイル システムの例を示します。local-users ディレクトリには 3 人のユーザ（admin、john、閲覧者）用のサブディレクトリがあります。

```
admin@apic1:local-users> pwd
/home/admin/aci/admin/aaa/security-management/local-users
admin@apic1:local-users> ls -al
total 3
drw-rw---- 1 admin admin 512 Apr 10 16:58 .
drw-rw---- 1 root  root  512 Apr  8 07:06 ..
drw-rw---- 1 root  root  512 Apr  8 07:06 admin
drw-rw---- 1 admin admin 512 Jan 28 20:16 john
-r--r----- 1 admin admin 197 Apr 10 16:58 summary
```

ロールベース アクセス コントロール（RBAC）では、他のユーザを管理する権限をユーザに与えることができます。この例では、admin と閲覧ユーザはルート（root）によって所有され、john は admin によって所有されています。

（注） このディレクトリには mo ファイルがありません。これは、このディレクトリ レベルには設定可能なプロパティが存在しないことを示しています。

```
admin@apic1:local-users> cd admin
admin@apic1:admin> pwd
/home/admin/aci/admin/aaa/security-management/local-users/admin
admin@apic1:admin> ls -al
total 4
drw-rw---- 1 admin admin 512 Jul 22 14:29 .
drw-rw---- 1 admin admin 512 Jul 22 14:29 ..
-rw-rw---- 1 admin admin 485 Jul 22 14:29 mo
drw-rw---- 1 admin admin 512 Jul 22 14:29 operational
drw-rw---- 1 admin admin 512 Jul 22 14:29 security-domains
drw-rw---- 1 admin admin 512 Jul 22 14:29 ssh-keys
-r--r----- 1 admin admin 493 Jul 22 14:29 summary
drw-rw---- 1 admin admin 512 Jul 22 14:29 user-certificates
```

GNU Bash シェルの理解

Bash（Bourne Again SHell）は、さまざまなオペレーティング システムでサポートされている UNIX シェル（コマンドライン インタプリタ）です。Bash インターフェイスを使用して、直接 APIC を設定したり、タスクを自動化する Bash シェル スクリプトを作成することができます。Bash はさまざまなコマンド ライン機能とスクリプト機能を備えています。

構文

Bash は sh 互換のコマンド言語インタプリタであり、標準入力やファイルから読み込んだコマンドを実行します。また、Bash には Korn および C（ksh および csh）シェルの有用な機能が組み込まれています。Bash は、最終的には、IEEE POSIX Shell and Tools 仕様（IEEE Working Group 1003.2）を忠実に実装することを目指しています。

Bash は、次のようなさまざまな機能をサポートしています。

- コマンドラインの編集
- サイズに制限がないコマンド履歴
- ジョブ制御
- シェル機能およびエイリアス
- サイズに制限がないインデックス配列
- 2 ～ 64 までのあらゆる数値を底とする整数演算

Bash シェルの詳細については、http://www.gnu.org/software/bash/bash.html を参照してください。

Bash 拡張機能

APIC には 次のような Bash シェルの拡張機能が含まれています。

ネットワークの命名規則

通常、ネットワーク オペレーティング システムでは、インターフェイス、ネットワーク アドレス、その他の設定を区切る文字としてフォワードスラッシュ（/）が使用されています。しかし、Bash シェルでは、ファイル名でのフォワード スラッシュの使用は制限されています。Bash ではエスケープ文字が用意されていますが、APIC ファイル システムでは区切り文字としてコロン（:）を使用することで、ネットワークの命名を簡易化しています。以下に例を上げて、この区切り文字の使用方法を説明します。

インターフェイス名

APIC Bash 拡張機能では、コロン（:）を使用してインターフェイス名を区切ります。たとえば、インターフェイス Ethernet 1/46 は **eth1:46** と記述されます。

次の例は、あるノードのインターフェイスの出力を示しています。

```
admin@apic1:physical-interfaces> pwd
/aci/fabric/inventory/fabric-pod-1/fabric-node-17/interfaces/physical-interfaces
admin@apic1:physical-interfaces> ls
eth1:1   eth1:17  eth1:24  eth1:31  eth1:39  eth1:46  eth1:53  eth1:60
eth1:10  eth1:18  eth1:25  eth1:32  eth1:4   eth1:47  eth1:54  eth1:7
eth1:11  eth1:19  eth1:26  eth1:33  eth1:40  eth1:48  eth1:55  eth1:8
eth1:12  eth1:2   eth1:27  eth1:34  eth1:41  eth1:49  eth1:56  eth1:9
eth1:13  eth1:20  eth1:28  eth1:35  eth1:42  eth1:5   eth1:57  summary
eth1:14  eth1:21  eth1:29  eth1:36  eth1:43  eth1:50  eth1:58
eth1:15  eth1:22  eth1:3   eth1:37  eth1:44  eth1:51  eth1:59
eth1:16  eth1:23  eth1:30  eth1:38  eth1:45  eth1:52  eth1:6
admin@apic1:physical-interfaces>
```

ネットワーク アドレスの命名

APIC Bash 拡張機能では、コロン（:）を使用してネットワーク アドレスを区切ります。たとえば、ネットワーク 192.168.1.0 とサブネット 255.255.255.0 は次のように記述されます。

```
192.168.1.0:255.255.255.0
```

コマンドの補完

APIC は、標準の Linux コマンドや「コマンド リファレンス」に記載されている APIC 固有のコマンドに対して、タブ補完機能をサポートしています。コマンドまたはオプションの一部を入力して最後に Tab キーを押すと、コマンドの完全形、またはその後に続く可能性があるキーワードや引数の選択肢が表示されます。

たとえば、次のように、Tab キーを使用して利用可能なディレクトリを表示することができます。

```
admin@apic1:aci> cd tenants/ <Tab>
common/ infra/  mgmt/
```

コマンド履歴

APIC CLI は Bash シェルの履歴機能をサポートしています。コマンド履歴を表示する方法として、**history** コマンド以外にも、上矢印または下矢印キーを使用することができます。

履歴を閲覧して適切なコマンドを再び呼び出し、Enter キーを押すと、履歴内のコマンドを再入力できます。また、コマンドを再度呼び出した後、実行する前にコマンドを変更することもできます。

また、Ctrl-r キーを押して、必要なコマンドが表示されるまでその一部を入力することで、以前のコマンドを直接検索できます。

その他のコマンド履歴機能など、Bash シェルの詳細については、次の URL を参照してください。
http://www.gnu.org/software/bash/bash.html

コマンドのヘルプ

CLI には、次の 2 つの形式のコンテキスト ヘルプがあります。

- インライン ヘルプ：いつでも Esc キーを 2 回押して、その時点のコマンド構文で使用可能なオプションを表示できます。プロンプトに何も入力しないで Esc キーを 2 回押すと、その時点のコマンドモードで使用可能なコマンドがすべて一覧表示されます。コマンドの一部を入力して Esc キーを 2 回押すと、コマンド構文のその時点の位置で使用可能なキーワードと引数がすべて一覧表示されます。
- man ページ：コマンドプロンプトで、**man** と入力し、その後ろにコマンドを入力するか、/aci の下にある管理対象オブジェクト（MO）のパスを入力すると、UNIX 形式の man ページを表示することができます。man ページは、すべてのコマンドやスコープに対して利用できるわけではありません。

マウント ポイント

APIC CLI には 3 つのマウント ポイント（aci、mit、debug）があります。 以下のセクションで、マウント ポイントについて詳しく説明します。

APIC にログインすると、マウント ポイント aci、debug、mit がデフォルト ディレクトリとして表示されます。

```
admin@apic1:~> ls
aci  debug  mit
```

（注） 各ファイル システムへのリンクは、各ユーザのホーム ディレクトリに置かれています。

以下のセクションで、マウント ポイントについて詳しく説明します。

aci マウント ポイント

aci ファイル システムは、MO とプロパティを対話型ユーザ セッション向けの簡易形式にまとめます。aci マウント ポイントは APIC のプライマリ CLI インターフェイスであり、大部分のユーザを対象としています。

mit マウント ポイント

上級ユーザは管理情報ツリー（MIT）ファイル システムを使用し、MIT 内の MO を直接表示して設定することができます。mit ファイル システムのディレクトリ構造は aci と同じですが、MO はネイティブ MIT オブジェクトとして表示されます。

たとえば、mit マウント ポイントでは、管理ユーザは次のように表示されます。

```
admin@apic1:user-admin> pwd
/mit/uni/userext/user-admin
admin@apic1:user-admin> ls -ltr
total 4
drw-rw---- 1 root  root  512 Jan 27 15:08 userdomain-all
drw-rw---- 1 root  root  512 Jan 27 15:08 userdata
-r--r----- 1 root  root  665 Jan 27 15:08 mo
drw-rw---- 1 admin admin 512 Jan 28 17:56 history
drw-rw---- 1 admin admin 512 Jan 28 17:56 faults
```

（注） mit マウント ポイントは、MO の設定を十分に理解している上級ユーザを対象としています。

debug マウント ポイント

debug マウント ポイントでは、複数の APIC デバイス、リーフ デバイス、スパイン デバイスにわたって設定を表示してデバッグできます。debug マウント ポイントは、上級ユーザによるトラブルシューティング用です。

ロールベース アクセス コントロール

ロールベース アクセス コントロール（RBAC）では、ユーザにロールを割り当てることで、デバイス操作のアクセスを制限できます。アクセスが必要なユーザだけにアクセスを許可するように、カスタマイズすることが可能です。

権限とセキュリティの適用

ロールベース アクセス コントロール（RBAC）を使用すると、一連の権限を備えたロールを作成してユーザに割り当て、ユーザの権限を制御することができます。RBAC では、直接的に権限を設定するのではなく、ロールを割り当てることでユーザに権限を適用できます。

APIC CLI 内では、管理対象オブジェクト（MO）など、管理情報ツリー（MIT）の特定部分を操作する権限をユーザに付与することができます。

次の例は、APIC CLI 内の RBAC 権限を表示する **ls** コマンドの使用方法を示しています。このコマンドの出力は、ファイル、UNIX のファイル読み取り/書き込み/実行権限、およびファイルが最後に変更された日時を表示します。

```
admin@apic1:user-admin> ls -al
total 4
drw-rw---- 1 admin admin 512 Jul 22 14:25 .
drw-rw---- 1 admin admin 512 Jul 22 14:25 ..
-rw-rw---- 1 admin admin 421 Jul 22 14:25 mo
-r--r----- 1 admin admin 608 Jul 22 14:25 summary
drw-rw---- 1 admin admin 512 Jul 22 14:25 userdata
drw-rw---- 1 admin admin 512 Jul 22 14:25 userdomain-all
```

ユーザ管理

デフォルトでは、各ユーザにホームディレクトリ（/home/<username>）が与えられます。このディレクトリで、ユーザはサブディレクトリとファイルを作成することができます。/home/<username> 内で作成されたファイルはデフォルトの umask パーミッションを継承し、本人と管理者（admin）のみがアクセスできます。

ユーザは初めてログインしたときに、ファイルを保存する *userid* ディレクトリ（/home/jsmith など）を作成することをお勧めします。作成すると、APIC は、その */userid* ディレクトリをユーザのホーム ディレクトリとして扱うようになります。

第 2 章

APIC CLI の使用

- オブジェクト モデル CLI へのアクセス，9 ページ
- 管理対象オブジェクトの表示，10 ページ
- 管理情報ツリーでの移動，10 ページ
- 設定の入力，11 ページ
- 設定ウィザードの使用，12 ページ
- 設定テンプレートの作成，14 ページ
- コマンドのカスタマイズ，16 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。

```
apic1# bash
admin@apic1:~>
```

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@apic1: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
- controller
- eventlog
- fabric-policies
- faults
- faults-history
- firmware
- health
- health-history
- import-export

- l4-l7-inventory
- l4-l7-packages
- local-user
- pod
- schedulers
- security-domains
- switch
- tenant
- 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, （21 ページ）

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、指定された名前のスコープ（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
--------------------------------------------------------------------------------

mocreate 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
mocreate 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
mocreate 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。

- Name
- Alias
- Description
- Tags
- 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
mocreate MyCompany
cd MyCompany
moset description 'My Company Network'
```

```
moset alias Home
moset monitoring-policy default
moconfig commit
cd networking
cd private-networks
mocreate 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
mocreate 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」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
--------------------------------------------------------------------------------
mocreate 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
mocreate 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
mocreate 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>
```

ウィザードの詳しい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 l4-l7-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 l4-l7-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 .yml 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。

第 3 章

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，22 ページ
- attach，23 ページ
- auditlog，23 ページ
- create，24 ページ
- controller，25 ページ
- diagnostics，25 ページ
- eraseconfig，26 ページ
- eventlog，27 ページ
- faults，28 ページ
- firmware，29 ページ
- health，31 ページ
- loglevel，32 ページ
- man，33 ページ
- mobrowser，33 ページ
- moconfig，34 ページ
- mocreate，35 ページ
- modelete，36 ページ
- mofind，36 ページ
- moprint，37 ページ
- moquery，38 ページ

- moset, 40 ページ
- mostats, 41 ページ
- password, 43 ページ
- reload, 43 ページ
- scope, 44 ページ
- show, 44 ページ
- svcping, 46 ページ
- techsupport, 46 ページ
- trafficmap, 48 ページ
- troubleshoot eptoep session（IP および MAC）, 49 ページ
- troubleshoot epext session EP-to-External-IP and External-IP-to-EP, 50 ページ
- troubleshoot eptoep session <session name>, 50 ページ
- troubleshoot eptoep session <session name> atomiccounter, 51 ページ
- troubleshoot eptoep session <session name> traceroute, 52 ページ
- troubleshoot eptoep session <session name> traceroute protocol, 52 ページ
- troubleshoot eptoep session <session name> traceroute protocol tcp dst port, 53 ページ
- show troubleshoot eptoep, 53 ページ
- show troubleshoot eptoep session <session name>, 54 ページ
- version, 54 ページ
- where, 55 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
--------------------------------------------------------------------------------
mocreate 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（.yml）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。

controller

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。
controller-id	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
```

diagnostics

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

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。

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。

eraseconfig [setup]

構文の説明

setup	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** コマンドの使用方法を示しています。

```
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

controller	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。
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。
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。
interface-name	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。
module-id	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
```

faults

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

controller	コントローラの健全性ログを表示します。
controller-id	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。
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。
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。
interface-name	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。
module-id	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。
port-number	ポート番号。
fault-code	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
```

firmware

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。
image-name	イメージ ファイルの名前。
list	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。

controller	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-       catalog  1.0            (0.566)        7461         2014-01-
1.0.0-566.bin
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-       catalog  1.0            (0.566)        7461         2014-01-
1.0.0-566.bin
28T11:17:36.054+00:00
ifabric-k9-simsw-1.0.0-   switch   1.0            (0.559)        854412177    2014-01-
559.bin

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-              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

controller	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。
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。
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。
interface-name	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。
module-id	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。
port_id	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。
set	ノードのサービス ログ レベルを設定します。
node	ノードを指定します。
node-name	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。
dme-name	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。 • ae • 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。
topic-name	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

scope	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。
running	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

context	MO のコンテキスト。
name	（任意）MO の名前。
property-name	（任意）MO のプロパティを指定します。
property-value	（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、下位レベルのスコープを削除する場合に使用します。

modelete *mo-name*

構文の説明

mo-name	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。
package	MO のパッケージ名。
class	MO のクラス名。
mo-value	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）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*| **--klass** *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。
host-id	APIC のホスト名または IP アドレス。
--port または **–p**	RESTインターフェイスのポートを指定します。
portname	REST インターフェイスのポート番号。
--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。
dn	MO の DN。
--klass または **–c**	クエリーのクラス名を指定します。
classname	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。
Property	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。
attributes	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。
output	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。
username	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。
options	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

property-name	プロパティ名。
property-value	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

stats-class	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。
interval	サンプリング間隔。次の値を選択できます。 • 5min • 15min • 1h • 1d • 1w • 1mo • 1qtr • 1year デフォルト値は 5 分です。
location	統計情報の表示対象の場所を指定します。
location-name	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。
counter-name	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。 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。
values	表示する特定の値を指定します。

values-name	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。
date-from	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。
date-to	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。
thresholded-flags	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。
output-name	出力のタイプ。次の値を選択できます。 • table • graph

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

controller	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。
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。

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

context	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（.yml）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 .yml ファイルを参照してください。

/home/*username*/scopedefs/ ディレクトリに .yml 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.yml 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（.yml）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。
fname	リモート接続先の名前。
host	リモート ホストの名前。
list	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。
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。
remote	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。
remotepath	リモート接続先のパス。
remoteport	リモート ポート番号。
password	リモート接続先のパスポート。
protocol	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。
svcname	サービス名
switch	スイッチの障害を表示します。
username	リモート接続先のユーザ名。

使用上のガイドライン

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：シャ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
```

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

```
admin@apic1:~> techsupport controllers status
Nodeid: 1
Collection Time: 2015-04-15T20:53:10.220+00:00
Status: success
Detailed status: Task completed
Config Location:
/data/techsupport/dbgexp_tsod-ts_exp_pol_apic1_sysid-1_2015-04-15T20-53UTC.tgz on APIC 1
Db Location: /data/techsupport/dbgexp_tsod-ts_exp_pol_apic1_sysid-1_2015-04-15T20-53UTC_db.tgz
 on APIC 1
Log Location:
/data/techsupport/dbgexp_tsod-ts_exp_pol_apic1_sysid-1_2015-04-15T20-53UTC_logs.tgz on APIC
 1
```

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。
source-node-id	送信元ノードの名前。
destnode	宛先ノードを指定します。
dest-node-id	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 eptoep 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 eptoep session** の作成方法を示します。

```
admin@apic1:/> 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>
```

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

```
admin@apic1:/> troubleshoot eptoep 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 epext 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 epext session** の作成方法を示します。

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

troubleshoot eptoep 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 eptoep 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。
description	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。
monitor	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。
report	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 eptoep session <session name>** コマンドの使用方法を示しています。

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

troubleshoot eptoep 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 eptoep 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 eptoep session <session name> atomiccounter** コマンドの使用方法を示しています。

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

troubleshoot eptoep 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 eptoep 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 eptoep session <session name> traceroute** コマンドの使用方法を示しています。

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

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

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

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

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

show troubleshoot eptoep

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 eptoep** 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 eptoep** コマンドの使用方法を示しています。

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

show troubleshoot eptoep 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 eptoep 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。
audit	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。
faults	障害を表示します。
monitor	モニタ ステータスを表示します。
reports	レポートを表示します。
statistics	統計情報を表示します。
topology	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 eptoep session <session name>** コマンドの使用方法を示しています。

```
admin@apic1:/> show troubleshoot eptoep 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。

構文の説明

version {**controller**| **switch**} [*node-id*]	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。

例

次の例は、**version** コマンドの使用方法を示しています。

```
admin@apic1:~> version switch 101
node type  node id  node name  version
---------  -------  ---------  ----------------
leaf       101      leaf1      simsw-1.0(0.450)


admin@apic1:~> version
node type   node id  node name  version
----------  -------  ---------  ----------------
controller  1        apic1      1.0(0.450)
controller  2        apic2      1.0(0.450)
controller  3        apic3      1.0(0.450)
leaf        101      leaf1      simsw-1.0(0.450)
leaf        102      leaf2      simsw-1.0(0.450)
leaf        103      leaf3      simsw-1.0(0.450)
spine       104      spine1     simsw-1.0(0.450)
spine       105      spine2     simsw-1.0(0.450)
```

where

スコープの管理情報ツリー（MIT）のディレクトリ パスを表示するには、**where** コマンドを使用します。

where *scope-name*

構文の説明

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

例

次の例は、**where** コマンドの使用方法を示しています。

```
admin@apic1:~> where aaa local-users admin
/aci/admin/aaa/security-management/local-users/admin
```

索 引

A

aci ファイル システム 5
attach コマンド 23

B

Bash 3
  Bash シェル 3
    GNU Bash シェル 3
Bash シェル 3

C

command コマンド 23
controller コマンド 25

D

debug ファイル システム 5

E

eraseconfig コマンド 26
eventlog コマンド 27, 49, 50, 51, 52, 53, 54

F

faults コマンド 28
file system 2

H

health コマンド 31

L

loglevel コマンド 32

M

man コマンド 33
man ページ 5
mit ファイル システム 5
mobrowser コマンド 33
moconfig コマンド 34
mocreate コマンド 35
mofind コマンド 36
moprint 37
moprint コマンド 37
moquery コマンド 38
moset コマンド 40
mostats コマンド 41

P

Python API 2

S

scope コマンド 44
show コマンド 44

T

trafficmap コマンド 48

V

version コマンド 54

W

where コマンド 55

こ

コマンド モード 4
    説明 4
コマンド リファレンス、CLI 9, 21
コマンドのヘルプ 5
コマンド履歴 5

し

シェル スクリプト 2

て

データ管理エンジン 2

ほ

ホーム ディレクトリ 7

ま

マウント ポイント 5