



## 概要

---

この章の内容は、次のとおりです。

- [Cisco UCS C シリーズ ラックマウント サーバの概要, 1 ページ](#)
- [CIMC CLI, 2 ページ](#)

## Cisco UCS C シリーズ ラックマウント サーバの概要

Cisco UCS C シリーズ ラック マウント サーバには次のモデルがあります。

- Cisco UCS C200 ラックマウント サーバ
- Cisco UCS C210 ラックマウント サーバ
- Cisco UCS C220 ラックマウント サーバ
- Cisco UCS C240 ラックマウント サーバ
- Cisco UCS C250 ラックマウント サーバ
- Cisco UCS C260 ラックマウント サーバ
- Cisco UCS C460 ラックマウント サーバ



(注)

どの Cisco UCS C シリーズ ラック マウント サーバがこのファームウェア リリースでサポートされているかを判断するには、関連する『*Release Notes*』を参照してください。C シリーズのリリース ノートは、[http://www.cisco.com/en/US/products/ps10739/prod\\_release\\_notes\\_list.html](http://www.cisco.com/en/US/products/ps10739/prod_release_notes_list.html) で入手できます。

---

# CIMC CLI

CIMC CLI は、Cisco UCS C シリーズ サーバのコマンドライン管理インターフェイスです。SSH または Telnet を使用し、ネットワークを介して CIMC CLI を起動し、サーバを管理できます。デフォルトでは、Telnet アクセスはディセーブルになります。

CLI のユーザ ロールは、**admin**、**user**（制御は可能、設定は不可）、および **read-only** のいずれかになります。



(注) **admin** パスワードが失われたために回復する必要がある場合には、ご使用のプラットフォームの Cisco UCS C シリーズ サーバ インストールおよびサービス ガイドを参照してください。

## コマンドモード

CLI のコマンドモードは階層構造になっており、EXEC モードがこの階層の最高レベルとなります。高いレベルのモードは、低いレベルのモードに分岐します。**scope** コマンドを使用すると、高いレベルのモードから 1 つ低いレベルのモードに移動し、**exit** コマンドを使用すると、モード階層内の 1 つ高いレベルに移動します。**top** コマンドを実行すると、EXEC モードに戻ります。



(注) ほとんどのコマンドモードは、管理対象オブジェクトに関連付けられています。**scope** コマンドを実行すると、管理対象オブジェクトは作成されず、管理対象オブジェクトがすでに存在するモードにアクセスできるだけです。

各モードには、そのモードで入力できるコマンドのセットが含まれています。各モードで使用できるほとんどのコマンドは、関連付けられた管理対象オブジェクトに関係しています。割り当てられているロールによっては、あるモードで使用できるコマンドのサブセットにしかアクセスできない場合があります。アクセスできないコマンドは非表示になります。

各モードの CLI プロンプトには、モード階層における現在のモードまでのフルパスが表示されます。これにより、コマンドモード階層での現在位置がわかりやすくなります。また、階層内を移動する必要がある場合には、非常に便利な機能です。

## コマンドモード表

次の表に、最初の 4 レベルのコマンドモード、各モードへのアクセスに使用するコマンド、および各モードに関連付けられている CLI プロンプトを示します。

モード名	アクセスするコマンド	モード プロンプト
EXEC	任意のモードから <b>top</b> コマンド	#
bios	EXEC モードから <b>scope bios</b> コマンド	/bios #
advanced	BIOS モードから <b>scope advanced</b> コマンド	/bios/advanced #
main	BIOS モードから <b>scope main</b> コマンド	/bios/main #
server-management	BIOS モードから <b>scope server-management</b> コマンド	/bios/server-management #
certificate	EXEC モードから <b>scope certificate</b> コマンド	/certificate #
chassis	EXEC モードから <b>scope chassis</b> コマンド	/chassis #
adapter	シャーシモードから <b>scope adapter index</b> コマンド	/chassis/adapter #
host-eth-if	アダプタモードから <b>scope host-eth-if</b> コマンド	/chassis/adapter/host-eth-if #
host-fc-if	アダプタモードから <b>scope host-fc-if</b> コマンド	/chassis/adapter/host-fc-if #
port-profiles	アダプタモードから <b>scope port-profiles</b> コマンド	/chassis/adapter/port-profiles #
vmfex	アダプタモードから <b>scope vmfex index</b> コマンド	/chassis/adapter/vmfex #
dimmsummary	シャーシモードから <b>scope dimm-summary index</b> コマンド	/chassis/dimm-summary #
flexflash		/chassis/flexflash #

モード名	アクセスするコマンド	モードプロンプト
	シャーシモードから <b>scope flexflash index</b> コマンド	
operational-profiles	flexflash モードから <b>scope operational-profile</b> コマンド	/chassis/flexflash/operational-profile #
storageadapter	シャーシモードから <b>scope storageadapter slot</b> コマンド	/chassis/storageadapter #
physical-drive	storageadapter モードから <b>scope physical-drive</b> コマンド	/chassis/storageadapter/physical-drive #
virtual-drive	storageadapter モードから <b>scope virtual-drive</b> コマンド	/chassis/storageadapter/virtual-drive #
cimc	EXEC モードから <b>scope cimc</b> コマンド	/cimc #
firmware	cimc モードから <b>scope firmware</b> コマンド	/cimc/firmware #
import-export	cimc モードから <b>scope import-export</b> コマンド	/cimc/import-export #
log	cimc モードから <b>scope log</b> コマンド	/cimc/log #
server	ログモードから <b>scope server index</b> コマンド	/cimc/log/server #
network	cimc モードから <b>scope network</b> コマンド	/cimc/network #
ipblocking	ネットワークモードから <b>scope ipblocking</b> コマンド	/cimc/network/ipblocking #
tech-support	cimc モードから <b>scope tech-support</b> コマンド	/cimc/tech-support #
fault	EXEC モードから <b>scope fault</b> コマンド	/fault #
pef	障害モードから <b>scope pef</b> コマンド	/fault/pef #

モード名	アクセスするコマンド	モード プロンプト
http	EXEC モードから <b>scope http</b> コマンド	/http #
ipmi	EXEC モードから <b>scope ipmi</b> コマンド	/ipmi #
kvm	EXEC モードから <b>scope kvm</b> コマンド	/kvm #
ldap	EXEC モードから <b>scope ldap</b> コマンド	/ldap #
role-group	ldap モードから <b>scope role-group</b> コマンド	/ldap/role-group #
power-cap	EXEC モードから <b>scope power-cap</b> コマンド	/power-cap #
sel	EXEC モードから <b>scope sel</b> コマンド	/sel #
sensor	EXEC モードから <b>scope sensor</b> コマンド	/sensor #
snmp	EXEC モードから <b>scope snmp</b> コマンド	/snmp #
trap-destination	snmp モードから <b>scope trap-destination</b> コマンド	/snmp/trap-destination #
v3users	snmp モードから <b>scope v3users</b> コマンド	/snmp/v3users #
sol	EXEC モードから <b>scope sol</b> コマンド	/sol #
ssh	EXEC モードから <b>scope ssh</b> コマンド	/ssh #
user	EXEC モードから <b>scope user</b> <i>user-number</i> コマンド	/user #
user-session		/user-session #

モード名	アクセスするコマンド	モードプロンプト
	EXEC モードから <b>scope user-session session-number</b> コマンド	
vmedia	EXEC モードから <b>scope vmedia</b> コマンド	/vmedia #
xmlapi	EXEC モードから <b>scope xmlapi</b> コマンド	/xmlapi #

## コマンドの実行

任意のモードで Tab キーを使用すると、コマンドを実行できます。コマンド名の一部を入力して Tab を押すと、コマンド全体が表示されるか、または別のキーワードを選択するか引数値を入力する必要があるところまで表示されます。

## コマンド履歴

CLI では、現在のセッションで使用したすべてのコマンドが保存されます。上矢印キーまたは下矢印キーを使用すると、これまでに使用したコマンドを1つずつ表示できます。上矢印キーを押すと履歴内の直前のコマンドが、下矢印キーを押すと履歴内の次のコマンドが表示されます。履歴の最後に到達すると、下矢印キーを押しても次のコマンドが表示されなくなります。

履歴内のすべてのコマンドは、履歴を1つずつ表示し、目的のコマンドを再度呼び出し、Enter を押すだけでもう一度実行することができます。このコマンドは手動で入力したように表示されず、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。

## 保留コマンドのコミット、廃棄、および表示

CLI でコンフィギュレーションコマンドを入力する場合、**commit** コマンドを入力するまで、そのコマンドは適用されません。コミットされるまで、コンフィギュレーションコマンドは保留状態となり、**discard** コマンドを入力して廃棄できます。保留中のコマンドについては、アスタリスク (\*) がコマンドプロンプトの前に表示されます。この例に示すように、**commit** コマンドを入力するとそのアスタリスクは消えます。

```
Server# scope chassis
Server /chassis # set locator-led off
Server /chassis *# commit
Server /chassis #
```

複数のコマンドモードで保留中の変更を積み重ね、**commit** コマンド1つでまとめて適用できます。任意のコマンドモードで **show configuration pending** コマンドを入力して、保留中のコマンドを表示できます。



- (注) 複数のコマンドをまとめてコミットするのは、アトミック操作ではありません。失敗したコマンドがあっても、成功したコマンドは適用されます。失敗したコマンドはエラーメッセージで報告されます。

## コマンド出力形式

ほとんどの CLI **show** コマンドでは、オプションの **detail** キーワードを指定でき、出力情報は表ではなくリスト形式で表示されます。**detail** キーワードを使用すると、出力情報を表示するための2つの表示形式のいずれかを設定できます。次の形式を選択できます。

- **Default** : 簡単に確認できるよう、コマンド出力はコンパクトリストで表示されます。

次に、デフォルト形式のコマンド出力例を示します。

```
Server /chassis # set cli output default
Server /chassis # show hdd detail
Name HDD_01_STATUS:
  Status : present
Name HDD_02_STATUS:
  Status : present
Name HDD_03_STATUS:
  Status : present
Name HDD_04_STATUS:
  Status : present

Server /chassis #
```

- **YAML** : スクリプトによる解析を簡単に行うため、コマンド出力は、定義された文字列で区切られた YAML (YAML Ain't Markup Language) データ シリアル化言語で表示されます。

次に、YAML 形式のコマンド出力例を示します。

```
Server /chassis # set cli output yaml
Server /chassis # show hdd detail
---
  name: HDD_01_STATUS
  hdd-status: present
---
  name: HDD_02_STATUS
  hdd-status: present
---
  name: HDD_03_STATUS
  hdd-status: present
---
  name: HDD_04_STATUS
  hdd-status: present
...

Server /chassis #
```

YAML の詳細については、<http://www.yaml.org/about.html> を参照してください。

ほとんどの CLI コマンドモードで、**set cli output default** を入力してデフォルト形式を設定するか、**set cli output yaml** を入力して YAML 形式を設定することができます。

## CLI に関するオンラインヘルプ

いつでも、? 文字を入力することで、コマンド構文の現在の状態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されていない状態で?を入力すると、そのときのモードで使用できるコマンドがすべて表示されます。コマンドの一部が入力されているときに?と入力すると、コマンド構文のそのときの位置で使用できるキーワードと引数がすべて表示されます。