

ServerView Suite

ServerView Virtual-IO Manager CLI V3.0

Command Line Interface

製品名称の表記

本書では、本文中の製品名称を、次のように略して表記します。

製品名称	本文中の表記	
Microsoft® Windows Server® 2008 Standard Microsoft® Windows Server® 2008 Enterprise Microsoft® Windows Server® 2008 Datacenter Microsoft® Windows Server® 2008 Foundation Microsoft® Windows® Small Business Server 2008 Standard Microsoft® Windows® Small Business Server 2008 Premium	Windows Server 2008	Windows
Microsoft® Windows Server® 2008 R2 Standard Microsoft® Windows Server® 2008 R2 Enterprise Microsoft® Windows Server® 2008 R2 Datacenter Microsoft® Windows Server® 2008 R2 Foundation Microsoft® Windows® Web Server 2008 R2	Windows Server 2008 R2	
Microsoft® Windows Server® 2003, Standard Edition Microsoft® Windows Server® 2003, Enterprise Edition Microsoft® Windows Server® 2003, Enterprise Edition for Itanium-based Systems Microsoft® Windows® Small Business Server 2003	Windows Server 2003	
Microsoft® Windows Server® 2003, Standard x64 Edition Microsoft® Windows Server® 2003, Enterprise x64 Edition	Windows Server 2003 x64	
Microsoft® Windows Server® 2003 R2 Standard Edition Microsoft® Windows Server® 2003 R2 Enterprise Edition Microsoft® Windows® Small Business Server 2003 R2 Microsoft® Windows® Storage Server 2003 R2, Standard Edition	Windows Server 2003 R2	
Microsoft® Windows Server® 2003 R2 Standard x64 Edition Microsoft® Windows Server® 2003 R2 Enterprise x64 Edition	Windows Server 2003 R2 x64 または Windows Server 2003 R2	

製品名称	本文中の表記	
Red Hat Enterprise Linux 5	Red Hat Linux	Linux
	RHEL5	
Red Hat Enterprise Linux AS (v.4)	RHEL4	
Red Hat Enterprise Linux ES (v.4)		
SUSE Linux Enterprise Server 11	SuSE Linux	
	SuSE Linux SLES11 または SLES11	
	SuSE Linux SLES10 または SLES10	
SUSE Linux Enterprise Server 10		
VMware ESX 4	ESX4	VMware
VMware ESX 3.5	ESX3.5	

著作権および商標

Copyright © 2012 Fujitsu Technology Solutions GmbH.

All rights reserved

Microsoft、Windows、Windows Server、Hyper-V は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標あるいは商標です。

Red Hat および Red Hat をベースとしたすべての商標とロゴは、米国およびその他の国における Red Hat, Inc. の商標または登録商標です。

BrightStor、ARCserve は、CA, Inc の登録商標です。

VMware、VMware ロゴ、VMware ESXi、VMware SMP および VMotion は VMware,Inc の米国およびその他の国における登録商標または商標です。

その他の各製品名は、各社の商標、または登録商標です。

その他の各製品は、各社の著作物です。

目次

1	はじめに	9
1.1	前版からの変更点	10
2	アーキテクチャ	11
3	セキュリティ	13
4	インストールとアンインストール	15
4.1	必要条件	15
4.2	インストール手順	16
4.2.1	Windows 環境でのインストール手順	16
4.2.2	Linux 環境でのインストール手順	23
4.3	最初の起動と構成	24
4.3.1	初めて viomcli を呼び出す	24
4.3.2	サーバの構成を確認する	26
4.3.3	ユーザプロファイルを作成する	27
4.4	VIOM CLI のアンインストール	28
4.4.1	Windows での VIOM CLI のアンインストール	28
4.4.2	Linux での VIOM CLI のアンインストール	28
5	コマンドとオプション	29
5.1	本書の表記について	29
5.2	コマンドの構文	29
5.2.1	データ型の種類	30
5.3	viomcli オプション	31
5.4	出力処理のコマンド	33
5.5	CLI オブジェクトとメソッド	36
5.5.1	cliconfig オブジェクト	37

5.5.1.1	cliconfig restore	37
5.5.1.2	cliconfig set	38
5.5.1.3	cliconfig show	40
5.5.2	cliuser オブジェクト	41
5.5.2.1	cliuser add	41
5.5.2.2	cliuser remove	42
5.6	VIOM オブジェクトとメソッド	43
5.6.1	api オブジェクト	46
5.6.1.1	api get	46
5.6.2	config オブジェクト	47
5.6.2.1	config delete	47
5.6.2.2	config save	48
5.6.2.3	configs show	49
5.6.3	iochannels オブジェクト	50
5.6.3.1	iochannels add	50
5.6.3.2	iochannel get	60
5.6.3.3	iochannel remove	64
5.6.3.4	iochannel set	66
5.6.3.5	iochannels show	74
5.6.4	network オブジェクト	75
5.6.4.1	network add	75
5.6.4.2	network get	76
5.6.4.3	network remove	77
5.6.4.4	network set	78
5.6.4.5	networks show	79
5.6.5	node オブジェクト	80
5.6.5.1	node getconfig	81
5.6.5.2	node manage	83
5.6.5.3	node restore	84
5.6.5.4	node setauth	85
5.6.5.5	node setconfig	86
5.6.5.6	nodes show	88
5.6.5.7	node unmanage	89
5.6.6	ports オブジェクト	90
5.6.6.1	ports add	90
5.6.6.2	ports remove	92
5.6.7	power オブジェクト	93
5.6.7.1	power get	93
5.6.7.2	power set	94
5.6.8	profile オブジェクト	96
5.6.8.1	profile add	96

5.6.8.2	profile assign	98
5.6.8.3	profiles export	99
5.6.8.4	profile failover	100
5.6.8.5	profile get	101
5.6.8.6	profiles import	102
5.6.8.7	profile remove	103
5.6.8.8	profile set	104
5.6.8.9	profiles show	106
5.6.8.10	profile unassign	107
5.6.9	servicelan オブジェクト	108
5.6.9.1	servicelan add	108
5.6.9.2	servicelan get	109
5.6.9.3	servicelan remove	110
5.6.9.4	servicelan set	111
5.6.10	servicevlan オブジェクト	112
5.6.10.1	servicevlan add	112
5.6.10.2	servicevlan get	113
5.6.10.3	servicevlan remove	114
5.6.10.4	servicevlan set	115
5.6.11	session オブジェクト	116
5.6.11.1	session close	116
5.6.12	uplinkset オブジェクト	117
5.6.12.1	uplinkset add	117
5.6.12.2	uplinkset get	119
5.6.12.3	uplinkset remove	120
5.6.12.4	uplinkset set	121
5.6.12.5	uplinksets show	123
5.6.13	vlannet オブジェクト	124
5.6.13.1	vlannet add	124
5.6.13.2	vlannet get	126
5.6.13.3	vlannet remove	127
5.6.13.4	vlannet set	128
5.6.13.5	vlannets show	129

6 エラーとリターンコード 131

6.1	エラーコード	131
6.1.1	メジャーコード 1000	132
6.1.2	メジャーコード 1100	133
6.1.3	メジャーコード 1200	133
6.1.4	メジャーコード 1300	133

6.1.5	メジャーコード 1400	135
6.2	リターンコード	136
7	追加情報	137
7.1	コンソールログとログファイル	137
7.2	INI ファイル	138
7.3	環境変数	139
図	141

1 はじめに

VIOM CLI は、VIOM で使用するコマンドライン・インターフェース (CLI) で、これにより VIOM 管理タスクをスクリプト化します。他の VIOM スイート製品が提供するのとは、高いレベルでの言語サポートとグラフィック管理であるのに対し、VIOM CLI が提供するのとは、Windows と Linux プラットフォームでスクリプト言語やタスクの自動化を行うための使いやすいインターフェースです。

VIOM CLI は、次の 2 つの部分から構成されています。

- 非常にサイズが小さく、プラットフォーム依存の **viomcli** バイナリ
- プラットフォームに依存しないように、JAVA サービスとして設計されている VIOM CLI サーバ

上記によりパフォーマンスと柔軟性が高まります。クライアントとサーバは通信にソケットと単純なテキストベースのプロトコルを使います。そのため、プラットフォーム依存部分なしでも通信可能で、telnet を使ったり、直接サーバへ接続したりすることができます。

以降の説明では、コマンドライン・インターフェースのみを扱い、VIOM の各種機能については扱っておりません。VIOM の詳細説明については、『VIOM ユーザーガイド』をご覧ください。

1.1 前版からの変更点

本版は、ServerView Virtual-IO Manager CLI V3.0 を対象としており、2011 年 8 月版のオンラインマニュアル『PRIMERGY ServerView Suite, ServerView Virtual-IO Manager CLI V2.6』の更新版です。

ServerView Virtual IO-Manager V3.0 では、特に次の機能が追加されています。

- PRIMERGY ラックサーバの管理。RX200 S7、RX300 S7、RX350 S7、TX300 S7 のサポート。
- BX920 S3 および BX924 S3 サーバブレードのサポート。
- BX920 S3 および BX924 S3 での PY CNA メザニンカード 10 Gb 2 ポート (MC-CNA112E) のサポート。1 つの物理ポートにつき 2 つの物理機能の制限も、機能タイプの制限もなくなりました。
- BX900 S2 シャーシのサポート。

2 アーキテクチャ

VIOM CLI の機能は、2つのアプリケーションに分かれています。

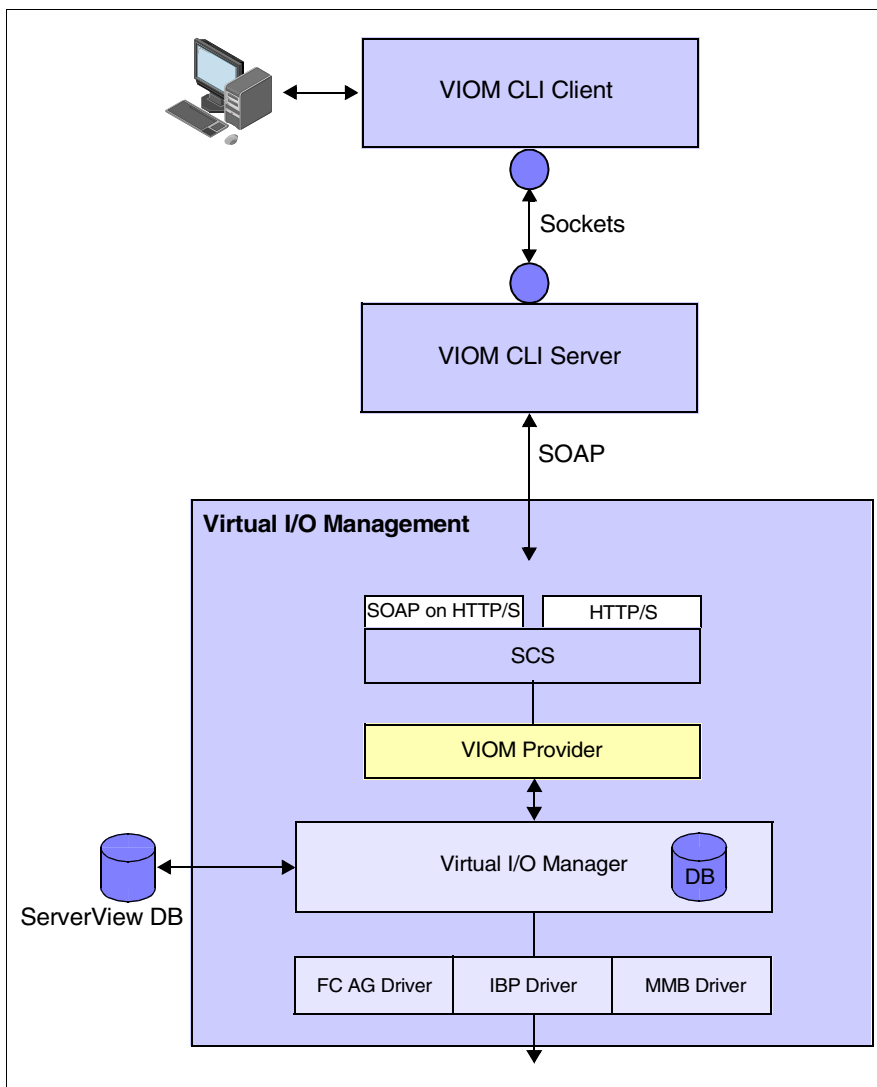


図 1: VIOM CLI アーキテクチャ

サイズが小さい **viomcli** バイナリはプラットフォーム依存で、ユーザまたはスクリプト言語のいずれかと、2 つ目の構成部分である **VIOM CLI** サーバとの間のインターフェースとして機能します。バイナリのサイズは小さく、その機能範囲も狭いので、読み込みが早く、ポータブル性も高くなります。バイナリの代わりに、単純な **telnet** や他のアプリケーションまたはスクリプトを使ってソケット通信を行うことも可能です。

VIOM CLI 機能の重要部分は **VIOM CLI** サーバと呼ばれ、プラットフォーム非依存の **Java** サービスとなるよう構築されています。サーバはソケットを通してクライアントと通信し、反対側にある **VIOM** とは、**SOAP over HTTP** を用いて通信します。サーバの役目は、認証とユーザプロファイルを管理し、入力情報の構文解析と検証を行い、**ServerView Connector Service (SCS)** を使って **SOAP** 通信を実行し、応答データとエラーメッセージを翻訳することです。

通常、情報の流れは次のようになります。

1. ユーザまたはスクリプトが **viomcli** を呼び出します。
2. **viomcli** がユーザの **INI** ファイルから設定を読み取ります。
3. **viomcli** が **VIOM CLI** サーバとの間のソケットを開き、すべてのコマンドラインパラメータをサーバへ渡します。
4. このコマンドラインを **VIOM CLI** サーバが構文解析します。
5. 認証情報が抽出され、ユーザプロファイルがロードされ、復号されます。
6. **SOAP** メッセージが作成されます。復号された認証情報が要求に挿入されます。
7. **SOAP** メッセージが **SCS** へ送信されます。
8. **SCS** から **SOAP** 応答を受信します。
9. リターンコードが評価されます。
10. そのデータとリターンコードが翻訳され、ソケットを介して **viomcli** へ送信されます。
11. **viomcli** があらかじめフォーマット設定されたテキストを出力します。
12. **viomcli** が抽出されたリターンコードを **OS** へ戻し、終了します。

3 セキュリティ

機密情報を保護するためのいくつかのメカニズムが実装されています。

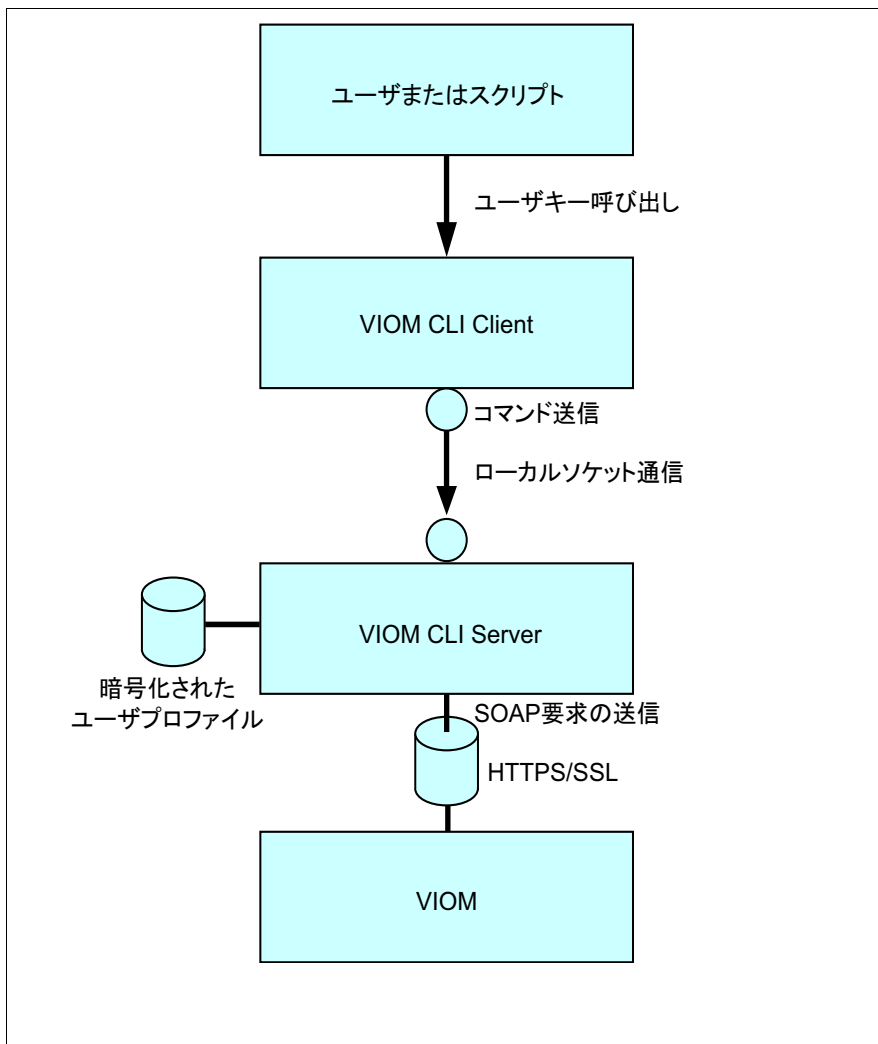


図 2: VIOM CLI セキュリティ概要

コマンドラインレベルでは、機密情報は必要とされません。すべてのセッションや認証データは **VIOM CLI** サーバにより処理され、暗号化されたユーザプロファイルストアに保管されます。各ユーザは自身の **SCS** 用認証データと自己定義キーを識別のためにサーバへ送信できます。後続のどの要求においても、このユーザキーのみが使用されます。そのため、スクリプトは、どのサーバやどのアクセスデータを使うかを意識することなく **VIOM** を使用することができます。

VIOM CLI のサーバ部分はシステムサービスとして動作し、接続を受け入れるのは、ローカルホストからのみです。そのため、サービスを制御するのは容易ですが、悪用するのは困難です。要求の間、サーバは提供されたユーザキーを使って該当ユーザのプロファイルを識別し、復号します。ユーザプロファイルはサーバプロセスのユーザコンテキストに保管されるので、呼び出しスクリプトから保護されます。

VIOM CLI サーバと **VIOM** の間の通信は、**SOAP** をベースにしており、セキュリティのために **HTTPS/SSL** を使用するよう設定できます。

4 インストールとアンインストール

VIOM CLI は、ServerView Suite DVD 1 でアドオンパッケージとして配布されます。Windows OS 用 MSI インストーラパッケージと Linux OS 用 RPM パッケージがご利用いただけます。

4.1 必要条件

VIOM CLI のインストールに必要な最低条件は、次のとおりです。

Windows

- OS
 - Windows XP
 - Windows Vista
 - Windows 7
 - Windows Server 2003 (32 / 64 bit)
 - Windows Server 2008 (32 / 64 bit)



日本では、Windows XP、Windows Vista および Windows 7 はサポートされません。

- Java Runtime 6

Linux

- OS
 - Red Hat Enterprise Linux 5 または 6
 - Suse Linux Enterprise Server 10 または 11



日本では、Suse Linux Enterprise Server はサポートされません。

- Java Runtime 6

4.2 インストール手順

次の 2 つのセクションでは、Windows および Linux OS 環境でのインストール手順を詳しく説明します。

管理サーバに旧バージョンの VIOM CLI がすでにインストールされている場合は、アップデートインストールが自動的に実行されます。

4.2.1 Windows 環境でのインストール手順

- ▶ Windows で管理者としてログインします。
- ▶ ServerView Suite DVD 1 を DVD-ROM ドライブにセットします。DVD が自動的に起動しない場合は、DVD-ROM のルートディレクトリにある **setup.exe** ファイルをクリックします。
- ▶ オプション「ServerView Software Product Selection」を選択します。
- ▶ 「Start」をクリックします。
- ▶ 次のウィンドウのメニューバーで必要な言語を選択してください。
- ▶ 「ServerView」→「Virtual-IO Manager」の順に選択します。

- ▶ ファイル `viomcli.exe` をダブルクリックし、インストールを開始します。



図 3: VIOM CLI インストールウィザードー開始ページ

VIOM CLI インストールウィザードが開きます。

- ▶ 「次へ」をクリックします。



図 4: VIOM CLI インストールウィザードー使用許諾契約書

該当項目を選択し、使用許諾契約に合意します。

- ▶ 「次へ」をクリックします。

図 5: VIOM CLI インストールウィザードーユーザ情報

お客様の名前、所属する会社名または組織名を入力します。また、この設定が現在のユーザのみに適用されるか、それともこのシステムで作業をするすべてのユーザに適用されるかを指定します。該当する項目を選択します。

4 インストールとアンインストール

- ▶ 「次へ」をクリックします。

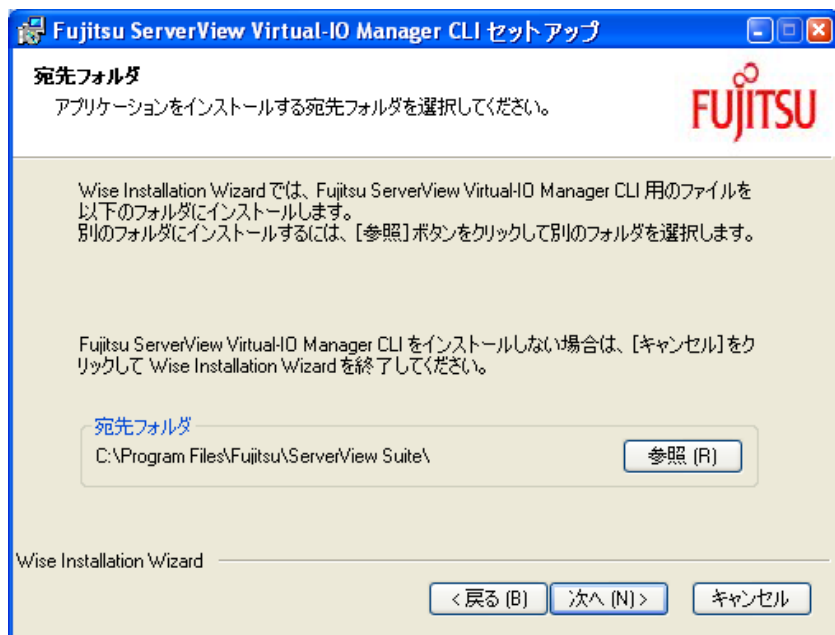


図 6: VIOM CLI インストールウィザードーインストール先フォルダ

VIOM CLI をインストールするフォルダを指定します。「参照」ボタンを使ってデフォルト設定を変更できます。

- ▶ 「次へ」をクリックします。



図 7: VIOM CLI インストールウィザードーインストールを開始する

設定を変更するには「戻る」をクリックし、インストールを開始するには「次へ」をクリックします。

4 インストールとアンインストール

- ▶ 「次へ」をクリックします。

インストールが完了すると、次の画面が表示されます。



図 8: VIOM CLI インストールウィザードーインストール完了

- ▶ 「終了」をクリックして、インストールを終了します。

4.2.2 Linux 環境でのインストール手順

Linux でルートユーザとしてシェルを開き、次の手順に従います。

- ▶ 配布メディア上の Linux ディレクトリへ切り替えます。
- ▶ 次のコマンドでインストールを開始します。

```
rpm -iv viomcli-<version>.i386.rpm
```

<version> については、VIOM CLI のバージョンとリリース番号を指定します（例：3.0-02）。

- ▶ インストールが完了したら、シェルから次のコマンドを実行し、VIOM CLI サーバを起動することができます。

```
/etc/init.d/viomclisrv start
```



ランレベル 3 と 5 での再起動後、VIOM CLI サーバの起動は自動的に行われます。

4.3 最初の起動と構成

新規にインストールした **VIOM CLI** を使用する前に、その構成を確認し、必要に応じて調整を行います。この章では、主な構築タスクの概略を説明します。**viomcli** コマンドについては、「[5 コマンドとオプション](#)」の章 (29 ページ) で説明します。

4.3.1 初めて **viomcli** を呼び出す

システムでコマンドプロンプトまたはシェルを開き、次のコマンドの実行を試みます。

```
viomcli cliconfig show
```

例

下記と類似した情報が表示されます。

```
ERROR: No INI file found,   code 1
server port                51000
charset                    windows-1252
language                   en
max. client connections    10
API pool size              10
console log level          IMPORTANT
file log level             DEBUG_EXTREME
log file                   C:\Documents and
Settings\viom\viomclisrv.log
```

```
0 0 OK
```

初めて **VIOM CLI** を呼び出したときは、一番上の行のエラーメッセージ「**No INI file found**」は関係ありません。**INI** ファイルは、自動的に作成されます。

このメッセージを再度受け取った場合は、次の原因が考えられます。

- **INI** ファイルが削除されました。
- 関連する権限をもっていません。
- **INI** ファイルを作成するのに十分なディスク容量がありません。

通常、**INI** ファイルはホームフォルダ内の **viomcli** サブディレクトリに作成されます。

VIOM CLI サーバへの接続が確立できなかったというエラーメッセージを受け取る場合もあります。これは、VIOM CLI サーバがまだ起動中であるために起こる場合もあります。サーバの起動は、実際の構成にもよりますが、30 秒から 1 分ほどかかります。

Windows 環境

- ▶ **Services Manager** ウィンドウを開き、**VIOM CLI Server** サービスが起動したか確認します。起動していない場合は、サービスを起動します。

Linux 環境

- ▶ 次のコマンドを入力します。

```
ps aux| grep -i viomcli
```

- ▶ このテキストを含むプロセスを見つけられない場合には、次のように呼び出して、VIOM CLI サーバを起動します。

```
/etc/init.d/viomclisrv start
```

- ▶ VIOM CLI サーバの起動に問題が残る場合は、サービスを実行しているユーザアカウント権限を確認し、エラーがないか、すべてのログファイルをチェックします。また、別のアプリケーションがサーバのデフォルトポート (**51000**) への接続をブロックしている可能性もあります。その場合、このアプリケーションを一時的に無効にし、[26 ページの「サーバの構成を確認する」の項](#) をご覧になり、サーバの設定を変更します。

4.3.2 サーバの構成を確認する

viomcli cliconfig show コマンドを実行すると、サーバの常時設定がすべて一覧で提供されます。

例

```
server port          51000
charset              windows-1252
language             en
max. client connections 10
API pool size        10
console log level     IMPORTANT
file log level        DEBUG_EXTREME
log file              C:\Documents and Settings\viom\viomclisrv.log
```

ここで重要なのは、「**server port**」と「**charset**」の値です。これらの値を確認し、必要があれば調整します。

すべての設定は、**viomcli cliconfig set** コマンドを使って変更可能です。

例

```
viomcli cliconfig set --port=51000 --maxclients=5
--poolsize=3 --levelcon=not levelfile=dbg
--logfile=c:\logfile.txt --charset=UTF-8 --lang=de
```

使用可能なオプションについては、[38 ページ](#)の「**cliconfig set**」の項で説明します。

Windows OS の場合、**charset** オプションをお使いのキーボードのコードページに設定します（例：日本の場合、**--charset=Windows-932**）。**cmd.exe** で各国の文字を使用するには、シェルで **chcp <codepage>** を実行し、TrueType フォントをコマンドプロンプトシステムメニューで選択します（「プロパティ」→「フォント」）。

Linux OS の場合、通常、**charset** オプションを「**UTF-8**」に設定すれば十分です。文字セットが正しく検出されない場合には、**LC_CTYPE** 変数が設定されていないか、または不適切に設定されている可能性があります。この場合、**LC_CTYPE** 変数（例：**LC_CTYPE=ja_JP.UTF-8**）を設定するか、または **cliconfig set** コマンドで正しい文字セットを設定してください。

選択したポート番号は、**49152** から **65535** までの範囲内にある必要がありますが、該当マシンの他のどのサービスも妨げないよう注意します。サーバポートを変更した場合には、サービスを再起動させ、設定変更を有効にする必要があります。クライアント **INI** ファイルの「**port**」の値も変更する必要があります。

4.3.3 ユーザプロフィールを作成する

viomcli を SCS や VIOM と関係して使用するには、*viomcli* ユーザアカウントを指定する必要があります。それには *viomcli* ユーザプロフィールの作成が必要です。そのためには、次のコマンドを実行します。

例

```
viomcli --user=administrator --password=admin --key=123
--server=123.45.78.12:3172 cliuser add
```

このコマンドで、パスワードが「**admin**」の「**administrator**」というユーザのプロファイルが作成されます。保存先サーバは **123.45.78.12** でポート **3172** を使用します。識別キーは **123** になります。これで、以下のように *viomcli* コマンドを実行できます。

```
viomcli --key=123 api get
```

または、短縮形を使って

```
viomcli -k 123 api get
```

このコマンドにより、短い **API** 情報要約を画面表示します。

Subject	Version	Build	Java
API	3.0.0	2012-02-06 17:42:02	
Manager	3.0.0	2012-02-06 17:42:07	1.6.0_30-b12

0 0 OK

--key または *-k* パラメータの値は、プロフィール作成中に割り当てた識別キーです。

4.4 VIOM CLI のアンインストール

VIOM CLI をアンインストールする前に、すべてのサーバの管理を解除します。VIOM CLI をアンインストールすると、すべての情報は失われます。そのため、後で使用する場合はサーバプロファイルを保存してください。

4.4.1 Windows での VIOM CLI のアンインストール

Windows の「スタート」メニューから VIOM CLI をアンインストールします。

– On Windows Server 2003 の場合

「スタート」→「設定」「コントロールパネル」→「プログラムの追加と削除」の順に選択します。

– Windows Server 2008 の場合

「スタート」→「コントロールパネル」→「プログラムと機能」の順に選択します。

製品のバージョンを表示するには、「表示」メニューから「詳細の選択」を選択します。「Version」オプションを選択して「OK」をクリックします。

4.4.2 Linux での VIOM CLI のアンインストール

VIOM CLI をアンインストールするには rpm パッケージマネージャを使用します。次のコマンドを使用して VIOM CLI をアンインストールします。

```
rpm -e ServerViewViomcli-<version>i386
```

<version> には、CLI のバージョン（3.0-02 など）を指定します。

5 コマンドとオプション

この章では、すべての *viomcli* コマンドとオプションの概略を述べます。

5.1 本書の表記について

斜体	コマンド、ファイル名、およびパス名は <i>斜体</i> で表記されています。
固定フォント	システム出力は、固定フォントで表記されています。
太字の固定フォント	キーボードから入力する必要があるコマンドは、太字の固定フォントで表記されています。
<abc>	山カッコは、実数値に置き換えられる変数を囲っています。
[パラメータ]	大カッコは、オプション（任意指定）パラメータとオプションを示すために使用されます。
[<value1> [,<value2>][,...]]	可変長のコンマ区切りのリストを示します。

表 1: 本書の表記

5.2 コマンドの構文

viomcli の一般的なコマンドの構文は、次のようになります。

構文

```
viomcli [options] [object method [parameters]]
```

viomcli バイナリを実行コマンド名のオプションを使って呼び出すことができます。オプションは実行コマンド名のすぐ後に続きます。これらのオプションについては、[31 ページの「viomcli オプション」](#)の項で説明します。

オプションに続き、オブジェクトと関連メソッドがある場合もあります。オブジェクトを指定したら、メソッドも必要になります。メソッドに応じて、0 から n までのパラメータがきます。オブジェクトとメソッドについては、[36 ページの「CLI オブジェクトとメソッド」](#)の項と [43 ページの「VIOM オブジェクトとメソッド」](#)の項で説明します。

5.2.1 データ型の種類

次のデータ型が本書で使用されています。

型	内容	使用法
string	文字列は、英数字で構成されます。 スペースを含む文字列は、二重引用符で囲みます。	-u admin --user=admin --comment="this is a comment"
integer	1 つの整数	-p 1 --port=1
boolean	1 つのブール値 (TRUE / FALSE)	-bootuse は FALSE を意味します。 +bootuse は TRUE を意味します。 -d は TRUE を意味します (bootuse の短縮形)。 この短縮形は、常に TRUE を意味します。
switch	切り替え。設定しなければ、デフォルトで FALSE になります。	-h --help
list	コンマで区切られたプロパティのリスト。各プロパティ値は、定義された順番で入力する必要があります。	--s fc,dc,1,2 --spec=fc,dc,1,2

表 2: データ型の種類

5.3 viomcli オプション

viomcli バイナリをいくつかのオプションを使って呼び出すことができます。こうしたオプションをここで説明します。

構文

```
viomcli[ --server=<string>][ --user=<string>]
[ --password=<string>] --key=<string>[ --help][ --version]
[ --output=<string>][ --clean][ --names][ --interactive]
[ --telnet][ --longtext][ --system]
```

パラメータ

これらのパラメータはすべて任意で指定できますが、**key** パラメータだけは例外です。**key** パラメータは、どの VIOM オブジェクトのメソッドにも必須です。

--server=<string> または短縮形 **-s <string>**

任意指定

VIOM サーバ名または IP のいずれかと、ポートをコロンで区切って指定します。次の例のようになります。

server:port

--user=<string> または短縮形 **-u <string>**

任意指定

ユーザ名

--user=<string> または短縮形 **-u <string>**

ユーザ名です。『*ServerView* でのユーザ管理』マニュアルで説明されている「Administrator」ロールを持つユーザ。

このバージョンの Virtual-IO Manager は、「Administrator」ロールを持つユーザしか起動できません。

--password=<string> または短縮形 **-p <string>**

任意指定

パスワード

--key=<string> または短縮形 **-k <string>**

後続の要求でユーザを識別するキー



- **-k** または **--key** オプションの指定は、すべての VIOM オブジェクトとメソッドで常に必要です。使用するユーザプロフィールを選択するのに必要になります。

- **telnet** モードや対話モードの場合、ユーザキーを 1 回入力すれば十分です。同じキーがそのセッションのステートメントのすべてに使用されます。

--help または短縮形 **-h**

任意指定

ヘルプを画面表示します。

--version または短縮形 **-v**

任意指定

バージョン情報を画面表示します。

--output=<string> または短縮形 **-o <string>**

任意指定

このメソッドの出力モードを変更します (*txt*, *csv*, *xml*)。このパラメータの指定は、定様式出力 (*txt*) とプログラムの構文解析可能な *CSV* または *XML* 出力とを切り替える場合に行います。*CSV* フォーマットと *XML* フォーマットは、ほとんどのリスト表示に対応します。

--clean または短縮形 **-c**

任意指定

指定すると、メッセージとエラーは画面表示されません。ヘッダーと空白は抑止されます。そのためプログラムやスクリプトにより出力処理は容易に行われます。

--names または短縮形 **-n**

任意指定

列名を表示して、ラベルを表示しません。

--interactive または短縮形 **-i**

任意指定

対話モードをアクティブにします。

--telnet または短縮形 **-t**

任意指定

telnet モードをアクティブにします。

--longtext または短縮形 **-l**

任意指定

長いテキストメッセージを表示します。

--system または短縮形 **-y**

任意指定

システムメッセージを表示します。

5.4 出力処理のコマンド

filter コマンドを使用すると、Java の正規表現を使って結果の 1 つまたはすべての列を検索したり、出力する列を 1 つまたは複数選択することができます。

構文

```
viomcli --key=<string> filter
[ --column=<integer1>[,<integer2>[,...]]]
[ --pattern=<string>][ --search=<integer>]
[ object method[parameter]]
```

パラメータ

--key=<key> または短縮形 **-k <key>**

要求でユーザを識別するキー

--column=<integer1>[,<integer2>[,...]] または短縮形 **-c <integer1>[,<integer2>[,...]]**

任意指定

結果で出力される 1 つまたは複数の列のリスト。複数の列をそれぞれコンマで区切って指定します。

--pattern= または短縮形 **-p <string>**

任意指定

正規表現パターン

--search=<integer> または短縮形 **-s <integer>**

任意指定

検索先の列

例 1

```
viomcli.exe -k 4321 filter -c 1,2,6 -p "BLADE_S.*" -s 5 nodes show
```

```
122 BX600-4           Managed
58  BX600-3           Unmanageable
26  BX600S3-2         Manageable
74  BX600S3-1         Unmanageable
42  BX600-2           Unmanageable
```

```
0 0 OK
```

この例では、*nodes show* メソッドが実行され、列 5 に適用される正規表現 *BLADE_S.** によって出力のフィルター処理が行われます。一致している行の列 1、2、6 のみが出力されます。

フィルターで、列番号の代わりに列名を指定することができます。この場合、*viomcli* コマンドを呼び出すのに **--names** または **-n** オプションを使用する必要があります。

構文

```
viomcli --key=<string> --names filter  
[ --column=<string1>[,<string2>[,...]]]  
[ --pattern=<string>][ --search=<integer>]  
[ object method[parameter]]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--names=<key> または短縮形 **-n <key>**
任意指定
コラム名を表示し、ラベルを表示しません。

--column=<string1>[,<string2>[,...]] または短縮形 **-c <string1>[,<string2>[,...]]**
任意指定
結果で出力される 1 つまたは複数の列のリスト。複数の列が、各列の
コンマ区切りのリストで指定されます。

--pattern=<string> または短縮形 **-p <string>**
任意指定
正規表現パターン

--search=<integer> または短縮形 **-s <integer>**
任意指定
検索先の列

次の例では、列名を使ったフィルターを示しています。

例2

```
viomcli -k 4321 -n filter -s SubType -p ".*SWITCH.*"
-c Id,Name,SubType,ManageStatus nodes show
```

Id	Name	SubType	ManageStatus
86	BX600S3-1\R01	SWITCH_SLOT	Manageable
119	BX600S3-1\S01	LAN_SWITCH	Unmanageable
140	BX600-4\S02	LAN_SWITCH	Managed
166	BX600S3-2\S03	FC_SWITCH	Manageable
38	BX600S3-2\R01	SWITCH_SLOT	Manageable
121	BX600S3-1\S03	FC_SWITCH	Unmanageable
57	BX600-2\R04	SWITCH_SLOT	Manageable

```
0 0 OK
SYCSC
```

空になっていない、特定の 1 列に対応する行のみを出力するには、「`.*`」のパターンを使います。つまり、その列の始まりから終わりまでに 1 文字もないと出力されません。次の例では、列名を使ったフィルターを示しています。

例3

```
viomcli -k 333 filter -s Java -p ".*" api get
Subject Version Build Java
Manager 3.0.0 2012-02-06 17:42:07 1.6.0_30-b12
```

```
0 0 OK
```

PRIMERGY ラックサーバを管理するときに、「**Manager**」列の IP アドレスを `node setauth` コマンドのトラップ送信先として使用します。

5.5 CLI オブジェクトとメソッド

コマンドライン・インターフェース（CLI）自体がサポートするオブジェクトと関連メソッドの数が限られています。下記テーブルは、CLI 自体がサポートするオブジェクトと関連メソッドのリストです。これらのオブジェクトとメソッドが説明されているセクションも示されています。

オブジェクト	メソッド	セクション
cliconfig	restore	37 ページの「cliconfig restore」の項
	set	38 ページの「cliconfig set」の項
	show	40 ページの「cliconfig show」の項
cliuser	add	41 ページの「cliuser add」の項
	remove	42 ページの「cliuser remove」の項

表 3: CLI オブジェクトとメソッド

5.5.1 cliconfig オブジェクト

cliconfig オブジェクトが提供するの、CLI の構成設定の保守を行うメソッドです。*cliconfig* メソッドでは、識別キーは不要です。

5.5.1.1 cliconfig restore

構成のデフォルト設定値に戻します。

構文

```
viomcli cliconfig restore
```

cliconfig restore メソッドには、パラメータがありません。

デフォルト設定値

構成のデフォルト設定値は、下記のとおりです。

パラメータ	デフォルト
server port	51 000
char set	OS のデフォルトの文字セット
language	en (英語)
max. client connections	10
API pool size	10
console log level	NOTHING
file log level	IMPORTANT
log file	<userhome>/viomclisrv.log <userhome> : Windows では環境変数 HOMEPATH、LINUX では環境変数 Home

表 4: デフォルト設定値

5.5.1.2 cliconfig set

構成値を設定します。

構文

```
viomcli cliconfig set[ --port=<integer>][ --maxclients=  
<integer>][ --poolsize=<integer>][ --levelcon=<string>]  
[ --levelfile=<string>][ --logfile=<string>]  
[ --charset=string][ --lang=<string>]
```

パラメータ

--port=<integer> または短縮形 **-p <integer>**

任意指定

サーバポート（サーバが再起動した後で、アクティブになります）

可能な値：49152 - 65535

デフォルト：51000

--maxclients=<integer> または短縮形 **-m <integer>**

任意指定

接続可能な最大クライアント数

可能な値：1 - n

デフォルト：10



たくさんのクライアントが接続すると、サーバのスピードが落ち、メモリの使用量が多くなりますので注意してください。

--poolsize=<integer> または短縮形 **-s <integer>**

任意指定

オブジェクトプールのサイズ

デフォルト：10



プールサイズが大きいと起動時間が増え、サーバのスピードが落ち、メモリの使用量が多くなりますので、注意してください。

--levelcon=<string> または短縮形 **-c <string>**

任意指定

コンソールで画面表示されるメッセージのログレベルを指定します。

not

コンソールでの出力を不可にする（デフォルト）。

sev

重大メッセージ

imp
重要メッセージ

inf
通知メッセージ

dgb
デバッグメッセージ

ext
拡張デバッグメッセージ

--levelfile=<string> または短縮形 **-f <string>**

任意指定
ログファイルへ出力されるメッセージのログレベルを指定します。

not
ファイルへの出力を不可にする

sev
重大メッセージ

imp
重要メッセージ (デフォルト)

inf
通知メッセージ

dgb
デバッグメッセージ

ext
拡張デバッグメッセージ

--logfile=<string> または短縮形 **-l <string>**

任意指定
ログファイルのパスと名前 (サーバの再起動後、アクティブになります)
デフォルト : <userhome>/viomclisrv.log
(<userhome> : Windows では環境変数 *HOMEPATH*、LINUX では環境変数 *Home*)

--charset=<string> または短縮形 **-r <string>**

任意指定
クライアント文字セット。完全 16 ビット以外、Java プラットフォームの文字セットをすべてサポートしています。
デフォルト : OS のデフォルトの文字セット

--lang=<string> または短縮形 **-g <string>**

任意指定
言語を指定します。

de

ドイツ語

en

英語（デフォルト）

ja

日本語

例

```
viomcli config set --port=51000 --maxclients=5
--poolsize=3 --levelcon=not --levelfile=dbg
--logfile=c:\logfile.txt --charset=UTF-8 --lang=de
```

5.5.1.3 cliconfig show

CLI の構成設定を画面表示します。

構文

```
viomcli cliconfig show
```

cliconfig show メソッドには、パラメータがありません。

例

```
viomcli cliconfig show

server port          51000
charset              windows-1252
language             en
max. client connections 10
API pool size        10
console log level     IMPORTANT
file log level        DEBUG_EXTREME
log file              C:\Documents and Settings\viom\viomclisrv.log

0 0 OK
```


5.5.2 cliuser オブジェクト

cliuser オブジェクトが提供するののは、ユーザとユーザプロファイルの保守を行うメソッドです。

5.5.2.1 cliuser add

ユーザプロファイルを作成します。このメソッドのパラメータ値を *viomcli* オプションから読み取ります。

構文

```
viomcli --server=<server> --user=<user> --password=<password>  
--key=<key> cliuser add
```

パラメータ

--server=<server> または短縮形 **-s <server>**

VIOM サーバ名または IP のいずれかと、ポートをコロンで区切って指定します。

例

この構文は、通常の URL 構文に従い、下記のようになります。

--server=<url>

SCS サーバの URL、具体例 (IPv6 または IPv4) :

- http://[::FFFF:111.222.3.4]:3172
- http://111.222.3.4:3172
- https://111.222.3.4:3172 (SSL 接続の場合)

接頭部の「http://」は省略可能です。

--user=<user> または短縮形 **-u <user>**

ユーザ名

--user=<user> または短縮形 **-u <user>**

ユーザ名です。『*ServerView* のユーザ管理』マニュアルで説明されている AccessVIOM 権限を持つユーザ。

デフォルトでは、Administrator ロールを持つユーザにはこの権限があります。

--password=<password> または短縮形 **-p <password>**

パスワード

--key=<key> または短縮形 **-k <key>**

後続の要求でユーザを識別するキー

ここで設定したキーは、識別のために使用されます。後続の要求では、どれでも、このキーを指定する必要があります。

add メソッドにはこれ以上パラメータはありません。

例

```
viomcli -s 111.222.3.4:3172 -u administrator -p admin -k mykey cliuser add
```

この例では、111.222.3.4 が、VIOM マネージャが動作する管理サーバです。3172 が、ServerView Connector Service (SCS) のポートです。パスワードが「admin」のユーザ「administrator」は、以降のリクエストで「mykey」キーを使用します。

5.5.2.2 cliuser remove

ユーザプロファイルを削除します。このメソッドのパラメータ値を **viomcli** オプションから読み取ります。構文は次のようになります。

構文

```
viomcli --key=<key> cliuser remove
```

パラメータ

--key=<key> または短縮形 **-k <key>**

要求でユーザを識別するキー

remove メソッドには、これ以上パラメータがありません。

5.6 VIOM オブジェクトとメソッド

このセクションでは、VIOM で使用するオブジェクトと関連メソッドについて説明します。これらのメソッドを使用するには、VIOM ユーザーアカウントと `viomcli` ユーザープロファイルが必要になります。VIOM のオブジェクトと関連メソッド、それらが説明されているセクションを次の表に示します。

オブジェクト	メソッド	セクション
api	get	46 ページの「api get」の項
config	delete save	47 ページの「config delete」の項 48 ページの「config save」の項
configs	show	49 ページの「configs show」の項
iochannel	get remove set	60 ページの「iochannel get」の項 64 ページの「iochannel remove」の項 66 ページの「iochannel set」の項
iochannels	add show	50 ページの「iochannels add」の項 74 ページの「iochannels show」の項
network	add get remove set	75 ページの「network add」の項 76 ページの「network get」の項 77 ページの「network remove」の項 78 ページの「network set」の項
networks	show	79 ページの「networks show」の項
node	getconfig manage restore setauth setconfig unmanage	81 ページの「node getconfig」の項 83 ページの「node manage」の項 84 ページの「node restore」の項 85 ページの「node setauth」の項 86 ページの「node setconfig」の項 89 ページの「node unmanage」の項
nodes	show	88 ページの「nodes show」の項
ports	add remove	90 ページの「ports add」の項 92 ページの「ports remove」の項
power	get set	93 ページの「power get」の項 94 ページの「power set」の項

表 5: VIOM オブジェクトとメソッド

オブジェクト	メソッド	セクション
profile	add assign failover get remove set unassign	96 ページの「 profile add 」の項 98 ページの「 profile assign 」の項 100 ページの「 profile failover 」の項 101 ページの「 profile get 」の項 103 ページの「 profile remove 」の項 104 ページの「 profile set 」の項 107 ページの「 profile unassign 」の項
profiles	export import show	99 ページの「 profiles export 」の項 102 ページの「 profiles import 」の項 106 ページの「 profiles show 」の項
servicelan	add get remove set	108 ページの「 servicelan add 」の項 109 ページの「 servicelan get 」の項 110 ページの「 servicelan remove 」の項 111 ページの「 servicelan set 」の項
servicevlan	add get remove set	112 ページの「 servicevlan add 」の項 113 ページの「 servicevlan get 」の項 114 ページの「 servicevlan remove 」の項 115 ページの「 servicevlan set 」の項
session	close	116 ページの「 session close 」の項
uplinkset	add get remove set	117 ページの「 uplinkset add 」の項 119 ページの「 uplinkset get 」の項 120 ページの「 uplinkset remove 」の項 121 ページの「 uplinkset set 」の項
uplinksets	show	123 ページの「 uplinksets show 」の項
vlnnet	add get remove set	124 ページの「 vlnnet add 」の項 126 ページの「 vlnnet get 」の項 127 ページの「 vlnnet remove 」の項 128 ページの「 vlnnet set 」の項
vlnnets	show	129 ページの「 vlnnets show 」の項

表 5: VIOM オブジェクトとメソッド

このセクションの重要な用語

項目	構文	説明
Network ID	--id	作成されたネットワーク、VLAN ネットワーク、サービス LAN およびサービス VLAN の空間内における固有の整数
Network Name	--name	ネットワーク、VLAN ネットワーク、サービス LAN およびサービス VLAN の空間内における固有の名前

表 6: 重要な用語

5.6.1 api オブジェクト

api オブジェクトは、VIOM API に関する情報を取り出すためのメソッドを提供します。

5.6.1.1 api get

VIOM API 情報を画面表示します。

構文

```
viomcli -k <key> api get
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

api get メソッドには、パラメータがありません。

例

```
viomcli -k 4321 api get
Subject  Version  Build                Java                IP
-----|-----|-----|-----|-----
API      3.0.0      2012-02-06 17:42:02  1.6.0_30-b12       11.1.1.1
Manager  3.0.0      2012-02-06 17:42:07  1.6.0_30-b12       11.1.1.1

0 0 OK
```

The IP address in the 'Manager' row may be used as a trap destination in the *node setauth* command when managing rack servers.

5.6.2 config オブジェクト

config オブジェクトは、VIOM 設定を保存するためのメソッドを提供します。

5.6.2.1 config delete

管理サーバで、保存されている VIOM 設定ファイルを削除します。

構文

```
viomcli --key=<key> config delete --file=<string>
```

パラメータ

--key=<key> または短縮形 **-k <key>**

要求でユーザを識別するキー

--file=<string> または短縮形 **-f <string>**

バックアップファイルの名前とパス

例

```
viomcli -k 4321 config delete -f "backup\test.xml"
```

```
0 0 OK
```

5.6.2.2 config save

管理サーバで、VIOM 設定をファイルに保存します。

構文

```
viomcli --key=<key> config save --file=<string>[ --overwrite]  
[ --profiles][ --node=<integer1>][ --node=<integer2>] ...
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--file=<string> または短縮形 **-f <string>**
バックアップファイルの名前。パス名は任意。

--overwrite または短縮形 **-o**
任意指定
設定すると、既存のバックアップファイルは上書きされます。

--profiles または短縮形 **-p**
任意指定
設定すると、すべてのサーバプロファイルが保存されます。

--node=<integer> または短縮形 **-n <integer>**
ブレードまたは PRIMERGY ラックサーバのノード ID が保存されます。
node パラメータは複数回指定できます。

例

```
viomcli -k 4321 config save -f "backup\test.xml" --overwrite -p -n 59 -n 80 -n 123  
0 0 OK
```


5.6.2.3 configs show

保存されている VIOM 設定を、管理サーバで表示します。

構文

```
viomcli --key=<key> configs show
```

configs show メソッドには、パラメータがありません。

例

```
viomcli -k 4321 configs show
```

File	Created	Contains Profiles	Blade Servers
backup_2010_01_15.xml	2010-01-15 16:19:01	Yes	BX600-4,sqywwn00000, BX600S3, MCH 12345, Admin
testSuite/backup.xml	2010-01-14 19:50:57	Yes	BX600-4,sqywwn00000, BX600S3, MCH 12345, Admin

0 0 OK

5.6.3 iochannels オブジェクト

入出力チャンネルを保守するメソッドを提供します。

5.6.3.1 iochannels add

1 つまたは複数の入出力チャンネルをサーバプロファイルへ追加します。

構文

```
viomcli --key=<key> iochannels add --id=<integer>  
--spec=<list1> -|+iouse -|+bootuse[ --pxeboot=<list3>]  
[ --fcboot=<list4>][ --iscsiboot=<list5>]  
[ --fctarget=<list6>[ --fctarget=<list6>]]  
[ --iscsitarget=<list7>][ --vaddr=<list8>]  
[ --name=<string>][--svcnets=<list9>][--dcbgroup=<list10>]  
[ --bandwidth=<integer>][ --vlanid=<integer>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
サーバプロファイル ID

次のパラメータは同じパラメータグループに属し、それぞれの指定は複数回可能です。

--spec=<list1> または短縮形 **-s <list1>**
入出力チャンネルの指定

入出力チャンネルの指定は、次のコンマで区切られたプロパティ項目により行う必要があります。

<chantype>,<btype>,<bnum>,<port>[,<slotindex>][,<fnum>]

<chantype>

チャンネルタイプ (文字列)

lan

LAN

fc

ファイバーチャンネル

<i>isci</i>	iSCSI
<i>lanf</i>	CNA の LAN 機能
<i>fcf</i>	CNA の LAN 機能
<i>iscsif</i>	CNA の iSCSI 機能
<btype>	
入出力ボードタイプ（文字列）	
<i>dc</i>	拡張ボード
<i>ob</i>	オンボード
<i>ac</i>	拡張カード
<bnum>	
入出力ボード番号（整数）。メインボード入出力チャネル搭載のシステムボードが常にボード番号 1 になります。1 番目の拡張ボード（メザニンカード）がボード番号 1 になります。	
<port>	
入出力ボード番号（整数）。ボードの 1 番目の入出力チャネルがポート番号 1 になります。	
<slotindex>	
サーバプロファイルが 2 つ以上のサーバスロットに対応する場合、このインデックスで関連するスロットを指定します。デフォルトは「ru」。これは、シングルスロットまたは PRIMERGY ラックサーバのプロファイルの値でもあります。有効なインデックスは次のとおりです。	
<i>ru</i>	右上のスロット（デフォルト）
<i>lu</i>	左上のスロット
<i>rl</i>	右下のスロット

//

左下のスロット

<fnun>

CNA の `iochannel` の機能番号。デフォルト値は 1 です。

-|+ iouse または短縮形 **-u**

入出力チャネルの使用 (`-iouse: no`, `+iouse: yes`, `-u: yes`) : 入出力チャネルを有効にするか無効にするかを定義します。これにより LAN ポートまたは FC ポートを有効/無効にします。

-|+ bootuse または短縮形 **-d**

ブートデバイスの使用 (`-bootuse: no`, `+bootuse: yes`, `-d: yes`) : 入出力チャネルをブートデバイスとして使用するかどうかを定義します。最大 4 つまで入出力チャネルをブートデバイスとして定義することができます。

--bootenv=<list2> または短縮形 **-e <list2>**

`bootenv` オプションは無効になりました。互換性を保つ目的でのみサポートされています。代わりに、3 つのオプション `pxeboot`、`fcboot`、および `iscsiboot` が利用できます。

--pxeboot=<list3> または短縮形 **-p <list3>**

任意指定

ブート環境

PXE ブートのブート環境。入出力チャネルタイプが `lan` の場合にのみ設定できます

ブート環境の指定を次のプロパティにより行う必要があります。

<prio>

<prio>

ブート優先順 (整数)

可能な値は、1、2、3、4 です。ブート優先順位 1 が 1 番優先されます。この入出力チャネルが 1 番目のブートデバイスとされ、これに対し最初の試行が行われます。

--fcboot=<list4> または短縮形 **-f <list4>**

任意指定

ブート環境

SAN ブートのブート環境。入出力チャネルタイプが `fc` の場合にのみ設定できます。 `fctarget` は少なくとも 1 つ指定する必要があります。

ファイバーチャネルブート環境の指定を次のコンマで区切られたプロパティ項目により行う必要があります。

<prio>,<fcspeed>,<fctop>

<prio>

ブート優先順（整数）

可能な値は、1、2、3、4 です。ブート優先順位 1 が 1 番優先されます。この入出力チャンネルが 1 番目のブートデバイスとされ、これに対し最初の試行が行われます。

<fcspeed>

ファイバーチャンネル速度（整数）

このポートが使用する転送速度を指定します。

指定可能な値は次のとおりです。

0

オートネゴシエーション

転送速度は外部スイッチで調整されます。

1

1 Gbit/s 全二重

2

2 Gbit/s 全二重

4

4 Gbit/s 全二重

8

8 Gbit/s 全二重

<fctop>

ファイバーチャンネルトポロジ（整数）

外部 SAN ネットワークとのポート接続のタイプを指定します。

指定可能な値は次のとおりです。

0

自動（はじめにループで接続を試みます）

4

ポイントツーポイント

8

自動（はじめにポイントツーポイントで接続を試みます）

12

アービトラリーティッドループ

--iscsiboot=<list5> または短縮形 **-c <list5>**

任意指定

ブート環境

ブート環境の指定を次のコンマで区切られたプロパティ項目により行う必要があります。

iSCSI ブートのブート環境。入出力チャネルタイプが **lan** の場合にのみ設定できます

<prio>,<initiator>[,<address>,<subnet>,<gateway>]

<prio>

ブート優先順（整数）

可能な値は、**1**、**2**、**3**、**4** です。ブート優先順位 **1** が 1 番優先されます。この入出力チャネルが 1 番目のブートデバイスとされ、これに対し最初の試行が行われます。

<initiator>

iSCSI ターゲットへの接続に使用される iSCSI イニシエータの名前

<address>

任意指定

このポートに使用される、静的なクライアント IP アドレス (IPv4 アドレス、ドット区切り 10 進表記法)。ポートは全 iSCSI セッションを通じてこのアドレスを使用します。

<subnet>

任意指定

IP サブネットマスク（ドット区切り 10 進表記法）。このポートを接続するために使用するネットワークの IP サブネットマスクを指定します。

<gateway>

任意指定

ネットワークゲートウェイの IP アドレス（ドット区切り 10 進表記法）。選択した iSCSI ブートポート以外のサブネットワークに iSCSI ターゲットが存在する場合に指定が必要です。



address、**subnet**、**gateway** を指定しなかった場合は、アドレス、サブネット、ゲートウェイは DHCP サーバより取り込まれます。

--fctarget=<list6> または短縮形 **-t <list6>**

任意指定

ファイバーチャネルターゲット

ファイバーチャネルターゲットを最大 **2** つまで定義できます。ファイバーチャネルターゲットの指定を次のコンマで区切られたプロパティ項目により行う必要があります。

<wwpn>,<lun>

<wwpn>

ワールドワイドポートネーム - WWPN (64 ビット) (文字列)

WWPN は、**16** 文字の **16** 進数で指定します。

(例 : 50060e8004578604)

<lun>

ロジカルユニットナンバー (整数)

1 番目に指定されたターゲットに対し、最初の試行が行われます。**2** 番目のターゲットにより **2** 回目のブート **lun** のアドレスが指定されます。**1** 番目のアドレスへのアクセスに失敗した場合には、この **2** 番目のアドレスへのアクセスを試みます。**2** 番目のターゲットの指定は任意です。

--iscsitarget=<list7> または短縮形 **-g <list7>**

任意指定

iSCSI ブートの iSCSI ターゲット。

iSCSI ターゲットを指定しない場合は、DHCP を使用して DHCP サーバから iSCSI ターゲットを読み出します。

iSCSI ターゲットの指定を次のコンマで区切られたプロパティ項目により行う必要があります。

[<name>,<address>,<port>,<lun>][,<chapname>,<chapsecret>[<mutualchap secret>]]

<name>

iSCSI ターゲットの名前

<address>

iSCSI ターゲットの IP V4 アドレス (ドット区切り 10 進表記)

<port>

iSCSI ターゲットのポート番号

<lun>

ロジカルユニットナンバー (整数)

<chapname>

CHAP ユーザ名

このパラメータを省略すると、認証が使用されません。この CHAP ユーザ名は、iSCSI ターゲットで設定した名前と同じものである必要があります。

<chapsecret>

CHAP パスワード

iSCSI ターゲットで設定したパスワードと同じものを指定します。指定する場合、文字数は **12 ～ 16** です。このパスワードは **<mutualchap secret>** で設定するパスワードとは別のものである必要があります。

<mutualchap secret>

このフィールドには相互 CHAP パスワードを入力します。iSCSI ターゲットで設定したパスワードと同じものを指定します。指定する場合、文字数は **12 ～ 16** です。

このパスワードは **<chapsecret>** で設定するパスワードとは別のものである必要があります。

--vaddr=<list8> または短縮形 -v <list8>

任意指定

この IO チャンネルに定義される仮想アドレス。

VIOM のインストール時に、仮想 MAC アドレスおよび WWN アドレスのアドレス範囲を選択した場合、デフォルトでは仮想アドレスは指定した範囲内になります。

インストール時にアドレス範囲を指定していない場合、またはアドレスをかつこで囲んでいる場合、VIOM による範囲チェックは行われません。

このパラメータを指定しない場合、設定されている範囲から仮想アドレスが自動的に生成されます。

WWN アドレス (World WideNode Name および World Wide Port Name) および **MAC アドレス**は、次の順序で指定します。

[<mac>] | [<wwnn>], [<wwpn>] [, <mac>]

ターゲット

チャンネルの場合、「lan」、「lanf」、「iscsi」、「iscsif」と入力します

<wwnn>,<wwpn>

チャンネルの場合、「fc」と入力します

<wwnn>,<wwpn>,<mac>

チャンネルの場合、「fcf」と入力します



アドレスを入力しない場合、事前に定義した範囲から自動的に生成されます。

--name=<string> または短縮形 **-n <string>**

任意指定

ネットワーク名

--svcnets=<list9> または短縮形 **-m<list9>**

任意指定

サービス LAN 名またはサービス VLAN 名のリストを、コロンで区切って指定します。

--vlanid=<integer> または短縮形 **-l <list9>**

任意指定

IO チャンネルの VLAN ID。

--bandwidth=<integer> または短縮形 **-h <list9>**

任意指定

帯域幅は 0 ～ 100 で、CNA 機能にのみ指定できます。指定すると、下記の *dbcgroup* パラメータの *weight* が上書きされます。

--dcbgroup=<list10> または短縮形 **-w <list10>**

任意指定

DCB パラメータのカンマ区切りのリスト。

<weight> [, *<priority level>*]

<weight>

帯域幅の予約パーセント（整数）。

入力できる値は 1 から 100 です。

weight は、*bandwidth* パラメータがない場合のみ指定する必要があります。

<priority level>

優先グループの優先レベル（整数）。

入力できる値は 1 から 7 です。デフォルトでは、ファイバチャネルの場合は 3、iSCSI の場合は 4 です。



- グループパラメータで IO チャンネルを定義します。グループパラメータは一緒に使用し、グループは必ずパラメータ **spec** で始める必要があります。IO チャンネルには 1 つ以上のパラメータグループを使用できます。
- **iouse** を **No** に設定すると、**bootuse**、**pxeboot**、**fcboot**、**iscsiboot**、**fctarget**、**iscsitarget**、**vaddr** を指定できません。
- **bootuse** を **Yes** に設定した場合、**pxeboot**、**fcboot**、または **iscsiboot** を指定する必要があります。
- **bootuse** を **Yes** に、**chantype** を **fc** に設定した場合、**fcboot** および 1 つ以上の **fctarget** を指定する必要があります。
- **bootuse** を **Yes** および **chantype** を **lan** に設定した場合、**pxeboot** または **iscsiboot** を指定する必要があります。

例 1

イニシエータ名を「Initiator」、VLAN ID を 11 として、入出力チャンネルに iSCSI ブートを設定し、ソースとターゲットシステムに DHCP を設定するには次のコマンドを使用します。

```
viomcli -k 4321 iochannels add -id=2 --spec=lan,ob,1,1 +iouse +bootuse --iscsiboot=1.initiator,,,,11 --name=Network_2
0 0 OK
```

例 2

```
viomcli -k 4321 iochannels add -id=2 --spec=lan,ob,1,1 +iouse +bootuse --iscsiboot=1,name,,,,12
--iscsitarget=,,,chapname,secret
0 0 OK
```

この例では、ターゲットシステムで DHCP が使用され、CHAP が認証方法として使用されています。

例 3 - 仮想アドレスの指定

次の例では、以下に示すアドレス範囲がインストール中に定義されているものとします。

- MAC アドレスの範囲
00:19:99:3E:D2:A1 ~ 00:19:99:3E:F1:E0
- WWN アドレスの範囲
50:01:99:93:ED:2A:10:00 ~ 50:01:99:93:EF:1E:0D:FF

以下では、サーバプロファイル名を「Profile_1」として、4 つの入出力チャンネル（LAN チャンネル 1 つ、FC チャンネル 3 つ）をサーバプロファイルに定義します。

```
viomcli -k 1234 profile add --name=Profile_1 +virt +boot
1
0 0 OK
```

この例では、出力されたサーバプロファイル ID は 1 です。

入出力チャンネルは次のように定義します。

```
viomcli -k 1234 iochannels add --id=1 --spec=lan,ob,1,1 +iouse
+bootuse --pxeboot=1 --vaddr=0019993ef000 --name=Network_1
0 0 OK
```

```
viomcli -k 1234 iochannels add --id=1 --spec=fc,dc,1,1 +iouse
+bootuse --fcboot=2,10,2 --vaddr=50019993EF000000 --fctarget=
00aabbccddeeff00,1 --name=Network_2
0 0 OK
```

```
viomcli -k 1234 iochannels add --id=1 --spec=fc,dc,1,2 +iouse
+bootuse --fcboot=3,10,2 --vaddr=,50019993EF000001 --fctarget=
00aabbccddeeff00,1 --name=Network_2
0 0 OK
```

```
viomcli -k 1234 iochannels add --id=1 --spec=fc,dc,1,3 +iouse
+bootuse --fcboot=4,10,2 --vaddr=50019993EF000002, 50019993EF000003
--fctarget=00aabbccddeeff00,1 --name=Network_2
0 0 OK
```

例 4

この例を、例 1 と比較します。ここでは、MAC アドレス 01:01:01:01:01:01 がインストール時に指定した範囲内ではないので、かつこで囲む必要があります。

```
viomcli -k 1234 iochannels add --id=1 --spec=lan,ob,1,1
+iouse +bootuse --pxeboot=1 --vaddr=(010101010101) --name=
Network_1
```

```
0 0 OK
```

5.6.3.2 iochannel get

入出力チャネルに関する情報を取得します。

構文

```
viomcli --key=<key> iochannel get --id=<integer>  
--spec=<list>[ --boot| --address| --dcbgroups]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
サーバプロファイル ID

--spec=<list> または短縮形 **-s <list>**
入出力チャネルの指定

入出力チャネルの指定を次のコンマで区切られたプロパティ項目により行う必要があります。

<chantype>,<btype>,<bnum>,<port>[,<slotindex>]
[,<fnum>]

<chantype>
チャネルタイプ (文字列)

lan

LAN

fc

ファイバチャネル

isci

iSCSI

lanf

CNA の LAN 機能

fcf

CNA の FC 機能

iscsif

CNA の iSCSI 機能

<btype>

入出力ボードタイプ（文字列）

dc

拡張ボード

ob

オンボード

ac

拡張カード

<bnum>

入出力ボード番号（整数）。オンボード入出力チャネル搭載のシステムメインボードが常にボード番号 1 になります。1 番目の拡張ボード（メザニンカード）がボード番号 1 になります。

<port>

入出力ポート番号（整数）。ボードの 1 番目の入出力チャネルがポート番号 1 になります。

<slotindex>

サーバプロファイルが 2 つ以上のサーバスロットに対応する場合、このインデックスで関連するスロットを指定します。デフォルトは *ru* で、単一のスロットプロファイルの値でもあります。

有効なインデックスは次のとおりです。

ru

右上のスロット（デフォルト）

lu

左上のスロット

rl

右下のスロット

ll

左下のスロット

<fnum>

CNA の *iochannel* の機能番号。デフォルト値は 1 です。

--boot または短縮形 **-b**
任意指定
ブート環境のみ表示します。

--address または短縮形 **-a**
任意指定
アドレス仮想化のみ表示します。

--dcbgroups または短縮形 **-w**
オプション
IO チャンネルに関連する優先グループを表示します。

例1

```
viomcli -k 4321 iochannel get -i 123 --spec=fc,dc,1,2
```

```
Server. Profile Id:    123
Network name:         net2
IO channel usage:     Yes
Boot device usage:    Yes
Service network name: svcnet1
Service network name: svcnet2
0 0 OK
```

例2

```
viomcli -k 4321 iochannel get --id=3 --spec=fcf,dc,1,1,ru,2 -b
```

Boot	Prio.	Link	Speed	FC	Topo.	WWPN	LUN
1		0		8		01:01:01:01:01:01:01:01	0
1		0		8		02:02:02:02:02:02:02:02	0

```
0 0 OK
```

例3

ここでの例では、**iochannel** を **Ethernet** 経由でファイバチャネルに使用します。

```
viomcli -k 4321 iochannel get --id=3 --spec=fcf,dc,1,1,ru,2 -dcbgroups
```

Priority	Level	Weight	Purpose
3		50	FCoE

```
0 0 OK
```

例 4

この例では、iSCSI ブートのブートパラメータ（58 ページの "例 2" で設定）を求めます。

```
viomcli -k 4321 iochannel get -i 2 --spec=lan,ob,1,1 --boot
```

DHCP	Initiator	Address	Netmask	Gateway	VLAN ID	DHCP	Target	Address	Port	LUN	CHAP
Yes	name				12	Yes					CHAP

Name	Secret	Mutual	Secret
chapname	secret		

0 0 OK
例 5

この例では、各入出力チャンネルに、59 ページの "例 3 - 仮想アドレスの指定" の、どの仮想アドレスが現在割り当てられているかを求めます。

```
viomcli -k 1234 iochannel get --id=1 --spec=lan,ob,1,1 -a
```

MAC	WWNN	WWPN
00:19:99:3e:f0:00		

0 0 OK

```
viomcli -k 1234 iochannel get --id=1 --spec=fc,dc,1,1 -a
```

MAC	WWNN	WWPN
	50:01:99:93:ef:00:00:00	50:01:99:93:ed:2a:10:00

0 0 OK

```
viomcli -k 1234 iochannel get --id=1 --spec=fc,dc,1,2 -a
```

MAC	WWNN	WWPN
	50:01:99:93:ed:2a:10:01	50:01:99:93:ef:00:00:01

0 0 OK

```
viomcli -k 1234 iochannel get --id=1 --spec=fc,dc,1,3 -a
```

MAC	WWNN	WWPN
	50:01:99:93:ef:00:00:02	50:01:99:93:ef:00:00:03

0 0 OK

5.6.3.3 iochannel remove

IO チャンネルを削除します。

構文

```
viomcli --key=<key> iochannel remove --id=<integer>  
--spec=<list>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
サーバプロファイル ID

--spec=<list1> または短縮形 **-s <list1>**
IO チャンネルの指定

IO チャンネルを、次のコンマで区切られたプロパティリストにより指定する必要があります。

<chantype>,<btype>,<bnun>,<port>[,<slotindex>][,<fnum>]

<chantype>

チャンネルタイプ (文字列)

lan

LAN

fc

ファイバチャンネル

iscsi

iSCSI

lanf

CNA の LAN 機能

fcf

CNA の FC 機能

iscsif

CNA の iSCSI 機能

<btype>

入出力ボードタイプ（文字列）

dc

拡張ボード

ob

オンボード

ac

拡張カード

<bnum>

入出力ボード番号（整数）。オンボード入出力チャネル搭載のシステムメインボードが常にボード番号 **1** になります。**1** 番目の拡張ボード（メザニンカード）がボード番号 **1** になります。

<port>

入出力ポート番号（整数）。ボードの **1** 番目の入出力 チャネルがポート番号 **1** になります。

<slotindex>

サーバプロファイルが 2 つ以上のサーバスロットに対応する場合、このインデックスで関連するスロットを指定します。デフォルトは「**ru**」。これは、シングルスロットプロファイルの値でもあります。

有効なインデックスは次のとおりです。

ru

右上のスロット（デフォルト）

lu

左上のスロット

rl

右下のスロット

ll

左下のスロット

<fnun>

CNA の **iochannel** の機能番号。デフォルト値は **1** です。

5.6.3.4 iochannel set

既存の入出力チャネルの値を設定します。

構文

```
viomcli --key=<key> iochannel set --id=<integer>
--spec=<list1> -|+iouse -|+bootuse[ -|+dcbase][ --pxeboot=
<list3>][ --fcboot=<list4>][ --iscsiboot=<list5>]
[ --fctarget=<list6>[ --fctarget=<list6>]]
[ --iscsitarget=<list7>][ --vaddr=<list8>]
[ --name=<string>][--svcnets=<list9>][--dcbgrou=<list10>]
[ --bandwidth=<integer>][ --vlanid=<integer>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
サーバプロファイル ID

次のパラメータは同じパラメータグループに属し、それぞれの指定は複数回可能です。

--spec=<list1> または短縮形 **-s <list1>**
入出力チャネル指定

入出力チャネルの指定を次のコンマで区切られたプロパティ項目により行う必要があります。

<chantype>,<btype>,<bnun>,<port>[,<slotindex>][,<fnum>]

<chantype>

チャネルタイプ (文字列)

iscsi

iSCSI

lanf

CNA の LAN 機能

fcf

CNA の FC 機能

iscsif

CNA の iSCSI 機能

<btype>

入出力ボードタイプ（文字列）

dc

拡張ボード

ob

オンボード

ac

拡張カード

<bnum>

入出力ボード番号（整数）。オンボード入出力チャネル搭載のシステムメインボードが常にボード番号 1 になります。1 番目の拡張ボード（メザニンカード）がボード番号 1 になります。

<port>

入出力ポート番号（整数）。ボードの 1 番目の入出力チャネルがポート番号 1 になります。

<slotindex>

サーバプロファイルが 2 つ以上のサーバスロットに対応する場合、このインデックスで関連するスロットを指定します。デフォルトは *ru* で、単一のスロットプロファイルの値でもあります。

有効なインデックスは次のとおりです。

ru

右上のスロット（デフォルト）

lu

左上のスロット

rl

右下のスロット

ll

左下のスロット

<fnun>

CNA の *iochannel* の機能番号。デフォルト値は 1 です。

-[+ iouse または短縮形 -u

入出力チャネルの使用（*-ious*: no、*+ious*: yes、*-u*: yes）：入出力チャネルを有効にするか無効にするかを定義します。これにより LAN ポートまたは FC ポートを有効／無効にします。

-|+ bootuse または短縮形 **-d**

ブートデバイスの使用 (*-bootuse: no*, *+bootuse: yes*, *-d: yes*) : 入出力チャンネルをブートデバイスとして使用するかどうかを定義します。最大4つまで入出力チャンネルをブートデバイスとして定義することができます。

-|+ dcbase または短縮形 **-b**

任意指定

iochannel の Data Center Bridging を有効または無効にします(「*-dcbase*」: 無効、「*+dcbase*」: 有効、「*-b*」: 有効)



「*-dcbase*」は優先グループを削除します。

--bootenv=<list2> または短縮形 **-e <list2>**

bootenv オプションは無効になりました。互換性を保つ目的でのみサポートされています。代わりに、3つのオプション *pxeboot*、*fcboot*、および *iscsiboot* が利用できます。

--pxeboot=<list3> または短縮形 **-p <list3>**

任意指定

ブート環境

ブート環境の指定を次のプロパティにより行う必要があります。

<prio>

<prio>

ブート優先順 (整数)

可能な値は、1、2、3、4です。ブート優先順位1が1番優先されます。この入出力チャンネルが1番目のブートデバイスとされ、これに対し最初の試行が行われます。

--fcboot=<list4> または短縮形 **-f <list4>**

任意指定

ブート環境

入出力チャンネルタイプが *fc* の場合にのみ設定できます。

ファイバーチャンネルブート環境の指定を次のコンマで区切られたプロパティ項目により行う必要があります。

<prio>, *<fcspeed>*, *<fctop>*

<prio>

ブート優先順 (整数)

可能な値は、1、2、3、4 です。ブート優先順位 1 が 1 番優先されます。この入出力チャンネルが 1 番目のブートデバイスとされ、これに対し最初の試行が行われます。

<fcspeed>

ファイバーチャネル速度（整数）
このポートが使用する転送速度を指定します。
指定可能な値は次のとおりです。

- 0
オートネゴシエーション
転送速度は外部スイッチで調整されます。
- 1
1 Gbit/s 全二重
- 2
2 Gbit/s 全二重
- 4
4 Gbit/s 全二重
- 8
8 Gbit/s 全二重

<fctop>

ファイバーチャネルトポロジー
外部 SAN ネットワークとのポート接続のタイプを指定します。
指定可能な値は次のとおりです。

- 0
自動（はじめにループで接続を試みます）
- 4
ポイントツーポイント
- 8
自動（はじめにポイントツーポイントで接続を試みます）
- 12
アービトラレーティッドループ

--iscsiboot=<list5> または短縮形 **-c <list5>**

任意指定
ブート環境
ブート環境の指定を次のコンマで区切られたプロパティ項目により行う必要があります。

<prio>,<initiator>[,<address>,<subnet>,<gateway>]

<prio>

ブート優先順（整数）

可能な値は、1、2、3、4 です。ブート優先順位 1 が 1 番優先されます。この入出力チャンネルが 1 番目のブートデバイスとされ、これに対し最初の試行が行われます。

<initiator>

iSCSI ターゲットへの接続に使用される iSCSI イニシエータ (iSCSI ブートの場合) の名前

<address>

任意指定

このポートに使用される、静的なクライアント IP アドレス (IPv4 アドレス、ドット区切り 10 進表記法)。ポートは全 iSCSI セッションを通じてこのアドレスを使用します。

<subnet>

任意指定

IP サブネットマスク（ドット区切り 10 進表記法）。このポートを接続するために使用するネットワークの IP サブネットマスクを指定します (iSCSI ブートの場合)。

<gateway>

任意指定

ネットワークゲートウェイの IP アドレス（ドット区切り 10 進表記法）。選択した iSCSI ブートポート以外のサブネットワークに iSCSI ターゲットが存在する場合に指定が必要です。



address、**subnet**、**gateway** を指定しなかった場合は、アドレス、サブネット、ゲートウェイは DHCP サーバより取り込まれます。

--fctarget=<list6> または短縮形 **-t <list6>**

任意指定

ファイバーチャネルターゲット

ファイバーチャネルターゲットを最大 2 つまで定義できます。ファイバーチャネルターゲットの指定を次のコマンドで区切られたプロパティ項目により行う必要があります。

<wwpn>,<lun>

<wwpn>

ワールドワイドポートネーム - WWPN（64 ビット）（文字列）

<lun>

ロジカルユニットナンバー（整数）

--iscsitarget=<list7> または短縮形 **-g <list7>**

任意指定

iSCSI ターゲット

iSCSI ターゲットを指定しない場合は、DHCP を使用して DHCP サーバから iSCSI ターゲットを読み出します。

iSCSI ターゲットの指定を次のコンマで区切られたプロパティ項目により行う必要があります。

[<name>,<address>,<port>,<lun>][,<chapname>,<chapsecret>[<mutualchap secret>]]

<name>

iSCSI ターゲットの名前

<address>

iSCSI ターゲットの IP V4 アドレス（ドット区切り 10 進表記）

<port>

iSCSI ターゲットのポート番号

<lun>

ロジカルユニットナンバー（整数）

<chapname>

CHAP ユーザ名。このパラメータを省略すると、認証が使用されません。iSCSI ターゲットで設定した名前と同じものを指定します。

<chapsecret>

CHAP パスワード

iSCSI ターゲットで設定したパスワードと同じものを指定します。指定する場合、文字数は 12 ～ 16 です。

<mutualchap secret>

このフィールドには相互 CHAP パスワードを入力します。iSCSI ターゲットで設定したものと同一パスワードを指定します。指定する場合、文字数は 12 ～ 16 です。

このパスワードは <chapsecret> で設定するパスワードとは別のものである必要があります。

--vaddr=<list8> または短縮形 **-v<list8>**

任意指定

この IO チャネルに定義される仮想アドレス。

VIOM のインストール時に、仮想 MAC アドレスおよび WWN アドレスのアドレス範囲を選択した場合、デフォルトでは仮想アドレスは指定した範囲内になります。

インストール時にアドレス範囲を指定していない場合、またはアドレスをカッコで囲んでいる場合、VIOM による範囲チェックは行われません。

このパラメータを指定しない場合、設定されている範囲から仮想アドレスが自動的に生成されます。

WWN アドレス (World WideNode Name および World Wide Port Name) および MAC アドレスは、次の順序で指定します。

[<mac>]| [<wwnn>],[<wwpn>][,<mac>]

ターゲット チャンネルの場合、「lan」、「lanf」、「iscsi」、「iscsif」と入力します

<wwnn>,<wwpn> チャンネルの場合、「fc」と入力します

<wwnn>,<wwpn>,<mac> チャンネルの場合、「fcf」と入力します

<mac>

MAC アドレス

MAC アドレスは、12 文字の 16 進数で指定します。

<wwnn>

ワールドワイドノードネーム - WWNN

WWNN は、16 文字の 16 進数で指定します。

<wwpn>

ワールドワイドポートネーム - WWPN (64 ビット) (文字列)

WWPN は、16 文字の 16 進数で指定します。



アドレスを入力しない場合、事前に定義した範囲から自動的に生成されます。

--name=<string> または短縮形 **-n <string>**

任意指定

ネットワーク名



同じ CNA ポートのさまざまな機能が同じネットワークを共有します。そのため、前のネットワークを上書きできます。

--svcnets=<list9>=*<list9>* または短縮形 *-m* *<list9>*

任意指定

サービス LAN 名またはサービス VLAN 名のリストをコンマで区切って指定します。

--vlanid=<integer> または短縮形 *-l* *<list9>*

任意指定

IO チャンネルの VLAN ID。

--bandwidth=<integer> または短縮形 *-h* *<list9>*

任意指定

帯域幅は 0 ～ 100 で、CNA 機能にのみ指定できます。指定すると、下記の *dbcgroup* パラメータの *weight* が上書きされます。

--dbcgroup=<list10> または短縮形 *-w* *<list10>*

任意指定

DCB パラメータのカンマ区切りのリスト。優先グループを指定した場合、*「+dcbuse」* も指定します。

<weight>[,<prioritylevel>]

<weight>

帯域幅の予約パーセント（整数）。

入力できる値は 1 から 100 です。

weight は、*bandwidth* パラメータがない場合のみ指定する必要があります。

<priority level >

優先グループの優先レベル（整数）。

入力できる値は 1 から 7 です。デフォルトでは、ファイバチャネルの場合は 3、iSCSI の場合は 4 です。



- *iouse* を *No* に設定する場合、*bootuse*、*pxeboot*、*fcboot*、*iscsiboot*、*fctarget*、*iscsitarget*、*vaddr* は指定できません。
- *bootuse* を *Yes* に設定する場合、*pxeboot*、*fcboot*、*iscsiboot* のいずれかを指定する必要があります。
- *chantype* を *fc* に設定する場合、*fcboot* および 1 つ以上の *fctarget* を指定する必要があります。
- *chantype* を *lan* に設定する場合、*pxeboot* または *iscsiboot* のみを指定します。

例

```
viomcli -k 4321 -o csv iochannel set -i 2 --spec=fc,dc,1,2 +iouse -bootuse -n net3
0 0 OK
```

5.6.3.5 iochannels show

サーバプロファイルの入出力チャネルを表示します。

構文

```
viomcli --key=<key> iochannels show --id=<integer>
```

パラメータ

- key=<key> または短縮形 -k <key>
要求でユーザを識別するキー
- id=<integer> または短縮形 -i <integer>
サーバプロファイル ID

例

```
viomcli -k 123 iochannels show --id=2
Ch.Type Index Brd.Type Brd.Number Port Number FunctionNumber Netw.Name IO Ch. Usage Boot Usage DCB Usage Boot Prio.
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
LAN          OnBoard 1         1         1         Net_2      Yes      Yes      No      1
LAN          OnBoard 1         2         1         Net_2      Yes      No       No
LAN    r1    OnBoard 1         1         1         Net_2      Yes      Yes      No      2
LAN    r1    OnBoard 1         2         1         Net_2      Yes      No       No

FC Boot Cfg.
|-----|
1
1
1
1
0 0 OK
```

このコマンド出力は、ダブルハイトの BX960 CPU ブレードの、サーバプロファイルの入出力チャネルを示します。Index 列はマルチスロット CPU ブレードにのみ使用され、iochannels add コマンドで記述されるスロットインデックスです。

5.6.4 network オブジェクト

VIOM ネットワークの保守を行うメソッドを提供します。

5.6.4.1 network add

新しいネットワークを作成します。このコマンドにより新規に作成されたネットワークの ID を標準出力へ返します。

構文

```
viomcli --key=<key> network add --name=<string>
--node=<integer>[ --uset=<integer>][+|-isol]
[ --comment=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--name=<string> または短縮形 **-n <string>**
ネットワーク名

--node=<integer> または短縮形 **-o <integer>**
スイッチスロットのノード ID。ネットワークを追加するには、IBP コネクショングレードをこのスイッチスロットへ差し込む必要があります。

スイッチスタッキングの場合、ここにマスタースイッチのノード ID を指定します。スイッチスタッキングの詳細については、『Virtual-IO Manager 取扱説明書』を参照してください。

--uset=<integer> または短縮形 **-u <integer>**
任意指定
アップリンクセット ID

+|-isol または短縮形 **-j**
ネットワークのダウンリンクアイソレーション（「-isol」：なし、「+isol」：あり、「-j」：あり）。デフォルトはダウンリンクアイソレーションなしです。

--comment=<string> または短縮形 **-c <string>**
任意指定
コメント

例

```
viomcli -k 4321 network add -n test_net -o 134
```

```
5
0 0 OK
```

この例では、「5」が新規に作成されたネットワークの ID で、標準出力に書き込まれます。コマンドステータス情報（「0 0 OK」）は、標準エラーへ書き込まれます。エラーが発生すると（リターンコードが 0 でない）、ID は標準出力へ書き込まれません。

5.6.4.2 network get

ネットワーク情報を取得します。

構文

```
viomcli --key=<key> network get --id=<integer>[ --comment]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
ネットワーク ID

--comment または短縮形 **-c**
任意指定
コメントのみを表示します。

例

```
viomcli -k 4321 network get -i 14
Netw. Name  Netw. Id  Node Id  Uplink Set Id  Slots  Isolation
|-----|-----|-----|-----|-----|-----|
test_net_2  14        134      null          No
```

```
0 0 OK
```

5.6.4.3 network remove

ネットワークを削除します。

構文

```
viomcli --key=<key> network remove --id=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
ネットワーク ID

例

```
viomcli -k 4321 network remove -i 14
```

```
0 0 OK
```

5.6.4.4 network set

既存のネットワークの値を設定します。

構文

```
viomcli --key=<key> network set --id=<integer>  
[ --name=<string>][ --uset=<integer>][+|-isol]  
[ --comment=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
ネットワーク ID

--name=<string> または短縮形 **-n <string>**
任意指定
ネットワーク名

--uset=<integer> または短縮形 **-u <integer>**
任意指定
アップリンクセット ID

+|-isol または短縮形 **-j**
ネットワークのダウンリンクアイソレーション（「-isol」：なし、
「+isol」：あり、「-j」：あり）。

--comment=<string> または短縮形 **-c <string>**
任意指定
コメント

例

```
viomcli -k 4321 network set -i 14 -n test_net2 -c "my test network II"  
0 0 OK
```

5.6.4.5 networks show

ノードのネットワークを表示します。

構文

```
viomcli --key=<key> networks show --node=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--node=<integer> または短縮形 **-o <integer>**
スイッチスロットのノード ID

例

```
viomcli -k 4321 networks show -o 134
```

Networks:

=====

Netw. Name	Netw. Id	Node Id	Uplink Set Id	Slots
NET_1	5	134	5	
NET_2	6	134	6	
NET_3	7	134	7	
NET_4	8	134	8	
test_net	13	134	null	
test_net_2	14	134	null	

VLAN Networks:

=====

Netw. Name	Netw. Id	Node Id	Uplink Set Id	Slots	VLAN Id

Service LANs:

=====

Netw. Name	Netw. Id	Node Id	Uplink Set Id	Slots	VLAN Id

Service VLANs:

=====

Netw. Name	Netw. Id	Node Id	Uplink Set Id	Slots	VLAN Id

0 0 OK

5.6.5 node オブジェクト

VIOM ノードの保守を行うメソッドを提供します。

ノードはすべてのタイプのデバイスを指し、各ノードはそれぞれ固有のノード ID により識別されます。ノードには、メインタイプ (Group、Slot など) とサブタイプ (Blade_Server、Server_Slot など) があります。

次のノードタイプが可能です。

メインタイプ	サブタイプ
Slot	Server_Slot
Slot	Switch_Slot
Switch	LAN_Switch
Switch	FC_Switch
Switch	Switch
Group	Blade_Server
MMB	MMB
Server	Server
Server	Blade

表 7: ノードタイプ

ノードは別のノードに関連付けられている場合があります。例えば、Group タイプのノードオブジェクトは、スロットなどのターゲットに対し、「provider」という関連付けが割り当てられています。

5.6.5.1 node getConfig

ノードに関する設定情報を表示します。

構文

```
viomcli --key=<key> node getConfig --node=<integer>[ --assoc]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--node=<integer> または短縮形 **-n <integer>**
表示されるノード ID

--assoc または短縮形 **-a**
任意指定
関連付けのみ表示します。

例1

```
viomcli -k 4321 node getConfig -n 135 -a
```

Node Id	Type	Direction
122	Group	Provider
140	Slot	Target
0 0	OK	

例2

```
viomcli -k 4321 node getconfig -n 122

node Id:          122
node type:        GROUP BLADE_SERVER
node name:        BX600-4
node status:      OK
Serial Number:    sqywn00000
VIOM manage status: Manageable
VIOM fault status: Ok
VIOM fault causes:
system:          BX600-4
UUID:
product:         PRIMERGY BX600 Management Blade S3
firmware:        3.08
housing:         BX600S3
admin IP:        111.22.33.44
administrator:   Martin Muster
location:        MCH 12345
firmware:        3.08
admin firmware:  2.09
housing:         BX600S3
IP:              111.22.33.43
admin IP:        111.22.33.44
```

Associations:

Node Id	Type	Direction
131	Group	Target
107	Group	Target
136	Group	Target
126	Group	Target
139	Group	Target
127	Group	Target
123	Group	Target
130	Group	Target
135	Group	Target
140	Group	Target
128	Group	Target
134	Group	Target
129	Group	Target
124	Group	Target
137	Group	Target
132	Group	Target
133	Group	Target
106	Group	Target
125	Group	Target
138	Group	Target

0 0 OK

5.6.5.2 node manage

ノードを管理します。管理可能な「Group Blade_Server」または「Server Server」タイプのノードのみ管理できます。

構文

```
viomcli --key=<key> node manage --node=<integer>[ --force]
```

パラメータ

--key=<key> または短縮形 **-k <key>**

要求でユーザを識別するキー

--node=<integer> または短縮形 **-n <integer>**

管理するノード ID (ブレードサーバまたは PRIMERGY ラックサーバ)

--force または短縮形 **-f**

任意指定

ノードが別の **Virtual-IO Manager** に管理されている場合でも強制的に行います。別の管理用サーバによってすでに管理されているノードを指定した場合、このサーバを管理する要求は失敗し、対応するメッセージが表示されます。この場合でも、**--force** パラメータにより、そのノードを管理できます。**force** オプションをデフォルト設定するのはお勧めしません。このオプションを使用する前に、必ず他の管理用サーバ/パソコンが使用されていないことを確認してください。

例

```
viomcli -k 4321 node manage -n 17 -f
```

```
0 0 OK
```

5.6.5.3 node restore

VIOM の内部設定を反映するためにノードをリストアします。

構文

```
viomcli --key=<key> node restore --node=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--node=<integer> または短縮形 **-n <integer>**
リストアするノード ID (ブレードサーバ、スイッチスロット、サーバスロット、PRIMERGY ラックサーバ)

例

```
viomcli -k 4321 node restore -n 140
```

```
0 0 OK
```

5.6.5.4 node setauth

ノードの認証情報を設定します。

構文

```
viomcli --key=<key> node setauth --node=<integer>
--prot=<string> --user=<string> --pass=<string>
--port=<integer>[ --trap=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**

要求でユーザを識別するキー

--node=<integer> または短縮形 **-n <integer>**

認証データを設定するノードのノード ID。これは、MMB、スイッチ、および PRIMERGY ラックサーバのノードに使用できます。現在、VIOM で管理されるブレードサーバシャーシ、ブレードサーバの MMB、IBP コネクションモジュール、および PRIMERGY ラックサーバに、認証データを設定する必要があります。

--prot=<string> または短縮形 **-r <string>**

プロトコル:

tel

telnet

ssh

セキュアシェル

ipmi

LAN 経由の IPMI

--user=<string> または短縮形 **-u <string>**

ユーザ名

--pass=<string> または短縮形 **-w <string>**

ユーザのパスワード

--port=<integer> または短縮形 **-p <integer>**

プロトコルポート

--trap=<integer> または短縮形 **-t <integer>**

プロトコルポート

SNMP トラップを送信するために、管理対象の PRIMERGY ラックサーバが使用する IP アドレス。46 ページの「[api get](#)」の項を参照してください。



- SWITCH タイプのノードの場合、デフォルト設定は、プロトコルが **telnet** でポート **23**、またはプロトコルが **SSH** でポート **22** です。**telnet** に別のポートを使用するようコネクションブレードが設定されている場合、対応するポートを指定してください。
- MMB タイプのノードの場合、デフォルト設定は、プロトコルが **telnet** でポート **3127**、またはプロトコルが **SSH** でポート **22** です。**telnet** に別のポートを使用するよう **MMB** が設定されている場合、対応するポートを指定してください。
- PRIMERGY ラックサーバノードの場合、デフォルトポートが **623** の LAN 経由の IPMI タイプである必要があります。

例

```
viomcli -k 4321 node setauth -n 17 --prot=tel -u test --pass=test1  
--port=3172
```

```
0 0 OK
```

5.6.5.5 node setconfig

既存のノードの値を設定します。

構文

```
viomcli --key=<key> node setconfig --id=<integer>[ --spare]
```

パラメータ

--key=<key> または短縮形 **-k <key>**

要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**

ノード ID

--spare または短縮形 **-s**

任意指定

設定すると、このノードがフェイルオーバーでの予備のノードとなります（サーバスロットのみ）。



「spare」が設定されない場合、既存の予備設定は削除されます。

例

```
viomcli -k 4321 node setconfig -i 17 -spare
```

```
0 0 OK
```

5.6.5.6 nodes show

すべてのノードを表示します。

構文

```
viomcli --key=<key> nodes show
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

nodes show メソッドには、パラメータがありません。

例

```
viomcli -k 4321 nodes show
```

Change 1

Id	Node Name	State	Main Type	Sub Type	VIOM State	Assoc. Type
62	BX600-3\F03	Unknown	SLOT	SERVER_SLOT		Group, Provider, 58 Slot, Target, 105
30	BX600S3-2\F03	Unknown	SLOT	SERVER_SLOT		Group, Provider, 26 Slot, Target, 92
31	BX600S3-2\F04	Unknown	SLOT	SERVER_SLOT		Group, Provider, 26 Slot, Target, 93
95	TX300-RD	OK	SERVER	SERVER		
123	BX600-4\M00	OK	MMB	MMB		Group, Provider, 122
113	BX600S3-1\C03	Unknown	SERVER	BLADE		Group, Provider, 74 Slot, Provider, 78
86	BX600S3-1\R01	Unknown	SLOT	SWITCH_SLOT		Group, Provider, 74

5.6.5.7 node unmanage

ノードの管理を解除します。

構文

```
viomcli --key=<key> node unmanage --node=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--node=<integer> または短縮形 **-n <integer>**
管理を解除するノード ID

例

```
viomcli -k 4321 node unmanage -n 17
```

```
0 0 OK
```

5.6.6 ports オブジェクト

VIOM アップリンクポートの保守を行うメソッドを提供します。

5.6.6.1 ports add

1 つまたは複数のアップリンクポートをアップリンクセットに追加します。

構文

```
viomcli --key=<key> ports add --uid=<integer> --port=<list>  
[ --port=<list>] ...
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--uid=<integer> 短縮形 **-u <integer>**
アップリンクセット ID

次のパラメータの指定は、複数回可能です。

--port=<list> または短縮形 **-p <list>**

追加するアップリンクポート。ポートの指定を次のコンマで区切られたプロパティ項目により行う必要があります。

<portnumber>,<backup>[,<slotid>]

<portnumber>

アップリンクポート番号

アップリンクポート番号は、IBP に関係なく必ず 1 から始まります。たとえば、**--port=1,N** は、IBP10/6 のポート 0/11 または IBP30/12 のポート 0/31 を意味します。

<backup>

バックアップとして機能する（ブール）：

Y

はい

N

いいえ

<slotid>

スイッチスタッキングの場合、これは追加するポートのスロット ID です。デフォルトは、マスタースイッチスロットのノード ID です。

例

```
viomcli -k 4321 ports add --uid=1 --port=2,Y --port=3,N
```

```
0 0 OK
```

5.6.6.2 ports remove

アップリンクセットから 1 つまたは複数のポートを削除します。

構文

```
viomcli --key=<key> ports remove --uid=<integer>  
--port=<list>[ --port=<list>] ...
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--uid=<integer> または短縮形 **-u <integer>**
アップリンクセット ID

次のパラメータの指定は、複数回可能です。

--port=<list> または短縮形 **-p <list>**
削除するアップリンクポート。ポートは、次のようにコンマ区切りの
プロパティリストで指定します。

<portnumber>[,<slotid>]

<portnumber>

アップリンクポート番号

アップリンクポート番号は、IBP に関係なく必ず 1 から始まります。たとえば、**--port=1,N** は、IBP10/6 のポート 0/11 または IBP30/12 のポート 0/31 を意味します。

<slotid>

スイッチスタッキングの場合、これは追加するポートのスロット ID です。デフォルトは、マスタースイッチスロットのノード ID です。

例

```
viomcli -k 4321 ports remove --uid=10 -p 2 --port=3
```

```
0 0 OK
```

5.6.7 power オブジェクト

電源管理のためのメソッドを提供します。

5.6.7.1 power get

ノードの電源情報を表示します。

構文

```
viomcli --key=<key> power get --node=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**

要求でユーザを識別するキー

--node=<integer> または短縮形 **-n <integer>**

電源情報表示の対象となるノード ID (サーバブレード、PRIMERGY ラックサーバ)

例

```
viomcli -k 4321 power get -n 138
```

Node Id	Power Mode	Boot Mode	Status
138	Off	Normal	PowerOnPending

```
0 0 OK
```

5.6.7.2 power set

電源設定の値を設定します。

構文

```
viomcli --key=<key> power set --node=<integer>  
--power=<string> --boot=<string>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--node=<integer> または短縮形 **-n <integer>**
電源設定の対象となるノード ID (サーバブレード、PRIMERGY ラックサーバ)

--power=<string> または短縮形 **-p <string>**
パワーモード

on
スイッチをオンにする

gs
シャットダウン後、電源断を行う

pc
パワーサイクルを行う

rst
リセットする

fo
強制オフを行う

--boot=<string> または短縮形 **-b <string>**
ブートモード

norm
標準

pxe1
PXE LAN 1

pxe2
PXE LAN 2

<i>pxe3</i>	PXE LAN 3
<i>pxe4</i>	PXE LAN 4
<i>pxe5</i>	PXE LAN 5
<i>pxe6</i>	PXE LAN 6
<i>pxe7</i>	PXE LAN 7
<i>pxe8</i>	PXE LAN 8
<i>pci1</i>	PCI 1
<i>pci2</i>	PCI 2
<i>inv</i>	インベントリ
<i>init</i>	init

例

```
viomcli -k 4321 power set -n 138 -p pc -b pxe1  
0 0 OK
```

5.6.8 profile オブジェクト

サーバプロファイルの保守を行うメソッドです。

5.6.8.1 profile add

新しいサーバプロファイルを作成します。このコマンドにより新規に作成されたサーバプロファイルの ID が標準出力へ返されます。

構文

```
viomcli --key=<key> profile add --name=<string> -[+virt  
-[+boot[ --smux=<string>][ --comment=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--name=<string> または短縮形 **-n <string>**
サーバプロファイル名

-[+virt または短縮形 **-v**
仮想 IO の使用 (*-virt*: no、*+virt*: yes、*-v*: yes)

-[+boot または短縮形 **-b**
ブートメニューの使用 (*-boot*: no、*+boot*: yes、*-b*: yes)

--smux または短縮形 **-s<string>**
任意指定。
メザニンカードをスイッチファブリックに接続します。
ここでは、LAN 拡張ボードの経路が割り当てられるファブリックを指定します。

fabric3

すべてのパスはファブリック 3 へ経路が割り当てられます。

fabric4

標準

すべてのパスはファブリック 4 へ経路が割り当てられます。

fabric34

LAN1 はファブリック 3 へ、LAN2 はファブリック 4 へ経路が割り当てられます。

--comment=<string> または短縮形 **-c <string>**
任意指定
コメント

例 1

```
viomcli -k 4321 profile add -n test3 -v +boot -c "my comment"

17
0 0 OK
```

この例では、「17」が新規に作成されたサーバプロファイルの ID で、標準出力に書き込まれます。コマンドステータス情報（「0 0 OK」）は、標準エラーへ書き込まれます。エラーが発生すると（リターンコードが 0 でない）、ID は標準出力へ書き込まれません。

例 2

```
viomcli -k 1234 profile add -n Profile1 -v -s fabric3 -c "my
comment"

8
0 0 OK
```

この例では、両方のメザニンカードがファブリック 3 に接続されるサーバプロファイルが作成されます。

5.6.8.2 profile assign

サーバプロファイルを既存のノードに割り当てます。プロファイルはサーバスロットノードまたは **PRIMERGY** ラックサーバノードに割り当てられます。プロファイルをサーバスロットに割り当てる前に、**IO** チャンネルをプロファイルに追加する必要があります。

プロファイルをサーバスロットに割り当てる場合に、そのロットが空いていない場合、接続されているサーバノードは、電源がオフであり、そのプロファイルの指定の **IO** チャンネルをサポートしている必要があります。

マルチスロットプロファイルの場合、プロファイルをマルチスロット **CPU** ブレード (**BX960** など) の右上のスロットに割り当てる必要があります。

プロファイルを **PRIMERGY** ラックサーバに割り当てる場合、サーバは電源がオフであり、そのプロファイルの指定の **IO** チャンネルをサポートしている必要があります。

構文

```
viomcli --key=<key> profile assign --id=<integer>  
--node=<integer>[ --nodwnlinks][ --nopaths][ --nospare]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
サーバプロファイル ID

--node=<integer> または短縮形 **-n <integer>**
サーバスロットまたは **PRIMERGY** ラックサーバのノード ID

--nodwnlinks または短縮形 **-d**
任意指定
設定すると、ダウンリンクを無視します。

--nopaths または短縮形 **-p**
任意指定
設定すると、非 **IBP** ハードウェアのネットワークパスを無視します。

--nospare または短縮形 **-s**
任意指定
設定すると、予備スロット設定を無視します。

例

```
viomcli -k 4321 profile assign -i 6 -n 37 --nodwnlinks -p -s  
0 0 OK
```

5.6.8.3 profiles export

1 つまたは複数のサーバプロファイルを XML 形式でエクスポートします。デフォルトでは、出力は標準出力で、「> ファイル名」を使うと XML ファイルに切り替えることができます。

構文

```
viomcli --key=<key> --output=xml profiles export  
--id=<list>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--output=xml または短縮形 **-o xml**
このメソッドの出力モードを XML 出力に変更し、バイナリで解析できるようにします。

--id=<list> または短縮形 **-i <list>**
どのサーバプロファイルをエクスポートするか指定します。

<id1>[,<id2>,...]

サーバプロファイル ID。

コンマで区切って複数のサーバプロファイルを指定することもできます。

all

all を設定すると、すべてのサーバプロファイルをエクスポートします。

例

```
viomcli -k 4321 -o xml profiles export --id=1 > abc.xml  
0 0 OK
```

1 番目のサーバプロファイルが、XML ファイル *abc.xml* にエクスポートされます。

5.6.8.4 profile failover

サーバプロファイルのサーバ切り替えを開始します。

マルチスロットプロファイルの場合、プロファイルをマルチスロット CPU ブレード (BX960 など) の右上のスロットにサーバを切り替える必要があります。

構文

```
viomcli --key=<key> profile failover --node=<integer>  
--profile=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--node=<integer> または短縮形 **-n <integer>**
ノード ID

--profile=<integer> または短縮形 **-p <integer>**
サーバプロファイル ID

例

```
viomcli -k 4321 profile failover -n 135 -p 1
```

```
0 0 OK
```

5.6.8.5 profile get

サーバプロファイル情報を取得します。

構文

```
viomcli --key=<key> profile get --id=<integer>[ --comment]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
サーバプロファイル ID

--comment または短縮形 **-c**
任意指定
コメントのみを表示します。

例

```
viomcli -k 4321 profile get --id=7
```

Id	Profile Name	I/O Virt. Usage	Node Id	Boot M. Usage
7	test3	Yes		Yes

```
0 0 OK
```

5.6.8.6 profiles import

XML ファイルにエクスポートされたサーバプロファイルを編集し、**profiles import** を使って再びインポートすることができます。通常、入力は標準入力になります。XML ファイルからサーバプロファイルをインポートしたい場合は、標準入力を XML ファイルに切り替える必要があります。OS により、標準入力を切り替える方法は次のようになります。

- Windows の場合
コマンドの最後に << ファイル名 > を追加します。
- Linux の場合
コマンドの先頭に **cat <ファイル名>.xml |** を追加します。



サーバプロファイルをファイルからインポートする前に、XML エディタを使って、XML ファイルの構文を提供の XML schema と照合します。XML schema は、VIOM Web GUI をインストールする際インストールされます。

サーバプロファイルのインポートとエクスポート、およびエクスポートファイルの形式に関して、詳しくは **ServerView Virtual-IO Manager** のユーザーマニュアルの該当箇所をご覧ください。

構文

```
viomcli --key=<key> profiles import
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

例1 (Windows の場合)

ファイル **serverprofile1** からサーバプロファイルをインポートするコマンドは次のようになります。

```
viomcli -k 4321 profiles import < serverprofil1
```

```
0 0 OK
```

例2 (Linux の場合)

ファイル **serverprofile1** からサーバプロファイルをインポートするコマンドは次のようになります。

```
cat serverprofil1|viomcli -k 4321 profiles import  
  
0 0 OK
```

5.6.8.7 profile remove

既存のサーバプロファイルを削除します。サーバプロファイルが割り当てられている場合には、削除できません。

構文

```
viomcli --key=<key> profile remove --id=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
サーバプロファイル ID

例

```
viomcli -k 4321 profile remove --id=7  
  
0 0 OK
```

5.6.8.8 profile set

既存のサーバプロファイルの値を設定します。

構文

```
viomcli --key=<key> profile set --id=<integer>  
[ --name=<string>][ -|+virt ][ -|+boot][ --smux=<string>]  
[ --comment=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
サーバプロファイル ID

--name=<string> または短縮形 **-n <string>**
任意指定
サーバプロファイル名

-|+virt または短縮形 **-v**
任意指定
仮想 IO の使用 (**-virt**: no、**+virt**: yes、**-v**: yes)

-|+boot または短縮形 **-b**
任意指定
ブートメニューの使用 (**-boot**: no、**+boot**: yes、**-b**: yes)

--smux または短縮形 **-s<string>**
任意指定
メザニンカードをスイッチファブリックに接続します。
ここでは、LAN 拡張ボードの経路が割り当てられるファブリックを指定します。

fabric3

すべてのパスはファブリック 3 へ経路が割り当てられます。

fabric4

標準

すべてのパスはファブリック 4 へ経路が割り当てられます。

fabric34

LAN1 はファブリック 3 へ、LAN2 はファブリック 4 へ経路が割り当てられます。

--comment=<string> または短縮形 **-c <string>**
任意指定
コメント

例 1

```
viomcli -k 4321 profile set --id=7 +boot -c "my new comment"

0 0 OK
```

例 2

```
viomcli -k 1234 profile set --id=8 --smux=fabric34

0 0 OK
```

この例では、サーバプロファイル ID 8 ([97 ページの "例 2"](#) をご覧ください) の、メザニンカードからスイッチファブリックへの接続を変更します。変更後、LAN1 はファブリック 3 へ、LAN2 はファブリック 4 へ経路が割り当てられます。

5.6.8.9 profiles show

すべてのサーバプロファイルを表示します。

構文

```
viomcli --key=<key> profiles show
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

「profiles show」メソッドには、パラメータがありません。

例1

```
viomcli -k 4321 profiles show
```

Id	Profile Name	IO Virt. Usage	Node Id	Boot M. Usage	SMUX
1	2LAN_virt	Yes		Yes	fabric4
2	2LAN_virt_1	Yes		No	fabric4
3	2LAN_virt_2	Yes		No	fabric4
4	2LAN_virt_3	Yes		No	fabric4

```
0 0 OK
```

例2

[105 ページの "例2"](#) のサーバプロファイルの場合、出力は次のようになります。

```
viomcli -k 1234 profiles show
```

Id	Profile Name	IO Virt. Usage	Node Id	Boot M. Usage	SMUX
8	Profile_1	No		No	fabric34

```
0 0 OK
```

5.6.8.10 profile unassign

ノードへのサーバプロファイルの割り当てを解除します。プロファイルがサーバスロットに割り当てられている場合に、そのスロットが空いていない場合、プロファイルのスロットへの割り当てを解除できるように、接続されているサーバは、電源がオフである必要があります。プロファイルが **PRIMERGY** ラックサーバに割り当てられている場合、プロファイルの割り当てを解除するために、サーバの電源がオフである必要があります。

構文

```
viomcli --key=<key> profile unassign --id=<integer>  
--node=<integer>[ --force]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
サーバプロファイル ID

--node=<integer> または短縮形 **-n <integer>**
サーバスロットまたは **PRIMERGY** ラックサーバのノード ID

--force または短縮形 **-f**
任意指定
プロファイルの割り当てを強制解除します。

例

```
viomcli -k 4321 profile unassign -i 6 -n 37
```

```
0 0 OK
```

5.6.9 servicelan オブジェクト

サービス LAN を管理するメソッドを提供します。

5.6.9.1 servicelan add

新しいサービス LAN を作成します。このコマンドにより、新規に作成されたサービス LAN のネットワーク ID を標準出力へ返します。

構文

```
viomcli --key=<key> servicelan add --name=<string>  
--node=<integer> --uset=<integer> --vlan=<integer>  
[ +|-isol][ --comment=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--name=<string> または短縮形 **-n<string>**
ネットワーク名

--node=<integer> または短縮形 **-o<integer>**
スイッチスロットのノード ID

スイッチスタッキングの場合、ここにマスタースイッチのノード ID を指定します。スイッチスタッキングの詳細については、『Virtual-IO Manager 取扱説明書』を参照してください。

--uset=<integer> または短縮形 **-u<integer>**
アップリンクセット ID

--vlan=<integer> または短縮形 **-v<integer>**
VLAN ID

+|-isol または短縮形 **-j**
任意指定
サービス LAN ネットワークのダウンリンクアイソレーション（「-isol」：なし、「+isol」：あり、「-j」：あり）。デフォルトはダウンリンクアイソレーションなしです。

--comment=<string> または短縮形 **-c<string>**
任意指定
コメント

例

```
viomcli -k 4321 servicelan add -n slan_test -o 134 -u 13 -v 1
1
0 0 OK
```

5.6.9.2 servicelan get

サービス LAN に関する情報を取得します。

構文

```
viomcli --key=<key> servicelan get --id=<integer>[ --comment]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i<integer>**
サービス LAN のネットワーク ID

--comment=<string> または短縮形 **-c<string>**
任意指定
コメントのみを表示します。

例

```
viomcli -k 4321 servicelan get -i 1
```

Netw.Id	Netw.Name	Node Id	Uplink Set Id	Slots	VLAN Id	Isolation
1	slan_test	211	1		1	No

```
0 0 OK
```

5.6.9.3 servicelan remove

既存のサービス LAN を削除します。

構文

```
viomcli --key=<key> servicelan remove --id=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i<integer>**
サービス LAN のネットワーク ID

例

```
viomcli -k 4321 servicelan remove -i 15
```

```
0 0 OK
```

5.6.9.4 servecelan set

既存のサービス LAN の値を設定します。

構文

```
viomcli --key=<key> servicelan set --id=<integer>
[ --name=<string>][ --uset=<integer>][ --vlan=<integer>]
[ +|-isol][ --comment=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i<integer>**
サービス LAN のネットワーク ID

--name=<string> または短縮形 **-n<string>**
任意指定
ネットワーク名

--uset=<integer> または短縮形 **-u<integer>**
任意指定
アップリンクセット ID

--vlan=<integer> または短縮形 **-v<integer>**
任意指定
VLAN ID

+|-isol または短縮形 **-j**
任意指定
サービス LAN ネットワークのダウンリンクアイソレーション (「-isol」:
なし、「+isol」「-j」:あり、:あり)。

--comment=<string> または短縮形 **-c<string>**
任意指定
コメント

例

```
viomcli -k 4321 servicelan set -i 15 -n "slan_test_new" -u 13
0 0 OK
```

5.6.10 servicevlan オブジェクト

サービス VLAN を管理するメソッドを提供します。

5.6.10.1 servicevlan add

新しいサービス VLAN を作成します。このコマンドにより、新規に作成されたサービス VLAN のネットワーク ID を標準出力へ返します。

構文

```
viomcli --key=<key> servicevlan add --name=<string>  
--node=<integer> --uset=<integer> --vlan=<integer>  
[+|-isol][ --comment=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--name=<string> または短縮形 **-n<string>**
ネットワーク名

--node=<integer> または短縮形 **-o<integer>**
スイッチスロットのノード ID

スイッチスタッキングの場合、ここにマスタースイッチのノード ID を指定します。スイッチスタッキングの詳細については、『Virtual-IO Manager 取扱説明書』を参照してください。

--uset=<integer> または短縮形 **-u<integer>**
アップリンクセット ID

--vlan=<integer> または短縮形 **-v<integer>**
VLAN ID

+|-isol または短縮形 **-j**
任意指定
サービス VLAN ネットワークのダウンリンクアイソレーション（「-isol」：なし、「+isol」：あり、「-j」：あり）。

--comment=<string> または短縮形 **-c<string>**
任意指定
コメント

例

```
viomcli -k 4321 servicevlan add -n svlan_test -o 211 -u 1 -v
11
2
0 0 OK
```

5.6.10.2 servicevlan get

サービス VLAN に関する情報を取得します。

構文

```
viomcli --key=<key> servicevlan get --id=<integer>
[ --comment]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i<integer>**
サービス VLAN のネットワーク ID

--comment=<string> または短縮形 **-c<string>**
任意指定
コメントのみを表示します。

例

```
viomcli -k 4321 servicevlan get -i 2
```

Netw.Id	Netw.Name	Node Id	Uplink Set Id	Slots	VLAN Id	Isolation
2	svlan_test	211	1		11	No

```
0 0 OK
```

5.6.10.3 servicevlan remove

既存のサービス VLAN を削除します。

構文

```
viomcli --key=<key> servicevlan remove --id=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i<integer>**
サービス VLAN のネットワーク ID

例

```
viomcli -k 4321 servicevlan remove -i 15
```

```
0 0 OK
```

5.6.10.4 servicevlan set

既存のサービス VLAN の値を設定します。

構文

```
viomcli --key=<key> servicevlan set --id=<integer>
[ --name=<string>][ --uset=<integer>][ --vlan=<integer>]
[ +|-isol][ --comment=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**

要求でユーザを識別するキー

--id=<integer> または短縮形 **-i<integer>**

サービス VLAN のネットワーク ID

--name=<string> または短縮形 **-n<string>**

任意指定

ネットワーク名

--uset=<integer> または短縮形 **-u<integer>**

任意指定

アップリンクセット ID

--vlan=<integer> または短縮形 **-v<integer>**

任意指定

VLAN ID

+|-isol または短縮形 **-j**

任意指定

サービス VLAN ネットワークのダウンリンクアイソレーション

(「-isol」: なし、「+isol」: あり、「-j」: あり)。デフォルトはダウンリンクアイソレーションなしです。

--comment=<string> または短縮形 **-c<string>**

任意指定

コメント

例

```
viomcli -k 4321 servicevlan set -i 15 -n "svlan_test_new" -u
13
```

```
0 0 OK
```

5.6.11 session オブジェクト

ServerView Connector Service (SCS) セッションを管理するメソッドを提供します。

5.6.11.1 session close

SCS セッションを閉じます。通常このメソッドは必要ありませんが、セキュリティ上の理由から、SCS セッションをただちに中断する場合にこのメソッドを使います。

構文

```
viomcli --key=<key> session close
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

「session close」メソッドには、その他のパラメータはありません。

例

```
viomcli -k 4321 session close
```

```
0 0 OK
```

5.6.12 uplinkset オブジェクト

アップリンクセットを保守するメソッドを提供します。

5.6.12.1 uplinkset add

新しいアップリンクセットを作成します。このコマンドにより新規に作成されたアップリンクセットの ID を標準出力へ返します。

構文

```
viomcli --key=<key> uplinkset add --name=<string>
--nodeid=<integer>+|-fover +|-ldp +|-larp +|-igmp
--speed=<string>[ --dcbgroup=<list1>][ --dcbgroup=<list2>]
...
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--name=<string> または短縮形 **-n <string>**
アップリンクセット名

--nodeid=<integer> または短縮形 **-o <integer>**
スイッチスロットのノード ID。新しいアップリンクセットを作成する場合、IBP コネクションブレードをスイッチスロットに差し込む必要があります。

スイッチスタッキングの場合、ここにマスタースイッチのノード ID を指定します。スイッチスタッキングの詳細については、『Virtual-IO Manager 取扱説明書』を参照してください。

-|+fover または短縮形 **-f**
フェイルオーバーの使用 (*-fover*: no、*+fover*: yes、*-f*: yes)

-|+ldp または短縮形 **-l**
リンクダウン通知の使用 (*-ldp*: no、*+ldp*: yes、*-l*: yes)

-|+larp または短縮形 **-a**
リンク・アグリゲーション・コントロール・プロトコルの使用 (*-larp*:no、*+larp*:yes、*-a*:yes)

-|+igmp または短縮形 **-g**
IGMP スヌーピングの使用 (*-igmp*:no、*+igmp*:yes、*-g*:yes)

--speed=<string> または短縮形 **-s <string>**
リンクスピードは次のとおりです。

auto

オートネゴシエーション

10m

10 M ビット/秒

100m

100 M ビット/秒

--dcbgroup=<list> または短縮形 **-w <list>**
任意指定
DCB パラメータのカンマ区切りのリスト。

<weight>[,<purpose>][,<priority level>]

<weight>

帯域幅の予約パーセント（整数）。

入力できる値は 1 から 100 です。

<purpose>

オプション

fcoe

FCoE

iscsi

iSCSI

<priority level >

オプション

優先グループの優先レベル（整数）。

入力できる値は 1 から 7 です。デフォルトでは、FCoE の場合は 3、iSCSI の場合は 4 です。1 以上の優先レベルは、カンマで区切って指定できます。



「*--dcbgroup*」パラメータは、各優先グループの一度のコマンドラインに複数回表示されます。

例

```
viomcli -k 4321 uplinkset add -n test -o 29 +f +l -s auto
```

```
4
0 0 OK
```

この例では、「4」が新規に作成されたアップリンクセットの ID で、標準出力に書き込まれます。コマンドステータス情報（「0 0 OK」）は、標準エラーへ書き込まれます。エラーが発生すると（リターンコードが 0 でない）、ID は標準出力へ書き込まれません。

5.6.12.2 uplinkset get

アップリンクセット情報を取得します。

構文

```
viomcli --key=<key> uplinkset get --id=<integer>
[ --dcbgroups]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
アップリンクセット ID

--dcbgroups または短縮形 **-w**
任意指定
アップリンクセットに関連するすべての優先グループを表示します。

例 1

```
viomcli -k 4321 uplinkset get --id=10
```

```
Id   Set Name   Link Speed   Failover   Link Dwn. Prop.   Used by Netw.
|---|-----|-----|-----|-----|
10   NET_2      Auto        Yes        Yes                10

0 0 OK
```

例2

ここでの例では、アップリンクセットを **Ethernet** 経由でファイバチャネルに使用します。

```
viomcli -k 4321 uplinkset get --id=10 --dcbgroups
```

Priority Level	Weight	Purpose
[3]	50	FCoE

```
0 0 OK
```

5.6.12.3 uplinkset remove

既存のアップリンクセットを削除します。

構文

```
viomcli --key=<key> uplinkset remove --id=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
アップリンクセット ID

例

```
viomcli -k 4321 uplinkset remove --id=1
```

```
0 0 OK
```


5.6.12.4 uplinkset set

既存のアップリンクセットの値を設定します。

構文

```
viomcli --key=<key> uplinkset set --id=<integer>
[ --name=<string>][ --nodeid=<integer>][ +|-fover ][ +|-ldp]
[ -|+dcbuse][ --speed=<string>][ --dcbgroup=<list1>]
[ --dcbgroup=<list2>] ...
```

パラメータ

--key=<key> または短縮形 **-k <key>**

要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**

アップリンクセット ID

--name=<string> または短縮形 **-n <string>**

任意指定

アップリンクセット名

--nodeid=<integer> または短縮形 **-o <integer>**

任意指定

ノード ID

-|+fover または短縮形 **-f**

任意指定

フェイルオーバーの使用 (*-fover: no*、*+fover: yes*、*-f: yes*)

-|+ldp または短縮形 **-l**

任意指定

リンクダウン通知の使用 (*-ldp: no*、*+ldp: yes*、*-l: yes*)

-|+ dcbuse または短縮形 **-b**

任意指定

Data Center Bridging を有効または無効にします (「**-dcbuse**」: 無効、
「**+dcbuse**」: 有効、「**-b**」: 有効)



「**-dcbuse**」は、アップリンクセットのすべての優先グループを削除します。

--speed=<string> または短縮形 **-s <string>**

任意指定

リンクスピードは次のとおりです。

auto

オートネゴシエーション

10m

10 M ビット/秒

100m

100 M ビット/秒

--dcbgroup=<list> または短縮形 **-w <list>**

任意指定

DCB パラメータのカンマ区切りのリスト。

<weight>[,<purpose>][,<priority level>]

<weight>

帯域幅の予約パーセント（整数）。

入力できる値は 1 から 100 です。

<purpose>

オプション

fcoe

FCoE

iscsi

iSCSI

<priority level >

オプション

優先グループの優先レベル（整数）。

入力できる値は 1 から 7 です。デフォルトでは、FCoE の場合は 3、iSCSI の場合は 4 です。1 以上の優先レベルは、カンマで区切って指定できます。



「--dcbgroup」パラメータは、各優先グループの一度のコマンドラインに複数回表示されます。

既存の優先グループが削除されます。

例

```
viomcli -k 4321 uplinkset set --id=10 --name="NET_3"

0 0 OK
```

5.6.12.5 uplinksets show

ノードのアップリンクセットを表示します。

構文

```
viomcli --key=<key> uplinksets show --node=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
 要求でユーザを識別するキー

--node=<integer> または短縮形 **-n <integer>**
 スイッチスロットのノード ID

例

```
viomcli -k 4321 uplinksets show -n 29
```

Uplink Set Id	Set Name	Link Speed	Failover	Link Dwn. Prop.	Used by Netw.	LACP	IGMP Snooping	DCB
1	test	Auto	No	Yes	No	No	Yes	No

0 0 OK

5.6.13 vlannet オブジェクト

VLAN ネットワークを保守するメソッドを提供します。

5.6.13.1 vlannet add

新しい VLAN ネットワークを作成します。このコマンドにより新規に作成された VLAN ネットワークのネットワーク ID を標準出力へ返します。

構文

```
viomcli --key=<key> vlannet add --name=<string>  
--node=<integer>--uset=<integer> --vlan=<integer> +|-native  
[ +|-isol][ --comment=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--name=<string> または短縮形 **-n <string>**
ネットワーク名

--node=<integer> または短縮形 **-o <integer>**
スイッチスロットのノード ID

スイッチスタッキングの場合、ここにマスタースイッチのノード ID を指定します。スイッチスタッキングの詳細については、『Virtual-IO Manager 取扱説明書』を参照してください。

--uset=<integer> または短縮形 **-u <integer>**
アップリンクセット ID

--vlan=<integer> または短縮形 **-v <integer>**
VLAN ID

-|+native または短縮形 **-a**
ネイティブ (-native: no、+native: yes、-a: yes)

+|-isol または短縮形 **-j**
任意指定
VLAN ネットワークのダウンリンクアイソレーション (「-isol」: なし、「+isol」: あり、「-j」: あり)。デフォルトはダウンリンクアイソレーションなしです。

--comment=<string> または短縮形 **-c <string>**
任意指定
コメント

例

```
viomcli -k 4321 vlannet add -n vlan_test -o 134 -u 13 -v 1 +native -c vtest
```

```
8  
0 0 OK
```

この例では、「8」が新規に作成された VLAN ネットワークの ID で、標準出力に書き込まれます。コマンドステータス情報（「0 0 OK」）は、標準エラーへ書き込まれます。エラーが発生すると（リターンコードが 0 でない）、ID は標準出力へ書き込まれません。

5.6.13.2 vlannet get

VLAN ネットワークに関する情報を取得します。

構文

```
viomcli --key=<key> vlannet get --id=<integer>[ --comment]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
ネットワーク ID

--comment または短縮形 **-c**
任意指定
コメントのみを表示します。

例

```
viomcli -k 4321 vlannet get -i 15
```

Netw. Id	Netw. Name	Node Id	Uplink Set Id	Slots	VLAN Id	Native	Isolation
15	vlan_test	134	13		1	Yes	No

0 0 OK

5.6.13.3 vlannet remove

VLAN ネットワークを削除します。

構文

```
viomcli --key=<key> vlannet remove --id=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
ネットワーク ID

例

```
viomcli -k 4321 vlannet remove -i 15
```

```
0 0 OK
```

5.6.13.4 vlannet set

既存の VLAN ネットワークの値を設定します。

構文

```
viomcli --key=<key> vlannet set --id=<integer> --name=<string>  
--uset=<integer> -|+native[ --vlan=<integer>]  
[ +|-isol][ --comment=<string>]
```

パラメータ

--key=<key> または短縮形 **-k <key>**
要求でユーザを識別するキー

--id=<integer> または短縮形 **-i <integer>**
ネットワーク ID

--name=<string> または短縮形 **-n <string>**
ネットワーク名

--uset=<integer> または短縮形 **-u <integer>**
アップリンクセット ID

-|+native または短縮形 **-a**
ネイティブ (-native: no、+native: yes、-a: yes)

--vlan=<integer> または短縮形 **-v <integer>**
任意指定
VLAN ID

+|-isol または短縮形 **-j**
任意指定
VLAN ネットワークのダウンリンクアイソレーション (「-isol」: なし、
「+isol」: あり、「-j」: あり)。

--comment=<string> または短縮形 **-c <string>**
任意指定
コメント

例

```
viomcli -k 4321 vlannet set -i 15 -n "vlan_test_new" -native -u 13  
0 0 OK
```


5.6.13.5 vlannets show

ノードの VLAN ネットワークを表示します。

構文

```
viomcli --key=<key> vlannet show --node=<integer>
```

パラメータ

--key=<key> または短縮形 **-k <key>**
 要求でユーザを識別するキー

--node=<integer> または短縮形 **-o <integer>**
 スイッチスロットのノード ID

例

```
viomcli -k 4321 vlannets show -o 134
```

Netw. Id	Netw. Name	Node Id	Uplink Set Id	Slots	VLAN Id	Native	Isolation
15	vlan_test	134	13		1	Yes	No

0 0 OK

6 エラーとリターンコード

コマンドを実行後、必ず次のフォーマットでメッセージを受け取ります。

<major code> <minor code> <message string>

コマンドの実行が成功した場合、次のメッセージが表示されます。

0 0 OK

コマンドの実行エラーの場合には、メッセージ文字列にはエラーメッセージ文字列が含まれます。通常、この文字列は短いテキストメッセージです。

viomcli オプションの **--longtext** または **--system** を設定すれば、長いテキストまたはシステムメッセージを取得できます ([31 ページの「viomcli オプション」の項](#)をご覧ください)。

6.1 エラーコード

VIOM CLI エラーコードはメジャーコードとマイナーコードに分かれます。次のメジャーコードカテゴリーは VIOM CLI により定義されています。

メジャーコード	カテゴリー
< 1000	VIOM/SCS 固有のエラーメッセージ
1000	一般的なエラー
1100	CLI オプションエラー
1200	CLI コマンドエラー
1300	「オブジェクト値欠落」エラー
1400	「オブジェクト値不明」エラー

表 8: エラーコード

エラーコードとメッセージは **STDERR** へ出力されます。その他の出力はすべて **STDOUT** へ出力されます。

6.1.1 メジャーコード 1000

メジャーコード 1000 は一般エラーのカテゴリです。次のマイナーコードが定義されています。

マイナーコード	名前	内容
0	unknown error	このエラーの原因は不明。詳細はサーバログファイルを参照。
1	call failed	SCS 機能呼び出しに失敗。詳細はサーバログファイルを参照。
2	I/O error	SCS との通信が確立できなかった、または通信が切断された。
3	option value is missing	所定のオプションの値が指定されなかった。
4	type mismatch	入力されたオプション値のタイプが所定のものではない。
5	general parsing error	不明なオプションが入力された。
6	unparsable part	入力された文字の並びが構文解析できなかった。
7	syntax error	所定のオプションまたは値が指定されなかった。
8	access to file denied	ファイルへのアクセスが拒否された。
9	unsupported charset	指定した文字セットがサポートされていない。
10	unsupported language	指定した言語がサポートされていない。
11	open session failed	SCS へのセッションを開こうとしたが失敗した。

表 9: メジャーコード 1000

6.1.2 メジャーコード 1100

メジャーコード 1100 は CLI オプションエラーのカテゴリです。次のマイナーコードが定義されています。

マイナーコード	名前	内容
1	invalid output format	レンダリングに使用する出力フォーマットが指定されたが、サポートされていない。
2	incomplete access data	ユーザキーが入力されていない、不適正なキー入力された、または入力されたキーに対応するユーザプロファイルがない。
3	duplicate user key	入力されたユーザキーはすでに存在し、新しいプロファイルに使用することができない。

表 10: メジャーコード 1100

6.1.3 メジャーコード 1200

メジャーコード 1200 は、CLI コマンドエラーのカテゴリです。次のマイナーコードが定義されています。

マイナーコード	名前	内容
1	unknown logger level	指定されたログレベルが不明。

表 11: メジャーコード 1200

6.1.4 メジャーコード 1300

メジャーコード 1300 は VIOM オブジェクトとメソッドに関するエラーのカテゴリで、オブジェクト値が欠落していることを示します。次のマイナーコードが定義されています。

マイナーコード	名前	内容
1	missing node ID	ノード ID の指定が必要。
2	missing uplink set ID	アップリンクセット ID の指定が必要。
3	missing password	パスワードの指定が必要。
4	missing port	ポート（1 つ）の指定が必要。
5	missing ports	ポート（複数）の指定が必要。
6	missing protocol type	プロトコルタイプの指定が必要。
7	missing user name	ユーザ名の指定が必要。
8	missing profile name	サーバプロファイル名の指定が必要。
9	missing virtual usage	仮想化を使用するための値の指定が必要。
10	missing profile id	サーバプロファイル ID の指定が必要。
11	missing network name	ネットワーク名の指定が必要。
12	missing network id	ネットワーク ID の指定が必要。
13	missing VLAN id	VLAN ID の指定が必要。
14	missing vlan native	
15	missing uplink set name	アップリンクセット名の指定が必要。
16	missing failover usage	フェイルオーバーを使用するための値の指定が必要。
17	missing link down propagation	リンクダウン通知のための値の指定が必要。
18	missing link speed	リンクスピードの指定が必要。
19	missing channel type	チャネルタイプの指定が必要。
20	missing board number	ボード番号の指定が必要。
21	missing board type	ボードタイプの指定が必要。
22	missing port number	ポート番号の指定が必要。
23	missing power mode	パワーモードの指定が必要。
24	missing boot menu usage	ブートメニュー使用に関する指定が必要。
25	missing file name	ファイル名の指定が必要。

表 12: メッセージコード 1300

6.1.5 メジャーコード 1400

メジャーコード 1400 は VIOM オブジェクトとメソッドに関するエラーのカテゴリで、指定されたオブジェクト値が不明であることを示します。次のマイナーコードが定義されています。

マイナーコード	名前	内容
1	unknown speed value	指定されたスピード値が不明である。
2	unknown protocol value	指定されたプロトコル値が不明である。
3	unknown io channel type	指定された入出力チャンネルタイプが不明である。
4	unknown power mode	指定されたパワーモードが不明である。
5	unknown boot mode	指定されたブートモードが不明である。

表 13: メジャーコード 1400

6.2 リターンコード

viomcli バイナリのリターンコードは 0 から 255 までの範囲で、メジャーとマイナーエラーコードから計算されます。

メジャーコード	マイナーコード	リターンコード	内容
< 1000	all	1	すべてが VIOM/SCS 固有のエラーメッセージ
1000	0	21	不明エラー
1000	1	22	呼び出し失敗
1000	2	23	入出力エラー
1000	3	24	オプション値が欠落
1000	4	25	タイプミスマッチ
1000	5	26	一般的な構文解析エラー
1000	6	27	構文解析不可部分
1000	7	28	構文エラー
1000	8	29	ファイルへのアクセス拒否
1000	9	30	文字セットがサポートされていない
1000	10	31	言語がサポートされていない
1000	11	32	オープンセッション失敗
1100	all	2	すべてが CLI オプションエラー
1200	all	3	すべてが CLI コマンドエラー
1300	all	4	すべてが「オブジェクト値欠落」エラー
1400	all	5	すべてが「オブジェクト値不明」エラー

表 14: リターンコード

リターンコード 0 は、エラーが発生せず、実行が成功したことを意味します。

7 追加情報

この章では下記について追加情報を記載します。

- コンソールログとログファイル
- INI ファイル
- VIOMCLI 環境変数
- telnet を使用する
- クライアントサーバプロトコル

7.1 コンソールログとログファイル

コンソールログとログファイルのフォーマットは同じです。列はスペースで区切られ、次のように出力されます。

```
2010-02-08 16:31:27.562 4 1 pool created
2010-02-08 16:31:27.593 3 1 server ready
2010-02-08 16:31:35.250 4 1 locking pool object 30035636
2010-02-08 16:31:35.281 4 12 server thread started
2010-02-08 16:31:35.281 5 12 socket streams opened
2010-02-08 16:31:35.296 5 12 Command Line Parser: found 4 elements
2010-02-08 16:31:35.296 5 12 Command Line Parser: element index 1
2010-02-08 16:31:35.296 5 12 Command Line Parser: 2 matches found
```

列	内容
1	日付
2	時刻
3	ログレベル 1 重大 2 重要 3 通知 4 デバッグ 低 5 デバッグ 最高
4	スレッド ID サーバのメインスレッドの場合、スレッド ID は 1 になります。サーバはクライアントごとに新しいスレッドを開始します。同じスレッド ID をもつメッセージはどれも同一のクライアントセッションにより生成されます。
5	メッセージ

表 15: 列について

上記の例では、「server ready」メッセージがメインスレッドにより記録されます。スレッド ID が 12 の「socket streams opened」メッセージは、クライアントからの要求を処理するワーカースレッドにより記録されます。ID が 12 の後続メッセージはどれも同一のクライアントセッションにより生成されます。このため、複数のスクリプトを同時に実行している場合、各メッセージの関連性を正確に追跡できます。

ログファイルは、10 MB を超えるとローテートされます。このファイルサイズは既定です。古い（ローテートされる）ログファイルの名前は、<logfile name>.0 になります。この名前のファイルがすでに存在する場合には、そのファイルに上書きします。

7.2 INI ファイル

通常、viomcli INI ファイルは <userhome>/viomcli ディレクトリに保存され、ファイル名は viomcli.ini になります（<userhome> : Windows では環境変数 HOMEPATH、LINUX では環境変数 Home）。

デフォルトの内容は次のとおりです。

```
[general]
debug=0
params=

[server]
addr=127.0.0.1
port=51000
eof=01 0D 0A 0D 0A
```

値	内容
debug	0 はデバッグなし（デフォルト） 1 を指定すると、viomcli バイナリのデバッグ出力がアクティブになります。
params	すべての呼び出しで使用されるコマンドラインパラメータ（出力オプション、ユーザキーなど） 例 : params=--key=1234 または params=-k 1234
addr	VIOM CLI サーバの IP アドレス （デフォルト 127.0.0.1、ローカルホスト）
port	VIOM CLI サーバのポート（デフォルト 51000）

表 16: INI ファイルの内容

値	内容
eof	EOF シーケンス。16 進値（デフォルト 01 0D 0A 0D 0A）。このシーケンスはサーバから送信の終わりを示すために送られます。

表 16: INI ファイルの内容

7.3 環境変数

すべての呼び出しで使用するオプションの指定は、INI ファイルの **params** 値により行う方法に加え、環境変数 **VIOMCLI** によっても行うことができます。

この場合、**VIOMCLI** 環境変数が最初に評価されるので、INI 設定が無効になります。また、環境変数や INI ファイルにより追加されたパラメータをコマンドラインのパラメータにより無効にすることができます。

例

Windows では、次のように設定します。

```
set VIOMCLI=--key=1234
```

または

```
set VIOMCLI=-k 1234
```




図 1:	VIOM CLI アーキテクチャ	11
図 2:	VIOM CLI セキュリティ概要	13
図 3:	VIOM CLI インストールウィザードー開始ページ	17
図 4:	VIOM CLI インストールウィザードー使用許諾契約書	18
図 5:	VIOM CLI インストールウィザードーユーザ情報	19
図 6:	VIOM CLI インストールウィザードー インストール先フォルダ	20
図 7:	VIOM CLI インストールウィザードー インストールを開始する	21
図 8:	VIOM CLI インストールウィザードーインストール完了	22

