

# iLO 5

## スクリプティング/コマンドラインガイド

### NEC NX7700xシリーズ

1. 概要
2. SMASH CLP の使用方法
3. IPMI の使用方法
4. Secure Shell の使用方法
5. iLO RESTful API の使用方法

本書での内容は、iLO、SMASH CLP、IPMI、Secure Shell、RESTful API について十分に理解されている方を対象に説明しています。これらに関する操作や不明点については、ご使用になるソフトウェアのマニュアルやインターフェースの規格などを参照してください。

本書は NX7700x サーバーにおいてオープンソースソフトウェアの動作保証を行うものではありません。

本書の内容は、将来予告なしに変更されることがあります。製品およびサービスに対する保証については、当該製品およびサービスの保証規定書に記載されています。本書のいかなる内容も、新たな保証を追加するものではありません。本書の内容につきましては万全を期しておりますが、本書中の技術的あるいは校正上の誤り、脱落に対して、責任を負いかねますのでご了承ください。

本書で取り扱っているコンピューターソフトウェアは秘密情報であり、その保有、使用、または複製には、弊社から使用許諾を得る必要があります。FAR 12.211 および 12.212 に従って、商業用コンピューターソフトウェア、コンピューターソフトウェアドキュメンテーション、および商業用製品の技術データ (Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items) は、ベンダー標準の商業用使用許諾のもとで、米国政府に使用許諾が付与されます。

他社の Web サイトへのリンクは、弊社の Web サイトの外に移動します。弊社は、弊社の Web サイト以外にある情報を管理する権限を持たず、また責任を負いません。

## 商標

Intel® およびインテルは、インテルコーポレーションまたはその子会社のアメリカ合衆国およびその他の国における商標または登録商標です。

Java は、Oracle および/またはその関連会社の登録商標です。

Linux® は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft® および Windows® は、米国および/またはその他の国における Microsoft Corporation の登録商標または商標です。

Red Hat® 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。

本製品は、日本国内で使用するための仕様になっており、日本国外で使用される場合は、仕様の変更を必要とすることがあります。

本書に掲載されている製品情報には、日本国内で販売されていないものも含まれている場合があります。

本書の内容の一部または全部を無断転載することは禁止されています。

# 目次

1. 概要 .....	1
スクリプティング/コマンドラインガイドの概要 .....	1
スクリプティング/コマンドラインユーティリティー .....	1
SMASH CLP .....	2
IPMI .....	2
Secure Shell .....	2
iLO RESTful API .....	2
2. SMASH CLP の使用方法 .....	3
SMASH CLP .....	3
SMASH CLP コマンドラインアクセス .....	3
コマンドラインの使用 .....	3
エスケープコマンド .....	4
基本コマンド .....	5
NIC 自動選択機能の使用 .....	8
各コマンドについて .....	8
ユーザーコマンド .....	9
iLO ネットワーク設定 .....	9
iLO アクセス設定 .....	12
iLO 内蔵ヘルス設定 .....	14
SNMP 設定 .....	16
アラートメール設定 .....	17
リモート Syslog 設定 .....	19
ライセンスコマンド .....	19
ディレクトリコマンド .....	20
仮想メディアコマンド .....	21
電源投入およびリセットコマンド .....	24
ファームウェアコマンド .....	25
イベントログコマンド .....	27
起動コマンド .....	28
LED コマンド .....	33
システムプロパティおよびターゲット .....	33
その他のコマンド .....	37
3. IPMI の使用方法 .....	38
IPMI ユーティリティー .....	38
IPMI ツールの基本的な使用方法 .....	38
Linux 環境での IPMI ツールの高度な使用方法 .....	39
Windows 環境での IPMIutil の高度な使用方法 .....	39
4. Secure Shell の使用方法 .....	40
SSH の概要 .....	40
サポートされる SSH 機能 .....	40
Secure Shell の使用 .....	40
SSH キーの管理 .....	41
PuTTY からの SSH キーのインポート .....	41
ssh-keygen を使用して生成された SSH キーのインポート .....	43

5. iLO RESTful API の使用方法 .....	44
RESTful API の紹介 .....	44
Redfish 準拠 .....	44
HATEOAS を使用して設計された REST API .....	44
RESTful API の主な利点 .....	44
リソースの操作 .....	44
リターンコード .....	45
RESTful API の使用に関するヒント .....	45
スキーマ .....	45
スキーマバージョンと互換性 .....	46
RESTful API のエラーメッセージ .....	46
iLO RESTful API の操作例 .....	47
CURL を使用した例 .....	47
Postman を使用した例 .....	49
iLO RESTful API の詳細 .....	54
/redfish/v1/ .....	54
/redfish/v1/AccountService/ .....	55
/redfish/v1/AccountService/Accounts/ .....	55
/redfish/v1/AccountService/Accounts/{item}/ .....	57
/redfish/v1/Chassis/1/ .....	57
/redfish/v1/Chassis/1/Power/ .....	58
/redfish/v1/Chassis/1/Power/FastPowerMeter/ .....	59
/redfish/v1/Chassis/1/Power/PowerMeter/ .....	60
/redfish/v1/Chassis/1/Power/Thermal/ .....	60
/redfish/v1/Managers/1/ .....	61
/redfish/v1/Managers/1/EthernetInterfaces/1/ .....	62
/redfish/v1/Managers/1/EthernetInterfaces/2/ .....	63
/redfish/v1/Managers/1/NetworkService/ .....	65
/redfish/v1/Managers/1/NetworkService/Actions/Oem/Hpe/HpeiLOManagerNetworkService.SendTestAlertMail/ .....	65
/redfish/v1/Managers/1/NetworkService/NetworkService/Actions/Oem/Hpe/HpeiLOManagerNetworkService.SendTestSyslog/ .....	66
/redfish/v1/Managers/1/SnmpService/ .....	67
/redfish/v1/Managers/1/SnmpService/Actions/HpeiLOSnmService.SendSNMPTestAlert/ .....	67
/redfish/v1/Managers/1/VirtualMedia/1/ .....	68
/redfish/v1/Managers/1/VirtualMedia/1/Actions/Oem/Hpe/HpeiLOVirtualMedia.InsertVirtualMedia/ .....	68
/redfish/v1/Managers/1/VirtualMedia/1/Actions/Oem/Hpe/HpeiLOVirtualMedia.EjectVirtualMedia/ .....	69
/redfish/v1/Managers/1/VirtualMedia/2/ .....	70
/redfish/v1/Managers/1/VirtualMedia/2/Actions/Oem/Hpe/HpeiLOVirtualMedia.InsertVirtualMedia/ .....	70
/redfish/v1/Managers/1/VirtualMedia/2/Actions/Oem/Hpe/HpeiLOVirtualMedia.EjectVirtualMedia/ .....	71
/redfish/v1/Systems/1/ .....	72
/redfish/v1/Systems/1/LogServices/IML/ .....	74
/redfish/v1/Systems/1/LogServices/IML/Entries/ .....	74
/redfish/v1/Systems/1/LogServices/IML/Entries/{item}/ .....	75
/redfish/v1/Managers/1/LogServices/IEL/ .....	75
/redfish/v1/Managers/1/LogServices/IEL/Entries/ .....	76
/redfish/v1/Managers/1/LogServices/IEL/Entries/{item}/ .....	76
/redfish/v1/Systems/1/Memory/ .....	77
/redfish/v1/Systems/1/Memory/{item}/ .....	78
/redfish/v1/Systems/1/NetworkAdapters/ .....	79
/redfish/v1/Systems/1/NetworkAdapters/{item}/ .....	79
/redfish/v1/Systems/1/PCIDevices/ .....	80
/redfish/v1/Systems/1/PCIDevices/{item}/ .....	81
/redfish/v1/Systems/1/Processors/ .....	81
/redfish/v1/Systems/1/Processors/{item}/ .....	82
/redfish/v1/Systems/1/SmartStorage/ArrayControllers/ .....	83
/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/ .....	83
/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/LogicalDrives/ .....	84

/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/LogicalDrives/{item}/.....	84
/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/DiskDrives/.....	85
/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/DiskDrives/{item}/.....	85
/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/StorageEnclosures/.....	86
/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/StorageEnclosures/{item}/.....	87
/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/UnconfiguredDrives/.....	87
/redfish/v1/Systems/1/SmartStorage/HostBusAdapters/.....	88
/redfish/v1/Systems/1/SmartStorage/HostBusAdapters/{item}/.....	88
/redfish/v1/Systems/1/SmartStorage/HostBusAdapters/{item}/DiskDrives/.....	89
/redfish/v1/UpdateService/.....	89
/redfish/v1/UpdateService/FirmwareInventory/.....	90
/redfish/v1/UpdateService/FirmwareInventory/{item}/.....	90
/redfish/v1/Managers/1/Actions/Oem/Hpe/HpeiLO.ClearRestApiState/.....	91
/redfish/v1/Managers/1/Actions/Oem/Hpe/HpeiLO.ResetToFactoryDefaults/.....	92
<b>用語集 .....</b>	<b>93</b>



# 1. 概要

## スクリプティング/コマンドラインガイドの概要

iLO 5 は、NX7700x サーバーをリモートから設定、アップデート、および操作するための複数の方法を提供します。iLO 5 ユーザーズガイドには、各機能についての説明と、ブラウザーベースの Web インターフェースおよび BMC Configuration Utility による機能の使用方法が記載されています。

詳しくは、iLO 5 ユーザーズガイドを参照してください。

(NX7700x シリーズポータル : <https://jpn.nec.com/nx7700x/>)

この iLO 5 スクリプティング/コマンドラインガイドには、コマンドラインインターフェースまたはスクリプティングインターフェースを通じて iLO 5 を使用するために利用できる構文およびツールに関する説明が記載されています。

このマニュアルでは、iLO 5 を単に iLO と表現しています。

GUIに加えて、iLO ファームウェアは、スクリプトおよびコマンドライン手順を使用して iLO とサーバーを設定および制御する複数の方法を提供します。

## スクリプティング/コマンドラインユーティリティー

ここでは、以下のスクリプティング/コマンドラインツールについて説明します。

- SMASH CLP
- IPMI(Intelligent Platform Management Interface)
- Secure Shell
- iLO RESTful API

## SMASH CLP

SMASH CLP は、データセンターの管理を統一するために DMTF で定義された業界標準プロトコルです。SMASH CLP 仕様により、データセンター内の異機種サーバーの管理が単純で直感的になります。SMASH CLP は、マネジメントプロセッサ（管理アクセスポイントと呼ばれます）とホストシステムを設定および制御するための標準化された一連のコマンドを提供します。iLO では、シリアルポートまたは SSH 経由で SMASH CLP にアクセスすることができます。

## IPMI

IPMI 仕様は、コンピューターシステムに対する一連の共通インターフェースを定義する規格です。システム管理者は、IPMI を使用してシステムを監視し、管理することができます。IPMI 2.0 では、必須となるシステムインターフェースに加え、オプションとして LAN インターフェースを定義しています。iLO プロセッサは、両方のインターフェースをサポートしています。

IPMI 仕様は、プラットフォーム管理の標準化インターフェースを定義します。IPMI 仕様は、次の種類のプラットフォーム管理を定義します。

- ファン、温度、電源装置などのシステム情報のステータスの監視
- システムのリセットおよび電源オン/オフ操作などのリカバリ機能
- 温度上昇読み取りやファン障害などの異常なイベントのロギング機能
- 障害のあるハードウェアコンポーネントの特定などのインベントリ機能

IPMI コマンドは、IPMITOOL、IPMIUTIL、OpenIPMI、FreeIPMI などのサードパーティまたはオープンソースユーティリティを使用して iLO に送信されます。

raw コマンドを発行する場合は、IPMI 仕様をよく理解している必要があります。詳しくは、Intel の Web サイト <https://www.intel.com/content/www/us/en/servers/ipmi/ipmi-home.html> の IPMI 仕様を参照してください。

## Secure Shell

SSH は、リモートマシンにログインし、コマンドを実行するための Telnet によく似たプログラムです。ただし、SSH には、認証、暗号化、およびデータの整合性機能が含まれるため安全です。iLO フォームウェアは、5 台の SSH クライアントからの同時アクセスをサポートします。SSH の接続および認証後、ユーザーは、コマンドラインインターフェースを使用できます。

## iLO RESTful API

RESTful API は、サーバー管理のための最新のインターフェースです。RESTful API は、基本的な HTTP メソッド（GET、PUT、POST、DELETE、および PATCH）を使用して、JSON 形式のリソースを、iLO 上の URI との間で送受信します。スクリプティング言語を使用して、RESTful API 向けの単純な REST クライアントを容易に開発することができます。Python などのほとんどの言語では、JSON を辞書のような内部データ構造に変換できるため、データに簡単にアクセスできます。これにより、サーバー管理用のツールを使用せずに、直接 RESTful API を呼び出すコードを書くことができます。



## 2. SMASH CLP の使用方法

### SMASH CLP

DMTF SMASH は、データセンターの管理を統一するための業界標準プロトコルです。

SMASH CLP 仕様により、データセンター内の異機種サーバーの管理が単純で直感的になります。

SMASH CLP は、マネジメントプロセッサ（管理アクセスポイントと呼ばれます）とホストシステムを設定および制御するための標準化された一連のコマンドを提供します。iLO では、SSH ポート経由で SMASH CLP にアクセスします。

### SMASH CLP コマンドラインアクセス

iLO 5 ファームウェアの機能を使用すると、SMASH CLP コマンドラインからサポートされているコマンドを実行できます。コマンドラインオプションには、次のいずれかのインターフェースからアクセスします。

- シリアルポート
- SSH を使用するネットワーク

### コマンドラインの使用

iLO にシリアルポートまたは SSH からアクセスすると、iLO の CLI プロンプトが表示されます。コマンドを実行する（またはリモートコンソールや VSP を終了する）たびに、次の例のような CLI プロンプトに戻ります。

```
iLO->
```

CLI コマンドを実行するたびに、次の一般形式に従う出力が返されます。

```
</>iLO-> CLI command
```

```
status=0  
status_tag=COMMAND COMPLETED  
Wed Aug 2 01:02:03 2017
```

```
...output returned...
```

```
</>iLO->
```

無効なコマンドを入力すると、次のように、status および status\_tag 値がエラーを示します。

```
</>iLO-> boguscommand
```

```
status=2  
status_tag=COMMAND PROCESSING FAILED  
error_tag=COMMAND NOT RECOGNIZED  
Wed Aug 2 01:02:03 2017
```

有効なコマンドに無効なパラメーターが適用された場合は、次のように応答が少し異なります。

```
</>iLO-> show /bad  
status=2  
status_tag=COMMAND PROCESSING FAILED  
error_tag=INVALID OPTION  
Wed Aug 2 01:02:03 2017
```

ログインしたユーザーの権限レベルがそのコマンドの実行に必要な権限と照合されます。権限レベルが一致する場合のみ、コマンドが実行されます。シリアルコマンドラインインターフェースステータスが、[有効 - 認証は不要]に設定されている場合は、すべてのコマンドが、権限レベルの照合なしで実行されます。

CLP コマンドの一般構文は、次のとおりです。

<verb> <target> <option> <property>

- **verb** - サポートされるコマンドは、次のとおりです。
  - cd
  - create
  - delete
  - help
  - load
  - reset
  - set
  - show
  - start
  - stop
  - exit
  - version
- **target** - デフォルトのターゲットは、/です。cd コマンドを使用するか、コマンドラインでターゲットを指定することによってターゲットを変更します。
- **option** - 有効なオプションは、次のとおりです。
  - -all
  - -a
- **property** - ターゲットの変更可能な属性です。
- **Output** - 出力の構文は、次のとおりです。
  - status
  - status\_tag
  - status\_msg

任意のコマンドに対して有効な Boolean 値は、yes、no、true、false、y、n、t、f、1、および 0 です。

## エスケープコマンド

エスケープキーコマンドは、頻繁に使用するタスクのショートカットです。

ESC R ESC r ESC R

システムをリセットします。

ESC ^

システムの電源を入れます。

ESC ESC

現在の行を削除します。

エスケープシーケンス文字の入力では、1 秒間のタイムアウトが設定されています。

## 基本コマンド

ここでは、コマンドラインで使用する基本コマンドを示します。

help

コンテキスト依存ヘルプおよびサポートされるすべてのコマンドを表示します。

help *command*

そのコマンド固有のヘルプメッセージを表示します。

exit

CLP セッションを終了します。

cd

このコマンドは、現在のデフォルトターゲットを設定します。コンテキストは、ディレクトリパスと同様に機能します。サーバーのルートコンテキストはスラッシュ (/) であり、これは CLP システムの起点です。コンテキストを変更すると、コマンドが短くなります。たとえば、現在の iLO ファームウェアバージョンを調べるには、次のコマンドを入力します。

```
cd /map1/firmware1
```

```
show
```

show

このコマンドは、収集ターゲットのプロパティの値または内容を表示します。

例 :

```
</>iLO-> show
```

```
status=0
```

```
status_tag=COMMAND COMPLETED
```

```
Wed Aug 2 01:02:03 2017
```

```
/
```

```
Targets
```

```
system1
```

```
map1
```

```
Properties
```

```
Verbs
```

```
cd version exit show
```

コマンドによって返される情報の最初の行は、現在のコンテキストです。この例では、/が現在のコンテキストです。コンテキストの後に、現在のコンテキストに対応するサブターゲット (Targets) とプロパティ (Properties) のリストが表示されます。コマンド (Verbs) セクションには、このコンテキストに対応するコマンドが表示されます。

showコマンドには、明示的または非明示的コンテキストや特定のプロパティを指定します。たとえば明示的コンテキストは/map1/firmware1 であり、現在のコンテキストに依存しませんが、非明示的コンテキストは、指定されるコンテキストが現在のコンテキストの子コンテキストであることを前提としています。現在のコンテキストが/map1 である場合、show firmware コマンドは、/map1/firmware1 のデータを表示します。プロパティが指定されていない場合は、すべてのプロパティが表示されます。/map1/firmware1 コンテキストの場合、version と date という2つのプロパティを指定できます。show /map1/firmware1 date を実行すると、日付だけが表示されます。

create

名前空間に MAP の新しいインスタンスを作成します。

delete

名前空間にある MAP のインスタンスを削除します。

load

バイナリイメージを指定した URL から MAP に移動させます。

reset

ターゲットを有効から一度無効にして、有効に戻します。

set

1つまたは複数のプロパティを指定した値に設定します。プロパティによっては、設定を反映するために自動的に iLO がリセットされる場合があります。

start

ターゲットを開始します。たとえば、サーバーの電源オンやLED点灯などを行います。

stop

ターゲットを終了します。たとえば、サーバーの電源オフやLED消灯などを行います。

version

このコマンドは、CLP 実装のバージョンまたはその他の CLP 要素のクエリを実行します。

例：

```
</>iLO-> version

status=0
status_tag=COMMAND COMPLETED
Wed Aug 2 01:02:03 2017

SM-CLP Version 1.0
```

## oemNEC\_ping

このコマンドは、指定した IP アドレスが iLO から通信可能か確認します。

例：

```
</>iLO-> oemNEC_ping 192.168.1.1
```

ここで 192.168.1.1 は、テスト対象の IP アドレスです。

## oemNEC\_clearRESTAPIstate

このコマンドは、RESTful interface にキャッシュされたデータをクリアします。RESTful API でトラブルが発生した場合にのみ実行してください。このコマンドの実行には数秒かかる場合があります。コマンド実行後、サーバーの再起動を行ってください。

例：

```
</>iLO-> oemNEC_clearRESTAPIstate
```

## oemNEC\_nicautosel

このコマンドを使用すると、iLO は起動時に iLO 専用ネットワークポートまたは iLO 共有ネットワークポートのいずれかを自動的に選択することができます。ポート上でネットワークアクティビティを確認し、ネットワークの動作が確認できた最初のポートを選択して使用します。この設定の変更は、iLO がリセットされるまで反映されません。

### コマンドフォーマット

```
oemNEC_nicautosel {disabled | linkact | rcvdata | dhcp} [sbvlan=<0-4094>] [sbport=<1 or 2>] [sbport_limit=<0 or 2>] [delay=<90-1800>]
```

引数	機能
disabled	NIC 自動選択を無効にします。
linkact	いずれの動作検出に対しても NIC の自動選択を有効にします。
rcvdata	受信データパケットの動作検出に対して NIC の自動選択を有効にします。
dhcp	DHCP のアドレス割り当てに対して NIC の自動選択を有効にします。
sbvlan	共有 NIC ポートに使用される VLAN タグを指定します。値を 0 (ゼロ) にすると、VLAN タグは無効になります。
sbport	iLO と共有する共有 NIC のポートを指定します。ご使用のサーバーとネットワークアダプターが 1 より大きい値をサポートできることを確認します。
sbport_limit	専用 NIC に加えて NIC 自動選択機能によってテストされる共有 NIC ポートを指定します。0 または 2 でなければなりません。 0 = 現在設定されているポートのみをテストします。 2 = 自動モード、共有 NIC ポート 1 と 2 の両方をテストします 指定しない場合、sbport_limit はデフォルトで 0 になります。
delay	検索中に次の NIC に移動する前に、各 NIC の接続をテストする秒数を指定します。 指定されていない場合、デフォルトの遅延時間は 90 秒になります。

例：

```
</>iLO-> oemNEC_nicautosel rcvdata
```

このコマンドを使用すると、受信データパケットの動作検出に対する機能を有効にできます。

## NIC 自動選択機能の使用

iLO NIC 自動選択を使用すると、iLO が iLO 専用ネットワークポートと iLO 共有ネットワークポートを自動的に選択できるようになります。起動時に、iLO は使用可能なポートのネットワークアクティビティを検索し、使用するポートを自動的に選択します。本機能の詳細は、iLO5 ユーザーガイドを参照してください。本機能を有効にするには、以下の操作を行います。

1. 両方の iLO ネットワークポートを設定します。

NIC の自動選択機能を有効にして使用する前に、両方の iLO ネットワークポートをそれぞれのネットワーク環境に合わせて設定する必要があります。

2. CLI コマンド `oemNEC_nicautosel` を使用して、機能を有効にします。

DHCP アドレス割り当てが動作検出に使用されている場合 (`oemNEC_nicautosel dhcp`) は、少なくとも 1 つのポートで適切な DHCP オプションを使用して有効にしてください。IPv4 または IPv6 のいずれかまたは両方のアドレス割り当て方法が使用できます。IPv4 および IPv6 の DHCP が有効な場合、アドレスを割り当てるプロトコルはどちらも、DHCP アドレス割り当て検出を使用して検索中に成功と見なされます。

3. 必要に応じてサーバーのケーブルを配線してから、iLO をリセットします。

NIC 自動選択への変更は、iLO がリセットされるまで反映されません。

## 各コマンドについて

以下の項では、コマンドラインで使用できる iLO の特殊コマンドについて説明します。以下に掲載している情報は、iLO ファームウェアのアップデートで予告なく変更される場合があります。また、機種によって設定可能な値が異なる場合があります。ご使用になる iLO の SMASH CLP 上で、`help` コマンドを使用して各コマンドの詳細をご確認ください。

- ユーザーコマンド
- iLO ネットワーク設定
- iLO アクセス設定
- iLO 内蔵ヘルス設定
- SNMP の設定
- アラートメール設定
- リモート Syslog 設定
- ライセンスコマンド
- ディレクトリコマンド
- 仮想メディアコマンド
- 電源投入およびリセットコマンド
- ファームウェアコマンド
- イベントログコマンド
- 起動コマンド

- LED コマンド
- システムプロパティおよびターゲット
- その他のコマンド

## ユーザーコマンド

ユーザーコマンドを使用すると、ユーザー設定を表示したり変更したりすることができます。表 1 に、ユーザーコマンドのプロパティを示します。ユーザー設定は、次の場所で確認できます。

```
/map1/accounts1
```

### ターゲット

すべてのローカルユーザーは、有効なターゲットです。たとえば、3つのローカルユーザーが Administrator、admin、および test というログイン名を持つ場合、有効なターゲットは、次のとおりです。

- Administrator
- admin
- test

### プロパティ

表 1 ユーザーコマンドのプロパティ

プロパティ	アクセス	説明
username	読み出し/書き込み	iLO 5 のログイン名に対応します。
password	読み出し/書き込み	現在のユーザーのパスワードに対応します。
name	読み出し/書き込み	ユーザーの名前を表示します。名前が指定されていない場合、パラメーターにはログイン名 (username) と同じ値が使用されます。この値は、iLO 5 ユーザー名プロパティに対応します。
group	読み出し/書き込み	権限レベルを指定します。有効な値は、次のとおりです。 <ul style="list-style-type: none"> <li>• admin</li> <li>• config</li> <li>• oemNEC_power</li> <li>• oemNEC_rc</li> <li>• oemNEC_vm</li> </ul> グループが指定されていない場合は、ユーザーに権限が割り当てられません。

例：

現在のパス：

```
/map1/accounts1
```

- `create username=lname1 password=password`

この例では、username がログイン名に対応します。

- `create /map1/accounts1 username=<lname1> password=<pwd12345> name=<lname1> group=<admin,config,oemNEC_vm,oemNEC_rc,oemNEC_power>`

この例では、lname1 がユーザーのログイン名です。

## iLO ネットワーク設定

ネットワークサブシステムは、次の場所で確認できます。

- /map1/enetport1
- /map1/dhccpendpt1
- /map1/dnsendpt1
- /map1/gateway1
- /map1/dnsserver1
- /map1/dnsserver2
- /map1/dnsserver3
- /map1/settings1
- /map1/vlan1

基本コマンドの oemNEC\_nicautosel も参照してください。

## ターゲット、プロパティ

**表 2 /map1/enetport1**

ターゲット	プロパティ
-	<ul style="list-style-type: none"> <li>• Autosense</li> <li>• PermanentAddress</li> <li>• LinkTechnology</li> <li>• Speed</li> <li>• SystemName</li> <li>• Fullduplex</li> </ul>
lanendpt1	<ul style="list-style-type: none"> <li>• EnabledState</li> <li>• MACAddress</li> <li>• LOMAndFlexibleLOMPortNumber</li> </ul>

例 :

```
set /map1/enetport1 Speed=100
```

```
set /map1/enetport1/lanendpt1/ipendpt1 IPv4Address=15.255.102.245
SubnetMask=255.255.248.0
```

**表 3 /map1/dhccpendpt1**

ターゲット	プロパティ
-	<ul style="list-style-type: none"> <li>• EnabledState</li> <li>• OtherTypeDescription</li> </ul>

**表 4 /map1/dnsendpt1**

ターゲット	プロパティ
-	<ul style="list-style-type: none"> <li>• EnabledState</li> <li>• HostName</li> <li>• DomainName</li> <li>• OtherTypeDescription</li> </ul>

**表 5 /map1/gateway1**

ターゲット	プロパティ
-------	-------



-	<ul style="list-style-type: none"> <li>• AccessInfo</li> <li>• AccessContext</li> <li>• DHCPOptionToUse</li> </ul>
---	--

**表 6 /map1/dnsserver1、dnsserver2、dnsserver3**

ターゲット	プロパティ
-	<ul style="list-style-type: none"> <li>• AccessInfo</li> <li>• AccessContext</li> </ul>

**表 7 /map1/settings1**

ターゲット	プロパティ
-	なし
DNSSettings1	<ul style="list-style-type: none"> <li>• DNSServerAddress</li> <li>• RegisterThisConnection</li> <li>• DomainName</li> <li>• DHCPOptionToUse</li> <li>• oemNEC_UseDHCPv4DomainName</li> </ul>
WINSSettingData1	<ul style="list-style-type: none"> <li>• WINSServerAddress</li> <li>• RegisterThisConnection</li> <li>• DHCPOptionToUse</li> </ul>
StaticIPSettings1	<ul style="list-style-type: none"> <li>• oemNEC_SRoutelAddress</li> <li>• oemNEC_Mask1Address</li> <li>• oemNEC_Gateway1Address</li> <li>• oemNEC_SRoutel2Address</li> <li>• oemNEC_Mask2Address</li> <li>• oemNEC_Gateway2Address</li> <li>• oemNEC_SRoutel3Address</li> <li>• oemNEC_Mask3Address</li> <li>• oemNEC_Gateway3Address</li> <li>• DHCPOptionToUse</li> </ul>

**表 8 /map1/vlan1**

ターゲット	プロパティ
-	<ul style="list-style-type: none"> <li>• EnabledState</li> <li>• VLANID</li> </ul>

## コマンド

ネットワーク設定は次のコマンドと親和性があります。

- cd
- version
- exit
- show
- set

1つまたは複数のプロパティをコマンドラインに指定します。複数のプロパティを同じコマンドラインで指定する場合は、スペースで区切る必要があります。

ネットワーク設定が適用されると、iLO ファームウェアはリセットされます。

たとえば次のコマンドは、iLO ネットワークポートを、サーバーマザーボード上の共有ネットワークポート NIC に設定します。この NIC は、iLO ドキュメントの他の場所では LOM と呼ばれます。このハードウェアは、一部のサーバーに搭載されています。

```
Set /map1/enetport1/lanendpt1 EnabledState=32768
```

次の例では、iLO ネットワークポートを、サーバーマザーボードの FlexibleLOM スロットに取り付けることができるオプションの共有ネットワークポート NIC に設定します。このスロットは標準の PCIe スロットではなく、マザーボード上の特殊なコネクタです。

このスロットとこのスロット用 NIC は、一部のサーバーに搭載されています。

```
Set /map1/enetport1/lanendpt1 EnabledState=32769
```

## iLO アクセス設定

iLO 設定コマンドを使用すると、iLO 設定の表示や修正ができます。表 9 に iLO 設定のプロパティを示します。iLO 設定は、次の場所で確認できます。

```
/map1/config1
```

### ターゲット

ターゲットは、ありません。

### プロパティ

表 9 iLO 設定のプロパティ

プロパティ	アクセス	説明
oemNEC_mapenable	読み出し/書き込み	iLO を有効または無効にします。Boolean 値を使用できます。
oemNEC_timeout	読み出し/書き込み	iLO Web インターフェースセッションまたはリモートコンソールセッションのアイドルタイムアウト設定（分単位）を表示または変更します。有効な値は、15、30、60、120 です。
oemNEC_rbsuenable	読み出し/書き込み	POST 実行中の BMC 構成ユーティリティーを有効または無効にします。Boolean 値を使用できます。
oemNEC_rbsulogin	読み出し/書き込み	ユーザーが BMC 構成ユーティリティーにアクセスしたときにユーザー認証情報プロンプトを表示するかどうかを指定します。Boolean 値を使用できます。
oemNEC_rbsushowip	読み出し/書き込み	POST 実行中の iLO IP アドレスの表示を有効または無効にします。Boolean 値を使用できます。
oemNEC_vsp_log_enable	読み出し/書き込み	サーバーからの仮想シリアルポートのログ記録を有効または無効にします。
oemNEC_httpport	読み出し/書き込み	HTTP ポートの値を設定します。
oemNEC_sslport	読み出し/書き込み	SSL ポートの値を設定します。
oemNEC_rcport	読み出し/書き込み	リモートコンソールポートの値を設定します。
oemNEC_vmpport	読み出し/書き込み	仮想メディアポートの値を設定します。
oemNEC_sshport	読み出し/書き込み	SSH ポートの値を設定します。
oemNEC_sshstatus	読み出し/書き込み	SSH を有効または無効にします。Boolean 値を使用できます。

oemNEC_serialcliststatus	読み出し/書き込み	CLIのステータス設定を表示または変更します。Boolean値を使用できます。
oemNEC_serialcliath	読み出し/書き込み	シリアルポートのCLIの認証ステータス設定を表示または変更します。Boolean値を使用できます。
oemNEC_serialclispeed	読み出し/書き込み	シリアルポートのCLIの速度設定を表示または変更します。
oemNEC_minpwdlen	読み出し/書き込み	最小パスワード長設定を表示または変更します。
oemNEC_enforce_aes	読み出し/書き込み	AES暗号化構成を表示または変更します。Boolean値を使用します。
oemNEC_authfailurelogging	読み出し/書き込み	認証失敗ログの設定を表示または変更します。
oemNEC_computer_lock	読み出し/書き込み	リモートコンソールコンピューターロック設定を表示または変更します。
oemNEC_hotkey_ctrl_t、 oemNEC_hotkey_ctrl_u、 oemNEC_hotkey_ctrl_v、 oemNEC_hotkey_ctrl_w、 oemNEC_hotkey_ctrl_x、 oemNEC_hotkey_ctrl_y	読み出し/書き込み	リモートコンソールのホットキー設定を表示または変更します。
oemNEC_irc_trust_enable	読み出し/書き込み	リモートコンソールのホットキー設定を表示または変更します。
oemNEC_snmp_access	読み出し/書き込み	iLO が外部の SNMP 要求に応答するかどうかを表示または変更します。
oemNEC_snmp_port	読み出し/書き込み	SNMPポートの設定を表示または変更します。
oemNEC_snmp_trapport	読み出し/書き込み	SNMPトラップのポート設定を表示または変更します。
oemNEC_fips_enable	読み出し/書き込み	FIPSモード設定を表示または変更します。
oemNEC_ipmi_dcmi_overlan_enable	読み出し/書き込み	IPMIアクセス設定を表示または変更します。
oemNEC_ipmi_dcmi_overlan_port	読み出し/書き込み	IPMIポート設定を表示または変更します。
oemNEC_webserver_status	読み出し/書き込み	Webサーバー設定の表示または変更します。
oemNEC_webgui_status	読み出し/書き込み	Web GUI設定の表示または変更します。
oemNEC_virtual_media_status	読み出し/書き込み	Virtual Media設定の表示または変更します。
oemNEC_remote_console_status	読み出し/書き込み	リモートコンソール設定の表示または変更します。

## コマンド

- cd
- version
- exit
- show
- set
- oemNEC\_loadSSHkey
- oemNEC\_resetHotkeys
- vsp

例：

```
set /map1/config1 oemNEC_mapenable=yes oemNEC_timeout=30
```

1つまたは複数のプロパティをコマンドラインに指定します。複数のプロパティを同じコマンドラインで指定する場合は、スペースで区切る必要があります。

例 :

```
set /map1/config1 oemNEC_computer_lock=windows
set /map1/config1 oemNEC_computer_lock=custom,l_gui,l
set /map1/config1 oemNEC_computer_lock=disabled
```

oemNEC\_computer\_lock で指定可能なキーのリストは、iLO 5 ユーザーズガイドを参照してください。スペースを含むキーの場合、スペースをアンダースコアに置き換える必要があります。

例 :

```
set /map1/config1 oemNEC_computer_lock=custom,SYS_RQ
```

## iLO 内蔵ヘルス設定

iLO 内蔵ヘルスコマンドを使用すると、ファン、温度センサー、電圧センサー、および電源装置のシステム内蔵ヘルス情報を表示することができます。表 10 に、iLO の内蔵ヘルスプロパティを示します。iLO 内蔵ヘルス CLP 設定は、次のとおりです。

- /system1/fan\*
- /system1/sensor\*
- /system1/powersupply\*

### ターゲット

- Fan
- Sensor
- Powersupply
- firmware1
- bootconfig1
- log1
- led1
- network1
- oemNEC\_vsp1
- oemNEC\_power1
- cpu\*
- memory\*
- slot\*
- swid\*

### プロパティ

表 10 内蔵ヘルスのプロパティ

プロパティ	アクセス	説明
-------	------	----

DeviceID	読み出し	ファン、センサー、または電源装置のラベル番号を表示します。
ElementName	読み出し	ファン、センサー、または電源装置の位置を表示します。
OperationalStatus	読み出し	ファン、センサー、または電源装置の動作ステータスを表示します。
VariableSpeed	読み出し	ファンが可変速度で動作しているかどうかを表示します。
DesiredSpeed	読み出し	現在のファン速度を表示します。
HealthState	読み出し	ファン、センサー、または電源装置のヘルスステータスを表示します。
RateUnits	読み出し	温度および電圧センサーの読み出し単位を表示します。
CurrentReading	読み出し	センサーの現在の読み出し値を表示します。
SensorType	読み出し	センサーの種類を表示します。
OemNEC_CautionValue	読み出し	温度センサー警告値を表示します。
OemNEC_CriticalValue	読み出し	温度センサーのクリティカル値を表示します。

注記: 表 10 に、すべてのターゲットから利用可能なすべての内蔵ヘルスプロパティを示します。実際に返されるプロパティは、コマンドによって異なります。

例: 次のコマンドは、システム fan1 のプロパティを表示します。

```
show /system1/fan1
```

例:

```
/system1/fan1
Targets
Properties
  DeviceID=Fan 1
  ElementName=System
  OperationalStatus=Ok
  VariableSpeed=Yes
  DesiredSpeed=40 percent
  HealthState=Ok
```

次のコマンドは、温度ゾーンプロパティの 1 つを表示します。

```
show /system1/sensor1
```

例:

```
/system1/sensor3
Targets
Properties
  DeviceID=01-Inlet Ambient
  ElementName=Ambient
  OperationalStatus=Ok
  RateUnits=Celsius
  CurrentReading=31
  SensorType=Temperature
  HealthState=Ok
  oemNEC_CautionValue=42
```

oemNEC\_CriticalValue=47

## SNMP 設定

SNMP 設定コマンドを使用すると、SNMP 設定を表示したり変更したりすることができます。表 11 に、SNMP コマンドのプロパティを示します。SNMP 設定は、次の場所で確認できます。

/map1/snmp1

### ターゲット

なし

### プロパティ

表 11 SNMP コマンドのプロパティ

プロパティ	アクセス	説明
accessinfo<n>	読み出し/書き込み	SNMP トラップ宛先アドレスを設定します。<n>は、1、2、3 です。
readcom<n>	読み出し/書き込み	SNMP 読み取りコミュニティアドレスを表示または変更します。<n>は、1、2、3 です。
trapcom<n>	読み出し/書き込み	SNMP トラップコミュニティアドレスを表示または変更します。<n>は、1、2、3 です。
oemNEC_ioloadert	読み出し/書き込み	iLO SNMP アラートを有効または無効にします。Boolean 値を使用できます。
oemNEC_systemlocation	読み出し/書き込み	サーバーの物理的位置を指定する文字列を表示または変更します。
oemNEC_systemcontact	読み出し/書き込み	システム管理者またはサーバーの所有者を指定する文字列を表示または変更します。
oemNEC_systemrole	読み出し/書き込み	サーバーの役割または機能を記述する文字列を表示または変更します。
oemNEC_systemroledetail<n>	読み出し/書き込み	サーバーが実行する可能性がある具体的なタスクを記述する文字列を表示または変更します。ここで、<n>は、1~5 です。
oemNEC_coldstarttrap	読み出し/書き込み	SNMP コールドスタートトラップブロードキャストの設定を表示または変更します。Boolean 値を使用できます。
oemNEC_trapsource	読み出し/書き込み	SNMP トラップソース設定を表示または変更します。

## コマンド

- cd
- version
- exit
- show
- set
- testtrap

例：

次のコマンドは、様々な SNMP のプロパティを設定します。

- `set /map1/snmp1 accessinfol=192.168.0.50 oemNEC_imdatalevel=Enabled`
- `set readcom1="public1"`
- `set trapcom1="trapcomm1"`
- `set oemNEC_systemlocation="NEC Data Center"`
- `set oemNEC_systemcontact="System Administrator"`
- `set oemNEC_systemrole="Brief Role Description [60 characters]"`
- `set oemNEC_systemroledetail1="Extended Role Description [100 characters]"`
- `oemNEC_systemroledetail1` に対して複数行を表示します。
  - `set oemNEC_systemroledetail1="Extended Role Description"`
  - `show - 100 文字の制限を満たすために、detail1 の既存の文字列にスペースが追加されています。`

1 つまたは複数のプロパティをコマンドラインに指定します。複数のプロパティを同じコマンドラインで指定する場合は、スペースで区切る必要があります。

## アラートメール設定

アラートメール設定コマンドを使用すると、アラートメール設定を表示したり変更したりすることができます。本機能を使用するためには、iLO ライセンスキーが必要です。表 12 に、アラートメールコマンドのプロパティを示します。アラートメール設定は、次の場所で確認できます。

```
/map1/oemNEC_alertmail1
```

### ターゲット

なし

### プロパティ

表 12 アラートメールコマンドのプロパティ

プロパティ	アクセス	説明
<code>oemNEC_alertmail_enable</code>	読み出し/書き込み	アラートメール機能の状態を表示または変更します。有効な値は <code>yes</code> または <code>no</code> です。
<code>oemNEC_alertmail_email</code>	読み出し/書き込み	メールアラートの送信先電子メールアドレスを表示または変更します。
<code>oemNEC_alertmail_sender_domain</code>	読み出し/書き込み	送信元 Email アドレスに設定されるドメイン名を表示または変更します。
<code>oemNEC_alertmail_smtp_server</code>	読み出し/書き込み	SMTP サーバーまたは MSA の IP アドレスまたは DNS 名を表示または変更します。
<code>oemNEC_alertmail_smtp_port</code>	読み出し/書き込み	SMTP サーバーが使用するポートを表示または変更します。

oemNEC_alertmail_smtp_auth_enable	読み出し/書き込み	SMTP 認証の有効/無効状態を表示または変更します。有効な値は yes または no です。
oemNEC_alertmail_smtp_auth_username	読み出し/書き込み	SMTP 認証に使用するユーザー名を表示または変更します。
oemNEC_alertmail_smtp_auth_password	読み出し/書き込み	SMTP 認証に使用するパスワード名を表示または変更します。

## コマンド

- cd
- version
- exit
- show
- set
- oemNEC\_sendTestAlertmail

例：

次のコマンドは、様々なアラートメールのプロパティを設定します。

- set /map1/oemNEC\_alertmail1 oemNEC\_alertmail\_smtp\_port=25
- set oemNEC\_alertmail\_smtp\_server=172.16.2.1
- set oemNEC\_alertmail\_email=" mailreceiver@example.com"
- set oemNEC\_alertmail\_sender\_domain="example.com"
- set oemNEC\_alertmail\_enable=yes

次のコマンドは、テスト通報を行います。

```
sendTestAlertmail
```

## アラートメールのサンプル

```
Subject: NEC iLO AlertMail-280: (CAUTION) System Fan Removed (Fan 4, Location System)
From: iLO hostname <hostname@example.com>
To: mailreceiver@example.com
-----
EVENT (15-Aug-2017 00:46): System Fan Removed (Fan 4, Location System)

Integrated Management Log Severity:CAUTION

iLO URL: https://hostname.example.com
iLO IP: https://172.16.0.1
iLO Name: hostname
iLO firmware: 1.10 Jun 07 2017

Server Model: NX7700x/A5010E-2
System ROM: U30 06/14/2017 Server UUID: 01234567-89AB-CDEF-0123-4367890ABCDE

PLEASE DO NOT REPLY TO THIS EMAIL. For more details about NEC iLO technology, visit: jpn.nec.com/nx7700x/
```



## リモート Syslog 設定

リモート Syslog 設定コマンドを使用すると、リモート Syslog 設定を表示したり変更したりすることができます。本機能を使用するためには、iLO ライセンスキーが必要です。表 13 に、リモート Syslog コマンドのプロパティを示します。リモート Syslog 設定は、次の場所で確認できます。

```
/map1/oemNEC_syslog1
```

### ターゲット

なし

### プロパティ

表 13 リモート Syslog コマンドのプロパティ

プロパティ	アクセス	説明
oemNEC_syslog_enable	読み出し/書き込み	リモート Syslog 機能の状態を表示または変更します。有効な値は yes または no です。
oemNEC_syslog_port	読み出し/書き込み	リモート Syslog を受信するポート番号を表示または変更します。
oemNEC_syslog_serveraddress	読み出し/書き込み	リモート Syslog サーバーのアドレスを表示または変更します。最大 127 文字です。

### コマンド

- cd
- version
- exit
- show
- set
- oemNEC\_sendTestSyslog

例：

次のコマンドは、様々なアラートメールのプロパティを設定します。

- set /map1/oemNEC\_syslog1 oemNEC\_syslog\_serveraddress=172.16.2.1
- set oemNEC\_syslog\_enable=yes

次のコマンドは、テスト通報を行います。

```
oemNEC_sendTestSyslog
```

## ライセンスコマンド

ライセンスコマンドを使用すると、iLO ライセンスを設定することができます。表 14 に、ライセンスコマンドのプロパティを示します。ライセンスコマンドは、次の場所で確認できます。

```
/map1/
```

### ターゲット

なし

## コマンド

表 14 ライセンスコマンド

コマンド	説明
cd	現在のディレクトリを変更します。
show	ライセンス情報を表示します。
set	現在のライセンスを変更します。

例：

```
set /map1 license=1234500000678910000000001
show /map1 license
```

## ディレクトリコマンド

ディレクトリコマンドを使用すると、ディレクトリ設定を表示したり変更したりすることができます。本機能を使用するためには、iLO ライセンスキーが必要です。ディレクトリコマンド設定は、次の場所で確認できます。

```
/map1/oemNEC_dircfg1
```

### ターゲット

表 15 に、ディレクトリコマンドのターゲットを示します。

表 15 ディレクトリコマンドのターゲット

ターゲット	説明
/map1/oemNEC_dircfg1/oemNEC_keytab1	このターゲットでは、指定されたURLからバイナリキータブファイルをロードするために、loadコマンドを使用できます。

### プロパティ

表 16 ディレクトリコマンドのプロパティ

プロパティ	アクセス	説明
oemNEC_dirauth	読み出し/書き込み	ディレクトリ認証を有効または無効にします。有効な値は、次のとおりです。 <ul style="list-style-type: none"><li>default_schema スキーマフリーディレクトリを使用します。</li><li>disabled ディレクトリベース認証を無効にします。</li></ul>
oemNEC_localacct	読み出し/書き込み	ローカルアカウント認証を有効または無効にします。このプロパティは、ディレクトリ認証が有効になっているときにのみ無効にできます。Boolean 値を使用できます。
oemNEC_dirsrvaddr	読み出し/書き込み	ディレクトリサーバーアドレスを設定します。DNS 名での指定を推奨します。
oemNEC_ldapport	読み出し/書き込み	ディレクトリサーバーのポートを設定します。

oemNEC_dirdn	読み出し/書き込み	スキーマフリーディレクトリ設定が使用されている場合、このフィールドは無視します。
oemNEC_usercntxt<n> <n>は、1~15 です。	読み出し/書き込み	ディレクトリユーザーコンテキストを指定します。
oemNEC_group<n>_name <n>は、1~6 です。	読み出し/書き込み	セキュリティグループの識別名を設定します。
oemNEC_group<n>_priv <n>は、1~6 です。	読み出し/書き込み	グループに関連付けられる権限を設定します。 次の値が有効です。  <ul style="list-style-type: none"> <li>• login</li> <li>• oemNEC_rc</li> <li>• oemNEC_power</li> <li>• oemNEC_vm</li> <li>• config</li> <li>• admin</li> </ul>
oemNEC_dir_kerberos_enabled	読み出し/書き込み	Kerberos 認証を有効または無効にします。 Boolean 値を使用できます。
oemNEC_dir_kerberos_kdc_port	読み出し/書き込み	ドメインコントローラーとの接続に使用するポート番号を指定します。 Kerberos ポート番号は 88 ですが、ドメインコントローラーを設定して別のポート番号を指定することもできます。
oemNEC_dir_kerberos_kdc_address	読み出し/書き込み	ドメインコントローラーの位置。ドメインコントローラーの位置は、IP アドレスまたは DNS 名で指定します。
oemNEC_dir_kerberos_realm	読み出し/書き込み	ドメインコントローラーが設定されている Kerberos レalmを指定します。指定されるドメインの Kerberos レalm名は、通常、大文字に変換されたドメイン名です。
oemNEC_dir_generic_ldap_enabled	読み出し/書き込み	スキーマフリーディレクトリの iLO 汎用 ldap 設定を表示または変更します。 Boolean 値を使用できます。このプロパティを変更するには、スキーマフリーディレクトリを設定する必要があります。

例 :

```
set /map1/oemNEC_dircfg1 oemNEC_dirauth=default_schema
oemNEC_dirsrvaddr=adserv.demo.com
```

追加の set コマンドを使用して、追加のグループを定義します。

1つまたは複数のプロパティをコマンドラインに指定します。複数のプロパティを同じコマンドラインで指定する場合は、スペースで区切る必要があります。

## 仮想メディアコマンド

iLO 仮想メディアは、CLP から操作することができます。本機能を使用するためには、iLO ライセンスキーが必要です。また、本機能を使用するためには「仮想メディア」権限が必要になります。表 17 に、仮想メディアコマンドのターゲットを示します。表 18 に、仮想メディアコマンドのプロパティを示します。仮想メディアサブシステムは、次の場所で確認できます。

```
/map1/oemNEC_vm1
```

詳しくは、iLO ユーザーズガイドを参照してください

## ターゲット

表 17 に、仮想メディアのターゲットを示します。

表 17 仮想メディアコマンドのターゲット

ターゲット	説明
/map1/oemNEC_vm1/floppydr1	仮想フロッピーディスクまたはUSBキードライブデバイス
/map1/oemNEC_vm1/cddr1	仮想CD-ROM デバイス

## プロパティ

表 18 仮想メディアコマンドのプロパティ

プロパティ	アクセス	説明
oemNEC_image	読み出し/書き込み	仮想メディアアクセスのためのイメージパスおよびイメージ名です。値は、最大長 80 文字の URL です。
oemNEC_connect	読み出し/書き込み	仮想メディアデバイスがすでに CLP またはスクリプト化可能な仮想メディア経由で接続されているかどうかを表示します。
oemNEC_boot	読み出し/書き込み	起動フラグを設定します。有効な値は、次のとおりです。 <ul style="list-style-type: none"><li>• Never デバイスから起動しません。値は、No_Boot と表示されます。</li><li>• Once デバイスから 1 回だけ起動します。値は、Once と表示されます。</li><li>• Always サーバーが再起動するたびにデバイスから起動します。値は、Always と表示されます。</li><li>• Connect 仮想メディアデバイスを接続します。oemNEC_connect を Yes に、oemNEC_boot を Always に設定します。</li><li>• Disconnect 仮想メディアデバイスの接続を切断し、oemNEC_boot を No_Boot に設定します。</li></ul>
oemNEC_wp	読み出し/書き込み	書き込み禁止フラグを有効または無効にします。Boolean 値を使用できます。仮想メディアデバイスの接続前に設定しておく必要があります。仮想メディアデバイスの接続中は変更できません。
vm_applet	読み出し	iLO 仮想メディアデバイスが IRC 経由で接続されているかどうかを表示します。これには、サポートされているライセンスキーが必要です。

## イメージの URL

oemNEC\_image の値は URL です。この URL は、最大長が 80 文字で、HTTP サーバー上の仮想メディアイメージファイルの位置を示し、スクリプト化可能な仮想メディアイメージの位置と同じフォーマットが適用されます。

URL の例 :

```
protocol://username:password@hostname:port/filename
```

- protocol - 必須フィールドで、HTTP または HTTPS を指定します。
- username:password - オプションフィールドです。
- hostname - 必須フィールドです。

- port - オプションフィールドです。
- filename - 必須フィールドです。

CLP では、URL の構文は、大まかにしか検証されません。必ず、URL が正しいことをよく確認してください。

例：

- `set oemNEC_image=http://imgserver.example.com/image/dosboot.bin`
- `set oemNEC_image=http://john:abc123@imgserver.example.com/VMimage/installDisk.iso`

## タスク

- 仮想フロッピーディスクにイメージを挿入するには、次のように入力します。

```
cd /map1/oemNEC_vm1/floppydr1
show
set oemNEC_image=http://my.example.com/floppyimg.bin
set oemNEC_boot=connect
show
```

この例では、次のコマンドが実行されます。

- 現在のコンテキストをフロッピーディスクに変更します。
  - メディアが使用中ではないことを確認するために現在のステータスを表示します。
  - 必要なイメージをドライブに挿入します。
  - メディアを接続します。起動設定は、常に自動的に接続されます。
- 仮想フロッピーディスクからイメージを取り出すには、次のように入力します。

```
cd /map1/oemNEC_vm1/floppydr1
set oemNEC_boot=disconnect
```

この例では、次のコマンドが実行されます。

- 現在のコンテキストをディスクまたは USB キードライブに変更します。
  - `disconnect` コマンドを発行します。これにより、メディアの接続が切断され、`oemNEC_image` がクリアされます。
- 仮想 CD-ROM から CD-ROM イメージを取り出すには、次のように入力します。

```
cd /map1/oemNEC_vm1/cddr1
show
set oemNEC_image=http://my.example.com/ISO/install_disk1.iso
set oemNEC_boot=connect
show
```

この例では、次のコマンドが実行されます。

- 現在のコンテキストを CD-ROM ドライブに変更します。
- メディアが使用中ではないことを確認するために現在のステータスを表示します。
- 必要なイメージをドライブに挿入します。

- メディアを接続します。起動設定は、常に、自動的に接続されます。
- 仮想 CD-ROM から CD-ROM イメージを取り出すには、次のように入力します。
 

```
cd /map1/oemNEC_vm1/cddr1
set oemNEC_boot=disconnect
```

 この例では、次のコマンドが実行されます。
  - 現在のコンテキストを CD-ROM ドライブに変更します。
  - disconnect コマンドを発行します。これにより、メディアの接続が切断され、oemNEC\_image がクリアされます。
- CD-ROM イメージを挿入し、1 回の起動に設定するには、次のように入力します。

```
cd /map1/oemNEC_vm1/cddr1
show
set oemNEC_image=http://my.example.com/ISO/install_disk1.iso
set oemNEC_boot=connect
set oemNEC_boot=once
show
```

この例では、次のコマンドが実行されます。

- 現在のコンテキストを CD-ROM ドライブに変更します。
- メディアが使用中ではないことを確認するために現在のステータスを表示します。
- 必要なイメージをドライブに挿入します。
- メディアを接続します。起動設定は、常に自動的に接続されます。
- 起動設定を Once に変更します。
- 単一コマンドで仮想 CD-ROM から CD-ROM イメージを取り出すには、次のように入力します。
 

```
set /map1/oemNEC_vm1/cddr1 oemNEC_boot=disconnect
```

ドライブが接続されていないときに接続の切断を試みると、エラーが表示されます。

## 電源投入およびリセットコマンド

開始およびリセットコマンドを使用すると、iLO のあるサーバーまたは iLO 自体の電源を入れたり再起動したりすることができます。表 19 に、電源投入およびリセットコマンドのプロパティを示します。

**表 19 電源投入およびリセットコマンド**

コマンド	説明
start	サーバーの電源を入れます。
stop	サーバーの電源を切ります。
reset hard	サーバーの電源を切って入れなおします。 パス指定時: <code>reset /system1 hard</code>
reset	system1 : サーバーをウォームブートします。

	map1 : iLO をリセットします。
reset soft	サーバーをウォームブートします。  パス指定時: reset /system1 soft

**表 20 手動リセットコマンド**

プロパティ	アクセス	説明
manual_iLO_reset	読み出し/書き込み	iLO のリセットを遅らせることができます。これは、複数のプロパティを変更する場合に便利です。有効な値は、yes（有効）または no（無効）です。これが有効になっている場合、iLO はユーザーがログアウトする、iLO から切断する、または 'reset /map1' コマンドを実行するときだけリセットします。

例 :

現在のターゲットが次のものである場合、以下のコマンドがサポートされます。

```
/system1
```

- start
- stop
- reset hard/reset soft/reset

現在のターゲットが次のものである場合、以下のコマンドがサポートされます。

```
/map1
```

- reset

以下のコマンドを使用して、manual\_iLO\_reset プロパティのステータスを設定します。

- set /map1/ manual\_ilo\_reset=yes
- set /map1/ manual\_ilo\_reset=no

## ファームウェアコマンド

ファームウェアコマンドを使用すると、iLO のファームウェアのバージョンを表示したり変更したりすることができます。表 21 に、ファームウェア更新のプロパティを示します。ファームウェア設定は、次の場所で確認できます。

```
/map1/firmware1
```

### ターゲット

ターゲットは、ありません。

**表 21 ファームウェア更新のプロパティ**

プロパティ	アクセス	説明
version	読み出し	現在のファームウェアのバージョンを表示します。
date	読み出し	現在のファームウェアのバージョンのリリース日を表示します。
name	読み出し	iLO ファームウェアの名前を表示します。

## コマンドフォーマット

```
load -source URL [ターゲット]
```

ここで、*URL* は、Web サーバー上にあるファームウェアアップデートイメージファイルの URL です。URL は、80 文字までに制限されています。

URL の例：

```
protocol://username:password@hostname:port/filename
```

- *protocol* - 必須フィールドで、HTTP または HTTPS を指定します。
- *username:password* - オプションフィールドです。
- *hostname* - 必須フィールドです。
- *port* - オプションフィールドです。
- *filename* - 必須フィールドです。

CLP では、URL 値の構文は、大まかにしか検証されません。必ず、よく見て、URL が正しいことを確認してください。

例：

```
load /map1/firmware1 -source  
http://imgserver.example.com/firmware/iloFWimage.bin
```

[ターゲット]フィールドは、次のとおりです。

/map1/firmware1 -すでに現在のターゲットである場合、このフィールドはオプションです。

---

注記: ロードされたファームウェアコンポーネントがシステム上にフラッシュされ、既存のバージョンを置き換えます。ファームウェアのフラッシュが成功すると、COMMANDCOMPLETED の status\_tag が表示されます。iLO のファームウェアがフラッシュされると、iLO がリセットされます。

Trusted Platform Module (TPM) がインストールされ有効化されている場合、ロードコマンドには URL の後に'-TPM\_force'オプションを含める必要があります。オプションを含めない場合、コマンドが失敗します。

- 
- ① 重要: TPM が有効な場合に適切な OS 暗号化手順を実行せずにアップグレードすると、データにアクセスできなくなります。回復キーがない場合、または BitLocker を一時停止していない場合、iLO をフラッシュしないでください。
- 

## iLO 以外のファームウェアコマンド

iLO 以外のファームウェアコマンドは、システムファームウェア更新をサポートし、次の場所にあります。

```
/system1/swid<N>
```

ここで、<N>は数値またはワイルドカード (\*) です。ワイルドカードを使用すると、システムにインストールされているすべてのファームウェアバージョンが表示され (iLO を含む)、ロードするファームウェアのソフトウェア ID 番号を識別できます。



システムにインストールされているファームウェアエンティティについての情報を表示 (show) するには、次のようにします。

```
</system1>iLO-> show swid*  
status=0  
status_tag=COMMAND COMPLETED  
Wed Aug 2 01:02:03 2017
```

```
/system1/swid1  
Targets  
Properties  
  name=iLO  
  VersionString=1.10 Jun 07 2017  
  oemNEC_VersionStrings=1.10 Jun 07 2017  
Verbs  
  cd version exit show load
```

iLO 以外のファームウェアをロードするとき、変更内容を適用するためにシステムを手動でリセットすることが必要な場合があります。

前述のコマンド形式に従ってください。例：

```
load -source http://192.168.1.1/images/fw/<firmware_filename>
```

次のタイプのファームウェアファイルがサポートされています。

- システム ROM
- システムプログラマブルロジックデバイス
- パワーマネジメントコントローラー
- Innovation Engine (IE)ファームウェア

## イベントログコマンド

Eventlog コマンドを使用すると、システムと iLO の両方のログを表示したり削除したりすることができます。表 22 に、イベントログコマンドのプロパティを示します。イベントログ設定は、次の場所で確認できます。

- /system1/log1 - Integrated Management ログ (IML)
- /map1/log1 - iLO イベントログ (IEL)

### ターゲット

```
record:1..n
```

ここで、 $n$  は、レコードの総数です。

### プロパティ

表 22 イベントログコマンドのプロパティ

プロパティ	アクセス	説明
number	読み出し	イベントのレコード番号を表示します。

severity	読み出し	イベントの重大度を表示します。重大度レベルは、informational、noncritical、critical、または unknown です。
date	読み出し	イベントの日付を表示します。
time	読み出し	イベントの時刻を表示します。
description	読み出し	イベントの説明を表示します。

例：

- `show /system1/log1 - IML` を表示します。
- `show /map1/log1 - IEL` を表示します。
- `show /system1/log1/recordn - IML` からレコード *n* を表示します。
- `show /map1/log1/recordn - IEL` からレコード *n* を表示します。
- `delete /system1/log1 - IML` を削除します。
- `delete /map1/log1 - IEL` を削除します。

## 起動コマンド

起動コマンドを使用すると、システムのブート順序を変更することができます。起動設定は、次の場所で確認できます。

```
/system1/bootconfig1
```

## レガシーモード

### ターゲット

```
bootsource<n>
```

ここで、*n* は、起動ソースの総数です。

起動ソースのターゲットと、対応する起動ソースの値は変更されません。

起動ソースの値は以下のとおりです。

- `bootsource1` : BootFmCd
- `bootsource2` : BootFmFloppy
- `bootsource3` : BootFmDrive
- `bootsource4` : BootFmUSBKey
- `bootsource5` : BootFmNetwork

---

注記: システム上で起動可能なネットワークデバイスが有効になっていない場合、`BootFmNetwork` 起動ソースがターゲットのリストに表示されないことがあります。

---

## プロパティ

表 23 /system1/bootconfig1 のプロパティ

プロパティ	アクセス	説明
oemNEC_bootmode	読み出し	システムブートモードを表示します。
oemNEC_secureboot	読み出し	UEFIが有効になっているシステムのシステムセキュアブート設定を表示します。
oemNEC_pendingbootmode	読み出し/書き込み	システムが再起動したときに適用される保留中のブートモードの変更を表示または変更します。

**表 24 bootsource<n>のプロパティ**

プロパティ	アクセス	説明
bootorder	読み出し/書き込み	特定の起動ソースのブート順序を設定します。
bootdevice	読み出し	システムブートソース情報を表示します。

例：

bootorder を設定するとき、最初に `show -all /system1/bootconfig1` と入力して、現在のブート順序を一覧表示します。次の出力例では、bootsource3 (BootFmUSBKey) が bootorder=1 と指定されているため、現在プライマリブートデバイスに設定されています。

```
</>iLO-> show -all /system1/bootconfig1
status=0
status_tag=COMMAND COMPLETED
Wed Aug 2 01:02:03 2017
```

```
/system1/bootconfig1
Targets
  bootsource1
  bootsource2
  bootsource3
  bootsource4
  bootsource5
Properties
  oemNEC_bootmode=Legacy
  oemNEC_secureboot=Not Available
  oemNEC_pendingbootmode=Legacy
Verbs
  cd version exit show set
```

```
/system1/bootconfig1/bootsource1
Targets
Properties
  bootorder=2
  bootdevice=BootFmCd
Verbs
  cd version exit show set
```

```
/system1/bootconfig1/bootsource2
Targets
Properties
  bootorder=3
  bootdevice=BootFmDisk
```

```

Verbs
  cd version exit show set

/system1/bootconfig1/bootsource3
  Targets
  Properties
    bootorder=1
    bootdevice=BootFmUSBKey
  Verbs
    cd version exit show set

/system1/bootconfig1/bootsource4
  Targets
  Properties
    bootorder=4
    bootdevice=BootFmNetwork1
  Verbs
    cd version exit show set

/system1/bootconfig1/bootsource5
  Targets
  Properties
    bootorder=5
    bootdevice=BootFmNetwork2
  Verbs
    cd version exit show set

```

ブート順序を変更するには、次のコマンドを入力します。

```
set /system1/bootconfig1/bootsource<n> bootorder=<num>
```

たとえば、bootsource1 (BootfmCd) をプライマリブートデバイスにするには、次のようにします。

```
</system1/bootconfig1>iLO-> set bootsource1 bootorder=1
```

```
status=0
status_tag=COMMAND COMPLETED
Wed Aug 2 01:02:03 2017
```

```
Bootorder being set.
```

```
bootsource1=BootFmCd          bootorder=1
bootsource3=BootFmUSBKey     bootorder=2
bootsource2=BootFmDisk       bootorder=3
bootsource4=BootFmNetwork1   bootorder=4
bootsource5=BootFmNetwork2   bootorder=5
```

特定のデバイスのブート順序を表示するには、次のコマンドを入力します。

```
show /system1/bootconfig1/bootsource<n>
```

たとえば、bootsource1 のブート順序を表示するには、次のように入力します。

```
</system1/bootconfig1>iLO-> show /system1/bootconfig1/bootsource1
```

```

status=0
status_tag=COMMAND COMPLETED
Wed Aug 2 01:02:03 2017

/system1/bootconfig1/bootsourcel
Targets
Properties
    bootorder=1
    bootdevice=BootFmCd
Verbs
    cd version exit show set

```

現在の起動モードを表示するには、次のコマンドを入力します。

```
show /system1/bootconfig1/ oemNEC_bootmode
```

保留中の起動モード（次の起動で有効になる）を表示するには、次のコマンドを入力します。

```
show /system1/bootconfig1/ oemNEC_pendingbootmode
```

## UEFI モード

### ターゲット

```
oemNEC_uefibootsource<n>
```

ここで、*n* は、起動ソースの総数です。

起動ソースのターゲットと、対応する起動ソースの値は変更されません。

### プロパティ

表 25 /system1/bootconfig1 のプロパティ

プロパティ	アクセス	説明
oemNEC_bootmode	読み出し	システムブートモードを表示します。
oemNEC_secureboot	読み出し	UEFI が有効になっているシステムのシステムセキュアブート設定を表示します。
oemNEC_pendingbootmode	読み出し/書き込み	システムが再起動したときに適用される保留中のブートモードの変更を表示または変更します。

表 26 oemNEC\_uefibootsource<n>のプロパティ

プロパティ	アクセス	説明
bootorder	読み出し/書き込み	特定の起動ソースのブート順序を設定します。
oemNEC_description	読み出し	UEFI ブートオプションの説明を表示します。

例：

UEFI モードの `bootorder` を設定するときは、最初に `show -all /system1/bootconfig1` と入力して、現在のブート順序を一覧表示します。UEFI をサポートするシステムでは、対象はそれぞれのブート順序で一覧表示されます。次の出力例では、`oemNEC_uefibootsourcel` が `bootorder=1` と指定されているため、現在プライマリブートデバイスに設定されています。

```
</system1/bootconfig1>iLO-> show -all /system1/bootconfig1
status=0
status_tag=COMMAND COMPLETED
Wed Aug 2 01:02:03 2017
```

```
/system1/bootconfig1
Targets
  oemNEC_uefibootsource1
  oemNEC_uefibootsource2
  oemNEC_uefibootsource3
Properties
  oemNEC_bootmode=UEFI
  oemNEC_secureboot=no
  oemNEC_pendingbootmode=UEFI
Verbs
  cd version exit show set
```

```
/system1/bootconfig1/oemNEC_uefibootsource1
Targets
Properties
  bootorder=1
  oemNEC_description=Red Hat Enterprise Linux
Verbs
  cd version exit show set
```

```
/system1/bootconfig1/oemNEC_uefibootsource2
Targets
Properties
  bootorder=2
  oemNEC_description=Generic USB Boot
Verbs
  cd version exit show set
```

```
/system1/bootconfig1/oemNEC_uefibootsource3
Targets
Properties
  bootorder=3
  oemNEC_description=Internal SD Card 1 : Generic USB3.0-CRW
Verbs
  cd version exit show set
```

UEFI 対応システムのブート順序を変更するには、次のコマンドを入力します。

```
set /system1/bootconfig1/oemNEC_uefibootsource<n> bootorder=<num>
```

たとえば、uefibootsource2 をプライマリブートデバイスにするには、次のようにします。

```
</system1/bootconfig1>iLO-> set oemNEC_uefibootsource2 bootorder=1
status=0
status_tag=COMMAND COMPLETED
Wed Aug 2 01:02:03 2017
```

Bootorder being set.

特定のデバイスのブート順序を表示するには、次のコマンドを入力します。

```
show /system1/bootconfig1/oemNEC_uefibootsource<n>/bootorder
```

システムセキュアブート設定を表示するには、次のコマンドを入力します。

```
show /system1/bootconfig1/oemNEC_secureboot
```

## LED コマンド

LED コマンドは、サーバーの UID LED の状態を変更するために使用します。コマンド

表 28 に、LED コマンドのプロパティを示します。LED 設定は、次の場所で確認できます。

```
/system1/led1
```

### プロパティ

表 27 /system1/led1 のプロパティ

プロパティ	アクセス	説明
enabledstate	読み出し	UID LED の状態を表示します。点滅の際には blinking となります。

### コマンド

表 28 LED コマンド

コマンド	説明
start	UID LED を点灯させます。
stop	UID LED を消灯させます。
show	UID LED のステータスを表示します。

例：

- `show /system1/led1` - 現在の UID LED のステータスを表示します。
- `start /system1/led1` - UID LED を点灯させます。
- `stop /system1/led1` - UID LED を消灯させます。

iLO の CLI のサポート簡素な UID CLI コマンドがサポートされます。

- `uid` - サーバー上の UID LED の現在の状態を表示します。
- `uid on` - UID LED を点灯させます。
- `uid off` - UID LED を消灯させます。

## システムプロパティおよびターゲット

この項で説明されているプロパティとターゲットは、サーバーに関する情報を提供します。

以下のプロパティは、次の場所で確認できます。

```
/system1/oemNEC_power1
```

### プロパティ

表 29 /system1/oemNEC\_power1 のプロパティ

プロパティ	アクセス	説明
oemNEC_PresentPower	読み出し	現在の電力読み取り値を表示します。
oemNEC_AvgPower	読み出し	過去 24 時間の電力測定値の平均を表示します。
oemNEC_MaxPower	読み出し	過去 24 時間の瞬間最高電力を表示します。
oemNEC_MinPower	読み出し	過去 24 時間に観測された最小電力を表示します。
warning_type	読み出し/書き込み	定義されたしきい値を消費電力が超えたときに SNMP アラートを送信できます。警告トリガーの種類を表示および変更します。
warning_threshold	読み出し/書き込み	消費電力のしきい値を表示および変更します。指定期間にわたって消費電力がこの値を超える場合、SNMP アラートがトリガーされます。
warning_duration	読み出し/書き込み	SNMP アラートがトリガーされるまでに消費電力が警告しきい値を超えていなければならない時間期間を表示および変更します。
oemNEC_powerreg	読み出し/書き込み	パワーレギュレーターモードの状態を表示および変更します。有効な値は、dynamic、min、max、os です。
oemNEC_pwracap	読み出し/書き込み	サーバーの消費電力上限設定を表示および変更します（ワット単位）。値が 0 の場合は、消費電力上限が無効になっていることを示します。値は、oemNEC_serverminpower 以上、oemNEC_powersupplycapacity 以下の整数の上限値でなければなりません。
oemNEC_powersupplycapacity	読み出し	サーバーのパワーサプライ容量を表示します（W 単位）。
oemNEC_servermaxpower	読み出し	サーバーの最大電力容量を表示します（W 単位）。
oemNEC_serverminpower	読み出し	サーバーの最小電力容量を表示します（W 単位）。
oemNEC_power_micro_ver	読み出し	パワーマイクロコントローラーのファームウェアバージョンを表示します。
oemNEC_auto_pwr	読み出し/書き込み	サーバーの自動電源投入設定を表示および変更します。有効な値は、ON、15、30、45、60、random、restore、off です。On - 最小の遅延で自動電源投入をオンにします。遅延自動電源オン設定は、15、30、45、60 秒、およびランダム遅延（最大 120 秒）を有効にすることができます（時間は iLO の起動が完了した後に開始）。Restore - 最後の電源状態をリストアします。Off - 自動電源投入をオフにします。

## コマンド

- cd
- version
- exit
- show
- set

例：

- `show /system1/oemNEC_power1 oemNEC_powerreg`
- `set /system1/oemNEC_power1 oemNEC_powerreg=<dynamic|max|min/os>`
- `show /system1/oemNEC_power1 oemNEC_pwracap`
- `set /system1/oemNEC_power1 oemNEC_pwracap=0`
- `show /system1/oemNEC_power1 oemNEC_power_micro_ver`



以下のプロパティは、次の場所で確認できます。

/system1

## プロパティ

表 30 /system1 のプロパティ

プロパティ	アクセス	説明
name	読み出し	システム名を表示します。
number	読み出し	システムのシリアル番号を表示します。
oemNEC_server_name	読み出し	ホストサーバー名文字列を表示します。この文字列の長さは最大50文字で、変更するには"iLO設定の構成"権限が必要です。
oemNEC_server_fqdn	読み出し/書き込み	このホストを識別するためにiLOが使用するサーバーの完全修飾ドメイン名 (FQDN) またはIPアドレスを表示および変更します。CLIは、FQDNまたはIPアドレスに最大232文字の入力を受け入れます。
enabledstate	読み出し	サーバーの電源が入っているかどうかを表示します。
processor_number	読み出し	システムの論理プロセッサ数を表示します。

例：

- show /system1
- show /system1 name
- set /system1 oemNEC\_powerreg=auto

CPU プロパティは/system1 のターゲットであり、システムプロセッサに関する情報を表示します。表 31 に、システム CPU のプロパティを示します。プロパティは、次の場所で確認できます。

/system1/cpun

ここで、*n* は、プロセッサ番号です。

## プロパティ

表 31 システム CPU のプロパティ

プロパティ	アクセス	説明
name	読み出し	プロセッサの製造元情報を表示します。
status	読み出し	ステータス情報を表示します。
number_cores	読み出し	プロセッサコアの数を表示します。
active_cores	読み出し	CPU内部のアクティブな論理コアの数を表示します。
threads	読み出し	CPU 内部の論理スレッドの数を表示します。
speed	読み出し	プロセッサの速度を表示します。
memory_technology	読み出し	プロセッサのメモリ機能を表示します。
cachememory1	読み出し	プロセッサの L1 キャッシュの容量を表示します。
cachememory2	読み出し	プロセッサの L2 キャッシュの容量を表示します。
cachememory3	読み出し	プロセッサの L3 キャッシュの容量を表示します。

例：

```

</i>iLO-> show /system1/cpul
status=0
status_tag=COMMAND COMPLETED
Thu Aug 31 15:05:01 2017

/system1/cpul
Targets
Properties
  name=Intel(R) Xeon(R) Gold 5122 CPU @ 3.60GHz
  status=OK
  number_cores=4
  active_cores=4
  threads=8
  speed=3600MHz
  memory_technology=64-bit Capable
  cachememory1=256KB
  cachememory2=4096KB
  cachememory3=16896KB
Verbs
  cd version exit show

```

memory プロパティは、システムメモリに関する情報を表示します。

表 32 に、システムメモリのプロパティを示します。プロパティは、次の場所で確認できます。

```
/system1/memoryn
```

ここで、*n* は、メモリ DIMM 番号です。

## プロパティ

**表 32 システムメモリのプロパティ**

プロパティ	アクセス	説明
size	読み出し	メモリの容量を表示します。
frequency	読み出し	メモリの速度を表示します。
location	読み出し	メモリの位置を表示します。

Slot プロパティは、システムスロットに関する情報を表示します。

表 33 に、システムスロットのプロパティを示します。プロパティは、次の場所で確認できます。

```
/system1/slotn
```

ここで、*n* は、スロット番号です。

## プロパティ

**表 33 システムスロットのプロパティ**

プロパティ	アクセス	説明
type	読み出し	スロットのタイプを表示します。
width	読み出し	スロットのバス幅を表示します。

name	読み出し	スロットの名称を表示します。
------	------	----------------

Firmware プロパティは、システム ROM に関する情報を表示します。

表 34 に、システムファームウェアのプロパティを示します。プロパティは、次の場所で確認できません。

```
/system1/firmware1
```

**表 34 システムファームウェアのプロパティ**

プロパティ	アクセス	説明
version	読み出し	システム ROM のバージョンを表示します。
date	読み出し	システム ROM のリリース日を表示します。

例 :

- `show /system1/cpu1` - CPU1 に関する情報を表示します。
- `show /system1/memory1` - メモリスロット 1 に関する情報を表示します。
- `show /system1/slot1` - スロット 1 に関する情報を表示します。
- `show /system1/firmware1` - システム ROM に関する情報を表示します。

例 :

```
/system1/firmware1
Targets
Properties
  version=U30
  date=06/14/2017
```

## その他のコマンド

その他のコマンドは次のとおりです。

```
start /system1/oemNEC_vsp1
```

仮想シリアルポートセッションを開始します。**Esc**(キーを押すと CLI セッションに戻ります)。

```
nmi server
```

NMI 生成機能で、オペレーティングシステムをデバッグのために停止します。このコマンドは、"仮想電源およびリセット"権限を持つユーザーのみが使用できます。

---

**注意:** 診断とデバッグのツールとしての NMI 生成機能は、主にオペレーティングシステムが使用不能になった場合に使用します。通常のサーバーの運用では、NMI 生成機能は使用しないでください。NMI ではオペレーティングシステムは適切にはシャットダウンされず、オペレーティングシステムがクラッシュします。このため、サービスとデータは失われます。本コマンドは、オペレーティングシステムが正常に動作せず、調査する場合にのみ使用してください。

---

## 3. IPMI の使用方法

### IPMI ユーティリティ

サーバープラットフォームで IPMI インターフェースをテストするには、Linux の IPMI ツールおよび Windows の IPMI ユーティリティアプリケーションを使用します。Linux の IPMI ツールは、スクリプティングをベースとしてプラットフォームを監視する環境で使用されます。

Windows の IPMI ユーティリティは、「インバンド」を使用する場合（またはコマンドプロンプトから実行する場合）に IPMI ドライバーに依存します。Windows の IPMI ドライバーは、Microsoft から提供されています。

Linux の IPMI ツールでも、「インバンド」を使用する場合は、IPMI ドライバー（ディストリビューションで提供される）が有効になっている必要があります。IPMI デバイスドライバーの、Linux オペレーティングシステム起動時の自動的な開始は、通常、有効になっていません。Linux コンソール（コマンドプロンプト）に root ユーザーとしてログインしている場合は、使用している Linux のバージョンに基づき、次のコマンドを使用して IPMI デバイスドライバーを起動してください。

- RHEL6 の場合 :

```
service ipmi start
```

- RHEL7 の場合 :

```
systemctl start ipmi.service
```

詳しくは、特定の Linux ディストリビューションに提供されている資料を参照してください。Linux の IPMI ツールには、IPMI 2.0 RMCP+プロトコルを使用してサーバーの電源オン/オフ機能およびプラットフォームのリモート監視機能を使用することができます。iLO ファームウェアは、高度な認証、暗号化、および統合のために IPMI 2.0 RMCP+プロトコルのみをサポートしています。従来の IPMI 1.5 IPMI over LAN プロトコルはサポートされていません。

---

注記：サーバーのオペレーティングシステムが動作中で iLO ヘルスドライバーが有効な場合は、インバンドインターフェースを介した IPMI トラフィックがヘルスドライバーのパフォーマンスとシステム全体のヘルスに影響を与える可能性があります。インバンドインターフェースを介して IPMI コマンドを実行しないでください。これはヘルスドライバーの監視に悪影響を与えることがあります。この制限には、IPMI パラメーター（たとえば、Set Watchdog Timer および Set BMC Global Enabled）を設定または変更するあらゆるコマンドが含まれています。単にデータを返す IPMI コマンド（たとえば、Get Device ID および Get Sensor Reading）は、どれでも安全です。

---

### IPMI ツールの基本的な使用方法

Linux の IPMI ツールについて詳しくは、Linux の man ページを参照してください。man ipmitool コマンドにより、このガイドに記載されていない範囲の説明を参照できます。Linux オペレーティングシステムから IPMI ツールを使用してシステムをローカル監視するには、IPMI ドライバーが有効になっている必要があります。一般的なインバンドコマンドには、以下のものがあります。

- iLO のステータスを取得する :

```
# ipmitool mc info
```

- iLO の監視対象センサーのステータスを取得する :

```
# ipmitool sensor list
```

- IPMI SEL の内容を取得する :

```
# ipmitool sel list
```

## Linux 環境での IPMI ツールの高度な使用方法

Linux の IPMI ツールには、IPMI 2.0 RMCP+プロトコルを使用して iLO と安全に通信する機能があります。これは、ipmitool lanplus プロトコル機能です。ほとんどのコマンドに、有効な iLO ユーザー名およびパスワードが必要です。一般的なアウトバンド（または IPMI over LAN）コマンドには、以下のものがあります。

- iLO の全般的なステータスを取得する :

```
# ipmitool -H IP Address or FQDN -I lanplus -U user name -P Password mc info
```

- NX7700x サーバーの電源を入れる :

```
# ipmitool -H IP Address or FQDN -I lanplus -U user name -P Password chassis power on
```

- NX7700x サーバーの UID をオンにする :

```
# ipmitool -H IP Address or FQDN -I lanplus -U user name -P Password chassis identify on
```

SEL エントリーおよび現在のセンサーの読み取り値を取得するなど、ほとんどの Linux の IPMI ツールコマンドはリモートから発行できます。IPMI 2.0 RMCP+プロトコルを有効にするには、次のパラメーターが必要です。

```
-I lanplus
```

## Windows 環境での IPMIutil の高度な使用方法

iLO へのリモート IPMI アクセスには、Windows の IPMIutil.exe アプリケーションを使用します。コマンドは異なりますが、同様の機能を提供します。

- iLO の全般的なステータスを取得する :

```
C:\> ipmiutil.exe health -N IP Address -J 3 -U user name -P Password
```

- NX7700x サーバーの電源を入れる :

```
C:\> ipmiutil.exe reset -u -N IP Address -J 3 -U user name -P Password
```

- NX7700x サーバーの電源を切る :

```
C:\> ipmiutil.exe reset -d -N IP Address -J 3 -U user name -P Password
```

- NX7700x サーバーの UID をオンにする :

```
C:\> ipmiutil.exe led -i5 -N IP Address -J 3 -U user name -P Password
```

---

注記:IPMI ユーティリティーアプリケーションが UID をオンにできるのは 5 秒間のみです。UID LED を点灯させ続けるには、4 秒間の遅延で繰り返し発行するようにコマンドのスクリプトを作成します。

---

注記:本書は NX7700x サーバーにおいてオープンソースソフトウェア（IPMI ドライバー、IPMI ツール、IPMIutil 等）の動作保証を行うものではありません。

---

## 4. Secure Shell の使用方法

### SSH の概要

SSH は、リモートマシンにログインし、コマンドを実行するための Telnet によく似たプログラムです。ただし、SSH には、認証、暗号化、およびデータの整合性機能が含まれるため安全です。iLO ファームウェアは、5 台の SSH クライアントからの同時アクセスをサポートします。SSH の接続および認証後、ユーザーは、コマンドラインインターフェースを使用できます。

iLO 5 は、次をサポートします。

- SSH プロトコルバージョン 2
- PuTTY は、インターネット上でダウンロードできる、SSH プロトコルの無償バージョンです。バージョン 0.54 以降の PuTTY を使用することをおすすめします。
- OpenSSH。インターネット上でダウンロードできる、SSH プロトコルの無償バージョンです。

ファームウェアをアップグレードすると、SSH 機能が使用できるようになるまで、60 秒程度かかります。この間、iLO は、1024 ビットの DSA キーを生成します。これらのキーは、iLO によって保存され将来使用されます。iLO が、工場出荷時のデフォルト設定にリセットされた場合、DSA キーは消去され、次の起動時に再生成されます。

### サポートされる SSH 機能

ライブラリは、SSH プロトコルのバージョン 2 (SSH-2) のみをサポートします。表 40 に、iLO によってサポートされている SSH 機能を示します。

表 40 サポートされる SSH 機能

項目	サポートされるアルゴリズム
サーバーホストキーのアルゴリズム	ssh-dsa
暗号化（両方向で同じセットをサポート）	3des-cbc、aes128-cbc
ハッシュ関数のアルゴリズム	hmac-sha1、hmac-md5
パブリックキーのアルゴリズム	ssh-dsa
キー交換	Diffie-hellman-group1-sha1
圧縮	なし
言語	英語
クライアント/ユーザー認証方式	パスワード
認証タイムアウト	2 分
認証試行回数	3

## Secure Shell の使用

### SSH の使用

1. SSH ウィンドウを開きます。
2. プロンプトが表示されたら、IP アドレスまたは DNS 名、ログイン名、およびパスワードを入力します。

## OpenSSH の使用

Linux マシンで OpenSSH クライアントを起動するには、次のように入力します。

```
ssh -l loginname ipaddress/dns name
```

## PuTTY の使用

- PuTTY セッションを開始するには、PuTTY をインストールしたディレクトリにある PuTTY アイコンをダブルクリックします。
- コマンドラインから PuTTY セッションを開始するには、以下の手順に従ってください。

- 次のように入力して、*host* という名前のサーバーへの接続を開始します。

```
putty.exe [-ssh | -rlogin | -raw] [user@]host
```

- 次のように入力して、*sessionname* という名前の保存された既存のセッションを開始します。

```
putty.exe -load session name
```

---

注記:本書は NX7700x サーバーにおいてオープンソースソフトウェア (OpenSSH、PuTTY 等) の動作保証を行うものではありません。

---

## SSH キーの管理

SSH キーベースの認証によって、SSH を介して LOM デバイスに接続します。

SSH キーは、SSH を対話モードで使用するために作成する必要はありません。対話モードでの SSH の使用については、「SSH の概要」を参照してください。

## PuTTY からの SSH キーのインポート

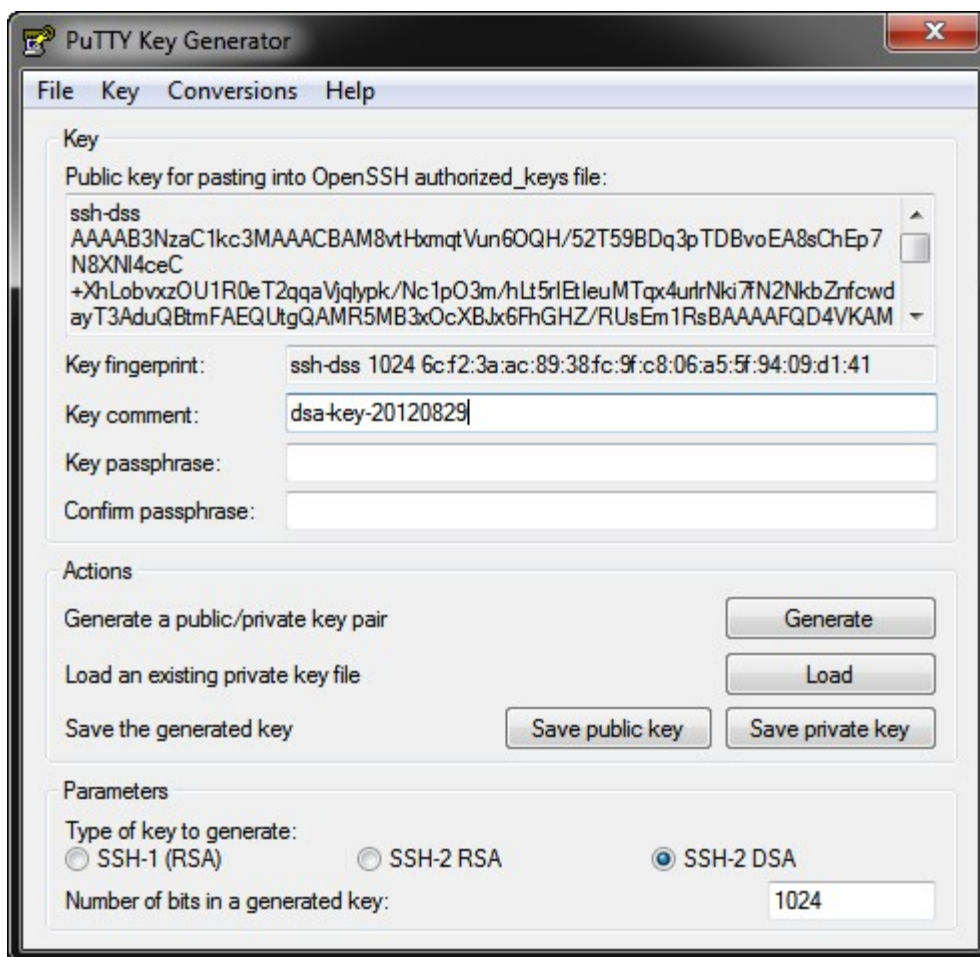
iLO インターフェースは、OpenSSH 2、RFC4716 の 2 つのキーフォーマットをサポートします。iLO ファームウェアでは、パブリックキーファイル情報を 1 行に記載します。PuTTY Key Generator ユーティリティ (puttygen.exe) を使用すると、iLO にインポートするための正しい形式のキーファイルを生成できます。

PuTTY から SSH キーを iLO にインポートするには、以下の手順に従ってください。

1. [PuTTY Key Generator]アイコンをダブルクリックしてユーティリティを起動します。
2. 生成するキーのタイプ (**ssh-2 RSA** または **SSH-2 DSA**) を選択します。
3. **[Generate]**をクリックします。

キーエリアで、マウスを移動してキーを生成します。キー生成プロセスが完了するまで、マウスを動かし続けてください。

図 1 PuTTY Key Generator

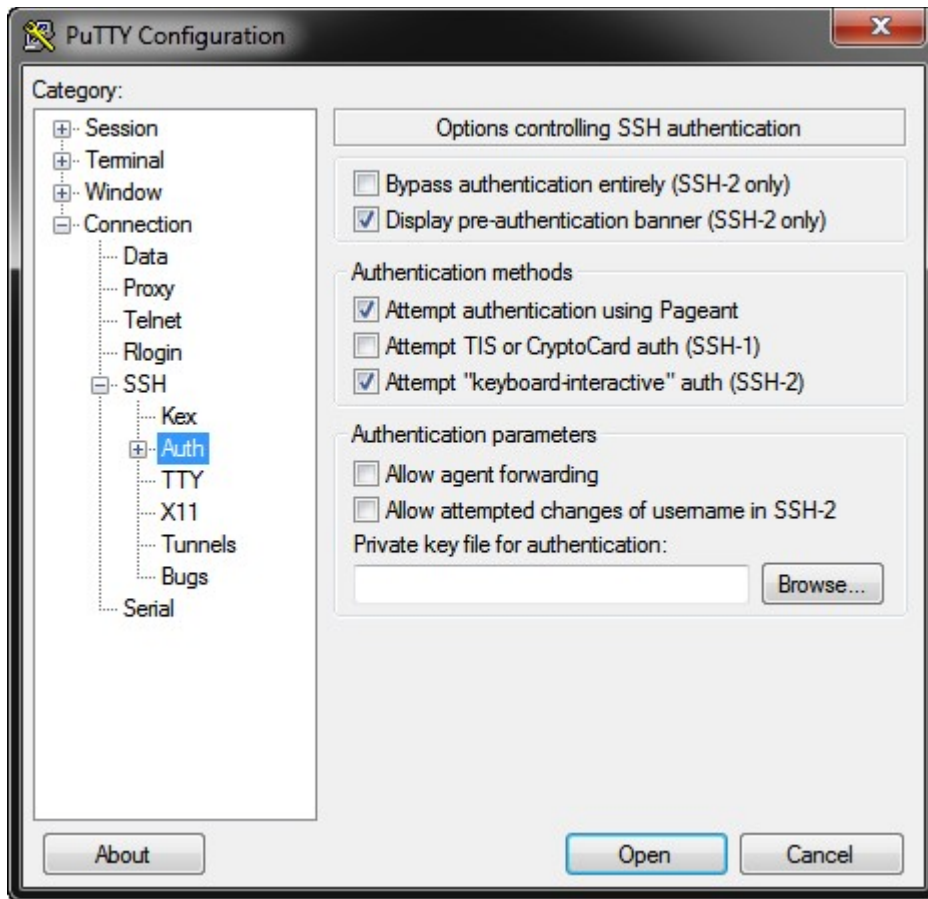


4. **[Save public key]**をクリックし、プロンプトが表示されたらファイル名を入力します。
5. **[Save private key]**をクリックし、プロンプトが表示されたらファイル名を入力します。
6. なお、キーパスフレーズを入力し、確認するオプションもあります。
7. テキストエディターでパブリックキーを開き、内容をクリップボードにコピーします。
8. iLO がまだ開いていない場合は、ログインします。
9. iLO の**[Security]**→**[Secure Shell Key]**/**[セキュリティ]**→**[セキュアシェルキー(SSH)]**ページで、**[Authorized SSH Keys]**/**[認証済み SSH キー]**リストからユーザーを選択し、**[Authorized New Key]**/**[新しいキーの認証]**をクリックします。
10. **[Public Key Import Data]**/**[パブリックキーインポートデータ]**ボックスが表示されます。
11. PEM エンコードされたパブリックキーをボックスに貼り付け、**[Import Public Key]**/**[パブリックキーのインポート]**をクリックします。
12. 認可された SSH キーのリストに、ユーザーの新しいパブリックキーハッシュが表示されます。
13. PuTTY を起動します。
14. **[Session]**を選択し、iLO の IP アドレスを設定します。



15. **[Connection]**→**[SSH]**→**[Auth]**を選択します。
16. **[Browse]**をクリックしてプライベートキーファイルを確認します。

図 2 PuTTY の設定ウィンドウ



17. **[Open]**をクリックします。
18. iLO ファームウェアからユーザー名が求められます。
19. パブリックキーに関連付けられたログオン名を入力します。
20. iLO のパブリックキーが、PuTTY のプライベートキーで認証されます。キーが一致したら、パスワードを使用せずに iLO にログインされます。

キーは、キーパスフレーズとともに作成することができます。キーパスフレーズを使用してパブリックキーを生成した場合は、iLO にログインする前にキーパスフレーズの入力を求められます。

### ssh-keygen を使用して生成された SSH キーのインポート

ssh-keygen を使用して SSH キーを生成し、key.pub ファイルを作成したら、以下の手順を実行してください。

1. key.pub ファイルの場所を確認してテキストエディターで開きます。ファイルは、テキスト ssh-dsa で始まります。
2. 保存して、ファイルを閉じます。

キーファイルをインポートして認証する準備ができました。

## 5. iLO RESTful API の使用方法

### RESTful API の紹介

RESTful API は、サーバー管理のための最新のインターフェースです。RESTful API は、基本的な HTTP メソッド (GET、PUT、POST、DELETE、および PATCH) を使用して、JSON 形式のリソースを、iLO 上の URI との間で送受信します。スクリプティング言語を使用して、RESTful API 向けの単純な REST クライアントを容易に開発することができます。Python などのほとんどの言語では、JSON を辞書のような内部データ構造に変換できるため、データに簡単にアクセスできます。これにより、サーバー管理用のツールを使用せずに、直接 RESTful API を呼び出すコードを書くことができます。

### Redfish 準拠

Redfish 標準に準拠しつつ、NEC 固有の機能のため RESTful API は拡張されています。

### HATEOAS を使用して設計された REST API

Representational State Transfer (REST) は、POST、GET、PUT、PATCH、および DELETE などの HTTP メソッドを使用してリソース上で実行された基本的な CRUD (生成、読み取り、更新、削除、およびパッチ) 操作を使用する Web サービスです。RESTful API は HATEOS (Hypermedia as the Engine of Application State) と呼ばれる REST アーキテクチャーを使用して設計されています。このアーキテクチャーにより、クライアントは、簡単な固定 URL (/redfish/v1/) および iLO データモデルで記載された他のいくつかの最上位 URI を使用して、iLO と情報のやり取りをすることができます。データモデルの残りの部分は、データで明確に識別する「リンク」をたどることで検出できます。これには、クライアントが一連の固定 URL を知る必要がないというメリットがあります。RESTful API を使用してタスクを自動化するスクリプトを作成する場合、この簡単な URL をハードコードして、タスクの実行に必要な RESTAPI URL を検出するスクリプトを設計するだけで済みます。REST の概念についての詳細は、以下をご参照ください。

- <https://ja.wikipedia.org/wiki/REST>
- <http://en.wikipedia.org/wiki/HATEOAS>

### RESTful API の主な利点

RESTful API は、iLO 5 向けの主な管理インターフェースになります。この機能セットは、従来の SMASH CLP や IPMI インターフェースよりも機能が強化されています。RESTful API を使用すると、サーバーの完全なインベントリの取得、電源の制御とリセット、BIOS や iLO の構成、イベントログの取得など、多くの機能を使用できます。RESTful API は、新しいソフトウェアインターフェースの共通パターンに移行するという、インターネットの傾向に対応したものです。さまざまな業界の多くの Web サービスでは RESTful API を使用しています。この API の実装が簡単で、使いやすく、以前の技術に比べて拡張性に優れているためです。現在では、サーバー管理用 API として RESTful API が主流になりつつあります。

### リソースの操作

操作	HTTP メソッド	説明
Create	POST	新しいリソースを作成するか、カスタムアクションを呼び出します。同期 POST は新しく作成したリソースを返します。
Read	GET	要求されたリソース表現を返します。

Update	PATCH または PUT	既存のリソースを更新します。スキーマで readonly = false と記述されているプロパティを変更できます。
Delete	DELETE	指定したリソースを削除します。

## リターンコード

リターンコード	説明
2xx	操作は成功しました。
4xx	クライアント側のエラーが、エラーメッセージとともに返されました。
5xx	iLO エラーがエラーメッセージとともに返されました。

**注記:** エラーが発生すると、リターンコード 4xx または 5xx が示され、ExtendedErrorJSON 応答が返されます。期待するリソースは返されません。

## RESTful API の使用に関するヒント

iLO RESTful API は、iLO Standard ライセンスのみで使用できます。ただし、Advanced や Scale out のライセンスがないと、データの一部の機能が使用できない場合があります。RESTful API にアクセスするためには、Postman または CURL などの HTTPS 対応のクライアントが必要です。

## スキーマ

各リソースタイプは、オブジェクト、許可されたプロパティ、要求されたプロパティ、タイプ、およびその他の情報のフォームを定義するスキーマファイルを備えています。データモデルは、ドラフト 4 の JSON スキーマ基準を採用して定義を行います。詳しくは、<http://json-schema.org> を参照してください。Python スキーマ検証用のパッケージをダウンロードするには、<https://github.com/Julian/jsonschema> を参照してください。パッケージは、対応するスキーマに対して他の言語でリソースを検証する場合にも使用できます。これは、業界標準のツールチェーンを活用する戦略の一部です。スキーマではリソースの class が定義され、クラスの各インスタンスは、利用可能なプロパティのサブセットを実装する場合があります。ただし、スキーマ内のすべてのプロパティが、必ずしもすべてのインスタンスに実装されるわけではありません。機種によっては、一部プロパティが動作しない場合があります。

**注記:** 各プロパティは、readonly = true または readonly = false として特定されます。これは、プロパティで PATCH 操作は実行できますが、PATCH を許可する Allow ヘッダーがリソースに存在する場合に限られることを示しています。2 つのリソースは同じ Type を共有する場合がありますが、readonly = false のプロパティに対しても、異なる Allow ヘッダー（PATCH 操作を有効または無効にするヘッダー）を持つ可能性があります。

**重要:** スキーマは Starter Pack 中の /software/Rest-classes-iLO5 フォルダに格納されています。また、ご利用の iLO ファームウェアから、ファームウェアがサポートしているスキーマを取得することも可能です。後述する CURL や Postman 等のクライアントを使用し、以下の URI へアクセスしてください。

<https://{iLO アドレス}/redfish/v1/Schemas/>

個々のスキーマの URI は iLO RESTful API の詳細に記載しています。

重要：スキーマに記載があっても、機種によってはサポートしていない機能があります。実際にお使いになる機種で、事前に十分な検証を行ってください。

## スキーマバージョンと互換性

RESTful API 内のリソースは、名前とバージョンでタイプ付けされます（例：

“ServiceRoot.1.0.0”）。これにより、正しいスキーマだけでなく、正確なバージョンも識別することができます。RESTful API を正しく使用するためには、iLO ファームウェアがサポートするスキーマのバージョンと同じものを参照する必要があります。

重要：iLO ファームウェアのバージョンによってサポートしている RESTful API が異なる場合があります。必ずご使用中の iLO ファームウェアのバージョンに対応したスキーマを参照してください。iLO ファームウェアがサポートするスキーマとは異なるバージョンのスキーマをお使いになった場合、動作は保障できません。

## RESTful API のエラーメッセージ

RESTful API を実行した場合、リターンコードとして 400 番台、500 番台の HTTP Status が返される場合があります。400 番台、500 番台はエラーを示すコードで、正しく API を実行できなかったことを意味します。この時、レスポンスの Body に JSON 形式でエラーの内容が返されます。このエラーの内容とスキーマファイルを組み合わせることで、より詳しいエラーの内容を確認することができます。

たとえば、エラーコード 400 と共に以下のエラーが返されたとします。

```
{
  "error": {
    "code": "iLO.0.10.ExtendedInfo",
    "message": "See @Message.ExtendedInfo for more information.",
    "@Message.ExtendedInfo": [
      {
        "MessageId": "Base.1.0.MalformedJSON"
      }
    ]
  }
}
```

この場合、“MessageID”の前半が参照すべきスキーマファイルを示しており、Base.v1\_0\_X.json（X は任意の数字）の MalformedJSON という項目に、より詳しい情報が記述されていることを意味しています。そこで、スキーマの Base.v1\_0\_X.json（スキーマの/RegistryStore/en/フォルダ以下に格納されています）を確認すると以下の記述があり、リクエストの Body で送った JSON に問題があったことがわかります。

```
"MalformedJSON": {
  "Description": "The request body contains malformed JSON.",
  "Message": "The request body submitted is malformed JSON and cannot be
    parsed by the receiving service.",
  "Severity": "Critical",
  "NumberOfArgs": 0,
  "Resolution": "Verify that the request body is valid JSON and
    resubmit the request."
},
```

次に、パスワード変更でエラーが発生した場合の例を示します。RESTful API 経由でユーザーのパスワードを”12345”に変更しようとした際に、エラーコード 400 と共に以下のエラーが返されたとします。

```
{
  "error": {
    "code": "iLO.0.10.ExtendedInfo",
    "message": "See @Message.ExtendedInfo for more information.",
    "@Message.ExtendedInfo": [
      {
        "MessageArgs": [
          "8",
          "39"
        ],
        "MessageId": "iLO.2.2.InvalidPasswordLength"
      }
    ]
  }
}
```

この場合、スキーマファイルの iLO.v2\_2\_X.json (X は任意の数字) の InvalidPasswordLength という項目に、より詳しい情報が記述されていることを意味しています。そこで、スキーマの iLO.v2\_2\_X.json (スキーマの /RegistryStore/en/ フォルダ以下に格納されています) を確認すると、以下の記述があります。

```
"InvalidPasswordLength": {
  "Description": "The password length is not valid.",
  "Message": "A valid password must contain between %1 to %2 characters.",
  "NumberOfArgs": 2,
  "ParamTypes": [
    "String",
    "String"
  ],
  "Resolution": "Retry the operation using a corrected password.",
  "Severity": "Critical"
},
```

ここで、iLO からの応答に記載された "MessageArgs" の "8" と "39" を iLO.v2\_2\_X.json の "Message" 文中に記述された %1 と %2 に当てはめて確認すると "A valid password must contain between 8 to 39 characters." となり、8 文字から 39 文字のパスワードしか受け付けられないよう iLO が設定されているためにパスワードの変更に失敗したことがわかります。この場合、iLO の最小パスワード長設定を変更するか、より長いパスワードを設定するかいずれかの対応を行うことで、パスワードの変更を行うことができます。

## iLO RESTful API の操作例

### CURL を使用した例

CURL は、RESTful API に簡単にアクセスできる多くのオペレーティングシステムで使用可能なコマンドラインユーティリティです。CURL は <http://curl.haxx.se/> から入手できます。CURL のすべての例がフラグ -insecure を使用していることに注意してください。これによって、CURL が HTTPS 証明書の検証をバイパスするようにします。実際の使用では、ユーザー提供証明書を使用するように iLO を構成する必要があり、このオプションは不要です。CURL が HTTP リダイレクト応答に従うよう強制する -L オプションを使用することにも注意してください。iLO がさまざまな項目の URI の場所を変更する場合、その新しい場所をクライアントに通知して自動的に新しいリンクをたどることができます。また、-u オプションでユーザー名とパスワードを指定します。

```
> curl https://myilo.example.com/redfish/v1/ -i -insecure -u
username:password -L
```

-i は、HTTP 応答ヘッダーを返し、--insecure は TLS/SSL 認定検証をバイパスし、-L はリダイレクトに従います。また、-u でユーザー名とパスワードを指定します。

```
{ "@odata.context": "/redfish/v1/$metadata#ServiceRoot.ServiceRoot", "@odata.
etag": "W/\"75421942\"", "@odata.id": "/redfish/v1/", "@odata.type": "#Serv
iceRoot.v1_1_0.ServiceRoot", "AccountService": {"@odata.id": "/redfish/v1/A
ccountService/"}, "Chassis": {"@odata.id": "/redfish/v1/Chassis/"}, "EventSe
rvice": {"@odata.id": "/redfish/v1/EventService/"}, "Id": "v1", "JsonSchemas
": {"@odata.id": "/redfish/v1/Schemas/"}, "Links": {"Sessions": {"@odata.id":
"/redfish/v1/SessionService/Sessions/"}, "Managers": {"@odata.id": "/redfi
sh/v1/Managers/"}, "Name": "HPE RESTful Root Service", "Oem": {"Hpe": {"@odat
a.context": "/redfish/v1/$metadata#HpeILOServiceExt.HpeILOServiceExt", "@o
data.type": "#HpeILOServiceExt.v2_0_0.HpeILOServiceExt", "Links": {"Resourc
eDirectory": {"@odata.id": "/redfish/v1/ResourceDirectory/"}, "Manager":
[ {"DefaultLanguage": "en", "FQDN": "myilo.example.bmc.com", "HostName": "myil
o", "Languages": [ {"Language": "en", "TranslationName": "English", "Version": "
1.15"}, {"Language": "ja", "TranslationName": "日本語", "Version": "1.15"}], "Ma
nagerFirmwareVersion": "1.15", "ManagerType": "iLO 5"}], "Moniker": {"ADVLIC
": "iLO Advanced", "BMC": "iLO", "BSYS": "", "CLASS": "Baseboard Management Con
troller", "FEDGRP": "DEFAULT", "IPROV": "NX7700XBUILDER", "PRODABR": "iLO", "PR
ODFAM": "Integrated Lights-Out", "PRODGEN": "iLO 5", "PRODNAM": "Integrated L
ights-Out 5", "PRODTAG": "NEC iLO", "STDLIC": "iLO Standard", "SUMABR": "SUM",
"SUMGR": "Smart Update Manager", "SYSFAM": "NX7700x", "VENDABR": "NEC", "VENDN
AM": "NEC Corporation", "WWW": "jpn.nec.com", "WWWAHSV": "jpn.nec.com/nx7700x
/", "WWWBMC": "jpn.nec.com/nx7700x/", "WWWDOC": "jpn.nec.com/nx7700x/", "WWWES
RS": "jpn.nec.com/nx7700x/", "WWWGLIS": "jpn.nec.com/nx7700x/", "WWWIOL": "jp
n.nec.com/nx7700x/", "WWWLIC": "jpn.nec.com/nx7700x/", "WWWLML": "jpn.nec.co
m/nx7700x/", "WWWPASS": "jpn.nec.com/nx7700x/", "WWWPRV": "jpn.nec.com", "WWW
QSPEC": "jpn.nec.com/nx7700x/", "WWWRESTDOC": "jpn.nec.com/nx7700x/", "WWWSP
P": "jpn.nec.com/nx7700x/", "WWWSWLIC": "jpn.nec.com/nx7700x/"}, "Sessions":
{"CertCommonName": "myilo.example.com", "CertificateLoginEnabled": false, "K
erberosEnabled": false, "LDAPAuthLicenced": true, "LDAPEnabled": false, "Local
LoginEnabled": true, "LoginFailureDelay": 0, "LoginHint": {"Hint": "POST to /S
essions to login using the following JSON object:", "HintPOSTData": {"Pass
word": "password", "UserName": "username"}}, "SecurityOverride": false, "Serve
rName": "localhost.localdomain", "Time": "2017-xx-xxTxx:xx:xxZ"}}, "Redfish
Version": "1.0.0", "Registries": {"@odata.id": "/redfish/v1/Registries/"}, "S
essionService": {"@odata.id": "/redfish/v1/SessionService/"}, "Systems": {"@
odata.id": "/redfish/v1/Systems/"}, "UUID": "xxxxxxxx-xxxx-xxxx-xxxx-xxxx
xxxxxxxx", "UpdateService": {"@odata.id": "/redfish/v1/UpdateService/"}}
```

注記:本書は NX7700x サーバーにおいてオープンソースソフトウェア (curl 等) の動作保証を行うものではありません。

## Postman を使用した例

Postman は RESTful API に簡単にアクセスできるソフトウェアです。POST/PATCH/DELETE などの任意のメソッドを送ることができ、レスポンスの JSON を整形して表示可能です。

<https://www.getpostman.com/>

© 2017 Postdot Technologies, Inc. All Rights Reserved

### GET

情報や設定値の取得をする際には、GET メソッドを使用します。GET メソッドを使用する際には、URI 入力欄の左側から、メソッドとして GET を選択します。

ここでは、ユーザー情報を取得する例をご紹介します。メソッドとして GET を選択し、URI 入力欄に "https://myilo.example.com/redfish/v1/AccountService/Accounts/" を入力します。続いて、Authorization の設定に移り、TYPE を Basic Auth に、Username と Password に iLO で有効なユーザー名とパスワードを入力します。最後に Send ボタンを押します。

The screenshot shows the Postman interface for configuring a GET request. The URL bar contains "https://myilo" and a dropdown arrow. Below it, the method is set to "GET" and the full URL "https://myilo.example.com/redfish/v1/AccountService/Accounts/" is entered. A blue "Send" button is on the right. Underneath, the "Authorization" tab is selected, showing a dropdown for "TYPE" set to "Basic Auth". To the right, there are input fields for "Username" (containing "User name") and "Password" (containing "Password").

リクエストが正しく処理されると、HTTP Status 200 が返され以下のような JSON フォーマットのレスポンスが取得できます。

```
{
  省略
  "@odata.type": "#ManagerAccountCollection.ManagerAccountCollection",
  "Description": "iLO User Accounts",
  "Members": [
    {
      "@odata.id": "/redfish/v1/AccountService/Accounts/1/"
    },
    {
      "@odata.id": "/redfish/v1/AccountService/Accounts/2/"
    }
  ],
  "Members@odata.count": 2,
  "Name": "Accounts"
}
```

この例では、iLO 用のユーザーが 2 つ作成されていることがわかります。

ユーザーの詳細情報を確認するためには、レスポンスに記載されたパスに対して GET を実行します。2 番目のユーザーの詳細情報を取得する場合、以下のように GET を実行します。

The screenshot shows a REST client interface. At the top, there is a URL bar containing "https://myilo" and a dropdown arrow. Below it, a request configuration bar shows the method "GET", the full URL "https://myilo.example.com/redfish/v1/AccountService/Accounts/2/", and a "Send" button. Underneath, there are tabs for "Authorization", "Headers", and "Body", with "Authorization" selected. The "Authorization" section contains a "TYPE" dropdown menu set to "Basic Auth", a "Username" field with the value "User name", and a "Password" field with the value "Password".

リクエストが正しく処理されると、HTTP Status 200 が返され以下のような JSON フォーマットのレスポンスが取得できます。

```
{
  省略
  "@odata.type": "#ManagerAccount.v1_0_0.ManagerAccount",
  "Id": 2,
  "Name": "User Account",
  "Oem": {
    "Hpe": {
      "@odata.type": "#HpeILOAccount.v2_0_0.HpeILOAccount",
      "LoginName": "sample",
      "Privileges": {
        "HostBIOSConfigPriv": true,
        "HostNICConfigPriv": true,
        "HostStorageConfigPriv": true,
        "LoginPriv": true,
        "RemoteConsolePriv": true,
        "SystemRecoveryConfigPriv": true,
        "UserConfigPriv": true,
        "VirtualMediaPriv": true,
        "VirtualPowerAndResetPriv": true,
        "iLOConfigPriv": true
      }
    }
  },
  "Password": null,
  "UserName": "sample"
}
```

レスポンス中の"@odata.type"を見ると、"#ManagerAccount.v1\_0\_0.ManagerAccount"と記載があります。そこで、スキーマファイル ManagerAccount.v1\_0\_0.json を確認すると、"UserName"がユーザー名であること、"Password"がパスワードであることがわかります。ただし、設定されているパスワードを取得することはできないので、ここでは null が入っています。スキーマファイルから、"Password"には null が入る場合があることがわかります。また、レスポンスの"Oem"の中には改めて"@odata.type"があり、"#HpeILOAccount.v2\_0\_0.HpeILOAccount"と記載されています。そこでスキーマファイルの HpeILOAccount.v2\_0\_0.json を参照すると、"LoginName"の説明や、各権限の範囲を確認することができます。



## PATCH

設定値の変更を行う際には、PATCH メソッドを使用します。PATCH メソッドを使用する際には、URI 入力欄の左側から、メソッドとして PATCH を選択します。

ここでは、sample ユーザーのパスワードを変更する例をご紹介します。sample ユーザーは iLO に登録された 2 番目のユーザーと仮定します。この場合、メソッドとして PATCH を選択し、URI 入力欄に "https://myilo.example.com/redfish/v1/AccountService/Accounts/2/" を入力します。続いて、Authorization の設定に移り、TYPE を Basic Auth に、Username と Password に iLO で有効なユーザー名とパスワードを入力します。

The screenshot shows a REST client interface with the following elements:

- URI: `https://myilo`
- Method: `PATCH`
- URI: `https://myilo.example.com/redfish/v1/AccountService/Accounts/2/`
- Buttons: `Send`
- Authorization: `Basic Auth`
- Username: `User name`
- Password: `Password`

先ほど GET した際に参照したスキーマファイル `ManagerAccount.v1_0_0.json` を確認すると、"Password" は "readonly": false、つまり書き換え可能であることがわかります。また、"type": ["string", "null"] となっていることから、string 型(文字列)を指定可能なことがわかります。そこで、Body を選択し、フォーマットに raw JSON(application/json) を選択したうえで以下の JSON リクエストを入力します。ここでは、パスワードとして "SamplePassword" を設定しています。

```
{
  "Password" : "SamplePassword"
}
```

The screenshot shows the REST client interface with the request body configured:

- URI: `https://myilo`
- Method: `PATCH`
- URI: `https://myilo.example.com/redfish/v1/AccountService/Accounts/2/`
- Buttons: `Send`
- Authorization: `Basic Auth`
- Format: `JSON(application/json)`
- Body content:

1	{
2	"Password" : "SamplePassword"
3	}

リクエストが正しく処理されると、HTTP Status 200 が返されます。これでパスワードが変更されました。

## POST

設定の新規追加を行う際には、POST メソッドを使用します。POST メソッドを使用する際には、URI 入力欄の左側から、メソッドとして POST を選択します。

ここでは、TestUser ユーザーを新規に作成する例をご紹介します。メソッドとして POST を選択し、URI 入力欄に” https://myilo.example.com/redfish/v1/AccountService/Accounts/”を入力します。続いて、Authorization の設定に移り、TYPE を Basic Auth に、Username と Password に iLO で有効なユーザー名とパスワードを入力します。

The screenshot shows a REST client interface with the following elements:

- URI input field: `https://myilo`
- Method dropdown: `POST`
- URI input field: `https://myilo.example.com/redfish/v1/AccountService/Accounts/`
- Send button: `Send`
- Tabbed interface with `Authorization` selected.
- Authorization TYPE dropdown: `Basic Auth`
- Username input field: `User name`
- Password input field: `Password`

ユーザーの詳細を GET した時に使用したスキーマファイル `ManagerAccount.v1_0_0.json` と `HpeiLOAccount.v2_0_0.json` を参照し、新規に作成するユーザーの設定値を記述します。例えば、ユーザー名を”TestUser”、権限を”SystemRecoveryConfigPriv”を除く全て、パスワードを”TestPassword”とした場合、以下のようになります。

```
{
  "Oem": {
    "Hpe": {
      "LoginName": "TestUser",
      "Privileges": {
        "HostBIOSConfigPriv": true,
        "HostNICConfigPriv": true,
        "HostStorageConfigPriv": true,
        "LoginPriv": true,
        "RemoteConsolePriv": true,
        "SystemRecoveryConfigPriv": false,
        "UserConfigPriv": true,
        "VirtualMediaPriv": true,
        "VirtualPowerAndResetPriv": true,
        "iLOConfigPriv": true
      }
    }
  },
  "Password": "TestPassword",
  "UserName": "TestUser"
}
```

次に、Body を選択し、フォーマットに raw JSON(application/json)を選択したうえで上記の JSON リクエストを入力します。

https://myilo

POST  https://myilo.example.com/redfish/v1/AccountService/Accounts/

Authorization Headers Body ●

form-data x-www-form-urlencoded raw ● binary JSON(application/json)

1	{
2	"Oem": {
3	"Hpe": {
4	"LoginName": "TestUser",
	省略

リクエストが正しく処理されると、HTTP Status 201 Created が返されます。これでユーザーが作成されました。

## DELETE

設定の削除を行う際には、DELETE メソッドを使用します。DELETE メソッドを使用する際には、URI 入力欄の左側から、メソッドとして DELETE を選択します。

ここでは、iLO の 3 番目のユーザーを削除する例をご紹介します。メソッドとして DELETE を選択し、URI 入力欄に” https://myilo.example.com/redfish/v1/AccountService/Accounts/3/”を入力します。続いて、Authorization の設定に移り、TYPE を Basic Auth に、Username と Password に iLO で有効なユーザー名とパスワードを入力します。

https://myilo

DELETE  https://myilo.example.com/redfish/v1/AccountService/Accounts/3/

Authorization ● Headers Body

TYPE	Username	<input type="text" value="User name"/>
<input type="text" value="Basic Auth"/> <input type="button" value="v"/>	Password	<input type="text" value="Password"/>

リクエストが正しく処理されると、HTTP Status 200 が返されます。これでユーザーが削除されました。

---

注記:本書は NX7700x サーバーにおいてオープンソースソフトウェア (Postman 等) の動作保証を行うものではありません。

---

## iLO RESTful API の詳細

iLO がサポートする RESTful API のうち、基本的なものをご紹介します。

RESTful API では、将来にわたり柔軟な機能の拡張を可能にするため、リソースが記載された URI は固定されていません。iLO が保障している URI は以下の 5 つだけです。下記以外にアクセスするには、まず下記の URI へアクセスし、データ構造を解析して目的のリソースのコレクションへ移動する必要があります。個々の URI をスクリプト等にハードコードしないでください。

- /redfish/v1/
- /redfish/v1/Systems/
- /redfish/v1/Chassis/
- /redfish/v1/Managers/
- /redfish/v1/Sessions/

以下に掲載している情報は、iLO ファームウェアのアップデートで予告なく変更される場合があります。ご利用中の iLO ファームウェアに対応するスキーマを必ずご確認ください。

### /redfish/v1/

iLO の大まかな情報を取得できます。

#### GET

- Web インターフェースデフォルト言語設定
- iLO FQDN (ホスト名+ドメイン名)
- iLO ホスト名
- インストール済み言語パックバージョン
- iLO Firmware バージョン
- 現在のセッション情報
- セキュリティオーバーライド状態
- サーバー名
- iLO 現在時刻

#### スキーマ

詳細は以下のスキーマを参照してください。

- Starter Pack

ServiceRoot.v1\_1\_0.json

- iLO ファームウェア

/redfish/v1/SchemaStore/en/ServiceRoot.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/AccountService/

認証失敗時の動作を設定できます。また、ローカルユーザーの最小パスワード長を設定できます。

### GET・PATCH

- 認証の失敗時の遅延時間
- 認証失敗ログ
- 遅延前の認証の失敗時
- 最小パスワード長

### スキーマ

詳細は以下のスキーマを参照してください。

- Starter Pack

AccountService.v1\_0\_2.json

HpeiLOAccountService.v2\_0\_0.json

- iLO ファームウェア

/redfish/v1/SchemaStore/en/AccountService.json

/redfish/v1/SchemaStore/en/HpeiLOAccountService.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/AccountService/Accounts/

ローカルユーザーを新規に作成することができます。

### POST

ローカルユーザーを新規に作成します。

- ユーザー名

- パスワード
- ログイン名
- 権限

例：

POST https://{iLO アドレス}/redfish/v1/accountservice/accounts/  
Content-Type : application/json

Body :

```
{
  "Oem": {
    "Hpe": {
      "LoginName": "TestUser",
      "Privileges": {
        "HostBIOSConfigPriv": true,
        "HostNICConfigPriv": true,
        "HostStorageConfigPriv": true,
        "LoginPriv": true,
        "RemoteConsolePriv": true,
        "SystemRecoveryConfigPriv": false,
        "UserConfigPriv": true,
        "VirtualMediaPriv": true,
        "VirtualPowerAndResetPriv": true,
        "iLOConfigPriv": true
      }
    }
  },
  "Password": "TestPassword",
  "UserName": "TestUser"
}
```

※各権限の詳細は iLO5 ユーザーズガイドを参照してください。

⇒ユーザーTestUserが追加されます。

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - ManagerAccount.v1\_0\_0.json
  - HpeiLOAccount.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/ManagerAccount.json
  - /redfish/v1/SchemaStore/en/HpeiLOAccount.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/AccountService/Accounts/{item}/

ローカルユーザーの参照と設定変更をすることができます。

### GET・PATCH

- ユーザー名
- パスワード
- ログイン名
- 権限

### DELETE

例：1番目のユーザーが削除。

```
DELETE https://{iLO アドレス}/redfish/v1/accountservice/accounts/1/  
Body  
なし
```

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
ManagerAccount.v1\_0\_0.json  
HpeiLOAccount.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/ManagerAccount.json  
/redfish/v1/SchemaStore/en/HpeiLOAccount.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Chassis/1/

サーバーのシャーシに関する情報を取得できます。

### GET

- Manufacture 名
- モデル名
- 型番
- シリアル番号

- サーバーヘルスステータス

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - Chassis.v1\_2\_0.json
  - HpeServerChassis.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/Chassis.json
  - /redfish/v1/SchemaStore/en/HpeServerChassis.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Chassis/1/Power/

サーバーの電力に関する設定の参照と変更をすることができます。パワーサプライの情報を取得できます。

## GET・PATCH

- 電力メーター有効/無効(Read Only)
- 高効率モード(Read Only)
- 電力しきい値超過による SNMP アラート設定
- 最大利用可能電力(Read Only)
- 消費電力現在値(Read Only)
- 消費電力上限設定
- 消費電力測定間隔(Read Only)
- 測定間隔内での最大/最小/平均消費電力(Read Only)
- パワーサプライ情報(Read Only)

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - Power.v1\_2\_1.json
  - HpePowerMetricsExt.v2\_0\_0.json



HpeServerPowerSupply.v2\_0\_0.json

- iLO ファームウェア

/redfish/v1/SchemaStore/en/Power.json

/redfish/v1/SchemaStore/en/HpePowerMetricsExt.json

/redfish/v1/SchemaStore/en/HpeServerPowerSupply.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Chassis/1/Power/FastPowerMeter/

過去 20 分の消費電力の履歴を取得できます。

### GET

- 過去 20 分間の最大消費電力
- 過去 20 分間の最小消費電力
- 環境温度
- 10 秒ごとの環境温度(Ambient 温度センサー)
- 10 秒ごとの平均消費電力
- 10 秒ごとの最小消費電力
- 10 秒ごとのピーク消費電力
- 消費電力上限設定値
- 測定日時

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack

HpePowerMeter.v2\_0\_0.json

- iLO ファームウェア

/redfish/v1/SchemaStore/en/HpePowerMeter.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Chassis/1/Power/PowerMeter/

過去 24 時間の消費電力の履歴を取得できます。

### GET

- 過去 24 時間の最大消費電力
- 過去 24 時間の最小消費電力
- 環境温度
- 5 分ごとの環境温度(Ambient 温度センサー)
- 5 分ごとの平均消費電力
- 5 分ごとの最小消費電力
- 5 分ごとのピーク消費電力
- 消費電力上限設定値
- 測定日時

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpePowerMeter.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpePowerMeter.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Chassis/1/Power/Thermal/

ファンの情報と温度センサーの情報を取得できます。

### GET

- ファンスピード(%)
- ファンステータス
- 温度センサー
- 温度ステータス
- 温度センサー閾値

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - Thermal.v1\_1\_0.json
  - HpeServerFan.v2\_0\_0.json
  - HpeSeaOfSensors.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/Thermal.json
  - /redfish/v1/SchemaStore/en/HpeServerFan.json
  - /redfish/v1/SchemaStore/en/HpeSeaOfSensors.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/

iLO でサポートする機能の基本設定の参照と変更を行えます。また、iLO のリセットを行えます。

## GET・PATCH

- SMASHCLP 有効/無効
- リモートコンソール有効/無効
- iLO Firmware バージョン(Read Only)
- フロントパネル USB (iLO Service Port) ステータス(Read Only)
- アイドル接続タイムアウト設定
- マウス、キーボードの持続接続設定
- iLO RBSU(BMC Configuration Utility)へのログイン要求
- シリアルコマンドラインインターフェース速度
- シリアルコマンドラインインターフェースステータス
- 仮想シリアルポートログ
- Web インターフェース有効/無効
- iLO RBSU(BMC Configuration Utility)有効/無効
- iLO セルフテスト結果(Read Only)

- iLO サービスポート設定
- シリアルコンソール(SOL)有効/無効
- UUID

## POST

### iLO のリセット

```
POST https://{iLO アドレス}/redfish/v1/Managers/1/
Content-Type : application/json
Body :
{
  "Action" : "Manager.Reset"
}
```

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - Manager.v1\_1\_0.json
  - HpeiLO.v2\_1\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/Manager.json
  - /redfish/v1/SchemaStore/en/HpeiLO.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/EthernetInterfaces/1/

iLO 専用ネットワークポート設定の参照と変更を行えます。

## GET・PATCH

- iLO 専用ネットワークポート有効/無効
- Auto Negotiation 有効/無効
- リンク速度 (Auto Negotiation 有効時は変更不可)
- Full Duplex 有効/無効 (Auto Negotiation 有効時は変更不可)
- FQDN(Read Only)※変更する際はドメイン名とホスト名を修正してください。
- ドメイン名
- ホスト名

- MAC アドレス(Read Only)
- DHCPv4 設定
- IPv4 アドレス
- IPv4 デフォルトゲートウェイ
- IPv4 DNS サーバー設定
- IPv4 静的経路構成設定
- WINS サーバー設定
- ゲートウェイ PING 有効/無効
- IPv6 有効/無効
- DHCPv6 設定
- IPv6 アドレス（動的割り当て）(Read Only)
- IPv6 アドレス（固定割り当て）
- IPv6 デフォルトゲートウェイ
- IPv6 DNS サーバー設定
- IPv6 SLAAC 有効/無効
- IPv6 静的経路構成設定

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
EthernetInterface.v1\_0\_3.json  
HpeiLOEthernetNetworkInterface.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/EthernetInterface.json  
/redfish/v1/SchemaStore/en/HpeiLOEthernetNetworkInterface.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Managers/1/EthernetInterfaces/2/

iLO 共有ネットワークポート設定の参照と変更を行えます。

**GET ・ PATCH**

- iLO 共用ネットワークポート有効/無効
- 使用ポート設定
- Auto Negotiation 有効/無効
- リンク速度 (Auto Negotiation 有効時は変更不可)
- Full Duplex 有効/無効 (Auto Negotiation 有効時は変更不可)
- VLAN 設定
- FQDN(Read Only)※変更する際はドメイン名とホスト名を修正してください。
- ドメイン名
- ホスト名
- MAC アドレス(Read Only)
- DHCPv4 設定
- IPv4 アドレス
- IPv4 デフォルトゲートウェイ
- IPv4 DNS サーバー設定
- IPv4 静的経路構成設定
- WINS サーバー設定
- ゲートウェイ PING 有効/無効

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - EthernetInterface.v1\_0\_3.json
  - HpeiLOEthernetNetworkInterface.v2\_0\_0.json
  - VlanNetworkInterface.v1\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/EthernetInterface.json
  - /redfish/v1/SchemaStore/en/HpeiLOEthernetNetworkInterface.json
  - /redfish/v1/SchemaStore/en/VlanNetworkInterface.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/NetworkService/

iLO が提供する各種サービスについての設定の参照と取得を行えます。

### GET・PATCH

- HTTP 設定
- HTTPS 設定
- IPMI 設定
- リモートコンソール設定
- 仮想メディア設定
- アラートメール設定
- リモートシスログ設定
- SNMP 設定（待ち受けポートと有効/無効のみ）
- SSH 設定
- iLO 連携マルチキャスト検出設定
- 仮想シリアルポートログ設定

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - ManagerNetworkProtocol.v1\_0\_0.json
  - HpeiLOManagerNetworkService.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/ManagerNetworkProtocol.json
  - /redfish/v1/SchemaStore/en/HpeiLOManagerNetworkService.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/NetworkService/Actions/Oem/Hpe/HpeiLOManagerNetworkService.SendTestAlertMail/

アラートメールのテストを行えます。

### POST

```
POST https://{iLO アドレス}/redfish/v1/Managers/1/NetworkService/
  Actions/Oem/Hpe/HpeiLOManagerNetworkService.SendTestAlertMail/
Content-Type : application/json
Body :
{ }
```

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
ManagerNetworkProtocol.v1\_0\_0.json  
HpeiLOManagerNetworkService.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/ManagerNetworkProtocol.json  
/redfish/v1/SchemaStore/en/HpeiLOManagerNetworkService.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Managers/1/NetworkService/NetworkService/Actions/Oem/Hpe/HpeiLOManagerNetworkService.SendTestSyslog/

リモート Syslog のテストを行えます。

## POST

```
POST https://{iLO アドレス}/redfish/v1/Managers/1/NetworkService/
  Actions/Oem/Hpe/HpeiLOManagerNetworkService.SendTestSyslog/
Content-Type : application/json
Body :
{ }
```

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
ManagerNetworkProtocol.v1\_0\_0.json  
HpeiLOManagerNetworkService.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/ManagerNetworkProtocol.json  
/redfish/v1/SchemaStore/en/HpeiLOManagerNetworkService.json



※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/SnmpService/

SNMP 通報設定の参照と取得を行えます。

### GET・PATCH

- SNMP 設定

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - HpeiLOSnmpService.v2\_0\_0.json
  - HpeiLOSnmpServiceExt.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/HpeiLOSnmpService.json
  - /redfish/v1/SchemaStore/en/HpeiLOSnmpServiceExt.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/SnmpService/Actions/HpeiLOSnmpService.SendSNMPTestAlert/

SNMP 通報のテストを行えます。

### POST

```
POST https://{iLO アドレス}/redfish/v1/Managers/1/SnmpService/
  Actions/HpeiLOSnmpService.SendSNMPTestAlert/
Content-Type : application/json
Body :
{ }
```

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - HpeiLOSnmpService.v2\_0\_0.json

HpeiLOSnmpServiceExt.v2\_0\_0.json

- iLO ファームウェア

/redfish/v1/SchemaStore/en/HpeiLOSnmpService.json

/redfish/v1/SchemaStore/en/HpeiLOSnmpServiceExt.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Managers/1/VirtualMedia/1/

仮想フロッピー/USB キーの情報を参照することができます。

#### GET・PATCH

- イメージ名
- 接続状況
- 接続元 (NotConnected/URI/Applet)

#### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack

VirtualMedia.v1\_0\_0.json

HpeiLOVirtualMedia.v2\_0\_0.json

- iLO ファームウェア

/redfish/v1/SchemaStore/en/VirtualMedia.json

/redfish/v1/SchemaStore/en/HpeiLOVirtualMedia.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Managers/1/VirtualMedia/1/Actions/Oem/Hpe/HpeiLOVirtualMedia.InsertVirtualMedia/

フロッピーイメージ(スクリプト方式)を挿入します。

#### POST

POST https://{iLO アドレス}/redfish/v1/Managers/1/VirtualMedia/1/  
Actions/Oem/Hpe/HpeiLOVirtualMedia.InsertVirtualMedia/

```
Content-Type : application/json
Body :
{
  "Image": "protocol://username:password@hostname:port/filename"
}
```

protocol の値は必須で、HTTP または HTTPS でなければなりません。

hostname と filename (img ファイル) の値は必須です。

username:hostname と port の値はオプションです。

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - VirtualMedia.v1\_0\_0.json
  - HpeiLOVirtualMedia.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/VirtualMedia.json
  - /redfish/v1/SchemaStore/en/HpeiLOVirtualMedia.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Managers/1/VirtualMedia/1/Actions/Oem/Hpe/HpeiLOVirtualMedia.EjectVirtualMedia/

フロッピーイメージ(スクリプト方式)を取り出します。

## POST

```
POST https://{iLO アドレス}/redfish/v1/Managers/1/VirtualMedia/1/
  Actions/Oem/Hpe/HpeiLOVirtualMedia.EjectVirtualMedia/
Content-Type : application/json
Body :
{ }
```

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - VirtualMedia.v1\_0\_0.json
  - HpeiLOVirtualMedia.v2\_0\_0.json
- iLO ファームウェア

/redfish/v1/SchemaStore/en/VirtualMedia.json

/redfish/v1/SchemaStore/en/HpeiLOVirtualMedia.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/VirtualMedia/2/

仮想 CD/DVD-ROM の情報を参照することができます。

### GET・PATCH

- イメージ名 (Read Only)
- 接続状況 (Read Only)
- 接続元 [NotConnected/URI/Applet] (Read Only)
- BootOnNextServerReset [true/false]

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
VirtualMedia.v1\_0\_0.json  
HpeiLOVirtualMedia.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/VirtualMedia.json  
/redfish/v1/SchemaStore/en/HpeiLOVirtualMedia.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/VirtualMedia/2/Actions/Oem/Hpe/HpeiLOVirtualMedia.InsertVirtualMedia/

CD/DVD-ROM イメージ (スクリプト方式) を挿入します。

### POST

```
POST https://{iLO アドレス}/redfish/v1/Managers/1/VirtualMedia/2/  
Actions/Oem/Hpe/HpeiLOVirtualMedia.InsertVirtualMedia/  
Content-Type : application/json
```

```
Body :
{
  "Image": "protocol://username:password@hostname:port/filename"
}
```

protocol の値は必須で、HTTP または HTTPS でなければなりません。

hostname と filename (iso ファイル) の値は必須です。

username:hostname と port の値はオプションです。

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - VirtualMedia.v1\_0\_0.json
  - HpeiLOVirtualMedia.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/VirtualMedia.json
  - /redfish/v1/SchemaStore/en/HpeiLOVirtualMedia.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

`/redfish/v1/Managers/1/VirtualMedia/2/Actions/Oem/Hpe/HpeiLOVirtualMedia.EjectVirtualMedia/`

CD/DVD-ROM イメージ(スクリプト方式)を取り出します。

## POST

```
POST https://{iLO アドレス}/redfish/v1/Managers/1/VirtualMedia/2/
  Actions/Oem/Hpe/HpeiLOVirtualMedia.EjectVirtualMedia/
```

Content-Type : application/json

```
Body :
{ }
```

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - VirtualMedia.v1\_0\_0.json
  - HpeiLOVirtualMedia.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/VirtualMedia.json

/redfish/v1/SchemaStore/en/HpeiLOVirtualMedia.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/

システムの基本的な情報の参照と変更を行えます。

### GET・PATCH

- Asset タグ
- BIOS バージョン(Read Only)
- ワンタイムブート設定
- UID LED
- Manufacturer 名(Read Only)
- モデル名(Read Only)
- シリアル番号(Read Only)
- 製品 ID (Read Only)
- ヘルスサマリー(Read Only)
- システム電源リストア設定
- パワーレギュレーター設定
- システム電源状態(Read Only)
- TPM ステータス(Read Only)
- UUID(Read Only)

### POST

#### 電源オン

POST https://{iLO アドレス}/redfish/v1/Systems/1/

Content-Type: application/json

Body :

```
{  
  "Action" : "ComputerSystem.Reset",  
  "ResetType" : "On"  
}
```

#### 電源強制オフ

POST https://{iLO アドレス}/redfish/v1/Systems/1/

```
Content-Type: application/json
```

```
Body :
```

```
{  
  "Action" : "ComputerSystem.Reset",  
  "ResetType" : "ForceOff"  
}
```

### 強制システムリセット

```
POST https://{iLO アドレス}/redfish/v1/Systems/1/
```

```
Content-Type: application/json
```

```
Body :
```

```
{  
  "Action" : "ComputerSystem.Reset",  
  "ResetType" : "ForceRestart"  
}
```

### NMI

```
POST https://{iLO アドレス}/redfish/v1/Systems/1/
```

```
Content-Type: application/json
```

```
Body :
```

```
{  
  "Action" : "ComputerSystem.Reset",  
  "ResetType" : "Nmi"  
}
```

### 安全な電源オフ

```
POST https://{iLO アドレス}/redfish/v1/Systems/1/
```

```
Content-Type: application/json
```

```
Body :
```

```
{  
  "Action" : "ComputerSystem.Reset",  
  "ResetType" : "PushPowerButton"  
}
```

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - ComputerSystem.v1\_3\_0.json
  - HpeComputerSystemExt.v2\_2\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/ComputerSystem.json
  - /redfish/v1/SchemaStore/en/HpeComputerSystemExt.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/Logservices/IML/

インテグレーションマネージメントログ (IML) をクリアすることができます。

### POST

```
POST https://{iLO アドレス}/redfish/v1/Systems/1/LogServices/IML/  
Content-Type: application/json  
Body :  
{  
  "Action" : "LogService.ClearLog"  
}
```

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
LogService.v1\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/LogService.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/LogServices/IML/Entries/

インテグレーションマネージメントログ (IML) の参照先一覧を取得することができます。

### GET

- IML の参照先一覧
- IML のログ数

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
LogEntryCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/LogEntryCollection.json



※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/LogServices/IML/Entries/{item}/

インテグレートドマネージメントログ (IML) の詳細を参照することができます。

### GET

- 説明
- カテゴリ
- クラス
- コード
- 回数
- ID
- 最終更新
- 深刻度

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - LogEntry.v1\_0\_0.json
  - HpeLogEntry.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/LogEntry.json
  - /redfish/v1/SchemaStore/en/HpeLogEntry.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/LogServices/IEL/

iLO イベントログ (IEL) をクリアすることができます。

### POST

```
POST https://{iLO アドレス}/redfish/v1/Managers/1/LogServices/IEL/  
Content-Type: application/json
```

```
Body :
{
  "Action": "LogService.ClearLog"
}
```

## スキーマ

詳細は以下のスキーマファイルを参照してください。

ファイル名のバージョンは、配布版と異なる場合があります。

- Starter Pack  
LogService.v1\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/LogService.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/LogServices/IEL/Entries/

iLO イベントログ (IEL) の参照先一覧を取得することができます。

### GET

- IEL の参照先一覧
- IEL のログ数

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
LogEntryCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/LogEntryCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/LogServices/IEL/Entries/{item}/

iLO イベントログ (IEL) の詳細を参照することができます。

## GET

- 説明
- カテゴリー
- クラス
- コード
- 回数
- ID
- 最終更新
- 深刻度

## スキーマ

詳細は以下のスキーマファイルを参照してください。

ファイル名のバージョンは、配布版と異なる場合があります。

- Starter Pack
  - LogEntry.v1\_0\_0.json
  - HpeLogEntry.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/LogEntry.json
  - /redfish/v1/SchemaStore/en/HpeLogEntry.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/Memory/

アドバンスドメモリプロテクション (AMP) 情報、実装されたメモリの実装位置、サイズ、速度、各 DIMM スロット情報への参照先一覧を取得することができます。

## GET

- AMP モードステータス
- 構成済み AMP モード
- サポートされる AMP モード
- 実装されたメモリー覧
- DIMM スロット数

- 各 DIMM スロット情報への参照先一覧

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - MemoryCollection.json
  - HpeAdvancedMemoryProtection.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/MemoryCollection.json
  - /redfish/v1/SchemaStore/en/HpeAdvancedMemoryProtection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

</redfish/v1/Systems/1/Memory/{item}/>

各 DIMM スロットに実装されたメモリの詳細を参照することができます。

## GET

- 搭載されているスロット、プロセッサ
- ステータス
- メモリモジュールのサイズ
- メモリモジュールの速度
- メモリモジュールのテクノロジー
- メモリモジュールのタイプ
- メモリモジュールのランク数
- メモリモジュールのエラー訂正
- メモリモジュールのデータ幅ビット
- メモリモジュールのバス幅ビット
- メモリチャネル番号
- メモリコントローラー番号
- メモリモジュールのベンダーID

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
Memory.v1\_1\_0.json  
HpeMemoryExt.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/Memory.json  
/redfish/v1/SchemaStore/en/HpeMemoryExt.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/NetworkAdapters/

実装されたネットワークアダプターの参照先一覧を取得することができます。

### GET

- ネットワークアダプターの参照先一覧
- ネットワークアダプターの数

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeBaseNetworkAdapterCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeBaseNetworkAdapterCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/NetworkAdapters/{item}/

実装されたネットワークアダプターの詳細を参照することができます。

### GET

- アダプター番号
- ネットワークアダプター名

- ファームウェアのバージョン
- ネットワークアダプターのステータス
- 各ポートのデュプレックス
- 各ポートの MAC アドレス
- 各ポートの送受信情報

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - HpeBaseNetworkAdapter.v2\_0\_0.json
  - HpeBaseNetworkAdapterExt.v2\_0\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/HpeBaseNetworkAdapter.json
  - /redfish/v1/SchemaStore/en/HpeBaseNetworkAdapterExt.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/PCIDevices/

PCI デバイスの参照先一覧を取得することができます。

## GET

- PCI デバイスの参照先一覧
- PCI デバイスの数

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - HpeServerPciDeviceCollection.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/HpeServerPciDeviceCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/PCIDevices/{item}/

PCI デバイスの詳細を参照することができます。

### GET

- デバイス名
- PCI バス番号
- PCI クラスコード
- デバイス ID
- デバイスタイプ
- インスタンス
- タイプ
- ファンクション番号
- PCI 構造体名
- UEFI デバイスパス
- ベンダーID

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeServerPciDevice.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeServerPciDevice.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/Processors/

実装されたプロセッサの参照先一覧を取得することができます。

### GET

- 実装されたプロセッサの参照先一覧
- 実装されたプロセッサの数

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
ProcessorCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/ProcessorCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/Processors/{item}/

実装されたプロセッサの詳細を参照することができます。

### GET

- プロセッサ名
- プロセッサのステータス
- プロセッサ速度
- プロセッサのコアおよびスレッド情報
- プロセッサのメモリ機能
- キャッシュサイズ

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
Processor.v1\_0\_0.json  
HpeProcessorExt.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/Processor.json  
/redfish/v1/SchemaStore/en/HpeProcessorExt.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。



## /redfish/v1/Systems/1/SmartStorage/ArrayControllers/

Smart アレイの参照先一覧を取得することができます。

GET
<ul style="list-style-type: none"><li>Smart アレイ一覧</li></ul>
スキーマ
詳細は以下のスキーマファイルを参照してください。 <ul style="list-style-type: none"><li>Starter Pack HpeSmartStorageArrayControllerCollection.json</li><li>iLO ファームウェア /redfish/v1/SchemaStore/en/HpeSmartStorageArrayControllerCollection.json</li></ul> ※ファイル名のバージョンは異なる場合があります。 ※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}//

Smart アレイの詳細情報を参照することができます。

GET
<ul style="list-style-type: none"><li>位置</li><li>ステータス</li><li>シリアル番号</li><li>モデル</li><li>ファームウェアバージョン</li><li>コントローラタイプ</li><li>キャッシュモジュールのステータス</li><li>キャッシュモジュールのメモリサイズ</li><li>暗号化ステータス</li><li>暗号化 ASIC</li><li>暗号化クリティカルセキュリティパラメータ NVRAM ステータス</li></ul>
スキーマ
詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeSmartStorageArrayController.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeSmartStorageArrayController.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/LogicalDrives/

論理ドライブの参照先一覧を取得することができます。

#### GET

- 論理ドライブ一覧

#### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeSmartStorageLogicalDriveCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeSmartStorageLogicalDriveCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/LogicalDrives/{item}/

論理ドライブの情報を取得することができます。

#### GET

- 論理ドライブ番号
- ステータス
- 容量
- フォールトトレランス
- 論理ドライブのタイプ
- 暗号化ステータス

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeSmartStorageLogicalDrive.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeSmartStorageLogicalDrive.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

`/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/DiskDrives/`

物理ドライブの参照先一覧を取得することができます。

## GET

- 物理ドライブ一覧

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeSmartStorageDiskDriveCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeSmartStorageDiskDriveCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

`/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/DiskDrives/{item}/`

物理ドライブの情報を取得することができます。

## GET

- ステータス
- シリアル番号
- モデル
- メディアタイプ

- 容量
- 位置
- ファームウェアバージョン
- ドライブの構成
- 暗号化ステータス

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeSmartStorageDiskDrive.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeSmartStorageDiskDrive.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/StorageEnclosures/

ドライブエンクロージャの参照先一覧を取得することができます。

## GET

- ドライブエンクロージャ一覧

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeSmartStorageStorageEnclosureCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeSmartStorageStorageEnclosureCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/StorageEnclosures/{item}/

ドライブエンクロージャの情報を取得することができます。

#### GET

- 位置
- ステータス
- ドライブベイ数
- シリアル番号
- モデル
- ファームウェアバージョン

#### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeSmartStorageStorageEnclosureCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeSmartStorageStorageEnclosure.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Systems/1/SmartStorage/ArrayControllers/{item}/UnconfiguredDrives/

未構成物理ドライブの参照先一覧を取得することができます。

#### GET

- 未構成物理ドライブ一覧

#### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeSmartStorageDiskDriveCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeSmartStorageDiskDriveCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください

## /redfish/v1/Systems/1/SmartStorage/HostBusAdapters/

直接接続ストレージの参照先一覧を取得することができます。

### GET

- 直接接続ストレージ一覧

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeSmartStorageHostBusAdapterCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeSmartStorageHostBusAdapterCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Systems/1/SmartStorage/HostBusAdapters/{item}/

直接接続ストレージの情報を取得することができます。

### GET

- コントローラー位置
- コントローラーステータス
- シリアル番号
- モデル
- ファームウェアバージョン

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeSmartStorageHostBusAdapter.v2\_0\_0.json

- iLO ファームウェア

/redfish/v1/SchemaStore/en/HpeSmartStorageHostBusAdapter.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/Systems/1/SmartStorage/HostBusAdapters/{item}/DiskDrives/

物理ドライブの参照先一覧を取得することができます。

#### GET

- 物理ドライブ一覧

#### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack

HpeSmartStorageDiskDriverCollection.json

- iLO ファームウェア

/redfish/v1/SchemaStore/en/HpeSmartStorageDiskDriveCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

/redfish/v1/UpdateService/

ファームウェアの更新および更新のステータスを参照することができます。

#### GET

- 更新のステータス
- 更新の進捗

#### POST

POST https://{iLO アドレス}/redfish/v1/UpdateService/

Content-Type: application/json

Body :

```
{
  "Action": "UpdateService.SimpleUpdate",
  "ImageURI": "protocol://username:password@hostname:port/filename"
}
```

protocol の値は必須で、HTTP または HTTPS でなければなりません。

hostnameとfilenameの値は必須です。  
username:hostnameとportの値はオプションです。

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
UpdateService.v1\_1\_0.json  
HpeiLOUpdateServiceExt.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/UpdateService.json  
/redfish/v1/SchemaStore/en/HpeiLOUpdateServiceExt.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/UpdateService/FirmwareInventory/

インストールされたファームウェアの参照先一覧を取得することができます。

## GET

- インストールされたファームウェアの参照先一覧
- インストールされたファームウェアの数

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
SoftwareInventoryCollection.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/SoftwareInventoryCollection.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/UpdateService/FirmwareInventory/{item}/

インストールされたファームウェアの詳細を参照することができます。



## GET

- ファームウェア名
- ファームウェアバージョン
- 位置

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
SoftwareInventory.v1\_0\_0.json  
HpeiLOSoftwareInventory.v2\_0\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/SoftwareInventory.json  
/redfish/v1/SchemaStore/en/HpeiLOSoftwareInventory.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/Actions/Oem/Hpe/HpeiLO.ClearRestApiState/

iLO の RESTful API のリセットを行えます。

## POST

```
POST https://{iLO アドレス}/redfish/v1/Managers/1/Actions/Oem/Hpe/  
HpeiLO.ClearRestApiState/  
Content-Type : application/json  
Body :  
{  
  "Action" : "HpeiLO.ClearRestApiState"  
}
```

## スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack  
HpeiLO.v2\_1\_0.json
- iLO ファームウェア  
/redfish/v1/SchemaStore/en/HpeiLO.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## /redfish/v1/Managers/1/Actions/Oem/Hpe/HpeiLO.ResetToFactoryDefaults/

工場出荷時デフォルト設定への iLO のリセットをすることができます。

注意: iLO を出荷時のデフォルト設定にリセットすると、ユーザーデータ、ライセンスデータ、構成設定、ログなど、すべての iLO 設定が消去されます。工場出荷時にライセンスキーがインストールされている場合には、ライセンスキーは保持されます。

### POST

```
POST https://{iLO アドレス}/redfish/v1/Managers/1/Actions/Oem/Hpe/  
    HpeiLO.ResetToFactoryDefaults/  
Content-Type: application/json  
Body :  
{  
    "ResetType" : "Default"  
}
```

### スキーマ

詳細は以下のスキーマファイルを参照してください。

- Starter Pack
  - Manager.v1\_1\_0.json
  - HpeiLO.v2\_1\_0.json
- iLO ファームウェア
  - /redfish/v1/SchemaStore/en/Manager.json
  - /redfish/v1/SchemaStore/en/HpeiLO.json

※ファイル名のバージョンは異なる場合があります。

※ご使用中の iLO ファームウェアバージョンに対応したスキーマを参照してください。

## 用語集

<b>3DES</b>	トリプル DES。Data Encryption Standard 暗号化アルゴリズム
<b>AES</b>	Advanced Encryption Standard
<b>AMP</b>	Advanced Memory Protection (AMP)は、搭載メモリに対してミラーリング等の制御をすることにより、強固な耐障害性を実現する技術です。
<b>API</b>	Application Programming Interface。アプリケーションプログラミングインターフェース
<b>BIOS</b>	Basic Input/Output System。基本入出力システム
<b>BMC</b>	Baseboard management controller
<b>CLP</b>	Command Line Protocol。コマンドラインプロトコル
<b>DCMI</b>	Data Center Manageability Interface。データセンター管理インターフェース
<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>DIMM</b>	Dual In-line Memory Module。デュアルインラインメモリモジュール。メモリチップを保持する小型回路基板。
<b>DMTF</b>	Distributed Management Task Force
<b>DNS</b>	Domain Name System。ドメインネームシステム
<b>DSA</b>	Digital Signature Algorithm。デジタル署名アルゴリズム
<b>FIPS</b>	Federal Information Processing Standard。連邦情報処理標準。
<b>FQDN</b>	Fully Qualified Domain Name。完全修飾ドメイン名
<b>iLO</b>	Integrated Lights-Out。標準インターフェース仕様の IPMI2.0 に準拠してハードウェアを監視する BMC です。
<b>IML</b>	Integrated Management Log。インテグレートドマネージメントログ
<b>IPMI</b>	Intelligent Platform Management Interface
<b>IRC</b>	Integrated Remote Console。統合リモートコンソール
<b>ISO</b>	International Organization for Standardization。国際標準化機構
<b>JSON</b>	JavaScript Object Notation。JavaScript オブジェクトの表記法
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>MAC</b>	Media Access Control
<b>MD5</b>	Message-Digest algorithm 5
<b>MSA</b>	Mail Submission Agent
<b>NIC</b>	Network interface card。ネットワークインターフェースカード。ネットワーク経由のデバイス間の通信を処理するデバイス。
<b>NMI</b>	Non-maskable interrupt。マスク不可能割り込み

<b>PuTTY</b>	SSH、Telnet、rlogin、およびロー TCP プロトコルのクライアントならびにシリアルコンソールクライアントとして機能できる端末エミュレーター。
<b>RBSU</b>	ROM-Based Setup Utility。ROM ベースセットアップユーティリティ。
<b>REST</b>	Representational State Transfer
<b>RSA</b>	パブリックキー暗号化用のアルゴリズム
<b>SHA</b>	Secure Hash Algorithm。セキュアハッシュアルゴリズム
<b>SLAAC</b>	Stateless Address Autoconfiguration
<b>SMASH</b>	Systems Management Architecture for Server Hardware
<b>SNMP</b>	Simple Network Management Protocol。簡易ネットワーク管理プロトコル
<b>SSH</b>	Secure Shell
<b>SSL</b>	Secure Sockets Layer
<b>TLS</b>	Transport layer security。トランスポート層セキュリティ
<b>TPM</b>	Trusted Platform Module
<b>UEFI</b>	Unified Extensible Firmware Interface
<b>UID</b>	Unit identification。ユニット識別子
<b>USB</b>	Universal serial bus。ユニバーサルシリアルバス。デバイスを接続するために使用されるシリアルバス規格。
<b>UUID</b>	Universally Unique Identifier。ユニバーサル一意識別子
<b>VSP</b>	Virtual Serial Port。仮想シリアルポート
<b>WINS</b>	Windows インターネットネームサービス

NEC NX7700x サーバ

iLO 5 スクリプティングコマンドラインガイド

2019 年 9 月 Rev.1.00

日本電気株式会社

東京都港区芝五丁目 7 番 1 号

TEL ( 03 ) 3454-1111 ( 大代表 )

落丁、乱丁はお取り替えいたします

© NEC Corporation 2019

日本電気株式会社の許可なく複製・改変などを行うことはできません。