Smart Update Manager CLI ガイド

部品番号: 30-2D16D6D4-002-ja-JP
発行: 2026 年 1 月
版数: 2

Smart Update Manager CLI ガイド

摘要

このガイドでは、SUM の CLI モードおよび入力ファイルモードを使用して、HPE ProLiant および Integrity サーバーにファームウェアアップデートを適用し、HPE ProLiant、HPE BladeSystem、HPE Synergy、HPE EdgeLine、および HPE Apollo のサーバーとインフラストラクチャにシステムソフトウェアアップデートを適用する方法について説明します。このガイドは、Microsoft Windows、Windows Server、Linux、Smart コンポーネント、VMware の構成および操作、ならびにアップデートの実行に伴うデータ消失の危険性について理解している担当者を対象にしています。

部品番号: 30-2D16D6D4-002-ja-JP
発行: 2026 年 1 月
版数: 2

ご注意

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。Hewlett Packard Enterprise 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、Hewlett Packard Enterprise 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、Hewlett Packard Enterprise の Web サイトの外に移動します。 Hewlett Packard Enterprise は、Hewlett Packard Enterprise の Web サイト以外の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

Adobe、Adobe ロゴ、Acrobat、および Adobe PDF ロゴは、米国またはその他の国における Adobe の登録商標または商標です。

AMD、AMD Arrow 記号、ATI、および ATI ロゴは、Advanced Micro Devices, Inc. の商標です。

Ampere®、Altra®、A®、および Ampere® ロゴは、Ampere Computing の登録商標または商標です。

目次

SUM CLI について

サブトピック

SUM CLI モードと入力ファイルモード

SUM の CLI モードと入力ファイルモードを使用して、ローカルおよびリモートのノードにベースラインからのアップデートを展開できます。

CLI モードでは、1 つのコマンドラインにすべてのパラメーターを追加し、ユーザーが操作することなく複数のノードでシーケンス全体を実行します。この方法では、`silent` コマンドが必要です。1 つのノードをアップデートする場合は、この方法を使用します。

入力ファイルモードを使用してすべてのパラメーターを 1 つのテキストファイルに追加し、このテキストファイルの名前を `inputfile` コマンドのパラメーターとして指定して、SUM を呼び出します。この方法では、`silent` コマンドが必要です。複数のノードをアップデートする場合は、この方法を使用します。

SUM の CLI モードと入力ファイルモードは、プロセス指向のツールです。SUM はコマンドを終了した後、その設定や結果を保存しません。いくつかの手順を実行し、その作業を後のために保存する必要がある場合は、GUI モードを使用します。

SUM CLI モード

注記
SUM 8.5.0 以降、対話型コマンドラインモードのアップデートは非推奨になりました。スクリプトが正しく機能するようにするには、対話型 CLI からレガシーコマンドラインに移行します。入力ファイルを含むコマンドラインは、ほとんどの顧客のニーズを満たすでしょう。また、コマンドライン入力や入力ファイルを変更することなく、SPP および SUM バージョン全体で大規模なアップデートを行うことができます。

使用しているサーバー環境に最も適した SUM の CLI モードを選択してください。

CLI

ノード数：1~3

長所：

- 1 つのコマンドでノードに対するすべての指示を与えます。

入力ファイル CLI

ノード数：1～50

長所：

- ベースラインとノードのすべての情報が含まれているファイルを作成します。
- 後で使用するためにファイルを保存します。
- 入力ファイルを呼び出す CLI コマンドを発行します。

アップデートのスケジュール

標準のオペレーティングシステムツールを使用して、CLI コマンドの実行スケジュールを設定します。たとえば、Windows ではタスクスケジューラを使用し、Linux システムでは cron を使用します。必ず、コマンドに `s`（サイレント）パラメーターを含めるようにしてください。タスクのスケジュールについて詳しくは、オペレーティングシステムのドキュメントを参照してください。

> **注記**
> 入力ファイルでは、スケジューリングパラメーターをサポートしています。

リモートノードでの Linux の root 認証情報

root 認証情報を提供するか、sudo 権限のあるユーザーを提供するか、リモートノードで root 以外の認証情報と root 認証情報を提供することができます。

スーパーユーザーの機能を使用するには、ユーザーをすべての root 特権を持つスーパーユーザーとして構成します。コンポーネントをアップデートするには root ユーザーとともに root 以外のユーザーを使用します。

サブトピック

Linux の sudo 認証情報を使用するための前提条件

root 認証情報での SSH キーファイルの使用

Linux の sudo 認証情報を使用するための前提条件

- 次のいずれかを指定します。
  - ユーザー名とパスワード
  - ユーザー名と SSH キーのファイルパス（PEM 形式）
- sudo ユーザーに `/var/tmp` ディレクトリへの書き込みアクセスを提供します。
- sudo ユーザーについては、/etc/sudoers ファイルにユーザーを追加します。以下に、各ユーザーの権限と指定値を示します。
  - ユーザー：`<Sudo_user>`（ユーザーアカウントの実際の名前）
  - 権限：`ALL`
  - 指定：`ALL`
- sudo コマンドの実行時にシステムが root ユーザーのパスワードではなく sudo ユーザーのパスワードを求めるように、`/etc/sudoers` ファイルのエントリーを編集します。
- 以下に、`/etc/sudoers` でコメント化または削除する権限を示します。
  - ユーザー：`ALL`
  - 権限：`ALL`
  - 指定：`ALL`

  

  注記

  このオプションは必ずすべてのシステム上で デフォルト の `targetpw` を指定して使用してください。
- ログイン認証情報の SSH キーを sudo と組み合わせて使用するときには、システムがユーザーパスワードを要求しないように、`/etc/sudoers` ファイルのエントリーを次のように編集します。
  - ユーザー：`sudo_user`
  - 権限：`ALL`
  - 指定：`NOPASSWD:ALL`

root 認証情報での SSH キーファイルの使用

このタスクについて

SUM では、パスワードを使用するか、SSH キーを指定することでログインできます。生成される SSH キーにはさまざまなタイプがあります。

SLES 15SP7、RHEL 9.6 以降など Linux の新しいバージョンでは、RSA キーは OpenSSH で非推奨またはデフォルトで無効になっています。`ECDSA` または `Ed25519` を使用することが推奨されます。

引き続き RSA を使用する場合は、`/etc/ssh/sshd_config` で明示的に再度有効にします。有効にするには、オペレーティングシステムのドキュメントを参照してください。

> **注記**
> アップデートするノードで FIPS モードが有効になっている場合は、RSA や ECDSA など FIPS 準拠のキータイプを使用します。FIPS モードについて詳しくは、適切なオペレーティングシステムのドキュメントを参照してください。

手順

. SUM を実行しているノードで、`ssh-keygen` コマンドを使用して必要な SSH キーペアを生成します。SUM は、DSA、RSA、ECDSA、Ed25519 など複数の暗号化キー形式をサポートします。適切なキー形式を選択し、対応するコマンドを使用します。

   a. RSA（デフォルトの OpenSSH 形式）の場合は、`ssh-keygen -t rsa` を使用します。

      これは最新の OpenSSH 形式で RSA キーペアを生成します。

   b. ECDSA キーの場合は、`ssh-keygen -t ecdsa` を使用します。

      これは楕円曲線デジタル署名アルゴリズム（ECDSA）キーペアを生成します。

   c. `Ed25519` キーの場合は、`ssh-keygen -t ed25519` を使用します。

      これは非常に安全で効率的な `Ed25519` キーペアを生成します。

   d. RSA キー（レガシー PEM 形式）の場合は、`ssh-keygen -t rsa -b 2048 -m PEM` を使用します。

      これはレガシー PEM 形式で RSA キーペアを生成します。パスフレーズはオプションです。

   e. DSA キーの場合は、`ssh-keygen -t dsa` を使用します。

これは DSA キーペアを生成します。

> **注記**
>
> DSA は安全性が低いと考えられており、レガシーユースケースにのみ推奨されます。

. プロンプトが表示されたら、キーを適切なディレクトリに保存します。デフォルトでは、キーはホームディレクトリの下の `.ssh` ディレクトリに保存されます。

RSA の場合は `~/.ssh/id_rsa.pub` です。

> **注記**
>
> root ユーザーの場合は、 `/root/.ssh` ディレクトリを使用します。

. 秘密キー付きの RSA の場合は、秘密キーファイルを開いて、キーが PEM 形式であることを確認します。

つまり、 `~/.ssh/id_rsa` を開き、
ファイルの先頭で次のテキストを確認します：

```
-----BEGIN RSA PRIVATE KEY-----
```

. ECDSA、Ed25519、OpenSSH 形式の RSA など他のキー形式の場合は、秘密キーファイルの先頭にある次のテキストを調べることで、キーのタイプを確認できます：

```
-----BEGIN OPENSSH PRIVATE KEY-----
```

. 必要に応じて、アップデートするノードで、ルートレベルに `.ssh` ディレクトリを作成します。ディレクトリでの権限レベルを 700 に設定します。

. アップデートするノードに `.ssh/authorized_keys` ファイルがあるかどうかを確認します。このファイルが存在しない場合は、このファイルを作成または追加します。 `authorized_keys` ファイルに対する権限を 640 に設定します。

> **注記**
>
> ファイルを付加すると、より多くのユーザーに秘密キーを使用する権限が与えられます。

. ホストノードの `.ssh/id_rsa.pub` （RSA キー形式の場合）の内容をコピーし、アップデートするノードの `.ssh/authorized_keys` ファイルに貼り付けます。

. ホストノードと、アップデートするノードの間で SSH パスを開きます。

a. `ssh root@10.0.0.1` と入力します。このパラメーターにより、シェルのパスが開きます。

b. `ssh root@10.0.0.1 uname` と入力します。このパラメーターにより、コマンドが実行され結果が戻されます。

. SUM には、`id_rsa` （RSA キー形式の場合）へのアクセスと、オプションのパスフレーズが必要です。

> **注記**
>
> その他のアプリケーションで PEM 形式の公開キーが必要な場合は、`ssh-keygen -e -f id_rsa.pub > id_rsa_pub.pem` と入力して変換できます。キーを開き、`----- BEGIN SSH2 PUBLIC KEY-----` がある PEM 形式であることを確認します。

タスクの結果

Windows システムで秘密キーを作成するには、PuTTY や PuTTY Key Generator （PuTTYGen） などのアプリケーションを使用します。詳しくは、https://www.digitalocean.com/community/tutorials/how-to-create-ssh-keys-with-putty-to-connect-to-a-vps を参照してください。

ファイルを PEM 形式にエクスポートするには、**Conversion > Export OpenSSH key** を選択します。

SUM の CLI コマンドと入力ファイルコマンドの概要

サブトピック

コマンドライン構文

SUM の CLI モードでは、1 つのコマンドラインにすべてのパラメーターを入力する必要があります。**Enter** キーを押すと、SUM はシーケンス全体を実行します。すべての CLI コマンドの実

行には、コマンドに `s` （サイレント）パラメーターが必要で、`/ s` （Windows）または `--s` （Linux）が必要です。

> **重要**
> コマンドラインモードは、ダブルバイト文字セットをサポートしていません。ダブルバイト文字セットを使用してコマンドラインに入力されたコマンドは、正しく表示されません。

SUM の一般的なコマンドライン構文は、以下のとおりです。

- Windows では、各引数の前にスラッシュ（/）を付けます。

```
smartupdate /s /use_location <baseline_directory>
```

- Linux では、各引数の前に 2 つのハイフン（--）を付けます。

```
smartupdate --s --use_location <baseline_directory>
```

コマンドラインの説明

`smartupdate`

SUM を起動します。

`--s` (Linux) **`/s`** (Windows)

CLI モードを開始します。

その他のインストールパラメーターについては、インストールパラメーターを参照してください。

`--target` (Linux) **`/target`** (Windows)

ターゲットノードの詳細を指定します。

その他のノードパラメーターについては、ノードパラメーターを参照してください。

`--on_failed_dependency` (Linux) **`/on_failed_dependency`** (Windows)

このパラメーターは、依存関係の問題を処理する方法を SUM に指示します。

エラーを無視するその他のパラメーターについては、エラーを無視するパラメーターを参照してください

`--r` (Linux) **`/r`** (Windows)

アップデートを展開した後にノードを再起動するかどうかを SUM に指示します。

その他の再起動パラメーターについては、再起動パラメーターを参照してください

`--v` (Linux) `/v` (Windows)

冗長ログの設定を使用するように SUM に指示します。

その他のログパラメーターについては、ログファイルのパラメーターを参照してください。

`--report` (Linux) `/report` (Windows)

SUM レポートを生成します。

その他のレポートパラメーターについては、レポートパラメーターを参照してください。

`--h` (Linux) `/h` (Windows)

SUM CLI のヘルプファイルを起動します。

入力ファイルの概要

入力ファイルには、構成の詳細とターゲットノード情報を含めます。アップデート用のユーザー認証情報とリモートホスト情報を追加できます。

> **重要**
> 入力ファイルはプレーンテキストファイルです。セキュリティを強化するため、認証情報はファイルから省略し、コマンドラインを使用して認証情報を SUM に渡してください。コマンドラインでユーザー認証情報を指定する場合は、入力ファイル内のすべてのノードで同じユーザー ID とパスワードを使用する必要があります。

入力ファイルで SUM CLI を使用すると、ノードをアップデートする `.txt` ファイルスクリプトを作成できます。入力ファイルを作成するには、プレーンテキストエディターを使用します。これはマークアップ言語と似ており、ヘッダーとトレーラーを一致させる必要があります。リストの値を取ることができるパラメーターには、カンマをリストの区切り文字として使用します。

スクリプトファイルを作成したら、`inputfile` <filename>パラメーターとして SUM コマンドラインにスクリプトファイルを追加します。たとえば入力ファイル `update.in` を実行するには、次のように入力します。

- Linux : `smartupdate --inputfile update.in`
- Windows : `smartupdate /inputfile update.in`

入力ファイルが SUM 実行可能ファイルと同じ場所に保存されていない場合は、そのファイルの場所へのフルパスを追加できます。スペースの入っているパスを指定するには、<filename>フィールドを二重引用符で囲みます。

> **注記**
>
> ファイアウォールポートを開くための入力ファイルパラメーターはありません。ファイアウォールを開く場合は、パラメーターとして `--open_firewall`（Linux）または `/open_firewall`（Windows）を、入力ファイルを呼び出すコマンドに追加します。以下に例を示します。
>
> - Linux: `smartupdate --inputfile update.in --open_firewall --s`
> - Windows: `smartupdate /inputfile update.in /open_firewall /s`

入力ファイルを編集せずに入力ファイルを変更するには、入力ファイルにすべてのコマンドラインオプションを使用します。コマンドラインオプションは、入力ファイル内のコマンドよりも優先されます。

入力ファイルのセクション

入力ファイルには、構成およびターゲット情報セクションが含まれています。入力ファイルでは、テキストの 1 行ごとに 1 つのパラメーターがサポートされます。

```
#Input file sample
SILENT = YES
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
SOURCEPATH = C:\SPP
FORCEALL = YES
REBOOTALLOWED = YES
REBOOTDELAY = 30
REBOOTMESSAGE = “Install complete, server will reboot in 30 seconds”
[TARGETS]
HOST = 192.168.1.1
UID = user
PWD = password
[END]
[TARGETS]
HOST = 192.168.1.2
HOST = 192.168.1.2
```

```
UID = user2
PWD = password2
NO_APP_ACCOUNT = YES
[END]
```

構成セクション

このセクションでは、入力ファイルのすべてのパラメーターを割り当てます。構成セクションの終了部から`TARGETS`セクションを開始します。

シャープ（#）記号で始まる行は注記です。1 行に複数のシャープ記号（#）を使用することはできません。

```
#Input file sample
SILENT = YES
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
SOURCEPATH = C:\SPP
FORCEALL = YES
REBOOTALLOWED = YES
REBOOTDELAY = 30
REBOOTMESSAGE = “Install complete, server will reboot in 30 seconds”
```

ターゲットセクション

このセクションは、アップデートする各ノードを定義します。

```
[TARGETS]
HOST = 192.168.1.1
UID = user
PWD = password
[END]
```

ターゲットセクション

```
[TARGETS_GROUP_HOME]
HOST = 192.168.1.2
HOST = 192.168.1.2
UID = user2
PWD = password2
[END]
```

複数のノードで同じユーザー認証情報を使用する場合、同じ`TARGETS`セクションにすべてのノードをリストします。

`[TARGETS]`タイトルにテキストを追加して、ターゲットの詳細を示すことができます。

SUM CLI コマンドタスクの例

サブトピック

CLI コマンドの例について

次のタスクは、CLI コマンドおよび入力ファイルを構成する方法の例を示します。使用できるパラメーターについて詳しくは、SUM CLI のパラメーターと入力ファイルのパラメーターを参照してください。

詳しくは

SUM CLI パラメーター
入力ファイルのパラメーター

入力ファイルコマンドの発行

手順

1. コマンドラインウィンドウを開きます。
2. SUM が配置されているディレクトリに移動します。
3. 使用しているオペレーティングシステムに応じて、次のコマンドを入力します。
   - Linux : `./ smartupdate --inputfile <path:/inputfile.txt>`

- Windows：`smartupdate /inputfile <path:\inputfile.txt>`

> **注記**
>
> 入力ファイルにパラメーター `SILENT = YES` が含まれていない場合は、CLI に `s`（サイレント）パラメーターを含めます。サイレントパラメーターを指定しない場合、SUM は入力ファイルを処理できません。

ローカルホストのアップデート

次の例では、ローカルホストにベースライン内のすべてのアップデートを適用します。

CLI モードで SUM と同じディレクトリにあるベースラインを使用する

- Linux：`./ smartupdate --silent`
- Windows：`smartupdate /silent`

CLI モードで SUM と異なるディレクトリにあるベースラインを使用する

- Linux：`./ smartupdate --silent --use_location <directorypath>`
- Windows：`smartupdate /silent /use_location <directorypath>`

入力ファイルモードで SUM と同じディレクトリにあるベースラインを使用する

```
SILENT = YES
[TARGETS]
HOST = localhost
UID = <userid>
PWD = <password>
[END]
```

SUM では、Windows ドメイン（たとえば、`domain1/userid1`）がサポートされています。

入力ファイルモードで SUM と異なるディレクトリにあるベースラインを使用する

```
SILENT = YES
SOURCEPATH = <baseline_path>
[TARGETS]
HOST = localhost
UID = <userid>
PWD = <password>
[END]
```

リモートノードのアップデート

次の例では、リモートノードにベースライン内のすべてのアップデートを適用します。

CLI モードで SUM を実行しているディレクトリと同じディレクトリからベースラインを使用する

- Linux：`./ smartupdate --s --target <ip_address> --targettype <type> --user <userid> --passwd <password>`
- Windows：`smartupdate /s /target <ip_address> /targettype <type> /user <userid> /passwd <password>`

SUM では、Windows ドメイン（たとえば、`domain1/userid1`）がサポートされています。

CLI モードで SUM と異なるディレクトリにあるベースラインを使用する

- Linux：`./ smartupdate --s --target <ip_address> --targettype <type> --user <userid> --passwd <password> --use_location <directorypath>`
- Windows：`smartupdate /s /target <ip_address> /targettype <type> /user <userid> /passwd <password> /use_location <directorypath>`

SUM では、Windows ドメイン（たとえば、`domain1/userid1`）がサポートされています。

入力ファイルモードで SUM を実行しているディレクトリと同じディレクトリからベースラインを使用する

```
SILENT = YES
ONFAILEDDEPENDENCY = OMITCOMPONENT
TARGETTYPE = <TYPE>
[TARGETS]
HOST = <node_ip_address>
UID = <userid>
PWD = <password>
[END]
```

入力ファイルモードで SUM と異なるディレクトリにあるベースラインを使用する

```
SILENT = YES
SOURCEPATH = <baseline_directory>
ONFAILEDDEPENDENCY = OMITCOMPONENT
TARGETTYPE = <TYPE>
[TARGETS]
HOST = <node_ip_address>
UID = <userid>
```

```
PWD = <password>
[END]
```

1 つのベースラインでの 2 つのノードのアップデート

CLI モードで 2 つのノードにベースライン内のすべてのアップデートを適用する

- Linux : `./ smartupdate --s --target <ip_address_1> --targettype <type_1> --user <userid_1> --passwd <password_1> --target <ip_address_2> --targettype <type_2> --user <userid_2> --passwd <password_2>`
- Windows : `smartupdate /s /target <ip_address_1> /targettype <type_1> /user <userid_1> /passwd <password_1> /target <ip_address_2> /targettype <type_2> /user <userid_2> /passwd <password_2>`

注記

これらの例は、SUM を実行しているディレクトリと同じディレクトリにあるベースラインを使用します。別のベースラインを使用する場合は、そのベースラインのディレクトリパスを指定した次のコマンドを含めます。

`/use_location <baseline_directory_path>`

入力ファイルモードで 2 つのノードにベースライン内のすべてのアップデートを適用する

```
SILENT = YES
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
ONFAILEDDEPENDENCY = OMITCOMPONENT
[TARGETS]
HOST = <ipaddress_1>
TARGETTYPE = <type_1>
UID = <userid_1>
PWD = <password_1>
[END]
[TARGETS]
HOST = <ipaddress_2>
TARGETTYPE = <type_2>
UID = <userid_2>
```

```
PWD = <password_2>
[END]
```

注記

両方のノードで同じユーザー認証情報を使用する場合、それらのノードにはユーザー認証情報を 1 回だけ指定します。2 番目の TARGETS エントリーを作成する代わりに、最初のノードエントリーに 2 番目のノードの IP アドレスを入力します。以下に例を示します。

```
[TARGETS]
HOST = <ipaddress_1>
HOST = <ipaddress_2>
TARGETTYPE = <type_1>
UID = <userid_1>
PWD = <password_1>
[END]
```

ノード上のソフトウェアのみのアップデート

これらの例を使用すると、SUM はリモートノード上のソフトウェアのみをアップデートします。

CLI モードで SUM と同じディレクトリにあるベースラインから 1 つのノードのソフトウェアをアップデートする

- Linux : `./ smartupdate --s --target <ip_address> --user <userid> --passwd <password> --softwareonly`
- Windows : `smartupdate /s /target <ip_address> /user <userid> /passwd <password> /softwareonly`

注記

別のベースラインを使用する場合は、次のコマンドを含めます。

- Linux : `--use_location <baseline_directory>`
- Windows : `/use_location <baseline_directory>`

入力ファイルモードで SUM と同じディレクトリにあるベースラインから 1 つのノードのソフトウェアをアップデートする

```
SILENT = YES
SOFTWAREONLY  = YES
```

```
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
ONFAILEDDEPENDENCY = OMITCOMPONENT
[TARGETS]
HOST = <ip_address>
UID = <userid>
PWD = <password>
[END]
```

この例は、SUM を実行しているディレクトリと同じディレクトリに配置されているベースラインを使用します。

別のベースラインを使用するには、入力ファイルに以下を追加します。

```
SOURCEPATH = <directory_path>
```

ノード上のファームウェアのみのアップデート

ノード上のファームウェアのみをアップデートするには、ソフトウェアのみを展開した例に以下の変更を加えます。

- **CLI モード** - コマンド `softwareonly` を削除し、`romonly` を追加します。
- **入力ファイルモード** - コマンド `SOFTWAREONLY` を削除し、`ROMONLY` を追加します。

ノードへの最新コンポーネントの展開

これらの例を使用すると、SUM は、複数のベースラインを含むディレクトリから最新のアップデートのみを適用します。

CLI モードでの最新コンポーネントの展開

- Linux : `./ smartupdate --s --target <ip_address> --user <userid> --passwd <password>`
- Windows : `smartupdate /s /target <ip_address> /user <userid> /passwd <password>`

入力ファイルモードでの最新コンポーネントの展開

```
SILENT = YES
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
```

```
ONFAILEDDEPENDENCY = OMITCOMPONENT
[TARGETS]
HOST = <ipaddress_1>
TARGETTYPE = <type_1>
UID = <userid_1>
PWD = <password_1>
[END]
```

ノード上での旧バージョンのコンポーネントの強制的な展開

以下の例では、SUM は、旧バージョンのコンポーネントのソフトウェア展開をノードで強制的に実行します。

CLI モードでの強制的な展開

- Linux：`./ smartupdate --f:bundle --softwareonly --use_location <baseline_directory_path> --s`
- Windows：`smartupdate /f:bundle /softwareonly /use_location <baseline_directory_path> /s`

入力ファイルモードでの強制的な展開

```
SILENT = YES
SOFTWAREONLY = YES
FORCEBUNDLE= YES
ONFAILEDDEPENDENCY = OMITCOMPONENT
[TARGETS]
HOST = <ip_address>
UID = <userid>
PWD = <password>
[END]
```

2 つのソフトウェアコンポーネントの展開

以下の例では、ローカルホストに 2 つのコンポーネントを展開します。また、コンポーネントがすでにインストールされており、それがソフトウェアコンポーネントである場合、再書き込みまたはダウングレードを強制的に実行します。

> **注記**
> この例では、指定されたコンポーネントがソフトウェアでない場合、SUM はそのコンポーネントを展開しません。

CLI モードでの 2 つのコンポーネントの展開

以下の例のいずれかを選択します。両方の例でタスクを実行します。

- Linux オプション 1：`./ smartupdate --f:software hponcfg-5.2.0-0.x86_64.rpm hp-smh-templates-10.6.1-1481.4.noarch.rpm --s`
- Linux オプション 2：`./ smartupdate --c hponcfg-5.2.0-0.x86_64.rpm --c hp-smh-templates-10.6.1-1481.4.noarch.rpm --f:software --s`
- Windows オプション 1：`smartupdate /f:software cp008097.exe cp008257.exe /s`
- Windows オプション 2：`smartupdate /c cp008097.exe /c cp008257.exe /f:software /s`

入力ファイルモードでの 2 つのコンポーネントの展開

```
SILENT = YES
FORCESOFTWARE  = YES
COMPONENTSLIST= cp008097.exe, cp008257.exe
ONFAILEDDEPENDENCY = OMITCOMPONENT
[TARGETS]
HOST = localhost
UID = <userid>
PWD = <password>
[END]
```

sudo を使用したリモートの Linux ノードのアップデート

以下の例では、SUM は `userid` を使用してリモートの Linux ノードにログインし、`sudo` 機能を使用してコマンドを実行します。SUM は、SUM を実行しているディレクトリ内のベースラインを使用します。

sudo を使用した CLI モードでのリモート Linux ノードのアップデート

- Linux：`./ smartupdate --target <ip_address> --targettype linux --username <userid> --passwd <password> --use_sudo --use_location <baseline_directory_path> --silent`
- Windows：`smartupdate /target <ip_address> /targettype linux /username <userid> /passwd <password> /use_sudo /use_location <baseline_directory_path> /silent`

sudo を使用した入力ファイルモードでのリモート Linux ノードのアップデート

```
SILENT = YES
SOURCEPATH = <baseline_path>
ONFAILEDDEPENDENCY = OMITCOMPONENT
TARGETTYPE = LINUX
[TARGETS]
HOST = <ip_address>
USESUDO = YES
UID = <userid>
PWD = <password>
[END]
```

sudo を使用した CLI モードでの複数のリモート Linux ノードのアップデート

- Linux：`./ smartupdate --s --f --target 10.0.1.15 --user sudouser1 --passwd password1 --targettype linux --use_sudo --target 10.0.1.17 --username sudouser2 --passwd password2 --targettype linux --use_sudo --use_location <baseline_directory_path>`
- Windows：`smartupdate /s /f /target 10.0.1.16 /user sudouser1 /passwd password1 /targettype linux /use_sudo /target 10.0.1.17 /username sudouser2 /passwd password2 /targettype linux /use_sudo /use_location C:\SPP\Swpackages`

スーパーユーザーの認証情報を使用したノードのアップデート

以下の例では、SUM は、リモートノードにログインし、スーパーユーザーの認証情報を使用してリモートノードをアップデートします。

CLI モードでのスーパーユーザーの認証情報の使用

- Linux : `./ smartupdate --target <ip_address> --targetype linux --username <userid> --passwd <password> --silent --su_username <su_userid> --su_password <su_password>`
- Windows : `smartupdate /target <ip_address> /targetype linux /username <userid> /passwd <password> /silent /su_username <su_userid> /su_password <su_password>`

入力ファイルモードでのスーパーユーザーの認証情報の使用

1 つのノードのアップデート

```
SILENT = YES
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
ONFAILEDDEPENDENCY=Omitcomponent
SUUSERNAME = <su_userid>
SUPASSWORD = <su_password>
[TARGETS]
HOST = <ip_address>
UID = <userid>
PWD = <password>
[END]
```

同じスーパーユーザー認証情報を使用する 2 つのノードのアップデート

```
SILENT = YES
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
ONFAILEDDEPENDENCY=Omitcomponent
SUUSERNAME = <su_userid>
SUPASSWORD = <su_password>
[TARGETS]
HOST = <ip_address_1>
UID = <userid_1>
PWD = <password_1>
[END]
[TARGETS]
HOST = <ip_address_2>
UID = <userid_2>
PWD = <password_2>
[END]
```

異なるスーパーユーザー認証情報を使用する 2 つのノードのアップデート

```
SILENT = YES
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
```

```
ONFAILEDDEPENDENCY=Omitcomponent
[TARGETS]
HOST = <ip_address_1>
UID = <userid_1>
PWD = <password_1>
SUUSERNAME = <su_userid_1>
SUPASSWORD = <su_password_1>
[END]
[TARGETS]
HOST = <ip_address_2>
UID = <userid_2>
PWD = <password_2>
SUUSERNAME = <su_userid_2>
SUPASSWORD = <su_password_2>
[END]
```

SSH キーを使用した Linux ノードのアップデート

次の例では、SUM で SSH キーを使用してログインし、リモート Linux ノードをアップデートすることができます。SUM は、DSA、RSA、ECDSA、および Ed25519 暗号化キー形式をサポートします。秘密キーを生成するには、root 認証情報での SSH キーファイルの使用を参照してください。

CLI モードでの SSH キーの使用：

- Linux：`./ smartupdate --target <ip_address> --targetype linux --username <userid> --use_sshkey privatekeyfile=/privatekeyfile/location passphrase=12345 --silent`
- Windows：`smartupdate /target <ip_address> /targetype linux /username <userid> /use_sshkey privatekeyfile=c:\privatekeyfile\location passphrase=12345 /silent`

入力ファイルモードでの SSH キーの使用：

1 つのノードのアップデート：

```
SILENT = YES
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
ONFAILEDDEPENDENCY=Omitcomponent
USE_SSHKEY = YES
[TARGETS]
HOST = <ip_address>
```

```
UID = <userid>
PRIVATEKEYFILE = c:\<keyfile_directory>
PASSPHRASE = <passphrase_string>
[END]
```

同じキーファイルと SSHKEY を使用する 2 つのノードのアップデート：

```
SILENT = YES
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
ONFAILEDDEPENDENCY=Omitcomponent
USE_SSHKEY=YES
PRIVATEKEYFILE = c:\<keyfile_directory>
PASSPHRASE = <passphrase_string>
[TARGETS]
HOST = <ip_address_1>
UID = <userid_1>
[END]
[TARGETS]
HOST = <ip_address_2>
UID = <userid_2>
[END]
```

異なる SSH キーファイルとパスフレーズを使用する 2 つのノードのアップデート：

```
SILENT = YES
IGNOREERRORS = ServerNotFound,BadPassword, FailedDependencies
SKIPTARGET = NO
ONFAILEDDEPENDENCY=Omitcomponent
USE_SSHKEY=YES
[TARGETS]
HOST = <ip_address_1>
UID = <userid_1>
PRIVATEKEYFILE = c:\<keyfile_directory>
PASSPHRASE = <passphrase_string>
[END]
[TARGETS]
HOST = <ip_address_2>
UID = <userid_2>
PRIVATEKEYFILE = c:\<keyfile_directory>
PASSPHRASE = <passphrase_string>
[END]
```

CLI のパラメーター

サブトピック

SUM CLI パラメーター

SUM は、次のコマンドライン引数を認識します。`--romonly`、`--softwareonly` など、一部の引数は一緒に使用することはできません。

サブトピック

ヘルプ

CLI のヘルプを表示するには、`smartupdate --s --h` と入力します。

構文

`--h[elp]`または`--?` (Linux)

`/h[elp]`または`/?` (Windows)

説明

コマンドラインのヘルプ情報を表示します。

インストールパラメーター

次のセクションでは、アップデートのインストール時に使用できる属性について説明します。

サブトピック

強制

構文

`--f[orce]` (Linux)

`/f[orce]` (Windows)

説明

現在インストールされているソフトウェアを再書き込みまたはダウングレードします。このパラメーターの役割は、`f:software` と同じです。

オプション

`f[orce]:bundle`

選択されているバンドルに含まれ、現在インストールされているコンポーネントを再書き込みまたはダウングレードします。

`f[orce]:rom`

現在インストールされ、適用可能なファームウェアコンポーネントを再書き込みまたはダウングレードします（ファームウェアにのみ適用）。

`f[orce]:software`

現在インストールされ、選択されているソフトウェアコンポーネントを再書き込みまたはダウングレードします。

`f[orce]:all`

現在インストールされ、適用可能なソフトウェアおよびファームウェアコンポーネントを再書き込みまたはダウングレードします。

ダウングレード

構文

`--g or --downgrade` (Linux)

`/g or /downgrade` (Windows)

説明

現在インストールされているバージョンよりも低いバージョンを利用できる、インストール対象のコンポーネントを選択します。このコマンドを発行するとアップグレードも行われます。`rewrite` と組み合わせることができます。

再書き込み

構文

`--e or --rewrite` (Linux)

`/e or /rewrite` (Windows)

説明

インストールされているバージョンよりも低いバージョンを利用できる、インストール対象のコンポーネントを選択します。このコマンドを実行すると、アップグレードも行われます。`downgrade` と組み合わせることができます。

ROM 展開のみ

構文

`--romonly` (Linux)

`/romonly` (Windows)

説明

インストール対象とみなされるコンポーネントをファームウェアコンポーネントのみに制限します。 `softwareonly` と一緒に使用することはできません。`softwareonly` または`romonly` を使用しない場合、すべてのコンポーネントがインストール対象となる可能性があるとみなされます。

ソフトウェア展開のみ

構文

`--softwareonly` (Linux)

`/softwareonly` (Windows)

説明

インストール対象とみなされるコンポーネントをソフトウェアコンポーネントのみに制限します。 `romonly` と一緒に使用することはできません。`softwareonly` または`romonly` を使用しない場合、すべてのコンポーネントがインストール対象であるとみなされます。

使用するネットワークポート

構文

`--port <n>` (Linux)

`/port <n>` (Windows)

説明

SUM が内部 HTTP サーバー用として使用するポートを 63001 から指定のポートに変更します。

SSL ポート

構文

`--ssl_port <n>` (Linux)

`/ssl_port <n>` (Windows)

説明

SUM が内部 HTTPS サーバー用として使用するポートを 63002 から指定のポートに変更します。

ファイアウォールのオープン

構文

`--open_firewall` (Linux)

`/open_firewall` (Windows)

説明

SUM に、外部アクセスに SUM で使用される HTTP および HTTPS ポートを開くことを許可します。リモートノード機能およびリモートブラウザーアクセスのためにポートを開くときにも使用します。

SUM は、ip_tables に DROP コマンドが含まれていても、ノードにアップデートを展開します。`open_firewall` を使用すると、DROP ルールはバイパスされます。

入力ファイルを呼び出すコマンドにこのパラメーターを含めます。SUM 入力ファイルは、ファイアウォールを開くパラメーターをサポートしていません。

サイレント展開

構文

`--s[ilent]` (Linux)

`/s[ilent]` (Windows)

説明

GUI CLI 出力を伴わないサイレントインストールが行われます。データはすべてログファイルに書き込まれます。`express_install` パラメーターは`silent` パラメーターと組み合わせて使用しないでください。

使用方法

SUM では、CLI コマンドにサイレントパラメーターが必要です。

展開するコンポーネントの指定

構文

`--c[omponent]<component_to_install>` または `<component1_to_install> <component2_to_install>` (Linux)

`/c[omponent]<component_to_install>` または `<component1_to_install> <component2_to_install>` (Windows)

説明

インストールするコンポーネントを指定します。

`c[omponent]` パラメーターを使用する場合は、パラメーターごとに 1 つの構成要素を指定します。複数の `c` パラメーターを、同じコマンドラインの個々のコンポーネントに含めることができます。

- `--c[omponent]` 引数を使用しない場合、複数のコンポーネントをスペースで区切り、コマンドライン上の他のすべての引数の後に指定します。
- デフォルトでは、SUM は、コマンドに記載されている順序でコンポーネントを展開します。SUM は、依存関係に基づいて展開の順序を変更します。
- 複数のコンポーネント（`--c[omponent]<component_to_install>` を参照）とバンドルを同一コマンドラインに指定します。1 つのコマンドラインにコンポーネントとバンドルが混在する場合、フィルタースイッチによりインストールされるコンポーネントとバンドルが制御されます。
- SUM は、コンポーネント名のみを使用します。完全なディレクトリパスを入力しても、SUM はパスを無視します。`--use_location` および `c` を使用する場合、SUM は、デフォルトのレポジトリと指定されたディレクトリの両方をチェックします。

5 つより多くの構成要素を指定する場合は、複数のコマンドを発行するか、入力ファイルを使用します。

例

`smartupdate --silent <other_params> --save_install_set --install_set_name Test1 --c[omponent]<component1_to_install>` (Linux)
`/smartupdate /silent <other_params> /save_install_set /install_set_name Test1 /c[omponent]<component1_to_install>` (Windows)

インストールするバンドルの定義

構文

`--b[undle] <bundlename>` (Linux)

`/b[undle] <bundlename>` (Windows)

説明

この引数は、インストールするバンドルを指定します。

複数のコンポーネントとバンドル（`b[undle] <bundlename>`を参照）を同一コマンドラインに指定します。1 つのコマンドラインにコンポーネントとバンドルが混在する場合、フィルタースイッチによりインストールされるコンポーネントとバンドルが制御されます。

ベースラインの場所の定義

構文

`--use_location "[filepath]"` (Linux)

`/use_location "filepath"\"file_share"` (Windows)

説明

SPP およびコンポーネントを含むディレクトリまたはファイル共有が SUM で使用されるように指定します。SUM は、ターゲットの **file_share** がマッピング済みファイル共有または UNC 形式であることを要求します。

この引数を指定しない場合は、SUM は、SUM を含むディレクトリをデフォルトで使用します。

ログインしたアカウントでこの場所にアクセスできることを確認してください。この場所が UNC ファイル共有である場合は、`unc_username` と `unc_password` を使用してファイル共有の認証情報を指定する必要があります。この認証情報を指定しない場合、SUM は現在の認証情報を使用して共有にアクセスします。

SUM では、CLI モードでの http ベースラインがサポートされていません。

コンポーネント構成のインポート

構文

`--import_configuration <directory_path>` (Linux)

`/import_configuration <directory_path>` (Windows)

説明

このパラメーターは、指定したベースライン内のコンポーネントのコンポーネント構成設定を含むディレクトリを示します。ベースライン内のコンポーネント構成は、インポートしたコンポーネント構成によって上書きされます。

UNC ユーザー認証情報

構文

`/unc_username` (Windows)

説明

ファイル共有にアクセスするためのユーザー名認証情報。SUM では、Windows ドメイン（たとえば、`domain1/userid1`）がサポートされています。

UNC パスワード認証情報

構文

`/unc_password` (Windows)

説明

ファイル共有にアクセスするためのパスワード認証情報。

管理エージェントコンポーネントなし

構文

`--no_mgmt` (Linux)

`/no_mgmt` (Windows)

説明

AMS、SNMP、および WBEM Provider の管理コンポーネントが自動的に選択されていないことを示します。サイレントモードでは、SUM は、どの管理コンポーネントもアップデートしません。

SNMP エージェントコンポーネントの使用

構文

`--use_snmp` (Linux)

`/use_snmp` (Windows)

説明

Gen8 および Gen9 サーバー:SUM が自動的に SNMP コンポーネントをインストール対象に選択します。

Gen10 以降のサーバー：SUM は適用外として SNMP コンポーネントを一覧表示します。

制限

Linux および Gen9 のみ：SUM が SNMP RPM および smh_templates RPM を選択します。

WMI エージェントコンポーネントの使用

構文

`/use_wmi`

説明

SUM が自動的に WBEM コンポーネントをインストール対象に選択します。

制限

Windows および Gen9 のみ。

AMS エージェントコンポーネントの使用

構文

`--use_ams` (Linux)

`/use_ams` (Windows)

説明

(Gen8 以降のみ) SUM が自動的に AMS コンポーネントをインストール対象に選択します。

高速インストール

構文

`--express_install` (Linux)

`/express_install` (Windows)

説明

GUI を起動し、Localhost ガイド付きアップデートを自動モードで開始します。このパラメーターは、`silent` パラメーターを使用しているときは無視されます。

エラーを無視するパラメーター

以下の属性は、SUM がエラーを処理する方法を決定します。

SUM では、これらの属性はコマンドに含まれているすべてのノードに適用されます。

サブトピック

TPM のバイパス
警告の無視
インストールの続行とエラーの無視
既存の接続の無効化
失敗した依存状態

TPM のバイパス

構文

`--tpmbypass` または `--ignore_tpm` (Linux)

`/tpmbypass` または `/ignore_tpm` (Windows)

説明

ターゲットの iLO 5 以降、Windows、Linux、または VMware ESXi で TPM を有効にした場合は、警告メッセージを無視し、コンポーネントのインストールを続行します。TPM について詳しく

は、https://www.hpe.com/support/SUMGen11-UG-en に公開されている Smart Update Manager 10.0 ユーザーガイドを参照してください。

> **注記**
> このパラメーターは、CLI コマンドの末尾で指定します。

警告の無視

構文

`--ignore_warnings` (Linux)

`/ignore_warnings` (Windows)

説明

SUM がノードに関する警告を受けた後でも、インストールの続行が許可されます。一部の警告は、次のとおりです。

- Serviceguard クラスターのアクティブメンバー：このオプションを指定せず、ノードが Serviceguard クラスターのアクティブなメンバーである場合、ノードはインストールフェーズまたは展開フェーズに入りません。
- TPM の警告
- 空きディスク領域の減少
- HPE OneView で管理されるシステム
- 保留中 iLO インストールキューの警告
- 例外タスクキューの警告
- iLO および iSUT の警告

インストールの続行とエラーの無視

構文

`--continue_on_error` `<error>` (Linux)

`/continue_on_error` `<error>` (Windows)

説明

インストールが継続され、エラーが無視されます。

オプション

有効な値は、次のとおりです。

`All`

任意の種類のエラーを返すリモートノードをバイパスし、他のノードで展開を続行します。

`ServerNotFound`

`ServerNotFound` オプションを使用すると、ファームウェアまたはソフトウェアを複数のリモートホストに同時に展開する場合に、アクティブではない、または利用できないリモートホストをバイパスすることができます。

`BadPassword`

`BadPassword` オプションを使用すると、提供された認証応報が誤っていると報告されたリモートノードをバイパスし、他のノードの処理を続行することができます。

`FailedDependencies`

`FailedDependencies` オプションを使用すると、失敗した依存関係のあるリモートノードをバイパスし、インストール準備ができている他のノードの処理を続行することができます。このパラメーターは、`on_failed_dependency:Force` または `:OmitComponent` を使用して上書きすることができます。

`iLOCACEnabled`

`iLOCACEnabled` オプションを使用して、iLO で CAC モードが有効になってるノードをバイパスし、他のノードで続行します。iLO 管理者認証情報は必要ありません。

`iLOHighSecurityMode`

`iLOHighSecurityMode` オプションを使用して、iLO で高セキュリティモードが有効になってるノードをバイパスし、他のノードで続行します。iLO 管理者認証情報は必要ありません。

`CHIFSignFailure`

`CHIFSignFailure` オプションを使用して、CHIF ドライバーコンポーネントの署名検証に失敗したときにノードをバイパスし、他のノードで続行します。

`CHIFInstallFailure`

`CHIFInstallFailure` オプションを使用して、CHIF ドライバーのインストールに失敗したときにノードをバイパスし、他のノードで続行します。

既存の接続の無効化

構文

`--override_existing_connection` (Linux)

`/override_existing_connection` (Windows)

説明

SUM に、進行中のセッションを無効化してリモートノードのインストールフレームワークを再初期化するよう指示します。

このパラメーターを指定しないと、SUM は、SUM セッションがリモートノードですでに実行されている場合に、このノードをスキップします。

失敗した依存状態

構文

`--on_failed_dependency` (Linux)

`/on_failed_dependency` (Windows)

説明

このパラメーターは、コンポーネントを省略するか、ホストへのインストールをスキップするかを選択するオプションを提供します。失敗した依存状態が発生した場合、SUM はインストールを試行しません。

オプション

`OmitHost`（デフォルト）

ターゲットが障害状態に設定されるため、インストールは試行されません。

`OmitComponent`

影響を受けるコンポーネントの選択が解除され、失敗した依存状態が発生していないアップデートによってプロセスが進められます。

`Force`

依存状態に障害があっても、すべてのアップデートが試行されます。

サンプル

`--on_failed_dependency:OmitHost` (Linux)

`/on_failed_dependency:OmitHost` (Windows)

`--on_failed_dependency:OmitComponent` (Linux)

`/on_failed_dependency:OmitComponent` (Windows)

`--on_failed_dependency:Force` (Linux)

`/on_failed_dependency:Force` (Windows)

再起動パラメーター

以下の属性は、SUM が再起動を実行する方法を決定します。

サブトピック

再起動

構文

`--r[eboot]` (Linux)

`/r[eboot]` (Windows)

説明

サーバー（リモートインストールではホストサーバー）は、次の条件で再起動します。

- `reboot` オプションを選択するか、そのオプションをコマンドライン引数内で使用する場合。
- インストール用に選択されたすべてのコンポーネントが正常にインストールされた場合。
- インストールされたコンポーネントのうち少なくとも 1 つで、インストールを完了するために再起動が必要な場合。

再起動メッセージ

構文

`--reboot_message "reboot message"` (Linux)

`/reboot_message "reboot message"` (Windows)

説明

再起動するサーバーに接続されているリモートコンソールに、指定した再起動メッセージを表示します。このパラメーターは `reboot` オプションと組み合わせて使用してください。そうでない場合は無視されます。

注記

SLES12 OS イメージには cli コマンドとしてパラメーターを実行できないという制約があるため、 `reboot_message` cli パラメーターは SLES12 OS では機能しません。

再起動遅延

構文

`--reboot_delay timeout_in_secs` (Linux)

`/reboot_delay timeout_in_secs` (Windows)

説明

`timeout_in_seconds` 変数で指定された時間だけ、サーバーの再起動を遅らせます。

デフォルトは 60 秒です。

Linux システムでは、再起動遅延の時間は秒単位から分単位に変換され、1 分未満（59 秒以下）の値は切り上げられます。

オプション

15～3600 までの値を指定できます。

要件

この引数は `reboot` オプションと組み合わせて使用してください。そうでない場合は無視されます。

常に再起動

構文

`--reboot_always` (Linux)

`/reboot_always` (Windows)

説明

SUM は、このコマンドラインオプションが渡されると常にサーバーを再起動します。

ノードパラメーター

以下の属性は、ノードのアップデートオプションを定義します。属性を指定するときに、ターゲットノードを指定してから属性を指定します。ターゲットノードの前に属性を指定しないでください。

以下に例を示します。

有効例：

```
smartupdate /s /target 10.0.1.2 /user root /password root
```

無効例：

```
smartupdate /s /user root /password root /target 10.0.1.2
```

グローバル属性を、属性を指定しないノードに適用できます。属性を適用する 1 つのターゲットノードとともに、コマンドラインの最後にグローバル属性を記述します。以下に例を示します。

```
smartupdate /s /target 10.0.1.3 /target 10.0.1.4 /target 10.0.1.5 /user root /password root
```

以下の属性は、グローバルに使用できます。

`username/user`、`password/passwd`、`use_location`、`current_credential`、`su_username`、`su_password` および `use_sshkey`。

サブトピック

- **ユーザー名**
- **ユーザーパスワード**
- **iLO ユーザー名**
- **iLO パスワード**
- **アプリケーションアカウントに代わる iLO 認証情報の使用 (NO_APP_ACCOUNT)**
- **SSH キーファイルの使用**
- **プライベートキーファイルの場所**
- **SSH パスフレーズ**
- **スーパーユーザー名**
- **スーパーユーザーのパスワード**
- **sudo 認証情報の使用**

ユーザー名

構文

`--user` **<username>**または`--username` **<username>**（Linux）

`/user` **<username>**または`/username` **<username>**（Windows）

説明

ユーザー ID を使用してリモートノードにログインする場合は、この引数を使用します。

要件

ユーザー名が root または管理者グループに属していることを確認してください。

SUM では、Windows ドメイン（たとえば、`domain1\userid1`）がサポートされています。

Windows ドメインの一部である Linux ノードを管理している場合は、次の形式を使用します。

`<domain>\<username>`

ユーザーパスワード

構文

`--password` **<password>**（Linux）

`/password` **<password>**（Windows）

smartupdate は、`passwd` と `pwd` もサポートします（Windows）

説明

`user` パラメーターで指定したユーザー ID のパスワード。システムはリモートノードへのログインにこのパスワードを使用します。

iLO ユーザー名

構文

`--ilo_username` **<username>** (Linux)

`/ilo_username` **<username>** (Windows)

説明

サーバーの iLO がセキュア標準モードまたは高セキュリティモードで構成されている場合、ホスト OS から iLO に接続するには、このパラメーターを使用します。iLO 管理者の認証情報を入力します。

このパラメーターが有効なのは、Windows、Linux、および不明のノードタイプに対してのみです。

要件

iLO 管理者アカウントに **iLO 設定の構成**の許可が付与されていることを確認してください。

iLO パスワード

構文

`--ilo_password` **<password>** (Linux)

`/ilo_password` **<password>** (Windows)

説明

`ilo_username` パラメーターで指定された iLO ユーザー名に関連付けられたパスワード。

アプリケーションアカウントに代わる iLO 認証情報の使用 (NO_APP_ACCOUNT)

構文：

`--no_app_account` (Linux)

`/no_app_account` (Windows)

説明

このパラメーターは、iL07 以降のバージョンでのみサポートされます。

SUM は iLO でアプリケーションアカウントを作成し、それを iLO との通信に使用します。このパラメーターは SUM に対し、iLO でアプリケーションアカウントを作成しないように指示します。このパラメータを使用する場合、SUM が iLO と通信するためには iLO 認証情報が必要です。

入力：
`smartupdate /s /target <ip> /user root /password root /ilo_username <ilo_username> /ilo_password <ilo_password> /no_app_account`。

SSH キーファイルの使用

構文

`--use_sshkey`（Linux）

`/use_sshkey`（Windows）

説明

SSH キーを使用して、Linux ノードに接続します。SUM は、他のタイプのノードについては、このパラメーターを無視します。

このパラメーターを指定する場合、SUM はプライベートキーファイルの場所を必要とします。キーファイル（PEM 形式）は、ユーザーが渡すことができるオプションのパラメーターです。

プライベートキーファイルの場所

構文

`privatekeyfile`

`smartupdate /target 10.0.1.2 /user root /use_sshkey privatekeyfile=c:/<keyfile_directory/ssh.pem> /r /s`

説明

専用の SSH キーファイルが生成されている場合、SUM によって、パスフレーズが必要であることが自動的に検出され、その入力が求められます。

要件

`use_sshkey` パラメーターとともに使用します。

SSH パスフレーズ

構文

`passphrase`

説明

このオプションを使用して、パラメーター`use_sshkey` とともに使用するパスフレーズと、オプションの`privatekeyfile` を指定します。

キーファイルがパスフレーズを要求するときに、パラメーターとして指定されていない場合、SUM はパスフレーズの入力を求めます。

```
smartupdate /target 10.0.1.2 /user root use_sshkey privatekeyfile=c:
/<keyfile_directory/ssh.pem> passphrase=12345 /r /s
```

要件

`use_sshkey` パラメーターとともに使用します。

スーパーユーザー名

構文

`--su_username` (Linux)

`/su_username` (Windows)

説明

`username` および`passwd` で指定された認証情報にコンポーネントをアップデートするための root 権限がない場合、リモートノードで root（スーパーユーザー）ユーザー名を使用してセッションを開始し、コンポーネントのインベントリとアップデートを実行します。

ローカルホストでは、このコマンドを使用しないでください。

スーパーユーザーのパスワード

構文

`--su_password` (Linux)

`/su_password` (Windows)

説明

`username` および `passwd` で指定された認証情報にコンポーネントをアップデートするための root 権限がない場合、リモートノードで root（スーパーユーザー）ユーザー名を使用してセッションを開始し、コンポーネントのインベントリとアップデートを実行します。

ローカルホストでは、このコマンドを使用しないでください。su を使用してローカルホスト上で SUM を起動するには、次の例を使用します。

```
su root
```

```
smartupdate --s --f --use_location /mnt/spp
```

sudo 認証情報の使用

構文

`--use_sudo` (Linux)

`/use_sudo` (Windows)

説明

リモートノードの username と password が sudo ユーザーの認証情報であることを指定します。

要件

`username` および `passwd` と併用すると、両方のパラメーターが sudo ユーザー認証情報であることを示します。

ターゲットアドレスの定義

構文

`--target` **“netAddress”** (Linux)

`/target` **“netAddress”** (Windows)

説明

リモートホスト（リモートサーバー、リモート iLO NIC ポートを使用可能）の IP アドレスまたは DNS 名。

ターゲットタイプの定義

構文

`--targettype` “type” (Linux)

`/targettype` “type” (Windows)

説明

スクリプト展開のインベントリ時間を短縮します。これはオプションのコマンドライン引数です。

設定可能な値

`Windows`

`Linux`

`iLO`

`targettype` は、`target` パラメーターと一緒に使用してください。`targettype` と `target` の順番は入れ替え可能です。ノード名にスペースが含まれる場合は、引用符 `" "` で名前を囲んでください。

ユーザーの現在の認証情報

構文

`/current_credential` (Windows のみ)

説明

ローカルホストの認証情報を使用してノードにアクセスでき、ノードごとに明示的にユーザー名とパスワードを入力する必要がなくなります。処理しているノードで現在の認証情報が有効であることが前提です (Windows ノードにのみ適用)。

iLO 5 以降のパラメーター

インストールセットの保存

構文

`--save_install_set` (Linux)

`/save_install_set` (Windows)

説明

このパラメーターは、iLO 5 以降のノード用の iLO レポジトリにインストールセットを保存します。

インストールセットの名前

構文

`--install_set_name` (Linux)

`/install_set_name` (Windows)

説明

このパラメーターは、iLO 5 以降のノード用の iLO レポジトリに、指定された名前を使用してインストールセットを保存します。

インストールセットの説明

構文

`--install_set_description` (Linux)

`/install_set_description` (Windows)

説明

このパラメーターは、iLO 5 以降のノード用の iLO レポジトリに、指定された説明を使用してインストールセットを保存します。

欠落しているコンポーネント署名のスキップ

構文

`--skip_missing_compsig` (Linux)

`/skip_missing_compsig` (Windows)

説明

ノード全体の展開を停止する代わりに、コンポーネントの署名が欠落しているコンポーネントをスキップします。

iLO レポジトリの手動管理

構文

`--manually_manage_ilo_repository` (Linux)

`/manually_manage_ilo_repository` (Windows)

説明

自動的に管理する代わりに、手動で iLO レポジトリを管理します。

> **注記**
> デフォルトのモードは自動です。SUM でアップロードされた新しいコンポーネントを収容するのに十分な領域が iLO NAND にない場合、SUM は（コンポーネントの合計サイズに基づいて）最小サイズのインストールセットを削除し、新しいコンポーネントおよびインストールセット用の領域を作成します。

保存された iLO の使用

構文

`--use_ilo_saved` (Linux)

`/use_ilo_saved` (Windows)

説明

ベースラインとして iLO レポジトリに保存されたインストールセットを使用します。SUM は、このインストールセットを iLO NAND で開始し、展開時に結果のタスクキューを処理します。

iLO のスキップ

構文

`--skip_ilo` (Linux)

`/skip_ilo` (Windows)

説明

このコマンドは、インストールセットから iLO Smart コンポーネントを削除するよう SUM に指示します （EFM のみ）。

前提条件のスキップ

構文

`--skip_prereqs` (Linux)

`/skip_prereqs` (Windows)

説明

SUM は、セルフインベントリ時のドライバーコンポーネントなどの前提条件のコンポーネント、インストールされているバージョンを最小アクティブバージョンにする必要がある場合のファームウェアの前提条件のコンポーネント、CHIF ドライバーを、リモート Windows ノードにインストールしません。

制限

CHIF ドライバーのインストールは、Windows ノードでのみ有効です。

実行パラメーターのテスト

構文

`--dryrun` (Linux)

`/dryrun` (Windows)

説明

この属性をその他の属性とともに使用し、実際の展開を除いてコマンドのすべてのプロセスを SUM で実行します。このパラメーターを使用して、リモートノードの接続と構成を確認し、SUM が何をアップデートするかを確認します。

要件

この引数は、テスト実行用にインストールをシミュレーションします。SUM は、どのアップデートも展開しません。

ログファイルのパラメーター

以下の属性は、SUM がログファイルを作成する方法を決定します。

サブトピック

ログパラメーター
冗長レベルのログ記録

ログパラメーター

ログディレクトリ

構文

`--logdir` **“path”**（Linux）

`/logdir` **“path”**（Windows）

説明

SUM からの出力をデフォルトディレクトリ以外のディレクトリにリダイレクトします。

Windows システムで実行されている SUM の場合、デフォルト位置は `%SYSTEMDRIVE%\CPQSYSTEM\sum\log\<netAddress>` で、リダイレクト先は `<path>\sum\log\<netAddress>` です。

SUM は、コンポーネントを処理するときに `%SYSTEMDRIVE%\CPQSYSTEM\` ディレクトリを作成します。SUM は、リダイレクトされたディレクトリに `smartupdate_log.txt`、`smartupdate_detail_log.txt`、および `smartupdate_InstallDetails.txt` ファイルのみを書き込みます。SUM は、その他のログをデフォルトディレクトリに書き込みます。

Linux システムで実行されている SUM の場合、デフォルト位置は `/var/log/sum/<netAddress>` で、リダイレクト先は `<path>/log/sum/<netAddress>` です。

デバッグログディレクトリ

構文

`--debuglogdir [directory_path]`（Linux）

`/debuglogdir [directory_path]`（Windows）

説明

このパラメーターは、デバッグログファイルが保存される SUM のディレクトリを割り当てます。

終了時にクリーンアップ

構文

`--cleanup_onexit`（Linux）

`/cleanup_onexit`（Windows）

説明

このパラメーターは、ターゲットおよびホストから以下のものを削除します。

- Linux フォルダー `/usr/lib/i386-linux-gnu/*` または `/usr/lib/x86_64-linux-gnu` にコピーされたファームウェア RPM。
- 抽出されたファームウェア RPM ディレクトリとフォルダーの内容。
- ベースラインインベントリの実行中に抽出されたコンポーネント。

- SUM が読み取り専用の場所から起動される場合、このパラメーターは「localsum」ディレクトリを削除します。

> **注記**
> このパラメーターでは、デバッグログファイルは削除されません。

終了時にすべてのログをクリーンアップ

構文

`--cleanupall_onexit` (Linux)

`/cleanupall_onexit` (Windows)

説明

このパラメーターは、ターゲットおよびホストからすべてのファイル（デバッグファイルを含む）を削除します。このパラメーターでは、ユーザーのログは削除されません。

冗長レベルのログ記録

構文

`--v[erbose]` または `--veryv[erbose]` (Linux)

`/v[erbose]` または `--veryv[erbose]` (Windows)

説明

SUM 実行ログファイル `sum_execution_log_*.log` の冗長レベルを設定します。ログファイルに維持される詳細レベルを上げることができます。デフォルトは通常の冗長レベルです。

レポートパラメーター

> **注記**
> 統合レポートとインストール済みレポートは、展開の完了後にのみ生成できます。

SUM はコマンドライン引数を使用して、指定されたシステムまたはリポジトリ位置に関するレポートを生成します。コマンドラインにその他の場所を指定しない場合は、SUM でローカルホストとデフォルトのレポジトリ位置（SUM が起動されたディレクトリ）が使用されます。適切な認証情報を提供する場合は、ノードを指定します。他のコマンドラインパラメーターを使用

してレポジトリを指定し、レポートを生成します。特定のコマンドについては、SUM CLI パラメーターを参照してください。

SUM は、JavaScript 対応 Web ブラウザーで表示できる XML または HTML ファイルとしてレポートを生成するか、CSV 形式をサポートする任意のアプリケーションで開くことができる CSV 形式のレポートを生成します。

HTML 形式のレポートは、`SUM_<タイプ>_Report_<日付>_<時刻>` という名前のディレクトリに生成されます。

デフォルトの場所は、SUM が起動された現在のワーキングディレクトリです。この場所が書き込み禁止になっている場合、SUM はレポートを SUM ログファイルと同じディレクトリに保存します。SUM がレポートを保存できる別のディレクトリを指定するには、`reportdir` パラメーターを使用します。

SUM レポートファイルは次のデフォルトのディレクトリにあります。

Windows：`C:\cpqsystem\sum\log`

Linux：`/var/log/sum`

SUM がレポートを生成するときに、SUM GUI は表示されません。SUM に、生成されたレポートのファイルの場所が表示されます。

サブトピック

ノード概要レポートの生成

構文

`--report` (Linux)

`/report` (Windows)

説明

ノードの概要、およびレポジトリ内のコンポーネントがノードに与える影響について説明した展開プレビューレポートを生成します。たとえば、各コンポーネントがノードに適用されるかどうかなどです。レポートは HTML、XML、および CSV 形式で生成され、`SUM_Deploy_preview_Report_<日付>_<時刻>.html`、`SUM_Deploy_preview_Report_<日付>_<時刻>.xml`、および `SUM_Deploy_preview_Report_<日付>_<時刻>.csv` というファイル名が付けられます。

SUM は、インベントリを実行し、レポートを生成してから終了します。どのアップデートも展開しません。

インベントリレポジトリレポートの生成

構文

`--inventory_report` (Linux)

`/inventory_report` (Windows)

説明

指定されたレポジトリ内のコンポーネントをリスト表示するレポートを生成します。レポートは HTML、XML、および CSV 形式で生成され、`SUM_Inventory_Report_<日付>_<時刻>.html`、`SUM_Inventory_Report_<日付>_<時刻>.xml`、および `SUM_Inventory_Report_<日付>_<時刻>.csv` というファイル名が付けられます。

SUM は、インベントリを実行し、レポートを生成してから終了します。どのアップデートも展開しません。

インストール済みファームウェアレポートの生成

構文

`--firmware_report` (Linux)

`/firmware_report` (Windows)

説明

インストール済みファームウェアとソフトウェアのリスト、およびノードの詳細を示すレポートを生成します。レポートは HTML、XML、および CSV 形式で生成され、`SUM_Firmware_Report_<日付>_<時刻>.html`、`SUM_Firmware_Report_<日付>_<時刻>.xml`、および `SUM_Firmware_Report_<日付>_<時刻>.csv` というファイル名が付けられます。

SUM は、インベントリを実行し、レポートを生成してから終了します。どのアップデートも展開しません。

失敗した依存関係のレポートの生成

構文

`--dependency_report` (Linux)

`/dependency_report` (Windows)

説明

すべてのノードの失敗した依存関係をリスト表示するレポートを生成します。レポートは HTML、XML、および CSV 形式で生成され、`SUM_FailedDependency_Report_<日付>_<時刻>.html`、`SUM_FailedDependency_Report_<日付>_<時刻>.xml`、および `SUM_FailedDependency_Report_<日付>_<時刻>.csv` というファイル名が付けられます。

SUM は、インベントリを実行し、レポートを生成してから終了します。どのアップデートも展開しません。

インストール済みアップデートレポートの生成

構文

`--installed_report` (Linux)

`/installed_report` (Windows)

説明

SUM のこのセッション中に、すべてのノードにインストールされているすべてのファームウェア、ソフトウェア、およびドライバーバージョンをリスト表示するレポートが生成されます。レポートは HTML、XML、および CSV 形式で生成され、`SUM_Installed_Report_<日付>_<時刻>.html`、`SUM_Installed_Report_<日付>_<時刻>.xml`、および `SUM_Installed_Report_<日付>_<時刻>.csv` というファイル名が付けられます。

SUM は、ノードのインベントリを作成し、アップデートを展開してから、レポートを生成します。

結合レポートの生成

構文

`--combined_report` (Linux)

`/combined_report` (Windows)

説明

このレポートは、1 つのファイルにすべてのレポートタイプを生成します。レポートは HTML、XML、および CSV 形式で生成され、`SUM_Combined_Report_<日付>_<時刻>.html`、`SUM_Combined_Report_<日付>_<時刻>.xml`、および `SUM_Combined_Report_<日付>_<時刻>.csv` というファイル名が付けられます。

SUM は、ノードのインベントリを作成し、アップデートを展開してから、レポートを生成します。

レポートディレクトリ

構文

`--reportdir` (Linux)

`/reportdir` (Windows)

説明

SUM のレポートの保存先ディレクトリを指定するには、すべてのレポートコマンドでこのパラメーターを使用します。例：`smartupdate --report --reportdir /etc/user/reports`

入力ファイルのパラメーター

複数のノードか個々のノード、またはノードグループのアップデートのスクリプトを作成するには、次のパラメーターを使用します。

サブトピック

入力ファイルの使用
入力ファイルの削除

入力ファイルの使用

構文

`--inputfile` **“filename”** (Linux)

`/inputfile` **“filename”** (Windows)

説明

このパラメーターがファイル名で使用されている場合、SUM は展開を実行するために、ファイルの内容を使用してターゲットおよびベースラインのリストを生成します。

詳しくは

入力ファイルのセクション

入力ファイルの削除

構文

`--deleteinputfile` (Linux)

`/deleteinputfile` (Windows)

説明

SUM に、入力ファイルを読み取り後に削除するように指示します。

リターンコード

SUM は Linux と Windows smart コンポーネントのリターンコードを、拡張されたリターンコードマッピングに統合しました。これらのリターンコードは、コンポーネントインストールのステータスを決定します。また、スクリプトのリターンコードを使用して、スクリプトの実行を制御したり、必要な分岐を決定したりできます。

Linux では、負のリターンコードは、256 から値を引くことにより、正のリターンコードに再計算されます。

`SUCCESS_NO_REBOOT`

Linux：0

Windows：0

テキスト：The installation was successful.（インストールに成功しました。）

`SUCCESS_REBOOT`

Linux：1

Windows：1

テキスト：The installation was successful, but a reboot is required.（インストールに成功しましたが、再起動が必要です。）

`SUCCESS_NOT_REQUIRED`

Linux：3

Windows：3

テキスト：The component was current or not required.（このコンポーネントは最新であるか、または不要です。）

FAILURE_GENERAL

Linux：255

Windows：-1

テキスト：A general failure occurred.（一般障害が発生しました。） For details, see the error log.（詳しくは、エラーログを参照してください。）

FAILURE_BAD_PARM

Linux：254

Windows：-2

テキスト：A bad input parameter was encountered.（不正な入力パラメーターが検出されました。）

FAILURE_COMPONENT_FAILED

Linux：253

Windows：-3

テキスト：The installation of the component failed or was blocked by a failed dependency.（コンポーネントのインストールが失敗したか、失敗した依存関係によってブロックされました。）

FAILURE_COMMAND_FAILED

Linux：252

Windows：-4

テキスト：The CLI command execution failed.（CLI コマンドの実行に失敗しました。）

Windows Smart コンポーネントのリターンコード

0

Smart コンポーネントはインストールされませんでした。詳しくは、ログファイルを参照してください。

1

Smart コンポーネントのインストールに成功しました。

2

Smart コンポーネントのインストールは成功しましたが、システムを再起動する必要があります。

3

必要なハードウェアが存在しないか、ソフトウェアが最新のものか、利用可能でないか、インストールするものがないためインストールは試行されませんでした。

Linux ソフトウェア RPM のリターンコード

0

Linux RPM のインストールに成功しました。

1

Linux RPM のインストールに失敗しました。

Linux Smart コンポーネントのリターンコードとファームウェア RPM のリターンコード

0

Smart コンポーネントのインストールに成功しました。

1

Smart コンポーネントのインストールは成功しましたが、システムを再起動する必要があります。

2

必要なハードウェアが存在しないか、ソフトウェアが最新のものか、インストールするものがないためインストールは試みられませんでした。

3

必要なハードウェアが存在しないか、ソフトウェアが最新のものか、利用可能でないか、インストールするものがないためインストールは試行されませんでした。

4

リモートノードにコンポーネントをインストールする場合に、このリターンコードは、ノードが見つからないことを示します。

5

インストールが開始する前に、ユーザーによってインストールが取り消されました。

6

依存関係が満たされていないか、またはインストールツールに問題があるため、インストーラーを実行できません。

7

インストールツールに問題はありませんが、実際のインストール操作が失敗しました。

VMware ESXi Smart コンポーネントのリターンコード

0

提供できるコンポーネントのインストールは成功しました。リブートは不要です。

1

提供できるコンポーネントのインストールは成功しました。インストールしたコンポーネントを有効にするために、再起動が必要です。

2

インストール予定のバージョンと既存のバージョンが一致するため、インストールは試みられませんでした。

3

次のいずれかの理由により、インストールは試みられませんでした。

- インストール予定のバージョンが、インストールされているバージョンより古い
- サポートされているハードウェアが存在しないか、有効になっていないか、またはインストールを試みることができない状態にある
- Smart コンポーネントが環境をサポートしない

4

リモートノードにコンポーネントをインストールする場合に、このリターンコードは、ノードが見つからないことを示します。

5

インストールが開始する前に、ユーザーによってインストールが取り消されました。

6

依存関係が満たされていないか、またはインストールツールに問題があるため、インストーラーを実行できません。

7

インストールツールに問題はありませんが、実際のインストール操作が失敗しました。

入力ファイルのパラメーターを使用した CLI

サブトピック

入力ファイルにおけるコンポーネント固有の構成

入力ファイルのコンポーネントの短い名前を使用して、コンポーネント構成を定義します。コンポーネントのファイル名は、リリースで変更される可能性があります。ファイル名を表示するには、SPP のリリースノートを参照してください。

hpsmh-windows-x64

HP System Management Homepage for Windows x64

hpinsightmgmtwbemprovider-windows-x64

HP Insight Management WBEM Provider for Windows Server x64 Edition

hpinsightmgmtagent-windows-x64

HP Insight Management Agents for Windows Server x64 Editions

hpmouse-linux

HP iLO High-Performance Mouse for Linux

hpsmh-linux-x64

HP System Management Homepage for Linux (AMD64/EM64T)

hpsnmpagent-rhel6-x64

HP SNMP Agents for Red Hat Enterprise Linux 6 (AMD64/EM64T)

hpsnmpagent-sles11-x64

HP SNMP Agents for SUSE Linux Enterprise Server 11 (AMD64/EM64T)

hpsnmpagent-rhel5-x64

HP SNMP Agents for Red Hat Enterprise Linux 5 (AMD64/EM64T)

hpsnmpagent-rhel7-x64

HP SNMP Agents for Red Hat Enterprise Linux 7 (AMD64/EM64T)

hpsnmpagent-sles12-x64

HP SNMP Agents for SUSE Linux Enterprise Server 12 (AMD64/EM64T)

hpqlogic-smartsan-windows-x64

HPE QLogic SmartSAN イネーブルメントキット for Windows 64 ビットオペレーティングシステム

hpemulex-smartsan-windows-x64

HPE Emulex SmartSAN イネーブルメントキット for Windows 64 ビットオペレーティングシステム

hpemulex-smartsan-linux

Emulex SmartSAN Enablement Kit for Linux

hpqlogic-smartsan-linux

QLogic SmartSAN Enablement Kit for Linux

サブトピック

入力ファイルでのコンポーネントの構成

入力ファイルでのコンポーネントの構成

このタスクについて

```
[COMPONENT_CONFIG=<component short name>]
[<PARAM_NAME1>]
Value1-line1
Value1-line2
[<PARAM_NAME1>]
Value1-line1
Value1-line2
[END_COMPONENT_CONFIG]
```

入力ファイルのエンコードについて

入力ファイルに UTF-8 形式を使用します。UTF-8 では、ダブルバイト文字を使用できます。

エラーのレポート

入力ファイルでエラーが発生すると、SUM は、-2 という値（不良パラメーター）を返して終了します。エラーの発生場所や性質は、ログファイル `smartupdate_execution_log_<date>_<time>.raw` で調べます。

SDR からの SUM およびコンポーネントのダウンロード

Linux システムを使用している場合は、SUM を RPM として SDR からダウンロードできます。一般的な YUM コマンドを使用して、SUM およびコンポーネントを SDR からダウンロードし、インストールすることができます。RPM として SUM をダウンロードし、インストールする方法について詳しくは、次の URL にある Linux best practices: Using Service Pack for ProLiant (SPP) and Software Delivery Repository (SDR)を参照してください。

https://www.hpe.com/info/spp/documentation

SDR の使用については、次を参照してください。

https://www.hpe.com/servers/sdr

入力ファイルのパラメーター

入力ファイルを使用する場合は、ターゲットノードおよび使用するベースラインを含める必要があります。ベースラインが含まれているディレクトリで SUM を実行する場合、SUM はデフォルトでそのディレクトリにあるベースラインを使用します。

以下の属性はセッションに関連しており、1 回のみ使用できます。

セッションの属性

`BUNDLESLIST`

設定可能な値：

`<bundlefilename>`

バンドルファイル名

指定されたベースラインの場所に複数のバンドルファイルがある場合に、アップデート対象として考慮されるバンドル xml ファイルのリスト。

CLEANUPALLONEXIT

設定可能な値：

`YES`、デフォルト = `NO`

このパラメーターは、すべてのファイル（デバッグファイルを含む）を削除します。このパラメーターでは、ユーザーのログは削除されません。

CLEANUPONEXIT

設定可能な値：

`YES`、デフォルト = `NO`

このパラメーターは、以下のものを削除します。

- Linux フォルダー `/usr/lib/i386-linux-gnu/*` または `/usr/lib/x86_64-linux-gnu` にコピーされたファームウェア RPM。
- 抽出されたファームウェア RPM ディレクトリとフォルダーの内容。
- ベースラインインベントリの実行中に抽出されたコンポーネント。
- SUM が読み取り専用の場所から起動される場合、このパラメーターは `localsum` ディレクトリを削除します。

> **注記**
> このパラメーターでは、デバッグログファイルは削除されません。

COMBINED_REPORT

設定可能な値：

`YES`、`NO`

このレポートは、1 つのファイルにすべてのレポートタイプを生成します。

SUM は、インベントリを実行し、アップデートを展開し、レポートを生成してから終了します。

SUM がレポートを保存する場所については、レポートパラメーターを参照してください。

COMPONENTSLIST

設定可能な値：

ファイル拡張子（`.exe`、`.rpm`、または `.scexe`）の付いたコンポーネント名

アップデートするコンポーネントのリストを制限します。

DEBUGLOGDIR

設定可能な値：

`<directorypath>`

デバッグファイルを保存する場所を SUM に指示します。

DELETEINPUTFILE

設定可能な値：

`YES`、デフォルト = `NO`

SUM に、入力ファイルを読み取り後に削除するように指示します。

DEPENDENCY_REPORT

設定可能な値：

`YES`、`NO`

すべてのノードの失敗した依存関係をリスト表示するレポートを生成します。

SUM は、インベントリを実行し、レポートを生成してから終了します。アップデートを展開しません。

SUM がレポートを保存する場所については、レポートパラメーターを参照してください。

DOWNGRADE

設定可能な値：

`YES`、`NO`

現在インストールされているバージョンよりも低いバージョンを利用できる、インストール対象のコンポーネントを選択します。これは、アップグレードとは別の機能です。`rewrite` と組み合わせることができます。

DRYRUN

設定可能な値：

`YES`、`NO`

テスト実行用にインストールをシミュレーションします。何もインストールされません。

FIRMWARE_REPORT

設定可能な値：

`YES`、`NO`

インストール済みファームウェアとインストール済みソフトウェアのリスト、およびノードの詳細を示すレポートを生成します。

SUM は、インベントリを実行し、レポートを生成してから終了します。アップデートを展開しません。

SUM がレポートを保存する場所については、レポートパラメーターを参照してください。

FORCEALL

設定可能な値：

`YES`、`NO`

現在インストールされ、選択されているソフトウェアコンポーネント、ファームウェアコンポーネント、およびバンドルを再書き込みまたはダウングレードします。

FORCEBUNDLE

設定可能な値：

`YES`、デフォルト = `NO`

選択されているバンドルに含まれ、現在インストールされているコンポーネントを再書き込みまたはダウングレードします。

FORCEROM

設定可能な値：

`YES`、`NO`

現在インストールされ、選択されているファームウェアコンポーネントを再書き込みまたはダウングレードします。

FORCESOFTWARE

設定可能な値：

`YES`、`NO`

現在インストールされ、選択されているソフトウェアコンポーネントを再書き込みまたはダウングレードします。

HOST

設定可能な値：

`IP アドレス`、`DNS 名`

リモートサーバー、リモート iLO NIC ポートの IP アドレスまたは DNS 名。

IGNOREERRORS

このパラメーターにより、インストールが継続され、エラーが無視されます。

設定可能な値：

`All` - エラーが発生したリモートノードを無視し、他のノードへの展開を続行します。

`ServerNotFound` - ファームウェアまたはソフトウェアを複数のリモートホストに同時に展開する場合に、アクティブではない、または利用できないリモートホストをバイパスします。

`BadPassword` - 提供された認証応報が誤っていると報告されたリモートノードをバイパスし、他のノードの処理を続行します。

`FailedDependencies` - 失敗した依存状態のあるすべてのノードを無視し、準備ができているノードによってプロセスが進められます。

`ILOCACENABLED` - iLO で CAC モードが有効になってるノードをバイパスし､他のノードで続行します。iLO 管理者認証情報は必要ありません。

`iLOHighSecurityMode` - iLO で高セキュリティモードが有効になってるノードをバイパスし、他のノードで続行します。iLO 管理者認証情報は必要ありません。

`CHIFSignFailure` - CHIF ドライバーコンポーネントの署名検証に失敗したときにノードをバイパスし、他のノードで続行します。

`CHIFInstallFailure` - CHIF ドライバーのインストールに失敗したときにノードをバイパスし、他のノードで続行します。

`IGNORETPM`

設定可能な値：

`YES`、`NO`

TPM を有効にした場合は、警告メッセージを無視し、コンポーネントのインストールを続行します。TPM について詳しくは、https://www.hpe.com/support/SUMGen11-UG-en に公開されている Smart Update Manager 10.0 ユーザーガイドを参照してください。

`IGNOREWARNINGS`

設定可能な値：

`YES`、`NO`

SUM がノードに関する警告を受けた後でも、インストールの続行が許可されます。一部の警告は、次のとおりです。

- Serviceguard クラスターのアクティブメンバー
- TPM の警告
- 保留中 iLO インストールキューの警告
- iLO タスクキューの例外
- iLO および iSUT の警告

`IMPORT_CONFIGURATION=<path>`

このパラメーターは、指定したベースライン内のコンポーネントのコンポーネント構成設定を含むディレクトリを示します。ベースライン内のコンポーネント構成は、インポートしたコンポーネント構成によって上書きされます。

INSTALLED_REPORT

設定可能な値：

`YES`、`NO`

すべてのノードにインストールされているすべてのファームウェア、ソフトウェア、およびドライバーバージョンをリスト表示するレポートが生成されます。

SUM は、インベントリを実行し、アップデートを展開し、レポートを生成してから終了します。

SUM がレポートを保存する場所については、レポートパラメーターを参照してください。

INVENTORY_REPORT

設定可能な値：

`YES`、`NO`

指定されたレポジトリ内のコンポーネントをリスト表示するレポートを生成します。

SUM は、インベントリを実行し、レポートを生成してから終了します。アップデートを展開しません。

SUM がレポートを保存する場所については、レポートパラメーターを参照してください。

LOGFILENAME = "path"

設定可能な値：

`ログファイル名`

SUM からの出力をデフォルトディレクトリ以外のディレクトリにリダイレクトします。

Windows コンポーネントの場合、デフォルト位置は`%SYSTEMDRIVE%\CPQSYSTEM\sum\log\<netAddress>`で、リダイレクト先は`<path>\sum\log\<netAddress>`です。

SUM は、コンポーネントを処理するときにディレクトリ`%SYSTEMDRIVE%\CPQSYSTEM\`を作成します。SUM は、リダイレクトされたディレクトリにファイル`smartupdate_log.txt`、`smartupdate_detail_log.txt`、および`smartupdate_InstallDetails.txt`のみを書き込みます。SUM は、その他のログをすべてデフォルトディレクトリに書き込みます。

Linux コンポーネントの場合は、デフォルト位置は`/var/log/sum/<netAddress>`で、リダイレクト先は`<path>/log/sum/<netAddress>`です。

NOMGMT

設定可能な値：

`YES`

SNMP および WBEM Provider を使用するコンポーネントを、コンポーネントの選択画面でオプションのアップデートとして指定します。

サイレントモードでは、SUM は、AMS、SNMP、または WBEM Provider をアップデートしません。

ONFAILEDDEPENDENCY

失敗した依存状態がコンポーネントに発生した場合のプロセスの進め方を SUM に指示します。

設定可能な値：

`OmitHost`（デフォルト）- ホストは障害状態に設定され、SUM はインストールを試行しません。

`OmitComponent`：影響を受けるコンポーネントの選択が解除され、失敗した依存状態が発生していないアップデートによってプロセスが進められます。

`Force`：失敗した依存状態を含むアップデートも含めすべてのアップデートのインストールが試みられます。

OPTIONS

設定可能な値：

1 つ以上の CLI スイッチ

入力ファイル内にある、構成設定を上書きする SUM CLI オプションを指定します。カンマを使用してパラメーターを区切ります。

LDU で以前にサポートされた `LSPOPTIONS` パラメーターに代わるパラメーターです。

REBOOTALLOWED

設定可能な値：

`YES`、`NO`

必要に応じて再起動できます。

REBOOTALWAYS

設定可能な値：

`YES`、`NO`

常にノードを再起動します。

REBOOTDELAY

設定可能な値：

秒数

再起動する前の待機時間です。

REBOOTMESSAGE

設定可能な値：

任意の文字列（256 文字以内）

再起動の前に表示するメッセージを作成します。

REPORT

設定可能な値：

`YES`、`NO`

ノードの概要のレポートリストとレポジトリ内のコンポーネントがノードに与える影響（たとえば、各コンポーネントがノードに適用されるかどうか）に関する説明を生成します。

SUM は、インベントリを実行し、レポートを生成してから終了します。アップデートを展開しません。

SUM がレポートを保存する場所については、レポートパラメーターを参照してください。

REPORTDIR

SUM のレポートの保存先ディレクトリを指定するには、すべてのレポートコマンドでこのパラメーターを使用します。

REWRITE

設定可能な値：

`YES`、デフォルト = `NO`

インストールされているバージョンと同じバージョンを利用できる、インストール対象のコンポーネントを選択します。これは、アップグレードとは別の機能です。`downgrade` と組み合わせることができます。

ROMONLY

設定可能な値：

`YES`、`NO`

インストール対象とみなされるコンポーネントをファームウェアコンポーネントのみに制限します。`softwareonly` と一緒に使用することはできません。`softwareonly` および `romonly` を使用しない場合、すべてのコンポーネントがインストール対象となる可能性があるとみなされます。

SCHEDULEDEPLOYFROM

設定可能な値：

以下に例を示します：`SCHEDULEDEPLOYFROM = MMDDYYYY HH:MM`

現在の展開にスケジュール開始時刻を追加します。

SCHEDULEDEPLOYTO

設定可能な値：

以下に例を示します：`SCHEDULEDEPLOYTO = MMDDYYYY HH:MM`

現在の展開にスケジュール終了時刻を追加します。

SILENT

設定可能な値：

`YES`

GUI 出力を伴わないサイレントインストールが行われます。データはすべてログファイルに書き込まれます。生成されるプロンプトではすべてデフォルトオプションが使用され、ユーザーの入力なしでインストールが続行されます。

> **注記**
> `NO` の値を入力すると、入力ファイルが正しく機能しません。SUM は GUI モードを起動しようとします。

SKIPTARGET

設定可能な値：

`YES`（デフォルト）、`NO`

デフォルト設定では、リモートノードで実行中のリモートセッションを SUM が認識した場合、リモートノードをスキップします。

リモートノードで既存の SUM セッションが進行しているときの動作を定義します。

既存の SUM セッションがある場合にホストをスキップするには、このパラメーターを使用します。`NO` は、進行中のセッションを無効化し、リモートホストのインストールフレームワークを再初期化します。

SOFTWAREONLY

設定可能な値：

`YES`、`NO`

インストール対象とみなされるコンポーネントをソフトウェアコンポーネントのみに制限します。`romonly` と一緒に使用することはできません。`softwareonly` または `romonly` を使用しない場合、すべてのコンポーネントがインストール対象となる可能性があるとみなされます。

TPMBYPASS

TPM 警告をバイパスするには、このパラメーターを使用します。

UPDATEEXISTINGRECOVERYSET

設定可能な値：

`YES`、`NO`

iLO リポジトリ内の既存のリカバリセットを、iLO、BIOS、CPLD、IE、および ME（インテルプロセッサーの場合、IE と ME）の新しいファームウェアバージョンを使用して、現在の展開からアップデートします。Gen10 以降のサーバーのみ。

USE_SSHKEY

設定可能な値：

`YES`、`NO`

SSH PEM 形式のキーファイルを Linux ノードに使用できます。ノード属性としてではなく、セッション属性として使用する必要があります。

USEAMS

設定可能な値：

`YES`

AMS コンポーネントをインストールするように SUM を指定します。

このオプションは、Gen8 以降のサーバーに適用されます。ProLiant G7 以前のサーバーにこのパラメーターを設定すると、このパラメーターは SUM で無視されます。

デフォルトでは、これらのコンポーネントは Gen8 以降のサーバーのみにインストールされます。このパラメーターは、Integrity サーバーには適用されません。

USECURRENTCREDENTIAL

設定可能な値：

`YES`、`NO`

ローカルホストの認証情報を使用してノードにアクセスでき、ノードごとに明示的にユーザー名とパスワードを入力する必要がなくなります。

ただし、アクセス対象のノードで現在の認証情報が有効であることが前提です（Windows にのみ適用）。

USESNMP

設定可能な値：

`YES`

SNMP コンポーネントをインストールするように SUM を指定します。

これらのコンポーネントはデフォルトではオプションで、このパラメーターを使用しない場合はインストールされません。このパラメーターは、Integrity サーバーには適用されません。

Gen9 のみ：SUM は、SNMP エージェントがインストールされるときに SMH テンプレートをインストールします。

注記

Gen10 以降のサーバーでは、SNMP エージェントがサポートされていません。

USEWMI

設定可能な値：

`YES`（Windows のみ）

WBEM コンポーネントをインストールするように SUM を指定します。

これらのコンポーネントはデフォルトではオプションで、このパラメーターを使用しない場合はインストールされません。

> **注記**
> Gen10 以降のサーバーでは、WBEM エージェントがサポートされていません。

VERBOSE

設定可能な値：

`YES`、`NO`

このパラメーターは、展開時に SUM が提供する詳細情報の量を決定します。

以下の属性はノードおよびベースラインに関連します。これらの属性をノードまたはベースラインごとに使用します。

ILO_PASSWORD

設定可能な値：

`<ilo password>`

`ILO_USERNAME` パラメーターに関連付けられている iLO 管理者のパスワード。

ILO_USERNAME

設定可能な値：

`<ilo username>`

iLO 管理者の認証情報を入力します。サーバーの iLO がセキュア標準モードまたは高セキュリティモードで構成されている場合、ホスト OS から iLO に接続するには、このパラメーターを使用します。

INSTALLSETDESCRIPTION

設定可能な値：

`<string>`

インストールセットの説明を設定します。

Gen10 以降のサーバーのみ。

INSTALLSETNAME

設定可能な値：

`<string>`

iLO に保存する必要があるインストールセットの名前を設定します。

Gen10 以降のサーバーのみ。

`MANUALLYMANAGEILOREPOSITORY`

設定可能な値：

`YES`、`NO`

`YES` に設定すると、SUM は新しいインストールセット用の領域を作成するために、iLO レポジトリ上のインストールセットを自動的に削除できません。

Gen10 以降のサーバーのみ。

デフォルトは自動です。つまり、SUM でアップロードされた新しいコンポーネントを収容する領域が iLO NAND にない場合、SUM は（コンポーネントの合計サイズに基づいて）最小サイズのインストールセットを削除し、新しいコンポーネントおよびインストールセット用の領域を作成します。

`PASSPHRASE`

設定可能な値：

`<passphrase=12345>`

SSHkey ファイルのパスフレーズ。`USE_SSHKEY` および `PRIVATEKEYFILE` パラメーターとともに使用します。

`PRIVATEKEYFILE`

設定可能な値：

`<c:/keyfile_directory/ssh.pem>`

プライベートキーファイルへのフルパスを指定します。

`USE_SSHKEY` パラメーターとともに使用します。

`PWD`

設定可能な値：

`<password>`

`UID` に指定されたユーザー ID のパスワードを使用します。

ノードにログインするためのパスワードを指定します。

`SAVEINSTALLSET`

設定可能な値：`YES`、`NO`

iLO レポジトリにインストールセットを保存します。

Gen10 以降のサーバーのみ。

SKIP_PREREQS

設定可能な値：

`YES`、`NO`（Windows および Linux）

`YES` と入力すると、前提条件のセルフインベントリコンポーネント（CHIF ドライバーなど）がインストールされません。

SKIPMISSINGCOMPSIG

設定可能な値：

`YES`、`NO`

`YES` は、コンポーネントの署名ファイルが欠落しているコンポーネントをスキップするように SUM に指示します。

Gen10 以降のサーバーのみ。

SOURCEPATH

設定可能な値：

`Directory path`、`UNC location`

単一のローカルのベースラインパスまたは UNC ファイル共有を指定します。この操作により、ローカルまたはデフォルトベースラインの代わりに、指定されたパスからインベントリが作成されます。

入力ファイルモードを使用した SUM CLI では、http ベースラインがサポートされていません。

STAGEONLY

設定可能な値：

`YES`、デフォルト = `NO`

インストールセットを作成して、適用可能なすべてのコンポーネントを iLO レポジトリにアップロードします。展開を実行するために作成されたインストールセットは開始されません。作成されたインストールセットを使用して、後でシステムをアップデートできます。

SUPASSWORD

設定可能な値：

`<superuserpassword>`

スーパーユーザーのパスワードを指定するには、この引数を使用します。

Linux ノードにログインしたら、通常の認証情報を使用して、これらの認証情報を使用するセッションを昇格できます。アップデートを実行するには、SUM に管理者レベルのアクセス権が必要です。

SUUSERNAME

設定可能な値：

`<superusername>`

スーパーユーザーのユーザー名を指定するには、この引数を使用します。

> **注記**
>
> `<superuserpassword>` および `<superusername>` を使用して、スーパーユーザーのユーザー名とパスワードを指定します。Linux ノードにログインしたら、通常の認証情報を使用して、これらの認証情報を使用するセッションを昇格できます。アップデートを実行するには、SUM に管理者レベルのアクセス権が必要です。

`TARGETTYPE`

ノードのタイプ（`--targettype Linux` (Linux)、`/targettype` (Windows)）を指定します。これにより、インベントリプロセスを短縮できます。

有効なノードのタイプは次のとおりです。

`Windows`

`Linux`

`iLO`

コマンドでグループを指定すると、SUM は、そのグループ内のすべてのノードが同じノードタイプであるとみなします。

`UID`

設定可能な値：

`<username>`

ノードにログインするためのユーザー ID を指定します。

SUM では、Windows ドメイン（たとえば、`domain1/userid1`）がサポートされています。

`UNC_PASSWORD`

設定可能な値：

`<password>`

UNC の場所にアクセスするためのパスワードを指定します。

`UNC_USERNAME`

設定可能な値：

`<username>`

`SOURCEPATH` で指定した UNC の場所のユーザー名認証情報を指定します。

`USEILOSAVED`

設定可能な値：

`<string>`

iLO レポジトリに保存した名前が指定されたインストールセットがノードのベースラインとして使用されます。

Gen10 以降のサーバーのみ。

`USESUDO`

設定可能な値：

`YES`、`NO`

`sudo` コマンドを使用できます。

リモートノードのユーザー名とパスワードが sudo ユーザーの認証情報であることを指定します。入力ファイルでユーザー名およびパスワードとともに `USESUDO` を指定すると、ユーザー名およびパスワードは sudo の認証情報であるとみなされます。

`NO_APP_ACCOUNT`

設定可能な値：

`YES`、`NO`、デフォルト = `NO`

SUM は iLO でアプリケーションアカウントを作成し、それを iLO との通信に使用します。このパラメーターは SUM に対し、iLO でアプリケーションアカウントを作成しないように指示します。このパラメータを使用する場合、SUM が iLO と通信するためには iLO 認証情報が必要です。

HPE コンピュートソフトウェアおよびファームウェア製品ドキュメントクイックリンク

このインフォグラフィックでは、記載されている HPE 製品のドキュメントへのクイックリンクを提供します。製品の理解、導入、およびトラブルシューティングに役立つドキュメントに簡単にアクセスできます。

各製品ごとのセクションには、現在のバージョン、過去の 2 つのメジャーバージョン、およびそれらの改訂版が記載されています。

Hewlett Packard Enterprise サポートセンターのナビゲーション

- Navigation and workspace
- Search and product knowledge

サブトピック

Smart Update Manager のクイックリンク
このページは、Smart Update Manager に関するすべてのドキュメントの包括的なリストです。各ドキュメントの最新バージョンへのクイックリンクを提供します。

Smart Update Manager のクイックリンク

このページは、Smart Update Manager に関するすべてのドキュメントの包括的なリストです。各ドキュメントの最新バージョンへのクイックリンクを提供します。

ドキュメント	10.x 以降のバージョン	9.x バージョン
Smart Update Manager リリースノート	12.4.0	9.1.0
	12.3.0	9.0.2
	12.2.0	9.0.1
Smart Update Manager ユーザーガイド	12.4.0	9.0.1
	12.3.0	
	12.2.0	
Smart Update Manager CLI ガイド	12.2.0	
	12.0.0	
	10.5.0	
Smart Update Manager お使いになる前に	12.0.0	
アラート - カスタマーアドバイザリ	すべてのバージョン	

Web サイト

一般的な Web サイト

Single Point of Connectivity Knowledge (SPOCK) ストレージ互換性マトリクス

https://www.hpe.com/storage/spock

テクニカルペーパーおよび分析レポート

https://www.hpe.com/us/en/resource-library

上記以外の Web サイトについては、サポートと他のリソースを参照してください。

サポートと他のリソース

サブトピック

Hewlett Packard Enterprise サポートへのアクセス
HPE 製品登録
アップデートへのアクセス
リモートサポート
保証情報
規定に関する情報
ドキュメントに関するご意見、ご指摘

Hewlett Packard Enterprise サポートへのアクセス

- ライブアシスタンスについては、Contact Hewlett Packard Enterprise Worldwide の Web サイトにアクセスします。

  https://www.hpe.com/info/assistance

- ドキュメントとサポートサービスにアクセスするには、Hewlett Packard Enterprise サポートセンターの Web サイトにアクセスします。

  https://www.hpe.com/support/hpesc

ご用意いただく情報

- テクニカルサポートの登録番号（該当する場合）
- 製品名、モデルまたはバージョン、シリアル番号
- オペレーティングシステム名およびバージョン
- ファームウェアバージョン
- エラーメッセージ
- 製品固有のレポートおよびログ

- アドオン製品またはコンポーネント
- 他社製品またはコンポーネント

HPE 製品登録

Hewlett Packard Enterprise サポートセンターおよび購入したサポートサービスのメリットを最大限に活用するため、契約と製品を HPESC のアカウントに追加してください。

- 契約と製品を追加すると、パーソナライゼーションの強化、ワークスペースのアラート機能、ダッシュボードを通じた有益な情報が提供され、環境の管理が容易になります。
- また、問題を自己解決するための推奨事項やカスタマイズされた製品知識も提供されるほか、ケースを作成する必要がある場合は、最適化されたケース作成によって解決までの時間が短縮されます。

契約と製品を追加する方法については、https://www.hpe.com/info/add-products-contracts を参照してください。

アップデートへのアクセス

- 一部のソフトウェア製品では、その製品のインターフェイスを介してソフトウェアアップデートにアクセスするためのメカニズムが提供されます。ご使用の製品のドキュメントで、ソフトウェアの推奨されるアップデート方法を確認してください。
- 製品のアップデートをダウンロードするには、以下のいずれかにアクセスします。

  Hewlett Packard Enterprise サポートセンター

  https://www.hpe.com/support/hpesc

  マイ HPE ソフトウェアセンター

  https://www.hpe.com/software/hpesoftwarecenter

- eNewsletters およびアラートをサブスクライブするには、以下にアクセスします。

  https://www.hpe.com/support/e-updates-ja

- お客様の資格情報を表示およびアップデートするには、または契約と標準保証をお客様のプロファイルにリンクするには、 Hewlett Packard Enterprise サポートセンター **More Information on Access to Support Materials** ページをご覧ください。

https://www.hpe.com/support/AccessToSupportMaterials

重要

Hewlett Packard Enterprise サポートセンターからアップデートにアクセスするには、製品資格情報が必要な場合があります。関連する資格情報で HPE アカウントをセットアップしておく必要があります。

リモートサポート

リモートサポートは、保証またはサポート契約の一部としてサポートデバイスでご利用いただけます。リモートサポートは、インテリジェントなイベント診断を提供し、ハードウェアイベントを Hewlett Packard Enterprise に安全な方法で自動通知します。これにより、ご使用の製品のサービスレベルに基づいて、迅速かつ正確な解決が行われます。Hewlett Packard Enterprise では、ご使用のデバイスをリモートサポートに登録することを強くお勧めします。

ご使用の製品にリモートサポートの追加詳細情報が含まれる場合は、検索を使用してその情報を見つけてください。

HPE リモート IT サポートサービス接続入門

https://support.hpe.com/hpesc/public/docDisplay?docId=a00041232ja_jp

HPE Tech Care Service

https://www.hpe.com/jp/techcare

HPE Complete Care Service

https://www.hpe.com/jp/completecare

保証情報

ご使用の製品の保証に関する情報を確認するには、**標準保証確認ツール**を参照してください。

規定に関する情報

安全、環境、および規定に関する情報については、Hewlett Packard Enterprise サポートセンターからサーバー、ストレージ、電源、ネットワーク、およびラック製品の安全と準拠に関する情報を参照してください。

https://www.hpe.com/support/Safety-Compliance-EnterpriseProducts

規定に関する追加情報

Hewlett Packard Enterprise は、REACH（欧州議会と欧州理事会の規則 EC No 1907/2006）のような法的な要求事項に準拠する必要に応じて、弊社製品の含有化学物質に関する情報をお客様に提供することに全力で取り組んでいます。この製品の含有化学物質情報レポートは、次を参照してください。

https://www.hpe.com/info/reach

RoHS、REACH を含む Hewlett Packard Enterprise 製品の環境と安全に関する情報と準拠のデータについては、次を参照してください。

https://www.hpe.com/info/ecodata

企業プログラム、製品のリサイクル、エネルギー効率などの Hewlett Packard Enterprise の環境に関する情報については、次を参照してください。

https://www.hpe.com/info/environment

ドキュメントに関するご意見、ご指摘

Hewlett Packard Enterprise では、お客様により良いドキュメントを提供するように努めています。ドキュメントを改善するために役立てさせていただきますので、何らかの誤り、提案、コメントなどがございましたら、 Hewlett Packard Enterprise サポートセンターポータル（https://www.hpe.com/support/hpesc）の**フィードバック**ボタンとアイコン（開いているドキュメントの下部にある）からお寄せください。このプロセスにより、すべてのドキュメント情報が取得されます。