Hewlett Packard Enterprise

Smart Update Manager 10.5.0 CLI Guide

Hewlett Packard Enterprise

Smart Update Manager 10.5.0 CLI Guide

Abstract

This document describes how to use SUM CLI and Input file modes to apply firmware updates to HPE ProLiant and Integrity servers, and apply system software updates to HPE ProLiant, HPE BladeSystem, HPE Synergy, HPE EdgeLine and HPE Apollo servers and infrastructure. This document is intended for individuals who understand the configuration and operations of Microsoft Windows, Windows Server, Linux, smart components, VMware, and the risk of data loss from performing updates.

Notices

Acknowledgments

Table of contents

Introduction to SUM CLI

Subtopics

SUM CLI and Input file modes

SUM CLI modes

Scheduling updates

Linux root credentials for remote nodes

SUM CLI and Input file modes

The SUM CLI and Input file modes allow you to deploy updates from a baseline to the local and remote nodes.

In CLI mode, add all parameters in one command line to execute the entire sequence on multiple nodes without any user interaction. This method requires the `silent` command. Use this method if you want to update one node.

Use Input file mode to add all parameters to a text file, and then call SUM using the `inputfile` command with the text file name as the parameter. This method requires the `silent` command. Use this method if you want to update multiple nodes.

The SUM CLI and Input file modes are process-oriented tools. SUM finishes the command, and then it does not save the settings or results. If you want to perform a few steps and save the work for later, use the GUI mode.

SUM CLI modes

> **NOTE:**
>
> Starting with SUM 8.5.0, the interactive command-line mode of update is deprecated. To ensure that the scripts work properly, migrate from interactive CLI to legacy command line. The command line with input file must satisfy most of the customer needs and also allows updates at scale across SPPs and SUM versions without having to change the command line inputs or the input file.

Choose the SUM CLI mode that fits your server environment best.

CLI

   Number of nodes: 1–3

   Strengths:

   - Use one command to issue all instructions to the nodes.

Input file CLI

   Number of nodes: 1–50

   Strengths:

   - Create a file that includes all information for baselines and nodes.

   - Save the files for future use.

   - Issue a CLI command that calls the Input file.

Scheduling updates

Scheduling updates

Use standard operating system tools to schedule SUM to run a CLI command. For example, use Task Scheduler for Windows or cron for Linux systems. Be sure to include the `s` (silent) parameter in the command. See the operating system documentation for more information about scheduling tasks.

> 📄 **NOTE:**
>
> Input file supports scheduling parameters.

Linux root credentials for remote nodes

You can provide root credentials, a user with sudo permissions, or provide nonroot with root credentials for remote nodes.

To use super user functionality, configure the user as a super user with all root privileges. You can also use nonroot user with a root user to update components.

Subtopics

Prerequisites for using Linux sudo credentials

Using an SSH keyfile with root credentials

Prerequisites for using Linux sudo credentials

- Provide one of the following:
  - Username and password
  - Username and SSH key file path (in PEM format)
- Give the sudo users write access to the `/var/tmp` directory.
- For sudo users, add the user to the `/etc/sudoers` file. The following shows the privileges and specifications for users.
  - User: `<Sudo_user>` (the actual name of the user account)
  - Privilege: `ALL`
  - Specification: `ALL`
- Edit the entry in the `/etc/sudoers` file so the system asks for the sudo user password instead of root user password when you run the sudo command.
- The following shows the privileges to comment or remove from `/etc/sudoers`.
  - User: `ALL`
  - Privilege: `ALL`
  - Specification: `ALL`

  > 📄 **NOTE:**
  >
  > Only use this option with `Defaults targetpw` on all systems.

- When using SSH keys for login credentials combined with sudo, edit the entry in the `/etc/sudoers` file so that the system does not ask for the user password as follows:
  - User: `sudo_user`

- Privilege: `ALL`

- Specification: `NOPASSWD:ALL`

Using an SSH keyfile with root credentials

About this task

SUM allows you to log in with a password or by providing an SSH key.

Procedure

1. On the node that is running SUM, type `ssh-keygen -t rsa`. SUM also supports DSA encryption key format. To create a DSA encryption key, type `ssh-keygen -t dsa`. A passphrase is optional.

2. Save the key in the directory `.ssh/id_rsa.pub`. Root users, use the directory `/root/.ssh`.

3. Verify that the private key is in PEM format by opening the key and viewing the top of the key for the following text:

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

4. If necessary, on the node you want to update, create an `.ssh` directory at the root level. Set the permissions level on the directory to 700.

5. Determine if the node you want to update has the filename `.ssh/authorized_keys`. If this file is not present, create or append this file. Set the permissions on the `authorized_keys` file to 640.

   > 📋 **NOTE:**
   >
   > If you append the file, more users are authorized to use the private key.

6. Copy the contents of `.ssh/id_rsa.pub` on the host node, and then paste them into the filename `.ssh/authorized_keys` on the remote node.

7. Open an SSH path between the host node and the node you want to update.

   a. Type `ssh root@10.0.0.1`. This parameter opens a shell path.

   b. Type `ssh root@10.0.0.1 uname`. This parameter executes the command and returns.

8. SUM requires access to `id_rsa` (private key file), and the optional passphrase.

   > 📋 **NOTE:**
   >
   > If other applications require the public key in PEM format, you can convert it by typing `ssh-keygen -e -f id_rsa.pub > id_rsa_pub.pem`. Open the key, and look at the PEM format for `---- BEGIN SSH2 PUBLIC KEY ----`.

Results

To create a private key on a Windows system, you can use an application like PuTTY or PuTTY Key Generator (PuTTYGen). For more information, see **https://www.digitalocean.com/community/tutorials/how-to-create-ssh-keys-with-putty-to-connect-to-a-vps**

To export the file to PEM format, select Conversion > Export OpenSSH key.

SUM CLI and input file commands overview

Subtopics

Command-line syntax

In SUM CLI mode, you must enter all parameters in one command line. When you press the Enter key, SUM executes the entire sequence. All CLI commands require the `s` (silent) parameter in the command, `/s` (Windows) or `--s` (Linux) to run.

> (i) **IMPORTANT:**
>
> Command-line mode does not support double-byte character sets. Any commands entered in the command line using a double-byte character set do not appear correctly.

The general command-line syntax for SUM is as follows:

- On Windows, use a slash before each argument.

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

- On Linux, use two hyphens before each argument.

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

Command-line description

`smartupdate`

> Launches SUM.

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

> Initiates the CLI mode.
>
> For more installation parameters, see Installation parameters.

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

> Provides details about the target node.
>
> For more node parameters, see Node parameters.

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

> This parameters tells SUM how to handle dependency issues.
>
> For more override error parameters, see Override error parameters.

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

> Tells SUM whether to reboot a node after deploying updates.
>
> For more reboot parameters, see Reboot parameters.

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

> Tells SUM to use verbose log settings.
>
> For more log parameters, see Log file parameters.

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

> Generates SUM reports.
>
> For more report parameters, see <u>Report parameters</u>.

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

> Launches the SUM CLI help file.

Input file overview

The Input files contain configuration details and target node information. You can include user credentials and remote host information for updates.

> (i) **IMPORTANT:**
>
> Input files are plain-text files, for greater security, omit the credentials from the file and pass them to SUM using the command line. If you provide user credentials on the command line, all nodes in the Input file must use the same user ID and password.

SUM CLI with Input files allows you to create `.txt` file scripts for updating nodes. Use a plain-text editor to create the Input file. It is similar to markup languages that require headers and trailers to match. Use commas as list separators for parameters that can take list values.

After creating your script file, add it as the `inputfile` **<filename>** parameter to a SUM command line. For example, to run the Input file `update.in`, type:

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

- Windows: `smartupdate /inputfile update.in`

You can add full paths to the Input file location when the file is not stored in the same location as the SUM executable files. Enclose the **<filename>** field in double quotes to enable paths with spaces.

> 📝 **NOTE:**
>
> There is no Input file parameter to open a firewall port. If you want to open the firewall, add the parameter `--open_firewall` (Linux) or `/open_firewall` (Windows) to the command that calls the input file. For example:
>
> - Linux: `smartupdate --inputfile update.in --open_firewall --s`
>
> - Windows: `smartupdate /inputfile update.in /open_firewall /s`

To modify an Input file without editing the file, use all the command-line options along with the Input file. The command-line options take precedence over the commands in the Input file.

Input file sections

An Input file contains Configuration and Targets information sections. An Input file only supports one parameter per line of text.

```
#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
[END]
```

Configuration section

This section assigns all parameters for an Input file. The configuration section ends at the beginning of the `TARGETS` section.

Lines that begin with a pound (#) character are notes. No more than 1 pound character is allowed on a line.

```
#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 section

This section defines each node that you are updating.

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

Targets section

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

If more than one node uses the same user credentials, list them all in the same `TARGETS` section.

You can add text to a `[TARGETS]` title to provide more target details.

SUM CLI command task examples

Subtopics

About CLI command examples

Issuing an Input file command

Updating the local host

About CLI command examples

The following tasks provide examples on how to compose CLI commands and Input files. For more information about parameters you can use, see SUM CLI parameters and Input file parameters.

Issuing an Input file command

Procedure

1. Open a command-line window.

2. Navigate to the directory where SUM is located.

3. Type the following command, depending on your operating system:

   - Linux: `./ smartupdate --inputfile <path:/inputfile.txt>`

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

     > 📄 **NOTE:**
     >
     > If the Input file does not contain the parameter `SILENT = YES`, include the CLI `s` (silent) parameter. If no silent parameter is provided, SUM cannot process the Input file.

Updating the local host

The following examples apply all updates in a baseline to the local host.

Using a baseline in the same directory as SUM in CLI mode

- Linux: `./ smartupdate --silent`

- Windows: `smartupdate /silent`

Using a baseline in a different directory than SUM in CLI mode

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

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

Using a baseline in the same directory as SUM in Input file mode

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

SUM supports Windows domains, for example, `domain1\userid1`.

Using a baseline in a different directory than SUM in Input file mode

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

Updating a remote node

The following examples apply all updates in a baseline to a remote node.

Using a baseline from the same directory running SUM in CLI mode

- 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 supports Windows domains, for example, `domain1\userid1`

Using a baseline in a different directory than SUM in CLI mode

- 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 supports Windows domains, for example, `domain1/userid1`

Using a baseline from the same directory running SUM in Input file mode

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

Using a baseline in a different directory than SUM in Input file mode

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

Updating two nodes with one baseline

Applying all updates in a baseline to two nodes in CLI mode

- **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>`

  > 📄 **NOTE:**
  >
  > These examples use the baseline in the same directory where SUM is running. If you want to use a different baseline, include the following command with the directory path for the baseline:
  >
  > `/use_location <baseline_directory_path>`

Applying all updates in a baseline to two nodes in Input file mode

```
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]
```

> **NOTE:**
> If both nodes use the same user credentials, only provide them once. Instead of creating a second `TARGETS` entry, enter the IP address for the second node in the first node entry. For example:

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

Updating only the software on a node

Using these examples, SUM updates only the software on a remote node.

Updating software on a node from a baseline in the same directory as SUM in CLI mode

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

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

> **NOTE:**
> If you want to use a different baseline, include the following command:
>
> - **Linux:** `--use_location <baseline_directory>`
>
> - **Windows:** `/use_location <baseline_directory>`

Updating software on a node from a baseline in the same directory as SUM in Input file mode

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

This example uses the baseline that is located in the same directory where SUM is running.

To use a different baseline, add the following to the Input file:

```
SOURCEPATH = <directory_path>
```

Updating only the firmware on a node

To update only the firmware on a node, make the following changes to the examples that deployed software only:

- **CLI mode**—remove the command `softwareonly` and add `romonly`.

- **Input file mode**—remove the command `SOFTWAREONLY` and add `ROMONLY`.

Deploying the latest components to a node

Using these examples, SUM only applies the latest updates from a directory that contains multiple baselines.

Deploying the latest components in CLI mode

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

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

Deploying the latest components in Input file mode

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

Forcing deployments to an earlier version of a component on a node

In the following examples, SUM forces software deployments of earlier versions of a component on a node.

Forcing deployments in CLI mode

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

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

Forcing deployments in Input file mode

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

Deploying two software components

In the following examples, deploys two components to the local host. The example also forces a rewrite or downgrade if the component is already installed and it is a software component.

> 📝 **NOTE:**
> In this example, if the listed components are not software, SUM does not deploy them.

Deploying two components in CLI mode

Select one of the following examples. Both examples perform the task.

- Linux option 1: `./ smartupdate  --f:software hponcfg-5.2.0-0.x86_64.rpm hp-smh-templates-10.6.1-1481.4.noarch.rpm --s`

- Linux option 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 option 1: `smartupdate  /f:software cp008097.exe cp008257.exe /s`

- Windows option 2: `smartupdate  /c cp008097.exe /c cp008257.exe /f:software /s`

Deploying two components in Input file mode

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

Using sudo to update a remote Linux node

In the following examples, SUM logs in to the remote Linux node with a `userid` and then executes commands using the `sudo` function. SUM uses the baseline in the directory running SUM.

Using sudo to update a remote Linux node in CLI mode

- 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`

Using sudo to update a remote Linux node in Input file mode

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

Using sudo to update multiple remote Linux nodes in CLI mode

- 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`

Using super user credentials to update nodes

In the following examples, SUM logs in to a remote node and then uses super user credentials to update a remote node.

Using super user credentials in CLI mode

- 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>`

Using super user credentials in Input file mode

Updating one node

```
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]
```

Updating two nodes that use the same super user credentials

```
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]
```

Updating two nodes that use different super user credentials

```
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]
```

Using an SSH Key to update Linux nodes

The following examples allow SUM to log in using an SSH Key and update remote Linux nodes. SUM supports DSA and RSA encryption key formats. Run the following command to generate a private key.

`ssh-keygen -t -dsa` **or** `ssh-keygen -t -rsa`

Using an SSH Key in CLI mode.

- **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`

Using an SSH Key in Input file mode.

Updating one node.

```
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]
```

Updating two nodes that use the SSHKEY and same keyfile.

```
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]
```

Updating two nodes that use different SSH Key files and passphrases.

```
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 parameters

Subtopics

Management agent installation parameters

SUM CLI parameters

Return codes

Windows smart-component return codes

Linux software RPM return codes

Linux smart-component and firmware RPM return codes

VMware ESXi smart-component return codes

Management agent installation parameters

To find out if you have any management agents installed, use SUM to create the Firmware Details Report. For example:

```
$ ./smartupdate --firmware_report --s
```
 (Linux)

```
C:\ smartupdate /firmware_report /s
```
 (Windows)

> **NOTE:**
> If the node already has management agents installed on it, SUM upgrades the agents if a newer version is available.
>
> For example, if the node has SNMP installed on it, and you use the `use_ams` parameter, AMS is installed and SNMP is upgraded.

The following describes what SUM deploys when you issue the given commands:

If you provide no management agent parameters.

> Gen8 and later servers installed agent: AMS

`--no_mgmt` (Linux) `/no_mgmt` (Windows)

> This parameter prevents any installation or upgrade of SNMP, AMS, and WBEM agents. It has no impact on any management agents installed on the system.

> Gen8 and Gen9 servers installed agent: None

`--use_ams` (Linux) `/use_ams` (Windows)

> Gen8 and later servers installed agent: AMS

`--use_snmp` (Linux) `/use_snmp` (Windows)

> **NOTE:**
>
> Gen10 and above servers do not support SNMP agents.

`/use_wmi` (Windows only)

> Gen8 and later servers installed agent: WBEM

> **NOTE:**
>
> Gen10 and above servers do not support WBEM agents.

`--use_ams --use_snmp` (Linux) `/use_ams /use_snmp` (Windows)

> Gen8 and Gen9 servers installed agents: AMS and SNMP

`/use_ams /use_wmi` (Windows)

> Gen8 and Gen9 servers installed agents: AMS and WBEM

`/use_wmi /use_snmp` (Windows)

> Gen8 and Gen9 servers installed agents: WBEM and SNMP

`/use_ams /use_wmi /use_snmp` (Windows)

> Gen8 and Gen9 servers installed agents: AMS, SNMP, and WBEM

Gen9 only: SUM installs SMH templates when it installs SMNP agents.

SUM CLI parameters

SUM recognizes the following command-line arguments. You cannot use some arguments, such as `--romonly` and `--softwareonly`, together.

Subtopics

[Help](#)

[Installation parameters](#)

[Override error parameters](#)

[Reboot parameters](#)

[Node parameters](#)

[Test run parameter](#)

[Log file parameters](#)

Help

View the help for the CLI by typing `smartupdate --s --h`.

Syntax

`--h[elp] or --?` **(Linux)**

`/h[elp] or /?` **(Windows)**

Description

Displays command-line help information.

Installation parameters

The following section lists attributes you can use while installing updates.

Subtopics

Force

Syntax

`--f[orce]` **(Linux)**

`/f[orce]` **(Windows)**

Description

Rewrites or downgrades an existing software installation. This parameter produces the same results as `f:software`.

Options

`f[orce]:bundle`

Rewrites or downgrades the existing installation of components in the selected bundle.

`f[orce]:rom`

Rewrites or downgrades the existing installation of the applicable firmware components (applies to firmware only).

`f[orce]:software`

Rewrites or downgrades the existing installation of the selected software components.

`f[orce]:all`

Rewrites or downgrades the existing installation of applicable software and firmware components.

Downgrade

Syntax

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

`/g or /downgrade` **(Windows)**

Description

Selects components for installation that have an available version lower than the currently installed version. Upgrades also occur when you issue this command. Can be combined with `rewrite`.

Rewrite

Syntax

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

`/e or /rewrite` **(Windows)**

Description

Selects components for installation that have an available version lower than the same as the installed version. Upgrades also occur when you issue this command. Can be combined with `downgrade`.

ROM deployment only

Syntax

`--romonly` **(Linux)**

`/romonly` **(Windows)**

Description

Limits components considered for installation to only firmware components. Cannot be used with `softwareonly`. If `softwareonly` or `romonly` are not used, all components are considered for possible installation.

Software deployment only

Syntax

`--softwareonly` **(Linux)**

`/softwareonly` **(Windows)**

Description

Limits components considered for installation to only software components. Cannot be used with `romonly`. If `softwareonly` or `romonly` are not used, all components are considered for installation.

Network port used

Syntax

`--port <n>` **(Linux)**

`/port <n>` **(Windows)**

Description

Changes the port SUM uses for the internal HTTP server from 63001 to the selected port.

SSL port

Syntax

`--ssl_port <n>` **(Linux)**

`/ssl_port <n>` **(Windows)**

Description

Changes the port SUM uses for the internal HTTPS server from 63002 to the selected port.

Open firewall

Syntax

`--open_firewall` **(Linux)**

`/open_firewall` **(Windows)**

Description

Allows SUM to open the HTTP and HTTPS ports used by SUM for external access. Also used to open the ports for remote node functionality and for remote browser access.

SUM deploys updates to nodes even when ip_tables include DROP commands. Use `open_firewall` to bypass the DROP rules.

Include this parameter with the command that calls an Input file. SUM Input files do not support a parameter to open a firewall.

Silent deployment

Syntax

`--s[ilent]` (Linux)

`/s[ilent]` (Windows)

Description

Causes the installation to run silently with no GUI CLI output. All data writes to the log file. Do not use the `express_install` parameter with the `silent` parameter.

Usage

SUM requires the silent parameter for CLI commands.

Specify components to deploy

Syntax

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

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

Description

Specifies which components to install.

If you use the `c[omponent]` parameter, specify only one component per parameter. You can include multiple `c` parameters with individual components in the same command line.

- If you do not use the `--c[omponent]` argument, specify multiple components separated by a blank space, and listed after all other arguments on the command line.

- By default SUM deploys components in the order listed in the command. SUM changes deployment order based on dependency issues.

- Specify multiple components (see `--c[omponent]<component_to_install>`) and bundles on the same command line. When you mix components and bundles on the command line, the filter switches control what components and bundles are installed.

- SUM only uses the component name. If you enter the entire directory path, SUM ignores the path. If you use `--use_location` and `c`, SUM checks both the default repository and the directory provided.

If you specify more than five components, either issue multiple commands or use an Input file.

Example

```
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)
```

Define bundles to install

Syntax

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

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

Description

This argument specifies bundles to install.

Specify multiple components and bundles (see `b[undle] <bundlename>`) on the same command line. When you mix components and bundles on the command line, the filter switches control which components and bundles are installed.

Define baseline location

Syntax

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

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

Description

Specifies a directory or file share that contains the SPP and components for use with SUM. SUM expects that the targeted **file_share** is a mapped file share or UNC format.

Without this argument, SUM defaults to the directory that contains SUM.

Make sure that the logged-in account has access to this location. If the location is a UNC file share, you must provide credentials to the file share using the `unc_username` and `unc_password`. If you do not provide these credentials, SUM uses the current credentials to access the share.

SUM does not support http baselines in CLI mode.

Import component configuration

Syntax

`--import_configuration <directory_path>` (Linux)

`/import_configuration <directory_path>` (Windows)

Description

This parameter specifies the directory that contains component configuration settings for components in the specified baseline. Imported component configurations override components configurations in the baseline.

UNC user credentials

Syntax

`/unc_username` (Windows)

Description

Username credentials for accessing the file share. SUM supports Windows domains, for example: `domain1/userid1`

UNC password credentials

Syntax

`/unc_password` **(Windows)**

Description

Password credentials for accessing the file share.

No management agent components

Syntax

`--no_mgmt` **(Linux)**

`/no_mgmt` **(Windows)**

Description

Indicates that management components for AMS, SNMP, and WBEM Providers are not selected automatically. In silent mode, SUM does not update any management components.

For more information on management options, see Management agent installation parameters .

Use SNMP agent components

Syntax

`--use_snmp` **(Linux)**

`/use_snmp` **(Windows)**

Description

Gen8 and Gen9 servers: SUM automatically selects SNMP components for installation.

Gen10 and above servers: SUM lists SNMP components as not applicable.

Restrictions

Linux and Gen9 only: SUM selects the SNMP RPMs and the smh_templates RPMs.

Use WMI agent components

Syntax

`/use_wmi`

Description

SUM automatically selects WBEM components for installation.

Restrictions

Windows and Gen9 only.

Use AMS agent components

Syntax

`--use_ams` (Linux)

`/use_ams` (Windows)

Description

(Gen8 and later only) SUM automatically selects AMS components for installation.

Express install

Syntax

`--express_install` (Linux)

`/express_install` (Windows)

Description

Starts the GUI and begins Localhost Guided update in automatic mode. This parameter is ignored when the `silent` parameter is used.

Override error parameters

The following attributes determine how SUM handles errors.

SUM applies these attributes to all nodes included in the command.

Subtopics

[Bypass TPM](#)

[Ignore warnings](#)

[Continue installation and ignore errors](#)

[Override existing connection](#)

[Failed dependencies](#)

Bypass TPM

Syntax

`--tpmbypass` or `--ignore_tpm` (Linux)

`/tpmbypass` or `/ignore_tpm` (Windows)

Description

If you have enabled TPM on target iLO 5 or above, Windows, Linux, or VMware ESXi, ignore the warning message and continue with component installation. For more information about TPM, see the Smart Update Manager 10.0 User Guide posted at https://www.hpe.com/support/SUMGen11-UG-en.

Ignore warnings

Syntax

`--ignore_warnings` for Linux

`/ignore_warnings` for Windows

Description

Allows installation to proceed after SUM receives warnings on nodes. Some warnings include:

- Active member of a Serviceguard cluster. If you do not specify this option, and the node is an active member of a Serviceguard cluster, the node does not enter the installation or deploy phases.

- TPM warning

- Low disk space

- A system managed by HPE OneView

- Pending iLO installation queue warning

- Exception task queue warning

- iLO and iSUT warnings

Continue installation and ignore errors

Syntax

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

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

Description

Causes the installation to continue and ignore errors.

Options

Valid values:

`All`

Bypasses remote nodes that return any kind of error and continues deployment on other nodes.

`ServerNotFound`

Use the `ServerNotFound` option to bypass inactive or unavailable remote hosts when you deploy firmware or software to multiple remote hosts at the same time.

`BadPassword`

Use the `BadPassword` option to bypass remote nodes that report incorrect credentials provided and continue with other nodes.

`FailedDependencies`

Use the `FailedDependencies` option to bypass remote nodes with failed dependencies and continue with other nodes that are ready to be installed. Override this parameter by using `on_failed_dependency:Force` or `:OmitComponent`.

iLOCACEnabled

Use the `iLOCACEnabled` option to bypass the nodes on which CAC mode is enabled in iLO and continue with other nodes. iLO Administrator credentials are not required.

iLOHighSecurityMode

Use the `iLOHighSecurityMode` option to bypass the nodes on which High Security mode is enabled in iLO and continue with other nodes. iLO Administrator credentials are not required.

CHIFSignFailure

Use the `CHIFSignFailure` option to bypass the nodes when Signature verification of a CHIF driver component has failed and continue with other nodes.

CHIFInstallFailure

Use the `CHIFInstallFailure` option to bypass the nodes when a CHIF driver installation has failed and continue with other nodes.

Override existing connection

Syntax

`--override_existing_connection` (Linux)

`/override_existing_connection` (Windows)

Description

Instructs SUM to override a session in progress and reinitialize the installation framework on the remote node.

If you do not include this parameter, SUM skips the node if a SUM session is already running on the remote node.

Failed dependencies

Syntax

`--on_failed_dependency` (Linux)

`/on_failed_dependency` (Windows)

Description

This parameter gives options to either omit the components or skip the installation on host when a failed dependency is encountered, and SUM does not attempt an installation.

Options

`OmitHost` (default)

Puts the target into a failure state, and no installation is attempted.

`OmitComponent`

Deselects the affected components and proceeds with any updates that do not have dependency failures.

`Force`

Attempts all updates, even with dependency failures.

Samples

`--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)

Reboot parameters

The following attributes determine how SUM performs reboots.

Subtopics

Reboot

Reboot message

Reboot delay

Reboot always

Reboot

Syntax

`--r[eboot]` (Linux)

`/r[eboot]` (Windows)

Description

The following conditions cause the server (or host server in a remote installation) to reboot:

- When you select the `reboot` option or use it in a command-line argument.

- When all components selected for installation are successfully installed.

- When at least one of the installed components requires a reboot to complete its installation.

Reboot message

Syntax

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

`/reboot_message "reboot message"` (Windows)

Description

Displays the specified reboot message on remote consoles connected to the server that you want to reboot. Use this parameter with the `reboot` option, or the argument is ignored.

> 📑 **NOTE:**
> The `reboot_message` cli parameter is not functional for SLES12 OS because the SLES12 OS image has a constraint that does not permit executing the parameter as a cli command.

Reboot delay

Syntax

`--reboot_delay timeout_in_secs` (Linux)

`/reboot_delay timeout_in_secs` (Windows)

Description

Delays the reboot of the server for the length of time that the `timeout_in_seconds` variable specifies.

Defaults to 60 seconds.

Linux systems convert the reboot delay time from seconds to minutes, and any value under a full minute (59 seconds or less) rounds to the next minute.

Options

Acceptable values are between 15 and 3600.

Requirements

Use this argument with the `reboot` option, or the argument is ignored.

Reboot always

Syntax

`--reboot_always` (Linux)

`/reboot_always` (Windows)

Description

SUM always reboots the server when this command line option is passed.

Node parameters

The following attributes define update options for nodes. When providing an attribute, provide the target node and then the attribute, do not specify the attribute before the target node.

For example:

Valid:

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

Invalid:

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

Global attributes can be applied to nodes that have no provided attributes. Provide global attributes at the end of the command line with one target node where the attributes apply. For example:

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

The following attributes can be used globally:

`username/user`, `password/passwd`, `use_location`, `current_credential`, `su_username`, `su_password` **and** `use_sshkey`.

Subtopics

User name

Syntax

`--user` **<username>** or `--username` **<username>** (Linux)

`/user` **<username>** or `/username` **<username>** (Windows)

Description

Use this argument to log in to the remote nodes with the user ID.

Requirements

Make sure that the user name belongs to the root or administrator group.

SUM supports Windows domains, for example: `domain1\userid1`

If you are managing a Linux node that is part of a Windows domain, use the following format:

`<domain>\<username>`

User password

Syntax

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

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

smartupdate also supports `passwd` and `pwd` (Windows)

Description

Password for the user ID specified in the `user` parameter. The system uses the password to log in to remote nodes.

iLO user name

Syntax

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

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

Description

Use this parameter to connect to iLO from the host OS when the iLO of server has been configured in High Security mode. Provide iLO Administrator credentials.

This parameter is valid only for Windows, Linux, and Unknown node types.

Requirements

Make sure that the iLO Administrator account has **Configure iLO Settings** permissions.

iLO password

Syntax

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

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

Description

Password associated with the iLO user name that is specified in `ilo_username` parameter.

Use SSH key file

Syntax

`--use_sshkey` (Linux)

`/use_sshkey` (Windows)

Description

Use SSH Key to connect to the Linux nodes. SUM ignores this parameter for other nodes types.

When you provide this parameter, SUM requires the Private key file location. A key file, in PEM format, is an optional parameter you can pass.

Private key file location

Syntax

`privatekeyfile`

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

Description

If a dedicated SSH key file is generated, then SUM automatically detects that a passphrase is required and prompts for one.

Requirements

Use with the parameter `use_sshkey`.

SSH passphrase

Syntax

`passphrase`

Description

Use this option to provide the passphrase that is used with the parameter `use_sshkey` and the option `privatekeyfile`.

If passphrase is required by the key file and is not provided as a parameter, SUM prompts for the passphrase.

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

Requirements

Use with the parameter `use_sshkey`.

Super user name

Syntax

`--su_username` (Linux)

`/su_username` (Windows)

Description

Use root (super user) username to start a session on a remote node, inventory, and update components when the credentials specified in `username` and `passwd` do not have root privileges to update components.

Do not use this command on the local host.

Super user password

Syntax

`--su_password` (Linux)

`/su_password` (Windows)

Description

Use root (super user) username to start a session on a remote node, inventory, and update components when the credentials specified in `username` and `passwd` do not have root privileges to update components.

Do not use this command on the local host. To launch SUM on the local host using su, use the following example:

`su root`

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

Use sudo credentials

Syntax

`--use_sudo` (Linux)

`/use_sudo` (Windows)

Description

Specifies that username and password for a remote node are sudo user credentials.

Requirements

When used with `username` and `passwd`, indicates that both parameters are sudo user credentials.

Define target address

Syntax

`--target` **"netAddress"** (Linux)

`/target` **"netAddress"** (Windows)

Description

The IP address or the DNS name of a remote host, which can be a remote server, remote iLO NIC port.

Define target type

Syntax

`--targettype` **"type"** (Linux)

`/targettype` **"type"** (Windows)

Description

Reduces inventory time for scripted deployments. It is an optional command-line argument.

Possible values

`Windows`

`Linux`

`iLO`

Use `targettype` with the `target` parameter. You can interchange the sequence of `targettype` and `target`. If the node name includes spaces, enclose the name in quotation marks:
" "

User current credentials

Syntax

`/current_credential` (Windows only)

Description

Enables the use of the localhost credential to access the nodes instead of providing the user name and password explicitly for each node. Assumes that the current credential is valid for the nodes you are processing. Applies to Windows nodes only.

iLO5 and above parameters

Save install set

Syntax

`--save_install_set` (Linux)

`/save_install_set` (Windows)

Description

This parameter saves the install set in the iLO repository for iLO5 and above nodes.

Install set name

Syntax

`--install_set_name` (Linux)

`/install_set_name` (Windows)

Description

This parameter saves the install set with the provided name in the iLO repository for iLO5 and above nodes.

Install set description

Syntax

`--install_set_description` (Linux)

`/install_set_description` (Windows)

Description

This parameter saves the install set with the provided description in the iLO repository for iLO5 and above nodes.

Skip missing component signatures

Syntax

`--skip_missing_compsig` (Linux)

`/skip_missing_compsig` (Windows)

Description

Skips components that are missing a component signature, instead of stopping the entire node deployment.

Manually manage iLO repository

Syntax

`--manually_manage_ilo_repository` (Linux)

`/manually_manage_ilo_repository` (Windows)

Description

Manage the iLO repository manually instead of automatically.

> 📝 **NOTE:**
>
> Default is automatic, which means that if iLO NAND does not have space to accommodate new components that SUM uploads, SUM deletes the install set with minimum size (based on total component size) and creates space for new components and install set.

Use iLO saved

Syntax

`--use_ilo_saved` (Linux)

`/use_ilo_saved` (Windows)

Description

Use an install set saved on the iLO repository as baseline. SUM will invoke this install set in iLO NAND and process the resultant task queue during deployment.

Skip iLO

Syntax

`--skip_ilo` (Linux)

`/skip_ilo` (Windows)

Description

This command tells SUM to remove iLO smart components from an install set. (EFM only)

Skip_prereqs

Syntax

`--skip_prereqs` (Linux)

`/skip_prereqs` (Windows)

Description

SUM does not install the prerequisite components, such as driver components during self-inventory, prerequisite components for a firmware if it needs the installed version to be at a minimum active version, the CHIF driver on the remote Windows node.

Restrictions

CHIF driver installation is valid for Windows nodes only.

Test run parameter

Syntax

`--dryrun` (Linux)

`/dryrun` (Windows)

Description

Use this attribute along with other attributes to have SUM run through all processes in the command, except the actual deployment. Use this parameter to verify connections and configurations for remote nodes and to see what SUM will update.

Requirements

This argument simulates the installation for a test run. SUM does not deploy any updates.

Log file parameters

The following attributes determine how SUM creates log files.

Subtopics

Log parameters

Logging verbosity level

Log parameters

Log directory

Syntax

`--logdir` **"path"** (Linux)

`/logdir` **"path"** (Windows)

Description

Redirects the output from SUM to a directory other than the default location.

For SUM running on Windows systems, the default location is `%SYSTEMDRIVE%\CPQSYSTEM\sum\log\<netAddress>` and the redirected location is `<path>\sum\log\<netAddress>`.

SUM creates the `%SYSTEMDRIVE%\CPQSYSTEM\` directory as it processes components. SUM writes only the `smartupdate_log.txt`, `smartupdate_detail_log.txt`, and `smartupdate_InstallDetails.txt` files to the redirected directory. SUM writes other logs to the default directory.

For SUM running on Linux systems, the default location is `/var/log/sum/<netAddress>` and the redirected location is `<path>/log/sum/<netAddress>`.

Debug log directory

Syntax

`--debuglogdir [directory_path]` (Linux)

`/debuglogdir [directory_path]` (Windows)

Description

This parameter assigns the directory where SUM saves debug log files.

Cleanup on exit

Syntax

`--cleanup_onexit` (Linux)

`/cleanup_onexit` (Windows)

Description

This parameter removes the following from target and host:

- Firmware RPMs copied to the Linux folder `/usr/lib/i386-linux-gnu/*` or `/usr/lib/x86_64-linux-gnu`.

- Extracted Firmware RPM directory and folder contents.

- Extracted components during baseline inventory.

- If SUM is run from a read-only location, this parameter deletes the 'localsum' directory.

> 📋 **NOTE:**
>
> This parameter does not delete the debug log files.

Cleanup all logs on exit

Syntax

`--cleanupall_onexit` **(Linux)**

`/cleanupall_onexit` **(Windows)**

Description

This parameter removes all files, including debug files from target and host. This parameter does not delete the user logs.

Logging verbosity level

Syntax

`--v[erbose]` **or** `--veryv[erbose]` **(Linux)**

`/v[erbose]` **or** `--veryv[erbose]` **(Windows)**

Description

Sets the verbosity level for the SUM execution log file, `sum_execution_log_*.log`. Allows you to increase the level of detail that the log files retain. Defaults to normal verbosity.

Report parameters

> 📋 **NOTE:**
>
> The combined report and installed reports can only be generated after completing a deployment.

SUM generates reports about the specified system or repository location using the command-line arguments. If you do not specify additional locations on the command line, SUM uses the local host and default repository locations (the directory where SUM was initiated). Specify a node if you provide the appropriate credentials. Specify a repository by using other command-line parameters to generate reports. For specific commands, see SUM CLI parameters.

SUM generates the reports as XML or HTML files that you can view in a JavaScript-enabled web browser, or a CSV-format report that you can open in any application that supports CSV format.

HTML reports are generated under a directory named `SUM_<Type>_Report_<date>_<time>`.

The location defaults to the present working directory where SUM is initiated. If that location is write-protected, SUM saves the report to the same directory as the SUM log files. Use the `reportdir` parameter to specify a different directory where SUM can save the reports.

The SUM report files are located in these default directories:

Windows: `C:\cpqsystem\sum\log`

Linux: `/var/log/sum`

When SUM generates the report, the SUM GUI does not appear. SUM displays the file location for the generated report.

Subtopics

Generate node summary report

Generate node summary report

Syntax

`--report` (Linux)

`/report` (Windows)

Description

Generates a deploy preview report that describes the node summary and how the components in the repository affect the node. For example, whether each component applies to the node. The report is generated in HTML, XML, and CSV formats with file names `SUM_Deploy_preview_Report_<date>_<time>.html`, `SUM_Deploy_preview_Report_<date>_<time>.xml`, and `SUM_Deploy_preview_Report_<date>_<time>.csv`.

SUM performs inventory, generates the report, and then exits. It does not deploy any updates.

Generate inventory repository report

Syntax

`--inventory_report` (Linux)

`/inventory_report` (Windows)

Description

Generates a report listing of the components in the specified repository. The report is generated in HTML, XML, and CSV formats with file names `SUM_Inventory_Report_<date>_<time>.html`, `SUM_Inventory_Report_<date>_<time>.xml` and `SUM_Inventory_Report_<date>_<time>.csv`.

SUM performs inventory, generates the report, and then exits. It does not deploy any updates.

Generate installed firmware report

Syntax

`--firmware_report` (Linux)

`/firmware_report` (Windows)

Description

Generates a report listing of the firmware, software installed, and details of the node. The report is generated in HTML, XML, and CSV formats with file names `SUM_Firmware_Report_<date>_<time>.html`, `SUM_Firmware_Report_<date>_<time>.xml`, and `SUM_Firmware_Report_<date>_<time>.csv`.

SUM performs inventory, generates the report, and then exits. It does not deploy any updates.

Generate failed dependencies report

Syntax

`--dependency_report` (Linux)

`/dependency_report` (Windows)

Description

Generates a report listing of the failed dependencies for all nodes. The report is generated in HTML, XML, and CSV formats with file names `SUM_FailedDependency_Report_<date>_<time>.html`, `SUM_FailedDependency_Report_<date>_<time>.xml`, and `SUM_FailedDependency_Report_<date>_<time>.csv`.

SUM performs inventory, generates the report, and then closes. It does not deploy any updates.

Generate installed updates report

Syntax

`--installed_report` (Linux)

`/installed_report` (Windows)

Description

Generates a report that lists all firmware, software, and driver versions installed on all nodes during this session of SUM. The report is generated in HTML, XML, and CSV formats with file names `SUM_Installed_Report_<date>_<time>.html`, `SUM_Installed_Report_<date>_<time>.xml`, and `SUM_Installed_Report_<date>_<time>.csv`.

SUM inventories the node, deploys updates, and then generates the report.

Generate combined report

Syntax

`--combined_report` (Linux)

`/combined_report` (Windows)

Description

This report generates all report types in one file. The report is generated in HTML, XML, and CSV formats with file names `SUM_Combined_Report_<date>_<time>.html`, `SUM_Combined_Report_<date>_<time>.xml`, and `SUM_Combined_Report_<date>_<time>.csv`.

SUM inventories the node, deploys updates, and then generates the report.

Report directory

Syntax

`--reportdir` (Linux)

`/reportdir` (Windows)

Description

Use this parameter with all report commands to specify a directory where SUM saves reports. For example, `smartupdate --report --reportdir /etc/user/reports`.

Input file parameters

Use the following parameters to script the update of multiple or individual nodes, or groups of nodes.

Subtopics

Use an input file

Delete an input file

Use an input file

Syntax

`--inputfile` **"filename"** (Linux)

`/inputfile` **"filename"** (Windows)

Description

When this parameter is used with file name, SUM uses the file contents to generate a list of targets and baselines for performing deployment. For more information, see Input file sections.

Delete an input file

Syntax

`--deleteinputfile` (Linux)

`/deleteinputfile` (Windows)

Description

Instruct SUM to delete the Input file after it has been read.

Return codes

SUM has consolidated return codes from Linux and Windows smart components into an enhanced return-code mapping. These return codes determine the status of the component installation. You can also use return codes in a script to control the execution of the script and determine any required branching.

In Linux, a negative return code is recalculated to a positive return code by subtracting the value from 256.

`SUCCESS_NO_REBOOT`

> **Linux:** 0
>
> **Windows:** 0
>
> **Text:** The installation was successful.

SUCCESS_REBOOT

>   **Linux:** 1
>
>   **Windows:** 1
>
>   **Text:** The installation was successful, but a reboot is required.

SUCCESS_NOT_REQUIRED

>   **Linux:** 3
>
>   **Windows:** 3
>
>   **Text:** The component was current or not required.

FAILURE_GENERAL

>   **Linux:** 255
>
>   **Windows:** -1
>
>   **Text:** A general failure occurred. For details, see the error log.

FAILURE_BAD_PARM

>   **Linux:** 254
>
>   **Windows:** -2
>
>   **Text:** A bad input parameter was encountered.

FAILURE_COMPONENT_FAILED

>   **Linux:** 253
>
>   **Windows:** -3
>
>   **Text:** The installation of the component failed or was blocked by a failed dependency.

FAILURE_COMMAND_FAILED

>   **Linux:** 252
>
>   **Windows:** -4
>
>   **Text:** The CLI command execution failed.

Windows smart-component return codes

0

>   The smart component was not installed. For more information, see the log file.

1

>   The smart component was installed successfully.

2

>   The smart component was installed successfully, but the system must be restarted.

3

>   The installation was not attempted because the required hardware is not present, the software is current or not applicable, or there is nothing to install.

Linux software RPM return codes

0

The Linux RPM installation was successful.

1

The Linux RPM installation failed.

Linux smart-component and firmware RPM return codes

0

The smart component was installed successfully.

1

The smart component was installed successfully, but the system must be restarted.

2

The installation was not attempted because the required hardware is not present, the software is current, or there is nothing to install.

3

The installation was not attempted because the required hardware is not present, the software is current or not applicable, or there is nothing to install.

4

If the component is being installed to a remote node, then the return code indicates that the node cannot be found.

5

A user canceled the installation before anything could be installed.

6

The Installer cannot run because of an unmet dependency or installation tool failure.

7

The actual installation operation (not the installation tool) failed.

VMware ESXi smart-component return codes

0

The installation of the deliverable was successful. No reboot is required.

1

The installation of the deliverable was successful. Reboot is required for the deliverable to be enabled.

2

The installation was not attempted because the version to be installed matches the version already installed.

3

The installation was not attempted because of one of the following:

- The version to be installed is older than the version already installed.

- The supported hardware is not present, not enabled, or in a state that an installation could not be attempted.

- The smart component does not support the environment.

4

    If the component is being installed to a remote node, then the return code indicates that the node cannot be found.

5

    A user canceled the installation before anything could be installed.

6

    The Installer cannot run because of an unmet dependency or installation tool failure.

7

    The actual installation operation (not the installation tool) failed.

CLI with input file parameters

Subtopics

Component-specific configuration in Input files

About Input file encoding

Error reporting

Downloading SUM and components from the SDR

Input file parameters

Component-specific configuration in Input files

Use component short names in Input files to define component configurations. The component filename might change in a release. See the SPP Release Notes to view the filename.

`hpsmh-windows-x64`

    HP System Management Homepage for Windows x64

`hpinsightmgmtwbemprovider-windows-x64`

    HP Insight Management WBEM Providers for Windows Server x64 Editions

`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 Enablement Kit for Windows 64 bit operating systems

hpemulex-smartsan-windows-x64

> HPE Emulex SmartSAN Enablement Kit for Windows 64 bit operating systems

hpemulex-smartsan-linux

> Emulex SmartSAN Enablement Kit for Linux

hpqlogic-smartsan-linux

> QLogic SmartSAN enablement kit for Linux

Subtopics

Configuring components with an Input file

Configuring components with an Input file

About this task

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

About Input file encoding

Use UTF-8 formatting for Input files. UTF-8 allows double-byte characters.

Error reporting

If errors occur in the Input file, SUM exits with a return value of −2 (bad parameter). Find the location and nature of the error in the log file `smartupdate_execution_log_<date>_<time>.raw`.

Downloading SUM and components from the SDR

If you are using a Linux system, you can download SUM as an RPM from the SDR. You can download and install SUM and components from the SDR using common YUM commands. For more information about downloading and installing SUM as an RPM, see Linux best practices: Using Service Pack for ProLiant (SPP) and Software Delivery Repository (SDR) at:

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

For information on using the SDR, see:

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

Input file parameters

When using an Input file, you must include the target nodes and baseline you want to use. If you are running SUM in a directory that includes a baseline, SUM uses the baseline in that directory by default.

The following attributes are related to the session, and you can only use them one time.

> **Session attributes**

`BUNDLESLIST`

> Possible values:
>
> `<bundlefilename>`
>
> Bundle file names
>
> List of bundle xml files to be considered for update if there are multiple bundle files in the provided baseline location.

`CLEANUPALLONEXIT`

> Possible values:
>
> `YES,` **Default =** `NO`
>
> This parameter removes all files, including debug files. This parameter does not delete the user logs.

`CLEANUPONEXIT`

> Possible values:
>
> `YES,` **Default =** `NO`
>
> This parameter removes the following:
>
> - Firmware RPMs copied to the Linux folder `/usr/lib/i386-linux-gnu/*` or `/usr/lib/x86_64-linux-gnu`.
>
> - Extracted Firmware RPM directory and folder contents.
>
> - Extracted components during baseline inventory.
>
> - If SUM is run from a read-only location, this parameter deletes the 'localsum' directory.
>
>   > 📄 **NOTE:**
>   > This parameter does not delete the debug log files.

`COMBINED_REPORT`

> Possible values:
>
> `YES` , `NO`
>
> This report generates all report types in one file.
>
> SUM performs the inventory, deploys updates, generates the report, and then exits.
>
> For information about where SUM saves reports, see Report parameters.

`COMPONENTSLIST`

Possible values:

Component names with file extensions `.exe`, `.rpm`, or `.scexe`.

Limits the list of components to be updated.

DEBUGLOGDIR

Possible values:

`<directorypath>`

Tells SUM where to save debug files.

DELETEINPUTFILE

Possible values:

`YES`, Default = `NO`

Instructs SUM to delete the Input file after it has been read.

DEPENDENCY_REPORT

Possible values:

`YES`, `NO`

Generates a report listing of the failed dependencies for all nodes.

SUM performs the inventory, generates the report, and then exits. It does not deploy updates.

For information about where SUM saves reports, see Report parameters.

DOWNGRADE

Possible values:

`YES`, `NO`

Selects components for installation that have an available version lower than the currently installed version. It is in addition to the upgrades. It can be combined with `rewrite`.

DRYRUN

Possible values:

`YES`, `NO`

Simulates the installation for a test run. Nothing is installed.

FIRMWARE_REPORT

Possible values:

`YES`, `NO`

Generates a report listing of the firmware installed, software installed, and details of the node.

SUM performs the inventory, generates the report, and then exits. It does not deploy updates.

For information about where SUM saves reports, see Report parameters.

FORCEALL

Possible values:

`YES`, `NO`

Rewrites or downgrades the existing installation of the selected software components, firmware components, and bundles.

FORCEBUNDLE

Possible values:

`YES`, Default = `NO`

Rewrites or downgrades the existing installation of components in the selected bundle.

FORCEROM

Possible values:

`YES`, `NO`

Rewrites or downgrades the existing installation of the selected firmware components.

FORCESOFTWARE

Possible values:

`YES`, `NO`

Rewrites or downgrades the existing installation of the selected software components.

HOST

Possible values:

`IP address`, `DNS name`

The IP address or the DNS name of a remote server, remote iLO NIC port.

IGNOREERRORS

This parameter causes the installation to continue and ignore errors.

Possible values:

`All` —Ignores remote nodes with an error and continues deployment to other nodes.

`ServerNotFound` —Bypasses inactive or unavailable remote hosts when deploying firmware or software to multiple remote hosts at the same time.

`BadPassword` —Bypasses remote nodes that report incorrect credentials provided and continue with other nodes.

`FailedDependencies` —Ignores any nodes with failed dependencies and proceeds with nodes that are ready.

`ILOCACENABLED` —Bypasses the nodes on which CAC mode is enabled in iLO and continues with other nodes. iLO Administrator credentials are not required.

`iLOHighSecurityMode` —Bypasses the nodes on which High Security mode is enabled in iLO and continues with other nodes. iLO Administrator credentials are not required.

`CHIFSignFailure` —Bypasses the nodes when Signature verification of a CHIF driver component has failed and continues with other nodes.

`CHIFInstallFailure` —Bypasses the nodes when a CHIF driver installation has failed and continues with other nodes.

IGNORETPM

Possible values:

`YES`, `NO`

If you have enabled TPM, ignore the warning message and continue with component installation. For more information about TPM, see the Smart Update Manager 10.0 User Guide posted at **https://www.hpe.com/support/SUMGen11-UG-en**

IGNOREWARNINGS

Possible values:

`YES`, `NO`

Allows installation to proceed after SUM receives warnings on nodes. Some warnings include:

- Active member of a Serviceguard cluster

- TPM warning

- Pending iLO installation queue warnings

- iLO task queue exceptions

- iLO and iSUT warnings

IMPORT_CONFIGURATION=<path>

This parameter specifies the directory that contains component configuration settings for components in the specified baseline. Imported component configurations override component configurations in the baseline.

INSTALLED_REPORT

Possible values:

`YES`, `NO`

Generates a report that lists all installed firmware, software, and driver versions installed on all nodes.

SUM performs the inventory, deploys updates, generates the report, and then exits.

For information about where SUM saves reports, see Report parameters.

INVENTORY_REPORT

Possible values:

`YES`, `NO`

Generates a report listing of the components in the specified repository.

SUM performs the inventory, generates the report, and then exits. It does not deploy updates.

For information about where SUM saves reports, see Report parameters.

LOGFILENAME = "path"

Possible values:

`Log file name`

Redirects the output from SUM to a directory other than the default location.

For Windows components, the default location is `%SYSTEMDRIVE%\CPQSYSTEM\sum\log\<netAddress>` and the redirected location is `<path>\sum\log\<netAddress>`.

SUM creates the directory `%SYSTEMDRIVE%\CPQSYSTEM\` when it processes components. SUM writes only the files `smartupdate_log.txt`, `smartupdate_detail_log.txt`, and `smartupdate_InstallDetails.txt` to the redirected directory. SUM writes all other log files to the default directory.

For Linux components, the default location is `/var/log/sum/<netAddress>` and the redirected location is `<path>/log/sum/<netAddress>`.

NOMGMT

Possible values:

`YES`

Specifies listing components using SNMP and WBEM Providers as optional updates on the Components Selection screen.

In silent mode, SUM does not update AMS, SNMP, or WBEM Providers.

ONFAILEDDEPENDENCY

Tells SUM how to proceed when a component has a failed dependency.

Possible values:

`OmitHost` (default)—Puts the host in a failure state, and SUM does not attempt installation.

`OmitComponent`—Clears the affected components and proceeds with any updates that do not have dependency failures.

`Force`—Attempts all updates, even if they have dependency failures.

OPTIONS

Possible values:

One or more CLI switches

Specify the SUM CLI options inside the Input file, which overrides the configuration settings. Separate parameters with a comma.

This parameter replaces the `LSPOPTIONS` parameter that was previously supported with LDU.

REBOOTALLOWED

Possible values:

`YES`,`NO`

Enables you to reboot, if required.

REBOOTALWAYS

Possible values:

`YES`,`NO`

It always reboots the node.

REBOOTDELAY

Possible values:

Time in seconds

Adds wait time before rebooting.

REBOOTMESSAGE

Possible values:

Any string (not exceeding 256 characters)

Creates a message to be displayed prior to rebooting.

REPORT

Possible values:

`YES`,`NO`

Generates a report listing of the node summary and describes how the components in the repository affect the node—for example, whether each component applies to the node.

SUM performs the inventory, generates the report, and then exits. It does not deploy updates.

For information about where SUM saves reports, see Report parameters.

REPORTDIR

Use this parameter with all report commands to specify a directory where you want SUM to save reports.

REWRITE

Possible values:

`YES`, Default = `NO`

Selects components for installation that have an available version that is the same as the installed version. This is in addition to upgrades. Can be combined with `downgrade`.

ROMONLY

Possible values:

`YES`,`NO`

Limits components considered for installation to only firmware components. Cannot be used with `softwareonly`. If `softwareonly` and `romonly` are not used, all components are considered for possible installation.

SCHEDULEDEPLOYFROM

Possible value:

For example: `SCHEDULEDEPLOYFROM = MMDDYYYY HH:MM`

Adds schedule start time to current deployment.

SCHEDULEDEPLOYTO

Possible value:

For example: `SCHEDULEDEPLOYTO = MMDDYYYY HH:MM`

Adds schedule end time to current deployment.

SILENT

Possible value:

`YES`

Causes the installation to run silently with no GUI output. Writes all data to the log file. Any generated prompts use the default option and continue the installation without user input.

> 📝 **NOTE:**
>
> If you enter the value `NO`, the input file will not work correctly. SUM will try to launch the GUI mode.

SKIPTARGET

Possible values:

`YES` (default), `NO`

The default setting skips a remote node if SUM finds a remote session running on the node.

Defines the behavior when a remote node has an existing SUM session in progress.

Use this parameter to skip the host if a SUM session already exists. A value of `NO` overrides the session in progress and reinitializes the installation framework on the remote host.

SOFTWAREONLY

Possible values:

`YES`, `NO`

Limits components considered for installation to only software components. Cannot be used with `romonly`. If `softwareonly` or `romonly` are not used, all components are considered for possible installation.

TPMBYPASS

Use this parameter to bypass TPM warnings.

UPDATEEXISTINGRECOVERYSET

Possible values:

`YES`, `NO`

Updates the existing recovery set in iLO Repository with new firmware versions of iLO, BIOS, CPLD, IE, and ME (IE and ME for Intel Processor) from current deployment. Gen10 and above servers only.

USE_SSHKEY

Possible values:

`YES`, `NO`

Allows you to use an SSH PEM-format keyfile for Linux nodes. Must be used as a session attribute, not as a node attribute.

USEAMS

Possible values:

`YES`

Specifies SUM to install the AMS component.

This option applies to Gen8 and later servers. If you set this parameter for a ProLiant G7 and earlier server, SUM ignores the parameter.

These components are only installed by default on Gen8 and later servers. This parameter does not apply to Integrity servers.

USECURRENTCREDENTIAL

Possible values:

`YES`, `NO`

Allows you to use the credentials of the local host to access the nodes instead of providing the user name and password explicitly for

each node.

Assumes that the current credentials are valid for the nodes that are being accessed. (Applies to Windows only.)

USESNMP

Possible values:

`YES`

Specifies SUM to install the SNMP component.

These components are optional by default and are not installed unless you use this parameter. This parameter does not apply to Integrity servers.

Gen9 only: SUM installs SMH templates when it installs SNMP agents.

> **NOTE:**
> Gen10 and above servers do not support SNMP agents.

USEWMI

Possible values:

`YES` (Windows only)

Specifies SUM to install the WBEM component.

These components are optional by default and are not installed unless you use this parameter.

> **NOTE:**
> Gen10 and above servers do not support WBEM agents.

VERBOSE

Possible values:

`YES`, `NO`

This parameter determines the amount of details SUM provides during deployment.

The following attributes are related to nodes and baselines. Use these attributes for each node or baseline.

ILO_PASSWORD

Possible values:

`<ilo password>`

The iLO Administrator password associated with `ILO_USERNAME` parameter.

ILO_USERNAME

Possible values:

`<ilo username>`

Provide iLO Administrator credentials. Use this parameter to connect to iLO from the host OS when the iLO of the server has been configured in High Security mode.

INSTALLSETDESCRIPTION

Possible value:

`<string>`

Set a description for the install set.

Servers from Gen10 and onwards only.

INSTALLSETNAME

Possible value:

`<string>`

Set the name for the install set that must be stored on the iLO.

Servers from Gen10 and onwards only.

MANUALLYMANAGEILOREPOSITORY

Possible values:

`YES` , `NO`

`YES` does not allow SUM to automatically remove install sets on the iLO Repository to create space for new install sets.

Servers from Gen10 and onwards only.

Default is automatic, which means that if iLO NAND does not have space to accommodate new components that SUM uploads, SUM deletes the install set with minimum size (based on total component size) and creates space for new components and install set.

PASSPHRASE

Possible values:

`<passphrase=12345>`

Passphrase for the SSHkey file. Use with the `USE_SSHKEY` and `PRIVATEKEYFILE` parameters.

PRIVATEKEYFILE

Possible values:

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

Provide the full path to the private key file.

Use with the `USE_SSHKEY` parameter.

PWD

Possible values:

`<password>`

Uses the password for the user ID specified in the `UID` .

Specifies the password for logging in to the nodes.

SAVEINSTALLSET

Possible values: `YES` , `NO`

Saves the install set on the iLO Repository.

Servers from Gen10 and onwards only.

SKIP_PREREQS

Possible values:

`YES` , `NO` (Windows and Linux)

If you enter `YES` , then prerequisite self-inventory components, such as the CHIF driver, are not installed.

SKIPMISSINGCOMPSIG

Possible values:

`YES` , `NO`

`YES` instructs SUM to skip components that are missing a component signature file.

Servers from Gen10 and onwards only.

SOURCEPATH

Possible values:

`Directory path` , `UNC location`

Provides a single local baseline path or a UNC file share. This action creates an inventory from the given path instead of the local or default baseline.

SUM CLI with Input File mode does not support http baselines.

STAGEONLY

Possible values:

`YES` , Default = `NO`

Uploads all applicable components to an iLO Repository by creating an install set. Does not initiate the install set created to perform deployment. The install set created can be used for updating the system later.

SUPASSWORD

Possible values:

`<superuserpassword>`

Use this argument to provide a superuser password.

After you log in to a Linux node, using the regular credentials, you can elevate the session using these credentials. SUM requires an Administrator level access to perform any updates.

SUUSERNAME

Possible values:

`<superusername>`

Use this argument to provide a superuser username.

> 📋 **NOTE:**
> `<superuserpassword>` and `<superusername>` are used to provide superuser username and password. After you log in to a Linux node, using the regular credentials, you can elevate the session using these credentials. SUM requires an Administrator level access to perform any updates.

TARGETTYPE

Specifies the type of node (`--targettype Linux` (Linux) `/targettype` (Windows), and can shorten the inventory process.

Valid node types are:

`Windows`

`Linux`

`iLO`

If you use the command with a group, SUM assumes that all nodes in the group are the same node type.

UID

Possible values:

`<username>`

Specifies the user ID for logging in to the nodes.

SUM supports Windows domains, for example: `domain1/userid1`

UNC_PASSWORD

Possible values:

`<password>`

Provides the password to access the UNC location.

UNC_USERNAME

Possible values:

`<username>`

Provides the user name credentials for the UNC location provided in the `SOURCEPATH` .

USEILOSAVED

Possible value:

`<string>`

Uses an install set with the provided name saved on the iLO Repository as a baseline for the nodes.

Servers from Gen10 and onwards only.

`USESUDO`

Possible values:

`YES`, `NO`

Allows use of the `sudo` command.

Specifies that username and password for a remote node are sudo user credentials. When you specify `USESUDO` in the Input file along with username and password, then the username and password are considered sudo credentials.

Website and support

Websites

Smart Update Manager	**www.hpe.com/servers/sum**
Smart Update Manager Information Library	**www.hpe.com/info/sum-docs**
Smart Update Tools	**www.hpe.com/servers/sut**
Smart Update Tools Information Library	**www.hpe.com/info/isut-docs**
Service Pack for ProLiant	**www.hpe.com/servers/spp**
Service Pack for ProLiant documentation	**www.hpe.com/info/spp/documentation**
Service Pack for ProLiant downloads	**www.hpe.com/servers/spp/download**
Service Pack for ProLiant custom downloads	**www.hpe.com/servers/spp/custom**
HPE SDR site	**downloads.linux.hpe.com**

For additional websites, see <u>Support and other resources</u>.

Subtopics

<u>Support and other resources</u>

Support and other resources

Subtopics

<u>Accessing Hewlett Packard Enterprise Support</u>

<u>Accessing updates</u>

<u>Remote support</u>

Customer self repair

Warranty information

Regulatory information

Documentation feedback

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:

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

- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:

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

Information to collect

- Technical support registration number (if applicable)

- Product name, model or version, and serial number

- Operating system name and version

- Firmware version

- Error messages

- Product-specific reports and logs

- Add-on products or components

- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.

- To download product updates:

  Hewlett Packard Enterprise Support Center

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

  My HPE Software Center

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

- To subscribe to eNewsletters and alerts:

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

- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the   Hewlett Packard Enterprise Support Center More Information on Access to Support Materials page:

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

> (i) **IMPORTANT:**
>
>   Access to some updates might require product entitlement when accessed through the   Hewlett Packard Enterprise Support Center. You must have an HPE Onepass set up with relevant entitlements.

Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which initiates a fast and accurate resolution based on the service level of your product. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

If your product includes additional remote support details, use search to locate that information.

HPE Get Connected

   https://www.hpe.com/services/getconnected

HPE Tech Care Service

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

HPE Complete Care

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

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR.

For more information about CSR, contact your local service provider.

Warranty information

To view the warranty information for your product, see the   **warranty check tool** .

Regulatory information

To view the regulatory information for your product, view the   Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products, available at the Hewlett Packard Enterprise Support Center:

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

Additional regulatory information

Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found at:

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

For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, see:

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

For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, see:

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

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, use the Feedback button and icons (at the bottom of an opened document) on the Hewlett Packard Enterprise Support Center portal (https://www.hpe.com/support/hpesc) to send any errors, suggestions, or comments. This process captures all document information.