Hewlett Packard Enterprise

Smart Update Manager 8.5.0 CLI and Interactive CLI Guide

Abstract

This document describes how to use SUM CLI, interactive 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, 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.

Part Number: 814105-404
Published: December 2019
Edition: 1

© Copyright 2014, 2019 Hewlett Packard Enterprise Development LP

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX® is a registered trademark of The Open Group.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Contents

Introduction to SUM CLI..... 5

About SUM and interactive CLI command-line options..... 5
SUM CLI modes..... 5
Scheduling updates..... 6
Linux root credentials for remote nodes..... 6
Prerequisites for using Linux sudo credentials..... 6
Using an SSH keyfile with root credentials..... 7

SUM CLI and Input file commands overview..... 8

Command-line syntax..... 8
Command-line description..... 8
Input file overview..... 9
Input file sections..... 9

SUM CLI command task examples..... 11

About CLI command examples..... 11
Issuing an Input file command..... 11
Updating the local host..... 11
Updating a remote node..... 12
Updating two nodes with one baseline..... 13
Updating only the software on a node..... 13
Updating only the firmware on a node..... 14
Deploying the latest components to a node..... 14
Forcing deployments to an earlier version of a component on a node..... 15
Deploying two software components..... 15
Using sudo to update a remote Linux node..... 16
Updating Superdome X partitions..... 16
Using super user credentials to update nodes..... 17
Using an SSH Key to update Linux nodes..... 18

CLI parameters..... 20

Management agent installation parameters..... 20
SUM CLI parameters..... 21
Help..... 21
Installation parameters..... 21
Override error parameters..... 26
Reboot parameters..... 28
Test run parameter..... 29
Node parameters..... 30
Log file parameters..... 37
Report parameters..... 38
Input file parameters..... 41
Superdome X server commands..... 41
Return codes..... 42
Windows smart-component return codes..... 43

Linux smart-component and firmware RPM return codes....43
Linux software RPM return codes....44
VMware ESXi smart-component return codes....44

CLI with Input file parameters....46
Component-specific configuration in Input files....46
Configuring components with an Input file....47
About Input file encoding....47
Error reporting....47
Input file parameters....47
Downloading SUM and components from the SDR....59

Running commands in interactive CLI mode....60
Deploying nodes....60
activate....60
add....61
configure....65
delete....66
deploy....67
findavailablenodes....69
generatereports....69
getattributes....71
getbaselines....72
getcomponentlogs....72
getcurrentlyinstalledversions....73
getenginestatus....74
getlogs....76
Getneededupdates....77
getnodes....78
inventory....80
login....81
setattributes....83
shutdownengine....91
Stage components....92
stageonly....92
validate baselines....92

Websites and support....94
Websites....94
Support and other resources....94
Accessing Hewlett Packard Enterprise Support....94
Accessing updates....95
Remote support....95
Warranty information....96
Regulatory information....96
Documentation feedback....96

Introduction to SUM CLI

About SUM and interactive CLI command-line options

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 or interactive CLI mode.

You can script most of the commands that are available in the SUM GUI in interactive CLI mode. Functionality available from interactive CLI mode includes:

- Adding nodes and baselines
- Deploying nodes
- Collecting log files
- Generating reports

NOTE: Do not use interactive CLI for performing updates. The interactive CLI access will be unsupported and deprecated in the future release of SUM.

SUM CLI modes

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

NOTE: You can use only 10 VMware nodes.

Strengths:

- Create a file that includes all information for baselines and nodes.
- Save files for future use.
- Issue CLI command that calls the Input file.

Interactive CLI

Number of nodes: 1–50

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.

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

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 ----`.

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

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.

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 display 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>
```

NOTE: If you update Onboard Administrator (Enclosure firmware) after the `/target` and OA IP address, you must provide the username and credentials of the OA administrator. The user ID must be an administrator equivalent ID and not an operator or user equivalent level ID.

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 **Report parameters**.

`--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.

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

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`

Updating Superdome X partitions

In the following examples, SUM deploys updates to the npar3 and blade 1/1 partitions of a Superdome X enclosure, and then reboots the partition to activate it.

Updating partitions in CLI mode

- **Linux**: `./smartupdate -silent --target <ip_address> --user <userid> --password <password> --ignore_warnings --update_type PARTITIONS --devicelist npar3,blade1/1 --rebootlist npar3 --use_location <baseline_directory_path>`
- **Windows**: `smartupdate /silent /target 16.17.18.19 /user Administrator / password 12345678 /ignore_warnings /update_type PARTITIONS /devicelist npar3,blade1/1 /rebootlist npar3 /use_location C:\Users\Administrator \Desktop\SDX\3.7.60`

Updating partitions in Input file mode

```
SILENT= YES
SOURCEPATH = <baseline_directory_path>
IGNOREWARNINGS = YES
[TARGET]
HOST = <ip_address>
UID = <userid>
PWD = <password>
UPDATETYPE = PARTITIONS
DEVICELIST=npar3,blade1/1
REBOOTLIST=npar3
[END]
```

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

Management agent installation parameters

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

- Linux: `$ ./smartupdate --firmware_report --s`
- Windows: `C:\ smartupdate /firmware_report /s`

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 servers do not support SNMP agents.

`/use_wmi` (Windows only)

Gen8 and later servers installed agent: WBEM

NOTE: Gen10 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.

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.

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 or interactive 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 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 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.

Bypass TPM

Syntax

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

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

Description

If you have enabled TPM on target iLO 5, Windows, Linux, or VMware ESXi, ignore the warning message and continue with component installation. For more information about TPM, see the *Smart Update Manager User Guide*.

NOTE: Provide this parameter at the end of the CLI command.

Ignore warnings

Syntax

`--ignore_warnings` (Linux)

`/ignore_warnings` (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
- Superdome X-specific warnings
- Low disk space
- A system managed by HPE OneView
- Pending iLO installation queue warning
- Exception task queue warning
- iLO 5 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.

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.

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.

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`, `use_sshkey`, `vcenter_host`, `vcenter_username`, and `vcenter_password`.

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, Virtual Connect Ethernet or Fibre Channel Module for c-Class BladeSystem, or BladeSystem OA.

When two OAs are in an enclosure, use the active OA as the argument. When specifying the IP address, use either the IPv4 or IPv6 format.

If you are adding a VC node, use the IP address of the primary Enet module.

Define target type

Syntax

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

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

Description

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

Possible values

`Windows`

`Linux`

`iLO`

`fc_switch`

`oa`

`virtual_connect`—Use the IP address of the primary VC Enet module.

`sas_switch`

`VMware`

`sdx`

`sdflex`

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.

OA username

Syntax

`--oa_username` (Linux)

`/oa_username` (Windows)

Description

Provides the associated OA user name credential for the VC target specified using the `target` command. You can specify only one set of OA credentials using command-line parameters. To add multiple VC nodes using the `target` parameter, make sure that the OA credentials for each specified VC are the same. You do not need to provide an OA network address associated with the VC. SUM obtains it from a specified VC node.

To update VCs with OAs having different credentials, use the corresponding `oa_username` and `oa_password`.

OA user password

Syntax

`--oa_password` (Linux)

`/oa_password` (Windows)

Description

Provides the password credential for OA associated with VC specified using the target command. Only one set of OA credentials can be specified with command-line parameters. Add multiple VC nodes to command-line parameters with the "node" parameter only if the credentials of OAs associated with specified VCs are the same. You do not need to provide an OA network address associated with VC. SUM queries it from a specified VC node.

To update VCs with OAs having different credentials, use the corresponding `oa_username` and `oa_password`.

Enable VC in FIPS mode

Syntax

`--encryption_key` (Linux)

`/encryption_key` (Windows)

Description

This parameter is used to provide the encryption key for FIPS-enabled VC.

Possible value

String of a minimum eight characters.

VC-Enet module activation order

Syntax

`--ethactorder` (Linux)

`/ethactorder` (Windows)

Description

Use this command to set the VC-Enet module activation order.

Possible values

`ODDEVEN` (Default)

`PARALLEL`

`SERIAL`

`MANUAL`

VC-FC activation order

Syntax

`--fcactorder` (Linux)

`/fcactorder` (Windows)

Description

Use this command to set the VC-FC activation order.

Possible values

`ODDEVEN`

`PARALLEL`

`SERIAL` (Default)

`MANUAL`

VC-Enet activation and reboot options

Syntax

`--ethactdelay` (Linux)

`/ethactdelay` (Windows)

Description

Use this command to specify the time, in minutes, between activating or rebooting VC-Enet modules.

Possible values

Maximum 60 minutes; the default is 0 minutes.

VC-FC activation and reboot options

Syntax

`--fcactdelay` (Linux)

`/fcactdelay` (Windows)

Description

Use this command to specify the time to wait between activating or rebooting VC-FC modules.

Possible values

Maximum 60 minutes; the default is 0 minutes.

VCenter Host

Syntax

`--vcenter_host` (Linux)

`/vcenter_host` (Windows)

Description

Enter the IP address for the VMware vCenter.

Use this command with VMware ESXi nodes to generate a ticket from VMware vCenter nodes that are in lockdown mode. This command applies to all targets in the command line. For non-VMware nodes, the command is ignored.

VCenter user name

Syntax

`--vcenter_username` (Linux)

`/vcenter_username` (Windows)

Description

Enter the username to log in to the VMware vCenter. See the `vcenter_host` for more information.

VCenter password

Syntax

`--vcenter_password` (Linux)

`/vcenter_password` (Windows)

Description

Enter the password to log in to the VMware vCenter. See the `vcenter_host` for more information.

iLO 5 parameters

Save install set

Syntax

`--save_install_set` (Linux)

`/save_install_set` (Windows)

Description

This parameter saves the install set in the iLO 5 repository.

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 5 repository

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 5 repository.

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)

Log file parameters

The following attributes determine how SUM creates log files.

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.

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.

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.

Superdome X server commands

The following attributes apply to Superdome X servers.

Firmware update type

Syntax

`--update_type` **<type>** (Linux)

`/update_type` **<type>** (Windows)

Description

Determines which Superdome X firmware is updated.

Options

`ALL` (Default)

`COMPLEX`

`PARTITIONS`

Example

`update_type PARTITIONS`

Update specific devices

Syntax

`--device_list` **<device[,device]...>** (Linux)

`/device_list` **<device[,device]...>** (Windows)

Description

Use this argument when the `update_type` argument is `PARTITIONS`. This argument specifies the subset of partitions or unassigned blades to update when you do not want to update all partitions or unassigned blades.

Options

`npar(number)`

`blade(enc)/(bay)`

Do not put any spaces between the parameters.

Example

```
device_list npar1,npar2,blade1/1,blade1/2
```

Partition reboot options

Syntax

`--reboot_list` **<npar[,npar]...>** (Linux)

`/reboot_list` **<npar[,npar]...>** (Windows)

Description

Use this argument when the `update_type` is `ALL` or `PARTITIONS`. This argument specifies specific partitions to reboot after the partition firmware has been updated.

Options

`npar` values are `npar(number)`. Do not put any spaces between parameters.

Example

```
reboot_list npar1,npar2
```

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 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 note, such as an OA or other network-based deployment, this 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.

Linux software RPM return codes

0

The Linux RPM installation was successful.

1

The Linux RPM installation 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, such as an OA or other network-based deployment, this 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

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.

To get a list of available configuration parameters for components in a baseline, use the following interactive CLI command:

- Linux: `smartupdate configure --show`
- Windows: `smartupdate configure /show`

NOTE: The interactive CLI does not support component short names, use the full component name when you show configuration parameters.

For more information on using the interactive CLI, see the *Smart Update Manager User Guide*.

`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

Configuring components with an Input file

```
[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`.

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 once.

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 value:

`<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**.

DEVICELIST

Possible values:

For example: `DEVICELIST=npar1,blade1/1`

Do not put spaces between the parameters.

For Superdome X servers only, use this argument when the `UPDATETYPE` is set to `PARTITIONS`. This argument specifies the subset of partitions or blades to update when you do not want to update all partitions. Valid devices are: `npar(number)` and `blade(enc)/(bay)`.

DOWNGRADE

Possible values:

`YES`, `NO`

Selects components for installation that have an available version lower than the currently installed version. This is in addition to upgrades. 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, Virtual Connect Ethernet, or Fibre Channel Module for c-Class BladeSystem or BladeSystem OA.

With two OAs in an enclosure, this parameter is the active OA. When specifying the IP address, use either the IPv4 or IPv6 format.

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 User Guide*.

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
- Superdome X-specific warnings
- Pending iLO installation queue warnings
- iLO task queue exceptions
- iLO 5 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 components 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 or the BladeSystem c-Class OA flash utility 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.

OAPWD

User can define OAPWD variable multiple times before each HOST variable.

Provides the password credentials for the OA associated with VC. Define a value for these variables before the HOST variable in the [TARGETS] section. This parameter applies only to VC target.

You can also use OAPASSWORD.

OAUID

User can define the OAUID variable multiple times before each HOST variable.

Provides the user name credentials for the OA associated with VC. Define a value for these variables before the HOST variable in the [TARGETS] section. This parameter applies only to VC firmware.

You can also use OAUSER or OAUSERNAME.

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.

REBOOTLIST

Possible values:

For example: REBOOTLIST=npar1,npar2

Do not put spaces between the parameters.

For Superdome X servers only, use this argument with the UPDATETYPE= ALL or UPDATETYPE= PARTITIONS argument. Specifies the partitions to reboot after updating the partition firmware. Valid npar values are npar(number).

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 or interactive CLI 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 servers only.

UPDATETYPE

Possible values:

`ALL`, `COMPLEX`, `PARTITIONS`

This argument determines which Superdome X firmware is updated.

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 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 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.

ETHACTDELAY

Use this command to specify the time, in minutes, between activating or rebooting VC-Enet modules.

Possible value:

0 (default) to 60 minutes.

ETHACTORDER

Use this command to determine VC-Enet module activation order.

Possible value:

ODDEVEN (default)

PARALLEL

SERIAL

MANUAL

FCACTDELAY

Use this command to specify the time to wait between activating or rebooting VC-FC modules.

Possible value:

0 (default) to 60 minutes.

FCACTORDER

Use this command to specify VC-FC activation order.

Possible value:

ODDEVEN (default)

PARALLEL

SERIAL

MANUAL

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.

Gen10 servers only.

INSTALLSETNAME

Possible value:

<string>

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

Gen10 servers 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.

Gen10 servers 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.

Gen10 servers only.

SKIP_PREREQS

Possible values:

`YES`, `NO` (Windows only)

If you provide the parameter `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.

Gen10 servers 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

fc_switch

OA

sdx

iLO

virtual_connect—Use the IP address of the primary VC Enet module.

sas_switch

VMware

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.

Gen10 servers 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.

VCENTER_HOST

Enter the IP address for the VMware vCenter.

Use this command with VMware ESXi nodes to generate a ticket from VMware vCenter server that is in lockdown mode.

VCENTER_PASSWORD

Enter the password to log in to the VMware vCenter.

See VCENTER_HOST for more information.

VCENTER_USERNAME

Enter the username to log in to the VMware vCenter.

See VCENTER_HOST for more information.

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

Running commands in interactive CLI mode

Deploying nodes

The following steps link to the basic instructions to deploy a node.

Procedure

1. **Add** a node and a baseline.
2. Perform **inventory**.
3. **Deploy** the updates.
4. **Generate** reports.

activate

Description

Activates any updates previously staged or installed, for example, rebooting or other processes based on specific devices.

Syntax

```
smartupdate activate --nodes <nodelist> --cancelpending
```

Parameters

`--nodes <nodelist>`

A list of the nodes (full hostname or IP address) on which to perform the activate nodes operation. If you do not provide any, the command applies to all nodes in the session.

Restrictions

The `activate` command only applies to server nodes.

Windows usage

```
C:\> smartupdate activate /nodes 10.0.0.1 10.0.0.2
```

Linux usage

```
$ ./smartupdate activate --nodes 10.0.0.1 10.0.0.2
```

`SUCCESS_NO_REBOOT`

Windows: 0

Linux: 0

Text: `The command was successful.`

`SUCCESS_REBOOT`

Windows: 1

Linux: 1

Text: `The command was successful but a reboot is required.`

`SUCCESS_NOT_REQUIRED`

Windows: 3

Linux: 3

Text: `The command completed successfully, but was not required or everything was already current.`

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: `A general failure occurred. For details, see the logs.`

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: `A bad input parameter was encountered. For details, see the logs.`

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: `The installation of the component failed.`

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: `The command failed. For details, see the logs.`

add

Description

Adds baselines or nodes to the current working set for the current session.

Syntax

```
smartupdate add --nodes
smartupdate add --baselines
```

Parameters

Parameters given immediately after the `/nodes` command, but used before a node, are considered global parameters. These parameters apply to all the nodes specified in the command. For example:

```
smartupdate add /nodes user=user1 password=password1 type=linux 192.168.1.2
192.168.1.10
```

The parameters user, password, and type apply to both nodes.

Parameters used after a specified node apply to that node only.

```
smartupdate add /nodes 192.168.1.15 user=user2 password=password2 type=linux
192.168.1.17 user=user3 password=password3 type=windows
```

The parameters `user2`, `password2`, and `linux` apply to node `192.168.1.15`. The parameters `user3`, `password3`, and `windows` apply to `192.168.1.17`.

--nodes <nodeslist>

Object to perform the add operation. The items in the list should be delimited by spaces. You can specify the user and password globally for all nodes.

```
domain_name: localhost
ip: 127.0.0.1
ipv6: ::1
```

Add either nodes or baselines. Do not add both in the same command.

If you are adding a VC node, use the IP address of the primary Enet module. All VC modules installed, including FC modules, are updated through the primary Enet module.

type = <node type>

This parameter defines the node type. Use this as both a local and global parameter. SUM supports node types:

- `windows`
- `linux`
- `ilo`
- `virtual_connect`

  Use the IP address of a primary Enet module. VC nodes require `oa_username` and `oa_password`.If you add a VC node, include the following attributes: `oa_username` and `oa_password`.
- `oa`
- `vmware`
- `sas_switch`
- `fc_switch`
- `sdx`
- `unknown`
- `sdflex`

skip_prereqs = <true|false>

By default SUM installs the CHIF driver on Windows nodes. Use the `true` option to skip installing the CHIF driver.

user=<user_name>

Example: `user=john_doe@example.com`

This provides the username credentials for the node. If you do not enter this on the command line, SUM prompts for the username.

Use this as both local and global parameters.

If the node is a Linux node and part of a Windows domain, use the following format for the username.

```
<domain>\<username>
```

password=<password>

Example: `password=userpassword`

The user password credentials for the node. If you do not enter this on the command line, SUM prompts for the password.

Use this as both local and global parameters.

SUM supports Windows domains accounts for user credentials.

ilo_username=<ilo username>

Use this parameter to connect to iLO from the host OS when the iLO of server has been configured in High Security mode. You must provide iLO Administrator credentials. This parameter is valid only for Windows, Linux, and Unknown node types. Make sure that the iLO Administrator account has **Configure iLO Settings** permissions in the iLO.

ilo_password=<ilo password>

The password for the user associated with the `ilo_username` parameter.

su_user=<super_userID>

Example: `user=su_john_doe@example.com`

The username for the super user.

su_password=<super_user_password>

Example: `password=su_userpassword`

The password for the super user.

use_sudo

Valid options: `TRUE`, `FALSE`

Use credentials as sudo credentials.

Potential options:

- `true`
- `false`

vcenter_host

The address of the VMware vCenter when VMware is in lock-down mode. Use these parameters along with `vcenter_username` and `vcenter_password` to establish a connection with a VMware host that is in lock-down mode.

vcenter_username

The user account to establish a connection with a VMware host.

vcenter_password

The account password to establish a connection with a VMware host.

use_sshkey

Potential options:

- `true`
- `false`

Linux nodes only. Use `privatekeyfile` and `passphrase` (optional) with this command.

privatekeyfile=<PEM_file_absolute_path>

Enter the absolute address for the private key. The private key must be in PEM format.

Use this as both local and global parameters.

passphrase=<passphrase>

(Optional) Enter the passphrase for the private key.

Use this as both local and global parameters.

--baselines <directory_path>

Use this parameter to add local baselines by specifying the directory. If you are adding an HTTP baseline, use the `save_location` parameter.

save_location=<working_location>

Directory where you want to save HTTP baselines you download.

type=<baseline_type>

This parameter defines the baseline type. SUM supports baseline types:

`http`

`unc` (Windows only)

NOTE: Make sure that the UNC directory is not open in the localhost system.

unc_username=<unc_username>

Enter the username for the UNC directory path. Valid for Windows only.

unc_password=<unc_password>

Enter the password for the UNC directory path. Valid for Windows only.

bundle=<bundle_name>

Enter the bundle file name you want to add. Use the command `getbaselines /list` to display the list of bundle file names. For example, `bundle=spp.2014.02.0.B`.

filter_os=<os_filter>

(HTTP baselines only) This attribute allows you to filter the components SUM downloads based on operating system. The valid options are:

`windows`

`linux`

`vmware`

Restrictions

- Do not add a baseline and node in the same command. Use two separate commands.
- SUM interactive CLI mode does not automatically add nodes or baselines. The interactive CLI mode does not automatically add associated nodes. Use GUI mode to add associated devices.

Usage

```
smartupdate add [--nodes n1 [type=<node_type>] [user=<username>]
[password=<userpassword>] [su_user=<su_user>] [su_password=<suuser_password>]
[use_sudo=true/false]
--baselines b1 [save_location=<directory>] [type=<baseline_type]
```

Windows usage

Windows sample command line

```
C:\>smartupdate add /nodes 10.0.1.15 type=windows
```

```
C:\>smartupdate add /baselines type=http save_location=c:\temp\spp
bundle=http://10.1.2.3/spp/hp/swpackages/bp002524.xml filter_os=linux

C:\>smartupdate add /nodes type=vc 10.0.1.16 user=Administrator
password=12iso*help oa_username=Administrator oa_password=12iso*help

C:\>smartupdate add /nodes 10.0.1.4 type=vmware vcenter_host=10.0.1.20
vcenter_username=Administrator vcenter_password=password

C:\>smartupdate add /nodes 10.1.2.3 type=virtual_connect
oa_username=Administrator oa_password=mypassword1
```

Linux usage

```
$ ./smartupdate add --nodes 10.0.1.16 type=linux

$ ./smartupdate add --baselines "/SPP/" "/tmp/myrepos/"

$ ./smartupdate add --baselines type=http save_location=c:\temp\spp
bundle=http://10.1.2.3/spp/hpe/swpackages/bp002524.xml filter_os=linux

$ ./smartupdate add --nodes type=vc 10.0.1.18 user=Administrator
password=12iso*help oa_username=Administrator oa_password=12iso*help

$ ./smartupdate add --nodes 10.0.1.4 type=vmware vcenter_host=10.0.1.20
vcenter_username=Administrator vcenter_password=password

$ ./smartupdate add --nodes 10.1.2.3 type=virtual_connect
oa_username=Administrator oa_password=mypassword1
```

NOTE: If the node type is Linux, you can use `su_user` and `su_password`, or use `use_sudo`.

For example:

```
smartupdate add /nodes 192.168.1.1 type=Linux user=user password=password
su_user=user1 su_password=password1

smartupdate add /nodes 192.168.1.1 type=Linux user=user password=password
use_sudo=true
```

configure

Description

Enables you to export, import, and display component configuration settings in a baseline.

Syntax

```
smartupdate configure --export baseline=<baselinename> dir=<directoryname>
[component=<componentname> includeCredentials=true|false>]

smartupdate configure --import baseline=<baselinename> dir=<directoryname>
[component=<componentname>]

smartupdate configure --show baseline=<baselinename>
[component=<componentname>
```

Parameters/Options/Flags/Strings/...

`/export`

This parameter exports component configuration to a specified directory.

`/import`

This parameter imports component configuration from a specified directory.

/show

This parameter displays the component configuration options for the components in a baseline.

baseline=<baselinename>

Use this parameter to specify the baseline that contains the component you want to import, export, or show.

dir=<directoryname>

Use this parameter to specify the directory where you want to export or from where you want to import component configuration settings.

component=<componentname>

This is an optional parameter. This specifies the name of the component that you want to import, export, or show. If you do not specify a component, SUM exports, imports, or shows all components in a baseline.

includeCredentials=<true|false>

This is an optional parameter. If you select `true`, SUM includes the user credentials when it exports configuration settings. The default for this parameter is `true`.

Windows usage

```
C:\>smartupdate configure /import baseline=C:\spp.2012.01 dir=C:\Components
```

In this example, all the component configurations present under `C:\Components` directory for `C:\spp.2012.01` baseline will be loaded into SUM.

```
C:\>smartupdate configure /import baseline=C:\spp.2012.01 dir=C:\Components
\cp022423_conf component=cp022423.exe
```

In this example, SUM only imports the configuration for the component `cp022423.exe` of the baseline `C:\spp.2012.01` in the directory `C:\Components`.

```
C:\>smartupdate configure /show baseline=C:\spp.2012.01
```

In this example, SUM displays all the component configurations from the baseline `C:\spp.2012.01`.

Linux usage

```
$ ./smartupdate configure --export baseline=/root/SPP.2012.01 dir=/root/
Components
```

In this example, SUM exports all the component configurations from the baseline `/root/SPP.2012.01` to the directory `/root/Components`.

```
$ ./smartupdate configure --export baseline=/root/SPP.2012.01 dir=/root/
Components component=hp-snmp-agents-10.10-2657.7.rhel6.i686.rpm
includeCredentials=false
```

In this example, SUM only exports the configuration for the component `hp-snmp-agents-10.10-2657.7.rhel6.i686.rpm` from the baseline `/root/SPP.2012.01` present in the directory `/root/Components`. SUM does not export credentials in this example.

```
$ ./smartupdate configure --show baseline=/root/SPP.2012.01 component=hp-
snmp-agents-10.10-2657.7.rhel6.i686.rpm
```

In this example, SUM displays only the configuration for the component `hp-snmp-agents-10.10-2657.7.rhel6.i686.rpm` from the baseline `/root/SPP.2012.01`.

delete

Description

Enables you to delete both selected and nonselected baselines, nodes, or both from the current session. You can delete an install set from an iLO Repository. You cannot delete a baseline or node performing a function on the node.

Syntax

```
smartupdate delete --nodes [nodelist]
```

```
smartupdate delete --baselines [baselinelist]
```

Parameters/Options/Flags/Strings/...

`--nodes [nodelist]`

Object to perform the delete operation. Delimit the list with a space. If you do not provide a nodelist, all nodes are deleted.

`--baselines [baselinelist]`

Object to perform the delete operation. Delimit the list with a space. If you do not provide a list of baselines, all baselines are deleted.

```
--baselines <install_set_name> node=<node IP>
```

Delete an install set saved on an iLO Repository. Provide the install set name and the iLO 5, or Gen10 server running Windows or Linux, IP address. SUM cannot delete a locked install set.

Restrictions

Do not delete a baseline and node in the same command.

Windows usage

```
C:\> smartupdate delete /baselines "spp.2012.01"
```

```
C:\> smartupdate delete /nodes 10.0.0.1
```

Linux usage

```
$ ./smartupdate delete --baselines "spp.2012.01"
```

```
$ ./smartupdate delete --nodes 10.0.0.1
```

deploy

Enables you to perform the deployment for nodes in the current working set for the current session. If you do not specify any nodes, the default is all in the currently selected list.

Syntax

```
smartupdate deploy [--nodes [nodelist]]
```

Parameters

`--nodes [nodelist]`

A list of the nodes to perform the deploy operation on. You can specify each node in three ways:

```
domain_name: localhost
```

```
ip: 127.0.0.1
```

```
ipv6: ::1
```

If you do not provide a nodelist, all nodes in the session are deployed.

`--cleanup_onexit`

Deletes the following files:

- Firmware RPMs copied to the Linux directory `/usr/lib/i386-linux-gnu` or `/usr/lib/x86_64-gnu`.
- Extracted firmware RPM directory and folder contents.
- Un-extracted components of a baseline.
- If SUM is launched from a read-only location, this parameter deletes the `localsum` directory.

`--cleanupall_onexit`

Deletes the following files:

- Firmware RPMs copied to the Linux directory `/usr/lib/i386-linux-gnu` or `/usr/lib/x86_64-gnu`.
- Extracted firmware RPM directory and folder contents.
- Un-extracted components of a baseline.
- If SUM is launched from a read-only location, this parameter deletes the `localsum` directory.
- Debug logs.

Restrictions

If there are no updates for a node, SUM might indicate that it deployed updates to the node, but nothing was deployed.

Windows usage

```
C:\> smartupdate deploy /nodes 10.7.0.0
```

Linux usage

```
$ ./smartupdate deploy --nodes 10.7.0.0
```

`SUCCESS_NO_REBOOT`

Windows: 0

Linux: 0

Text: `The command was successful.`

`SUCCESS_REBOOT`

Windows: 1

Linux: 1

Text: `The command was successful but a reboot is required.`

`SUCCESS_NOT_REQUIRED`

Windows: 3

Linux: 3

Text: `The command completed successfully, but was not required or everything was already current.`

`FAILURE_GENERAL`

Windows: -1

Linux: 255

Text: A general failure occurred. For details, see the logs.

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: A bad input parameter was encountered. For details, see the logs.

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: The installation of the component failed.

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: The command failed. For details, see the logs.

findavailablenodes

Enables you to search the network for available nodes. To be available, you must be able to reach nodes and another management system must not have exclusive control of the nodes. This parameter searches the range of IP addresses provided and accepts an optional port.

Syntax

```
smartupdate findavailablenodes --type ip|port [ipAddrLow ipAddrHi] [port]
```

Description

Parameters/Options/Flags/Strings/...

--type ip|port

If you pass the type option, the value must match one of two types:

1. ip—Enter a range of IP addresses that SUM will search. Options: ipAddrLow; ipAddrHi
2. port—Enter a port number along with an IP address range that SUM can search. Options: ipAddrLow; ipAddrHi; portNum

Windows usage

```
C:\>smartupdate findavailablenodes /type ip 10.7.85.0 10.7.85.128
```

Linux usage

```
$ ./smartupdate findavailablenodes --type port 10.7.85.0 10.7.85.128 445
```

generatereports

Enables you to generate reports for the SUM session. These reports include Inventory, Firmware, Failed Dependency, Deploy preview, Installed components, and Combined reports. The output defaults to attributes values and the type of report to all available.

Syntax

```
smartupdate generatereports [--type inventory | firmware | installables |
faileddep | installed | combined] [--output html | xml | csv] [--nodes
[nodelist]] [--baselines [baselinelist]]
```

Parameters/Options/Flags/Strings/...

--type

If you specify the `type` parameter, it must be one of the following:

- Inventory
- Firmware
- Installables
- Faileddep
- Installed
- Combined

If you do not provide a type, all types possible are generated.

--output

- `xml`
- `csv`
- `html`

If you do not provide an output type, all formats are produced.

--nodes [nodelist]

SUM generates a single report with details for all the nodes in the list. If you do not specify a list of nodes, all nodes in the current session are selected.

--baselines [baselinelist]

List of baselines to include in the reports. If you do not specify a list of baselines, all baselines in the current session are selected.

Windows usage

```
C:\>smartupdate generatereports /type inventory /nodes localhost
```

Linux usage

```
$ ./smartupdate generatereports --type faileddep --nodes 10.7.85.0
10.7.85.128
```

SUCCESS_NO_REBOOT

Windows: 0

Linux: 0

Text: `The command was successful.`

SUCCESS_REBOOT

Windows: 1

Linux: 1

Text: `The command was successful but a reboot is required.`

SUCCESS_NOT_REQUIRED

Windows: 3

Linux: 3

Text: The command completed successfully, but was not required or everything was already current.

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: A general failure occurred. For details, see the logs.

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: A bad input parameter was encountered. For details, see the logs.

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: The installation of the component failed.

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: The command failed. For details, see the logs.

getattributes

Enables you to view any of the attributes for the session or selected node. You can change attributes using the `SetAttributes` command.

Syntax

```
smartupdate getattributes [--nodes [nodelist]] [--session]
```

Parameters/Options/Flags/Strings/...

`--nodes [<nodelist>]`

A list of the nodes for which to get the attributes. If both nodes and session attributes are not specified, the command returns attributes for all nodes and sessions.

`--session`

Lists the attributes available in the SUM session.

Windows usage

```
C:\> smartupdate getattributes /nodes localhost
C:\> smartupdate getattributes /nodes 10.0.1.2
```

Linux usage

```
$ ./smartupdate getattributes --nodes localhost
```

```
$ ./smartupdate getattributes --nodes 10.0.1.4
```

getbaselines

Enables you to show the list of the baselines in the current SUM session.

Syntax

```
smartupdate getbaselines [<baselinelist --details>] [nodeip]
```

Parameters

`baselinelist [--details]`

Baseline location to use for performing the operation. Specify `--details` to get detailed baseline information including a list of components and their version values.

`nodeip <ipaddress>`

Lists the install sets on the node. (Gen10 iLO 5 nodes only)

Windows usage

```
C:\>smartupdate getbaselines c:\spp\baseline1 /details
```

Linux usage

```
$ ./smartupdate getbaselines
```

getcomponentlogs

Enables you to view the component logs for a node.

Syntax

```
smartupdate getcomponentlogs [--nodes [<nodelist>]] [--component
<componentname>]
```

Parameters

`--nodes [<nodelist>]`

Nodes for this request. If you do not provide a nodelist, all nodes in the session are used.

`--component <componentname>`

The name of the component that SUM will display the log. If you use this parameter, you must enter a component name. If you do not specify a component, SUM generates logs for all components.

Windows usage

```
C:\> smartupdate getcomponentlogs /nodes localhost
```

Linux usage

```
$ ./smartupdate getcomponentlogs --nodes 10.0.1.1 --component cp020024.exe
```

`SUCCESS_NO_REBOOT`

Windows: 0

Linux: 0

Text: `The command was successful.`

`SUCCESS_REBOOT`

Windows: 1

Linux: 1

Text: The command was successful but a reboot is required.

SUCCESS_NOT_REQUIRED

Windows: 3

Linux: 3

Text: The command completed successfully, but was not required or everything was already current.

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: A general failure occurred. For details, see the logs.

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: A bad input parameter was encountered. For details, see the logs.

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: The installation of the component failed.

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: The command failed. For details, see the logs.

getcurrentlyinstalledversions

Enables you to get the currently installed versions of components on nodes after node inventory in the current SUM session.

Syntax

```
smartupdate getcurrentlyinstalledversions [--nodes [<nodelist>]]
```

Parameters

--nodes [<nodelist>]

A list of the nodes to get the installed versions. If you do not provide a nodelist, all nodes in the session are used.

Windows usage

```
C:\> smartupdate getcurrentlyinstalledversions /nodes localhost
```

Linux usage

```
$ ./smartupdate getcurrentlyinstalledversions --nodes localhost
```

SUCCESS_NO_REBOOT

Windows: 0

Linux: 0

Text: The command was successful.

SUCCESS_REBOOT

Windows: 1

Linux: 1

Text: The command was successful but a reboot is required.

SUCCESS_NOT_REQUIRED

Windows: 3

Linux: 3

Text: The command completed successfully, but was not required or everything was already current.

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: A general failure occurred. For details, see the logs.

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: A bad input parameter was encountered. For details, see the logs.

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: The installation of the component failed.

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: The command failed. For details, see the logs.

getenginestatus

Enables you to retrieve the status information of the SUM session.

Syntax

```
smartupdate getenginestatus
```

Windows usage

```
C:/>smartupdate getenginestatus
```

Linux usage

```
$ ./smartupdate getenginestatus
```

Example output

```
No SUM engine found running.
```

Example output

```
Process name:
Copyright:
Version:
Active Sessions:
Engine status:
Total number of sessions:
```

SUCCESS_NO_REBOOT

Windows: 0

Linux: 0

Text: The command was successful.

SUCCESS_REBOOT

Windows: 1

Linux: 1

Text: The command was successful but a reboot is required.

SUCCESS_NOT_REQUIRED

Windows: 3

Linux: 3

Text: The command completed successfully, but was not required or everything was already current.

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: A general failure occurred. For details, see the logs.

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: A bad input parameter was encountered. For details, see the logs.

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: The installation of the component failed.

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: `The command failed. For details, see the logs.`

getlogs

Launches the `GatherLogs` function for SUM, which collects all logs in to a compressed file. By default, SUM saves the log files in one of the following locations:

- Windows
  - `C:\cpqsystem\log`
  - `C:\cpqsystem\sum\log`
  - `C:\Windows\Temp\SUM`
  - `%localappdata%\sum`
- Linux
  - `/var/log/sum`
  - `/var/cpq`
  - `/var/tmp/SUM`

Options

`debuglogdir <directorypath>`

Tells SUM to find the log files in a directory other than the default directory.

`logdir <directorypath>`

Tells SUM to find the log files in a directory other than the default directory.

Syntax

```
smartupdate getlogs
```

Windows usage

```
C:\> smartupdate getlogs
```

Linux usage

```
$ ./smartupdate getlogs
```

`SUCCESS_NO_REBOOT`

Windows: 0

Linux: 0

Text: `The command was successful.`

`SUCCESS_REBOOT`

Windows: 1

Linux: 1

Text: `The command was successful but a reboot is required.`

SUCCESS_NOT_REQUIRED

Windows: 3

Linux: 3

Text: The command completed successfully, but was not required or everything was already current.

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: A general failure occurred. For details, see the logs.

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: A bad input parameter was encountered. For details, see the logs.

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: The installation of the component failed.

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: The command failed. For details, see the logs.

Getneededupdates

Enables you to see the details of needed updates based on either the state of the current selections or items passed in as part of the command after node inventory. If you do not specify any nodes, the command uses the currently selected ones in the session.

Syntax

```
smartupdate getneededupdates [--nodes [<nodelist>]]
```

Description

Parameters

--nodes [<nodelist>]

A list of the nodes on which to perform the check for needed updates operation. If you do not specify a nodelist, all nodes in the session are used.

Windows usage

```
C:\> smartupdate getneededupdates /nodes localhost
```

Linux usage

```
$ ./smartupdate getneededupdates --nodes localhost
```

SUCCESS_NO_REBOOT

Windows: 0

Linux: 0

Text: The command was successful.

SUCCESS_REBOOT

Windows: 1

Linux: 1

Text: The command was successful but a reboot is required.

SUCCESS_NOT_REQUIRED

Windows: 3

Linux: 3

Text: The command completed successfully, but was not required or everything was already current.

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: A general failure occurred. For details, see the logs.

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: A bad input parameter was encountered. For details, see the logs.

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: The installation of the component failed.

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: The command failed. For details, see the logs.

getnodes

Enables you to show the list of the nodes in the current SUM session. Gen10 and iLO 5 nodes also list install sets in the iLO repository, components in each install set, and pending iLO task queue items.

Syntax

```
smartupdate getnodes [<nodelist> --details]
```

Parameters

--<nodeslist>

The nodes on which to perform the operation.

--details

Specify `[--details]` to get detailed information.

Restrictions

Windows usage

```
C:\>smartupdate getnodes 10.0.1.1 /details
```

Linux usage

```
$ ./smartupdate getnodes 10.0.1.1 --details
```

SUCCESS_NO_REBOOT

Windows: 0

Linux: 0

Text: `The command was successful.`

SUCCESS_REBOOT

Windows: 1

Linux: 1

Text: `The command was successful but a reboot is required.`

SUCCESS_NOT_REQUIRED

Windows: 3

Linux: 3

Text: `The command completed successfully, but was not required or everything was already current.`

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: `A general failure occurred. For details, see the logs.`

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: `A bad input parameter was encountered. For details, see the logs.`

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: `The installation of the component failed.`

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: `The command failed. For details, see the logs.`

inventory

Enables you to scan nodes for the currently installed version of firmware, drivers, and hardware components.

Syntax

```
smartupdate inventory [--nodes <nodelist>] [--baselines <baselinelist>]
```

Parameters

`--nodes [<nodelist>]`

A list of the nodes on which to perform the inventory command. You can specify each node in the following ways:

- `localhost`
- `127.0.0.1`
- `::1`

`--baselines [<baselinelist or install_set_name>]`

Specify the path of the SPP or components or iLO saved install set. If the baseline location includes spaces or special characters, use escape characters (Linux) or double-quotations (Windows).

To use a saved install set on the iLO Repository, include a node IP address or DNS name and the baseline install set name.

If you include a node list with the baseline, SUM performs inventory on the nodes with the baseline specified. If you do not include a node list, then SUM only performs inventory on the baseline.

Restrictions

Set the management options (`use_ams`, `use_wmi`, `use_snmp`, `no_mgmt`) with the `setattributes` command before performing inventory.

After performing successful inventory on a VC node using an encryption key, use the `setattributes` command to assign the `encryption_key` parameter details.

Windows usage

```
C:\>smartupdate inventory /nodes 10.0.1.1 /baselines C:\Users\Administrator
\Desktop\blds\ilo\26.33
```

```
C:\>smartupdate inventory /nodes 10.0.1.1 /baseline <install_set_name>
```

Linux usage

```
$ ./smartupdate inventory --nodes 10.0.1.1 --baselines /root/Desktop/rono/
blds/ilo
```

`SUCCESS_NO_REBOOT`

Windows: 0

Linux: 0

Text: `The command was successful.`

`SUCCESS_REBOOT`

Windows: 1

Linux: 1

Text: `The command was successful but a reboot is required.`

SUCCESS_NOT_REQUIRED

Windows: 3

Linux: 3

Text: `The command completed successfully, but was not required or everything was already current.`

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: `A general failure occurred. For details, see the logs.`

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: `A bad input parameter was encountered. For details, see the logs.`

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: `The installation of the component failed.`

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: `The command failed. For details, see the logs.`

login

Enables you to set the credentials used for the current session.

Syntax

Authenticating with credentials for the first time:

```
smartupdate login [--username <username> --password <password>] [--port
<port>] [--ssl_port <sslport>] [--ftp_port <ftpport>] [--open_firewall]
```

Authenticating with current credentials:

```
smartupdate login
```

Parameters

`--username`

Optional parameter for adding username.

`--password`

Optional parameter for adding user password.

--port

Optional parameter to define a port that SUM uses.

The allowed values are:

80, 1024-65535

--ssl_port

Optional parameter to define an SSL port that SUM uses.

The allowed values are:

443, 1024-65535

--ftp_port

Optional parameter to define the FTP port that SUM uses. Include this parameter when updating FC switches.

The allowed values are:

disabled, 21, 1024-65535

--debuglogdir <filepath>

Tells SUM where to save log files. Enter a directory path.

--open_firewall

SUM cannot open all firewall instances. Manually open the firewall if SUM cannot open a firewall.

Windows usage

```
C:\> smartupdate login /username rono /password pwd123 /ssl_port 9091
```

Linux usage

```
$ ./smartupdate login --username rono --password pwd123 --ssl_port 9091
```

SUCCESS_NO_REBOOT

Windows: 0

Linux: 0

Text: The command was successful.

SUCCESS_REBOOT

Windows: 1

Linux: 1

Text: The command was successful but a reboot is required.

SUCCESS_NOT_REQUIRED

Windows: 3

Linux: 3

Text: The command completed successfully, but was not required or everything was already current.

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: A general failure occurred. For details, see the logs.

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: A bad input parameter was encountered. For details, see the logs.

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: The installation of the component failed.

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: The command failed. For details, see the logs.

setattributes

Enables you to change the attributes of an SUM node or session. The setattributes command might return an invalid parameter notice if the permissions are not correct, or if either the parameter is not part of the requested set or if the value is not allowed.

NOTE: The setattributes options and their values are separated by the equals sign (=). Do not add a space before and after the equals sign in any option and value combination.

To view a list of attributes you can change, use the command getattributes. The available attributes change based on the nodes and baselines added in each session. For example:

smartupdate /getattributes /session (Windows)

smartupdate /getattributes /nodes (Windows)

Syntax

```
smartupdate setattributes --nodes [<nodelist>] attribute1=value1
attribute2=value2
```

```
smartupdate setattributes --session attribute1=value1
```

Assign session attributes in the interactive CLI:

smartupdate setattributes --session <attribute>=<value> (Linux)

smartupdate setattributes /session <attribute>=<value> (Windows)

Parameters

--nodes [<nodelist>] attribute=value

Specify the nodes and attributes that you want to apply to nodes.

--session attribute=value

Specify the session attribute you want to apply. Session attributes are global.

Session parameter options

auto_save_session

[true|false]

auto_save_session_time

<integer>—The time in minutes. The default is 15 minutes.

engine_mode

[debug|release]

software

[true|false]—This value defines the option for all nodes added in this session.

firmware

[true|false]—This value defines the option for all nodes added in this session.

rewrite

[true|false]—This value defines the option for all nodes added in this session.

downgrade

[true|false]—This value defines the option for all nodes added in this session.

language

[en|ja|cn]

verbosity

[normal|verbose|veryverbose]

- normal—The default amount of information provided by SUM about a deployment.
- verbose—Provides more details about a deployment if information is available.
- veryverbose—Provides more details about a deployment if information is available.

verbose

[true|false]

```
smartupdate setattributes /nodes verbose=true
```

Determines the amount of details SUM provides during deployment.

report_dir

[directory_path]—Directory path where you want SUM to save reports. For example: C:\SUM_reports\.

open_firewall

[true|false]

Optional parameter that allows SUM to open the HTTP and HTTPs ports for external communication. Also used to open the ports for remote node functionality and for remote browser access.

Node parameter options

Use_vCenter

Options for VMware nodes:

[true|false]

If you set this attribute as true, enter the following settings:

- `vcenter_host`—IP address for the VMware vCenter node.
- `vcenter_username`—Username for the VMware vCenter node.
- `vcenter_password`—Password for the user connecting to the VMware vCenter node.

These parameters can also be set as session attributes.

`encryption_key`

Option:

String of a minimum of eight characters.

Required parameter for VC nodes with FIPS mode enabled. SUM uses the encryption key to encrypt the configuration saved to disk during the update process. This value is independent and does not need to match the VC password.

Assign this parameter after successful completion of VC inventory.

`ethactorder`

Possible values:

`ODDEVEN/PARALLEL/SERIAL/MANUAL`, the default value is `ODDEVEN`.

Use this command to determine VC-Enet module activation order.

`ilo_username`

Possible values:

`<ilo username>`

iLO username with Administrator privileges for Windows and Linux Gen10 target nodes whose iLO 5 is in any of the high security modes.

`ilo_password`

Possible values:

`<ilo password>`

Password for iLO Administrator user.

`fcactorder`

Possible values:

`ODDEVEN/PARALLEL/SERIAL/MANUAL`, the default value is `SERIAL`.

Use this command to determine VC–FC activation order for VC target.

`ethactdelay`

Valid entries:

0–60 minutes. The default is 0 minutes.

Use this command to determine the time, in minutes, between activating or rebooting VC-Enet modules for VC target.

`fcactdelay`

Valid entries:

0–60 minutes. The default is 0 minutes.

Use this command to determine the time, in minutes, between activating or rebooting VC-FC modules.

`use_sshkey`

(Linux nodes only)

Potential options:

- `true`
- `false`

Other attributes:

- `privatekeyfile`

  – Enter the address for the private key. The private key must be in PEM format.

- `passphrase`

  – (Optional) Enter the passphrase for the private key.

These parameters can also be set as session attributes.

`no_mgmt`

Windows and Linux targets

Potential options:

- `true`
- `false`

No AMS, SNMP, or WBEM Providers are selected for deployment.

The attribute must be set before node inventory so that the install set is updated. Setting it after node inventory will not impact the install set created.

`use_snmp`

For Gen8/9 Windows and Linux targets.

Windows: Selects the HPE Insight Management agent (SNMP agent) for deploy along with other components.

Linux: Selects sm-templates and snmp-agent rpm components along with other components.

Potential options:

- `true`
- `false`

SUM deploys SNMP components.

The attribute must be set before node inventory so that the install set is updated. Setting it after node inventory will not impact the install set created.

`use_wmi`

(Windows only)

Potential options:

- `true`
- `false`

SUM deploys WBEM Provider components. (Gen8/9 only)

The attribute must be set before node inventory so that the install set is updated. Setting it after node inventory will not impact the install set created.

use_ams

(Gen9 and later)

Potential options:

- `true`
- `false`

SUM deploys AMS components to both Windows and Linux systems.

The attribute must be set before node inventory so that the install set is updated. Setting it after node inventory will not impact the install set created.

skip_prereqs = <true|false>

By default SUM installs the CHIF driver on Windows nodes. Use the `true` option to skip installing the prerequisite self-inventory components, such as the CHIF driver.

ignore_tpm

Potential options:

`[true|false]`

This parameter tells SUM to ignore the TPM warning message when a TPM or TM module is installed on a server.

This parameter is supported on server and iLO nodes.

on_failed_dependency

`[omithost|omitcomponent|force]`

- `omithost`—SUM skips the host if it discovers a dependency failure. (Default)
- `omitcomponent`—SUM skips components with failed dependencies. Other components on the node are deployed.
- `force`—SUM forces the components with a failed dependency to deploy.

ignore_warnings

Potential options:

`[true|false]`

This parameter tells SUM to ignore warnings if it encounters any. Some errors are:

- Pending iLO 5 task queue warning.
- Exception task queue warning.
- iLO 5 or iSUT warnings.
- TPM/TM warnings.
- Low disk space.

action

Valid options:

`ifneeded`, `always`, `never`

Determines the reboot options for this node.

delay

Valid entries:

60 - 3600 (in seconds)

Delays the reboot of the server for the length of time specified in seconds.

Defaults to 60 if the value provided is < 60 and 3600 if the value provided is > 3600.

message

Valid entries:

`<Reboot message>`

Displays the specified reboot message on remote consoles connected to the server that will be rebooted.

iLO 5 parameters

save_install_set <true/false>

`True`: Saves the install set on the iLO Repository.

`False`: (Default) Does not save the install set on the iLO Repository.

install_set_name <SPP_baseline>

Provide the name of the install set to save on the iLO Repository. Valid only for SPP baselines.

install_set_description <description>

Saves the description you assign the install set.

manually_manage_iLO_repository = <true>

`True`: Prevents SUM from automatically removing install sets to make room for new install sets.

`False` (Default)

update_existing_recoveryset<true/false>

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

`False` (Default): Does not update existing recovery set in iLO Repository.

Superdome X parameter options

skip_missing_compsig

Possible values: [TRUE|FALSE]

Skips deploying components that lack component signature files. The default value is false.

update_type

`[ALL/COMPLEX/PARTITIONS]`—This value tells SUM what to Superdome X nodes to update.

device_list

`[devicelist]`—This value tells SUM which Superdome X nodes you want to update.

reboot_list

`[rebootlist]`—This value tells SUM which Superdome X nodes to reboot after deployment.

Downgrade and rewrite parameter options

The following describes GUI downgrade and rewrite combinations and the corresponding interactive CLI parameters:

Software only

```
smartupdate setattributes /nodes localhost rewrite=false downgrade=false
firmware=false software=true
```

Firmware only

```
smartupdate setattributes /nodes localhost rewrite=false downgrade=false
firmware=true software=false
```

Firmware and software

```
smartupdate setattributes /nodes localhost rewrite=false downgrade=false
firmware=true software=true
```

Downgrade software

```
smartupdate setattributes /nodes localhost rewrite=false downgrade=true
firmware=false software=true
```

Downgrade firmware

```
smartupdate setattributes /nodes localhost rewrite=false downgrade=true
firmware=true software=false
```

Downgrade both

```
smartupdate setattributes /nodes localhost rewrite=false downgrade=true
firmware=true software=true
```

Rewrite software

```
smartupdate setattributes /nodes localhost rewrite=true downgrade=false
firmware=false software=true
```

Rewrite firmware

```
smartupdate setattributes /nodes localhost rewrite=true downgrade=false
firmware=true software=false
```

Rewrite both

```
smartupdate setattributes /nodes localhost rewrite=true downgrade=false
firmware=true software=true
```

Rewrite and downgrade software

```
smartupdate setattributes /nodes localhost rewrite=true downgrade=true
firmware=false software=true
```

Rewrite and downgrade firmware

```
smartupdate setattributes /nodes localhost rewrite=true downgrade=true
firmware=true software=false
```

Rewrite and downgrade both

```
smartupdate setattributes /nodes localhost rewrite=true downgrade=true
firmware=true software=true
```

Windows usage

```
C:\> smartupdate setattributes /nodes 10.0.1.5 firmware=true software=true
rewrite=true downgrade=true

C:\> smartupdate setattributes /nodes 10.0.1.5 update_type=all

C:\> smartupdate setattributes /session open_firewall=yes
```

```
C:\> smartupdate setattributes /nodes 10.0.0.4 use_vcenter=true
vcenter_host=10.0.0.1 vcenter_username=test vcenter_password=password
```

Linux usage

```
$ ./smartupdate setattributes --nodes 10.0.1.5 firmware=true software=true
rewrite=true downgrade=true

$ ./smartupdate setattributes --nodes 10.0.1.5 update_type=all

$ ./smartupdate setattributes -session open_firewall=yes

$ ./smartupdate setattributes --nodes 10.0.0.5 use_vcenter=true
vcenter_host=10.0.0.1 vcenter_username=test vcenter_password=password
```

SUCCESS_NO_REBOOT

Windows: 0

Linux: 0

Text: The command was successful.

SUCCESS_REBOOT

Windows: 1

Linux: 1

Text: The command was successful but a reboot is required.

SUCCESS_NOT_REQUIRED

Windows: 3

Linux: 3

Text: The command completed successfully, but was not required or everything was already current.

FAILURE_GENERAL

Windows: -1

Linux: 255

Text: A general failure occurred. For details, see the logs.

FAILURE_BAD_PARM

Windows: -2

Linux: 254

Text: A bad input parameter was encountered. For details, see the logs.

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: The installation of the component failed.

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: The command failed. For details, see the logs.

shutdownengine

Enables you to terminate the SUM engine. If other sessions are active, this fails unless you use the `--force` option.

CAUTION: Using `--force` can cause loss of information.

Syntax

```
smartupdate shutdownengine [--force]
```

Parameters

`--force`

Force shutdown even if other sessions are still active.

`--cleanup_onexit`

Deletes SUM files from remote nodes, except user and debug log files.

`--cleanupall_onexit`

Deletes SUM files, including installed firmware RPMs. Does not delete user log files.

Windows usage

```
C:/>smartupdate shutdownengine /force
```

Linux usage

```
$ ./smartupdate shutdownengine --force
```

`SUCCESS_NO_REBOOT`

Windows: 0

Linux: 0

Text: `The command was successful.`

`SUCCESS_REBOOT`

Windows: 1

Linux: 1

Text: `The command was successful but a reboot is required.`

`SUCCESS_NOT_REQUIRED`

Windows: 3

Linux: 3

Text: `The command completed successfully, but was not required or everything was already current.`

`FAILURE_GENERAL`

Windows: -1

Linux: 255

Text: `A general failure occurred. For details, see the logs.`

`FAILURE_BAD_PARM`

Windows: -2

Linux: 254

Text: `A bad input parameter was encountered. For details, see the logs.`

FAILURE_COMPONENT_FAILED

Windows: -3

Linux: 253

Text: `The installation of the component failed.`

FAILURE_COMMAND_FAILED

Windows: -4

Linux: 252

Text: `The command failed. For details, see the logs.`

Stage components

Syntax

`--stage_only` (Linux)

`/stage_only` (Windows)

Description

Enables you to upload all applicable components to an iLO Repository by creating a new install set. After staging, you can choose to deploy the node, which will invoke the new install set and complete deployment.

Example

`smartupdate --silent <other_params> --save_install_set --install_set_name Test1 --stage_only` (Linux)

`/smartupdate /silent <other_params> /save_install_set /install_set_name Test1 /stage_only` (Windows)

stageonly

Uploads all applicable components to an iLO Repository by creating a new install set.

Syntax

`--stage_only` (Linux)

`/stage_only` (Windows)

Possible values

YES, NO (default)

validate baselines

Description

Ensures that there are no issues with inventorying the bundle/component metadata after you add the baselines.

Syntax

`smartupdate validate --baselines [<baselinelocation>]`

Parameters

`baselinelocation`

Baseline location to use for performing the operation.

Windows usage

```
C:\>smartupdate validate /baselines c:\spp\baseline1
```

Linux usage

```
$ ./smartupdate validate --baselines /root/spp/baseline1
```

Websites and support

Websites

Hewlett Packard Enterprise Information Library	**www.hpe.com/info/EIL**
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 **Support and other resources**.

Support and other resources

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

  Hewlett Packard Enterprise Support Center: Software downloads

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

  Software Depot

  https://www.hpe.com/support/softwaredepot
- 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

IMPORTANT: Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Passport 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 will initiate a fast and accurate resolution based on your product's service level. 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.

Remote support and Proactive Care information

HPE Get Connected

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

HPE Proactive Care services

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

HPE Datacenter Care services

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

HPE Proactive Care service: Supported products list

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

HPE Proactive Care advanced service: Supported products list

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

Proactive Care customer information

Proactive Care central

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

Proactive Care service activation

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

Warranty information

To view the warranty information for your product, see the links provided below:

HPE ProLiant and IA-32 Servers and Options

https://www.hpe.com/support/ProLiantServers-Warranties

HPE Enterprise and Cloudline Servers

https://www.hpe.com/support/EnterpriseServers-Warranties

HPE Storage Products

https://www.hpe.com/support/Storage-Warranties

HPE Networking Products

https://www.hpe.com/support/Networking-Warranties

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, send any errors, suggestions, or comments to Documentation Feedback (**docsfeedback@hpe.com**). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.