HP System Software Manager

User Guide

About this guide

This guide uses the following styles to distinguish elements of text:

Style	Definition
`<variable>`	Variables or placeholders are enclosed in angle brackets. For example, replace `<pathname>` with the appropriate path, such as `C:\Windows\System`. When typing the actual value for the variable, omit the brackets.
`[optional parameters]`	Optional parameters are enclosed in square brackets. When specifying the parameters, omit the brackets.
`"literal value"`	Command line text that appears inside quotation marks should be typed exactly as shown, including the quotation marks.
Type	When you are instructed to type information, type the information without pressing the Enter key.
Enter	When you are instructed to enter information, type the information and then press the Enter key.

Table of contents

1   Getting started

What is HP System Software Manager?

HP System Software Manager (SSM) is a free utility that enables you to do the following:

- Deploy software update packages and system ROM updates from a single File Store to multiple desktop, workstation, and notebook computers simultaneously

- Run reports to compare software updates awaiting deployment against machines in the network to identify PCs in need of updates

- Create and deploy customized update packages

SSM complements software delivery products by making it easy to create distributable update packages and target those update packages to the applicable systems.

SSM can be integrated with a variety of software installation and delivery mechanisms, such as a logon script, software distribution programs, email, or local execution.

Download the latest version of SSM from http://www.hp.com/go/ssm. Software updates (SoftPaqs) supported by SSM are easily identified by this icon:

System requirements

File Store server	Client
Windows® XP Professional, Windows Vista®, Windows 7, Windows 8, or Windows 8.1	Windows 7, Windows 8, or Windows 8.1
Internet Explorer 4.01 or later	Internet Explorer 4.01 or later

*Running SSM in Windows PE is not supported. To flash the BIOS of systems running Windows PE, use the latest version of the HPQFlash 5.1 series and HP BIOS and Update Recovery 1.0.27.1 or higher. Support for Windows XP has been deprecated as of version 3.0.7.1.

Installing and running SSM

SSM has many powerful command-line options and usage models. The most basic steps to use SSM are outlined here. See for a complete list of command-line options.

1.   Download the SSM SoftPaq from http://www.hp.com/go/ssm.

2.   Run the SSM SoftPaq to extract the SSM program and support files to a directory.

3.   Create and share a directory on the server to be used as the File Store, such as `C:\SSMFS` shared as `\\MyServer\SSMFS`.

4.   Copy **SSM.exe** and **ssm.cab** to the File Store.

NOTE: To help organize the File Store, create subfolders with meaningful names to hold each SoftPaq, such as `\\MyServer\SSMFS\Audio\IDT\sp65011.exe`.

5. Place all SSM-supported SoftPaq files in the File Store. Each customized package must be located in a separate subdirectory. See Package deployment on page 8 for more information about custom packages.

6. On each client computer, run SSM in Update Mode by specifying the UNC path to the File Store on the command line:

   `\\MyServer\SSMFS\SSM.exe \\MyServer\SSMFS /accept`

   This step updates the system software (BIOS and drivers) on the client according to the contents of the File Store.

7. A log file of actions performed can be viewed in the user's temporary directory. A quick method to view the log file on the client system is to click **Start > Run**, then enter `%temp%\%computername%.htm`.

8. Review the log file to view the list of software applicable to the system, updates performed, and any errors. If no errors are found, then the log file does not include an error section. See Log file description on page 14 for more information.

Components of SSM

This section covers the various SSM modules and their functions.

Table 1-2 SSM modules and functions

Module	Function
SSM.exe	This is the SSM executable.
ssm.cab	This file contains the support files needed by SSM. These consist of the Microsoft® Data Access Object (DAO) DLLs, drivers to support ROM flash and device detection, and the SSMMain executable.
ssm.bin	This file is created when SSM is run in Administrative Mode. It contains runtime settings and the encrypted setup password. All settings in this file can be overridden by command-line parameters. See Command line usage on page 10 for more information.
ssmcva.mdb	This is the Update Information Database (UID) file that contains information about the software updates in a File Store. The UID is created in the File Store by running SSM in Administrative Mode on the server. If SSM is running in Update Mode and a UID is not present in the File Store, SSM scans the File Store and builds a temporary UID. Having a UID in the File Store reduces network traffic and improves performance by condensing the relevant File Store information into a single file. The file CCMCVARESULT.XML contains the UID data in a different format for use with HP Client Manager.
SSM log file	SSM creates a log file or updates an existing log file after updating the machine. The log file is created in the temporary directory and can be copied to a centralized log file path specified on the command line or ssm.bin file. See Log file description on page 14 for more information.
SSM trace file	When `/debug` is used on the command line, SSM adds a file to the temporary directory. This is useful when debugging SoftPaq CVA problems.

Features of SSM

Device driver updates

When searching for updates, SSM detects the currently installed hardware configuration and selects only the appropriate drivers for installation. The File Store can contain multiple video, NIC, and audio drivers, but SSM installs only the appropriate updates.

System ROM BIOS flash support

SSM detects the platform and BIOS version of the target system, then selects the appropriate ROM BIOS update package from the File Store. If a newer BIOS is available, then SSM flashes the new BIOS to the system and optionally reboots the system. The new BIOS is not effective until the system is rebooted.

NOTE: HP recommends that all BIOS firmware updates use HPQFlash as the installation mechanism. SSM now supports running HPQFlash and passes the setup password, if any, in an encrypted file.

Customized packages

If a particular SoftPaq does not contain the exact information or configuration information that is desired, a customized package can be created. Create a subdirectory containing the customized packages and a package description file (CVA). If the target system is out of date, SSM copies the files from the File Store to a temporary directory on the local system and installs the customized package. See Package deployment on page 8 for more information.

2 Modes of operation

There are two modes of operation: Administrative Mode on page 4 and Update Mode on page 6.

Administrative Mode

Use Administrative Mode to configure the simultaneous update of multiple networked client computers. Administrative Mode lets you do the following:

- Set specific runtime options for the utility when updating client computers

- Create a read-only network share for the File Store

- Create a centralized read-write network share in which to store log files

- Encrypt setup password in a configuration file

- Build the UID on the server

> **NOTE:** If a UID is present, it must be rebuilt each time a SoftPaq is added or removed from the File Store.

To run Administrative Mode, complete the following steps:

1. Create a directory to be used as a File Store (for example, `C:\SSMFS`). Updates will be stored in and accessed from this directory by client computers. SSM shares read access to this directory with everyone.

2. Copy **SSM.exe** and **ssm.cab** to the root of the File Store.

3. Place all SoftPaqs or customized packages in the File Store.

4. Double-click **SSM.exe** to launch SSM in Administrative Mode.

5. Read the license agreement and click **Accept** to proceed. A welcome message displays a brief introduction to the steps required to configure SSM using Administrative Mode.

> **NOTE:** Have the following information handy:
> - Location of the File Store where updates are to be kept
> - Location of the centralized log file share
> - Setup password (if one is set on client computers)

6. Click **Next**. The location of the File Store and the default share name are displayed. Optionally, you can specify a different File Store and share name.

7. Click **Next**. A message appears to remind you to copy all SoftPaqs to the File Store before proceeding.

8. Click **Next**. The **Advanced Client Options** page is displayed.

9. Click the check box next to each client option you want:

    - **Reboot client upon completion of updates**

After applying updates, reboot the client computer so that updates take effect.

- **Display update status bar on client system**

    Display a status bar on the client system so that individual users can see progress and completion of updates.

- **Force client systems to update**

    Apply updates to the client system, even if the client system already has a newer version of the software installed.

- **Create centralized logfile share**

    Create a log file for each client on a centralized share. If you check this option, enter the folder and share name for the centralized log files.

10. Click **Next**. If you want to use a setup password for all client systems, browse to the location of an encrypted password file. If an encrypted password file does not exist, click the **Launch** button to open the password creator utility.

> **TIP:** See the *HP BIOS Configuration Utility User Guide* for a complete list of supported characters in the setup password.

> **NOTE:** Setup passwords are used only for system ROM BIOS updates and system BIOS setting updates.

11. Click **Next**. A summary of all operations that will be performed is displayed.

12. Click **Next**. Indicate if you want SSM to check the HP website for updated CVA files when building the UID. CVA files are occasionally updated on the HP website to support new computers or correct minor problems with the CVA file. SSM obtains CVA files in the following order:

    - From the FTP site

    - From CVA files in the existing File Store

    - From within SoftPaqs in the File Store

> **NOTE:** If you modify CVA files in the File Store, either write-protect the CVA file or use the `/nocvaupdate` option to prevent overwriting the modified CVA file.

13. Click **Build Database** to begin building the software update database. If desired, click **Check ftp.hp.com for CVA file updates**.

14. Click **Finish** to complete the process. SSM creates the ssm.bin file.

You must rebuild the database each time you add or remove a SoftPaq from the File Store. Administrative Mode lets you rebuild the database manually. See for more information. The database can be rebuilt quickly by executing the following from the command line: `SSM /am_bld_db`.

UID performance considerations

Update Mode uses information from the database to determine which software is applicable for a specific client system. The database is built using information from the CVA files located in the File Store. The CVA specifies what languages, devices, and platforms a given SoftPaq supports.

SSM attempts to match EXE and CVA files in the File Store. If a CVA file does not exist for a given SoftPaq EXE file, SSM optionally attempts to download the CVA file from the HP website. SSM then extracts the CVA from the SoftPaq.

Store the SoftPaqs in the File Store exactly as downloaded; do not run or extract them within the File Store. HP recommends that you put the SoftPaq files in subdirectories with meaningful names, such as `\` `\MyServer\SSMFS\Audio\IDT\sp65011.exe`.

Customized packages should be extracted and stored as a CVA file in their own subfolder in the File Store. See Custom CVA files on page 17 for detailed information on setting up a customized CVA file.

You must rebuild the database each time you add or remove a SoftPaq from the File Store. Administrative Mode lets you rebuild the database manually. If a database file does not exist, Update Mode rebuilds the database each time SSM is run. This takes a little longer than running Update Mode with a database in the File Store. See Update Mode on page 6 for more information. The database can be rebuilt by executing the following from the command line: `SSM /AM_BLD_DB`.

Update Mode

Update Mode updates the client system from the File Store. It is invoked by specifying the File Store location as a command-line parameter. See Command line usage on page 10 for detailed information on command-line parameters.

The functions that can be performed using Update Mode include the following:

- **Display a Cancel option for low bandwidth users**—If Update Mode is executed with the optional `/cancel` parameter, SSM determines if a user-initiated cancellation of updates is permitted for all users or just those connecting with dial-up networking and displays the **Cancel** dialog if permitted.

- **Read settings from ssm.bin**—SSM searches the File Store for the file ssm.bin. The BIN file contains settings chosen through Administrative Mode and the encrypted setup password file. The BIN file also indicates that the administrator accepts the license agreement. The BIN file is optional. To accept the license agreement, use the `/accept` command line parameter. All other options can be specified on the command line.

- **Create or copy the UID**—SSM can operate without a pre-built UID, but performance is greatly improved with a pre-built UID. This method is slower, but ensures that the database is always up to date. SSM searches for a pre-built UID in the path specified on the command line.

  If the database is found, SSM copies it locally and uses it. If the database is not found, SSM builds a temporary database. SSM scans the File Store for EXE files and CVA files. For any EXE files that do not have a corresponding CVA file, SSM then attempts to extract a CVA file. If the File Store is across the network, this causes additional network traffic. After all the CVA files have been extracted, SSM creates a temporary database from the CVA files. After the update process is complete, the temporary database is deleted.

- **Query the UID**—SSM uses information from the UID to determine what updates are available and the revision levels of the software in those updates. It queries the database for packages that match the specific device, platform, operating system, and language of the current system. The results are considered possible updates.

- **Perform updates**—SSM then compares the revision level of the possible updates to the existing software. Only those update packages that are newer than the currently installed software are updated on the client system, unless the `/force` option is used. SSM executes the packages in the order they appear in the database. Multiple File Stores can be used to install packages in a particular order.

- **Update the log file**—A log file in HTML format is created for tracking purposes. By default, the log is stored in the current user's temporary directory on the client system, unless a different location is specified in the command line. Select **Create centralized logfile share** when running Administrative Mode to copy the log to the specified log file share on the server.

An optional progress indicator is displayed throughout the update process; do not restart the computer while SSM is running unless you are prompted to do so.

3 Package deployment

Can I use SSM to deploy customized software updates?

SSM requires detailed information about each software component in the form of a CVA file. In addition, any software deployed using SSM must meet the requirements described in this chapter. To create a customized software update or custom deliverable, do the following:

1. Create a CVA file for each deliverable and specify the required parameters in the CVA file.

2. Store the CVA file and related software components in a separate subfolder of the File Store.

What is a CVA file?

The CVA file describes the deliverable. It contains the name and description of the package and indicates which languages, systems, and devices are supported. It is a plain text file similar to a Windows INI file. SSM supports SoftPaq packages containing an embedded CVA file and customized packages stored in a subfolder with a customized CVA file. You can create a customized CVA file using any text editor, such as Notepad. See Custom CVA files on page 17 for detailed information needed to create your own CVA files.

Additional requirements for software packages

- **Post-operating system silent installation**—All drivers supported by SSM must have post-operating system installation mechanisms. This means that they must have a self-contained executable setup program such as an InstallShield program, a batch file, or Microsoft Installer. If the deliverable requires changes to INF files, then they need to be included in an executable that can be run by SSM. The only function of SSM is to detect the hardware and software and run a single command specified by the CVA file.

- **Detectable deliverable version**—To allow SSM to determine if a specific deliverable version is installed on a system, the deliverable must have a significant file. The significant file is used to determine if the software is already installed and, if so, which version is installed. The significant file must have the correct version in its resource header. Any time a deliverable's version and revision change, the value in the resource header of this significant file must change too.

- **Controlled reboot of system**—All packages must support a controlled reboot of the system. This means that they cannot reboot the system by default. If the vendor delivers an installation mechanism that requires a reboot, use the `/reboot` option for SSM. This allows multiple packages to be installed with a single reboot at the end of the process. If the application reboots the system, SSM might not complete successfully while waiting for the application to finish and might leave temporary files on the client system.

- **Installation mechanism**—Any installation mechanism provided must be able to gracefully recover from a failed installation by exiting without requiring user interaction. HP recommends using a mechanism that can log errors. The installation mechanism must support upgrading from previous versions.

- **Customized packages structure**—Each customized package supported through SSM must reside in its own separate folder within the File Store. Its associated CVA file must have a unique name and must reside in the root of this unique subfolder. See File Store on page 13 for additional information about the structure of the File Store.

4 Using SSM with commercial software distribution systems

A commercial software distribution system can be used to copy **SSM.exe**, **ssm.cab**, and the File Store contents to the target system and execute SSM. This ensures that there are no network interruptions while SSM is updating the target.

Another option is to use the commercial software distribution system to execute SSM remotely without copying the files to the target system. For this method, ensure that SSM has been run in Administrative Mode to create a UID file (`ssmcva.mdb`) in the File Store. Otherwise, SSM might have to copy each SoftPaq over the network to the target system to extract the CVA file to determine the SoftPaq properties. Another drawback of this method is that some software deployment programs can only launch in the context of the local system account, which does not have domain network file share rights. The `/INSTALL` command-line parameter is recommended with this method to prevent `ssm.cab` from being copied over the network every time.

A Command line usage

Command line syntax

To run SSM in Administrative Mode from the command line, use the following syntax:

`SSM [optional parameters]`

For example: `SSM /am_bld_db`

To run SSM in Update Mode from the command line, use the following syntax:

`SSM <File Store path> [optional parameters]`

For example: `SSM C:\SSMFS /accept /debug`

📝 **NOTE:**   The File Store path can be either a fully qualified Universal Naming Convention (UNC) path or a path that is relative to SSM.exe.

Command line parameters

Parameter	Modes	Description
`/accept` or `/a`	U	Accepts the license agreement. This allows SSM to be used in Update Mode without the ssm.bin file.
`/am_bld_db`	A	Silently rebuilds the UID in Administrative Mode.
`/cancel: <Numeric Value>[,modem]`	U	Allows a user to cancel an update. Enables a **BEGIN** and **CANCEL** button on the client screen. The numeric value is a timeout in seconds before SSM starts the update process. The `modem` flag allows this to take effect only when a modem connection is detected and to support mobile users.
`/CurrentSetupPasswordFile:"<filename>"` —or— `/cpwdfile:"<filename>"`	U	Specifies the current BIOS setup password by providing the name of encrypted password file. The current password file must be provided when changing either the BIOS settings or the password. You can specify one current password file only. **NOTE:** Requires SSM version 2.53.8.1 or later.
`/debug` or `/d`	U, A	Provides additional information in the log file and records database build problems in the file error_log.txt. It also creates a trace file (with extension .ssmtrace) with additional information useful in debugging SoftPaq or installation problems.
`/eDebug`, `/ExtendedDebug`, or `/extended_debug`	U	Provides greater details when an error is encountered. Extended debug is designed to be run in combination with a single SoftPaq.
`/force` or `/f`	U	Bypasses software version detection and installs all packages that match the hardware installed on a system, regardless of version or date. It is useful when an update was interrupted, or to ensure that all clients have the same revision of a deliverable.

Parameter	Modes	Description
		NOTE: This parameter does not force the installation of the driver, only the execution of the installation program. If the setup program of the software update performs version checking and does not install, then the driver is not installed.
`/help`	U	Displays an abbreviated list of command line parameters with short descriptions of their function. (Available in SSM version 3.0.1.1 or later.)
`/install`	U	Causes the installer stub to create a subfolder on the client and install a portion of the SSM engine and the support files in that folder. It also creates a set of registry flags with data regarding the current installation. In a network File Store share environment, this eliminates the need for SSM.exe to copy and extract ssm.cab every time it runs.
`/log: <path>`	U	Specifies a UNC path location of the client log file and debug trace file if `/debug` is specified. The command-line parameter overrides UNC path specified in Administrative Mode. The file name will be updated automatically, based on the computer name.
`/nocvaupdate` or `/nu`	A	Tells SSM not to check the HP FTP site for CVA updates. Used with `/am_bld_db`.
`/nodialog` or `/s`	U	Suppresses the progress dialog and overrides the `/title` parameter.
`/nolog`	U	Suppresses creation of the log file and overrides the `/log` parameter.
`/noreboot`	U	Suppresses a reboot of the machine by SSM. A reboot can be requested by an update through CVA or be requested through Administrative Mode. Overrides the `/reboot` parameter.
`/reboot` or `/rb`	U	Reboots the machine if updates were performed. If an installation mechanism requires a reboot, use the `/reboot` option. The reboot occurs after all packages listed in the File Store are processed by SSM. The `/reboot` option does not interrupt SSM processing.
`/report` or `/r`	U	Allows the administrator to determine what updates are available without making any changes to the client system. SSM performs detection and generates a log file, but does not execute any update packages. **NOTE:** SSM does not report possible `/repset` BIOS changes.
`/runonce`	U	Causes SSM to clear itself from the RunOnce registry key. This command-line switch is used as part of a preinstallation package.
`/sort:ascending` or `/sort:descending`	U	`/sort:ascending` installs starting with the lowest numbered SoftPaqs first; `/sort:descending` does the opposite. Custom SoftPaqs are always last.
`/timeout:<Numeric Value>` —or— `/to:<Numeric Value>`	U	Overrides the SSM default timeout of fifteen minutes before aborting the execution of a deliverable. The numeric value is the number of minutes for SSM to wait before killing the update packages installer. **NOTE:** A value of 0 means wait indefinitely on the package and is not suggested.

Parameter	Modes	Description
`/title:"<title>"` or `/ti:"<title>"`	U	Specifies the title for the slider bar dialog (must be specified in quotes).
`/useinstallpath:<path>`	U	Specifies the folder from which SSM will expand and run each SoftPaq installation. By default, SSM expands the SoftPaq to a temporary folder in the Windows temporary directory and cleans up the folder after installation. By specifying `/useinstallpath`, SSM expands each SoftPaq to a subfolder, executes its silent install command from this location, and does not delete this folder after installation. If the path contains spaces, be sure to enclose the argument in quotation marks, for example, `/useinstallpath:"C:\SSM SWSetup"`.

B   File Store

File Store structure

The File Store is a folder in which the software update packages are stored. It can be located on a local hard drive, a local CD, or a network share accessible to the client computers. The File Store itself can be read-only, but the location where the log files are to be stored must be read/write. To perform a remote update to a File Store using the `/AM_BLD_DB` option, the administrator must have read/write access to the network share containing the File Store.

The File Store folder contains SoftPaqs (not expanded), CVA files, and any configuration files.

See the following example of a linear File Store:

```
/FileStore

    spxxx1.exe

    spxxx2.exe
```

For each customized deliverable, there must be a unique subfolder within the File Store. Each subfolder contains the individual components of the software deliverable and a customized CVA file. See Custom CVA files on page 17 for detailed information about creating a customized CVA file.

See the following example of a nested File Store with customized packages:

```
/FileStore

    /ROMS

        spxxxx.exe

        spxxxx.cva

        spxxxx.cfg

    /Drivers

        spxxxx.exe

        spxxxx.cva

        spxxxx.cfg
```

C   Log file description

A log file in HTML format is created for tracking purposes. By default, the log is stored in the temporary directory on the client system, unless a different location is specified in the command line. If the **Create centralized logfile share** option is selected in Administrative Mode, the log is copied to the specified log file share on the server. It has a .htm extension and its name is based on the client computer's name. Creation of the log file can be suppressed altogether.

The log file contains the following information:

- **Date and time**—Lists when the log file was generated, the name of the computer that was updated, and the version of SSM and the command line used.

- **Possible Updates**—Lists the device identifiers and update packages that have been found in the database and could be updated on the client system.

- **Updates**—Lists the devices and update packages used to update the client system.

> 📝 **NOTE:**   If the significant file (as described in the DetailFileInformation section of the CVA file) is newer or the same as the update, then the update is neither installed nor listed in the Updates section. In such cases, the debug trace file has additional information about the significant file found, its version, and the reason why the update was not installed. For example, `Wed Apr 23 17:53:18 2014: SP12345.CVA` is older than or same as the installed version.

- **Errors**—Lists errors that occurred during the detection and upgrade process. This section of the log is generated only if errors occurred.

- **Client Information**—Lists the machine ID, system ROM version, system serial number, and asset tag information.

- **Detected Devices**—Lists any PCI or plug-and-play device IDs found in the system, the ROM name, and the date and system identifier.

See the log file to get the system ID information when generating CVA files.

For example, if DOMAIN\Jsmith is the current user on a Windows Vista machine named COMPUTER1, then the log file is located in the file `C:\Users\JSMITH\AppData\Local\Temp\COMPUTER1.HTM` by default.

The system environment variables Temp and COMPUTERNAME are useful for locating the log file.

Log file example

This log file shows SSM updating the ROM on an HP Compaq dc7900 system:

Log file generated on 05/20/11 09:15:36 PM for DC7900

- SSM Version: 2.15 A7 (2.15.0.7)

- Parameters: /TEMP:"C:\Users\ADMINI~1.NN\AppData\Local\Temp
  \SSMAFBE.tmp" /SSMFS:"C:\dc7900" /SSMPATH:"C:\dc7900" . /a

Possible Updates

- Update 1: HP Compaq Business Desktop System BIOS (786G1 BIOS) -
  Version: 1.23 A (SP47156.CVA)

Updates Performed

- Update 1: HP Compaq Business Desktop System BIOS (786G1 BIOS) -
  Version: 1.23 A (SP47156.CVA Syntax v.:Success (Return code=0x0))

Client Information

- System Manufacturer: Hewlett-Packard

- System Product: HP Compaq dc7900

- System Version:

- Base Board Manufacturer: Hewlett-Packard

- Base Board Product: 3031h

- Base Board Version:

- Machine ID: 0x3031

- BIOS Release Date: 08/25/2009

- BIOS Version: 786G1 v01.22

- System Serial Number: PPL3030036

- Asset Tag: PPL3030036

- OS: WV64PR - OEM

- LANG: US

- WMI Provider: Firmware Interface

- CPQMC DLL: Version 1.00 E8 (1.0.4.8)

- Driver: Version: 0.0.0.0

Detected Devices

- Device 0: "*ISATAP\0000","isatap.{BD8B9A30-A39E-4778-
  BB60-0EA04F5BE859}","Net"

- Device 1: "ACPI\FixedButton\2&daba3ff&1","@machine.inf,%acpi
  \fixedbutton.devicedesc%;ACPI Fixed Feature Button","System"

- Device 2: "ACPI\GenuineIntel_-_EM64T_Family_6_Model_23_0","Intel(R)
  Core(TM)2 Quad CPU Q9650 @ 3.00GHz","Processor"

- `Device 3: "ACPI\GenuineIntel_-_EM64T_Family_6_Model_23\_1","Intel(R) Core(TM)2 Quad CPU Q9650 @ 3.00GHz","Processor"`

- `Device 4: "ACPI\GenuineIntel_-_EM64T_Family_6_Model_23\_2","Intel(R) Core(TM)2 Quad CPU Q9650 @ 3.00GHz","Processor"`

- `Device 5: "ACPI\GenuineIntel_-_EM64T_Family_6_Model_23\_3","Intel(R) Core(TM)2 Quad CPU Q9650 @ 3.00GHz","Processor"`

- `Device 6: "ACPI\IFX0102\1","@tpm.inf,%std_desc%;Trusted Platform Module 1.2","SecurityDevices"`

- `Device 7: "ACPI\PNP0000\4&269b4552&0","@machine.inf, %*pnp0000.devicedesc%;Programmable interrupt controller","System"`

- `Device 8: "ACPI\PNP0003\1","@machine.inf,%*pnp0003.devicedesc %;Advanced programmable interrupt controller","System"`

- `Device 9: "ACPI\PNP0100\4&269b4552&0","@machine.inf, %*pnp0100.devicedesc%;System timer","System"`

- `Device 10: "ACPI\PNP0103\1","@machine.inf,%*pnp0103.devicedesc%;High precision event timer","System"`

- `Device 11: "ACPI\PNP0200\4&269b4552&0","@machine.inf, %*pnp0200.devicedesc%;Direct memory access controller","System"`

- `Device 12: "ACPI\PNP0303\4&269b4552&0","@keyboard.inf, %*pnp0303.devicedesc%;Standard 101/102-Key or Microsoft Natural PS/2 Keyboard","Keyboard"`

- `Device 13: "ACPI\PNP0501\1","Communications Port (COM1)","Ports"`

- `Device 14: "ACPI\PNP0700\4&269b4552&0","@fdc.inf,%fdc_devdesc %;Standard floppy disk controller","fdc"`

- `Device 15: "ACPI\PNP0800\4&269b4552&0","@machine.inf, %*pnp0800.devicedesc%;System speaker","System"`

- `Device 16: "ACPI\PNP0A08\2&daba3ff&1","@machine.inf, %*pnp0a03.devicedesc%;PCI bus","System"`

- `Device 17: "ACPI\PNP0B00\4&269b4552&0","@machine.inf, %*pnp0b00.devicedesc%;System CMOS/real time clock","System"`

- `...`

D   Custom CVA files

Sometimes it is necessary to modify a package description, or CVA, file to meet specific needs. This section defines the more relevant fields used by SSM. CVA files can be modified for existing SoftPaqs or for customized packages. A customized package is a directory in the File Store that contains a CVA file and the package setup files.

There are several key points to remember about CVA files:

- If a customized package is created, remove the **SoftPaq=** line from the CVA file.

- For a non-SoftPaq customized package, all files in the directory and subdirectory of the CVA are transferred as a unit during installation. Do not put customized packages in the top level of the File Store.

- If the CVA file and setup files are packaged with **Packaged for the Web** to create a single package file, the CVA file name must be the same as the Packaged for the Web file name.

CVA syntax

```
[CVA File Information]

CVATimeStamp=<YYYYMMDDhhmmss>

CVASyntaxVersion=2.0A1 or higher

[Software Title]

US=<Software Name/Title in US English>

[US.Software Description]

<General description of the software>

[General]

Version=<Software Version>

Revision=<Software Revision>

Pass=<Software Pass Number>

[DetailFileInformation]

<FileName>=<Path Information>, <Major Version>, <Minor Version>, <Major
Revision>, <Minor Revision>, [OS Target]

[SupportedLanguages]

Languages=<Comma Separated language values as described by the 2 letter
designator>

[Operating Systems]

<OS code>=<Minimum Service Release on which the package works on this OS>
```

```
[System Information]

SysID1=<System ID for the 1st system on which this deliverable operates>

SysName1=<Marketing name of the system>

SysIDn=<System ID for the nth system on which this package operates>

SysNamen=<Marketing name of the system><Comma separated list of models>


[SoftPaq]
```

SoftPaqNumber =<SoftPaq Number>

```
[Devices]

<PnP/PCI Device ID>=<Marketing designation of the device>

<PnP/PCI Device ID>=<Marketing designation of the device>


[US.Enhancements]

Item1=<Description of first enhancement in the SoftPaq>

Item2=<Description of second enhancement in the SoftPaq>

...

ItemN=<Description of Nth enhancement in the SoftPaq>


[Install Execution]

SilentInstall=<Relative path of Install executable><command line
parameters>

DelayAfterInstall=<Delay in seconds up to 10 seconds>
```

Field definitions

[CVA File Information]

```
CVATimeStamp=<YYYYMMDDhhmmss>

CVASyntaxVersion=2.0A1 or higher
```

This section is optional. This field has no effect on SSM's function. Every time the content of the CVA file changes (regardless of whether the actual deliverable changes), the CVA date and time are updated. This allows you to track the official versions of the CVA file.

[Software Title]

```
US=<Software Name/Title in US English>
```

This section is required, and has a maximum of 40 characters. This field identifies the software update.

[US.Software Description]

```
<General description of the software approved by marketing>
```

This section is recommended. It is a brief description of the software, in English.

[General]

```
Version=<Software Version>
```

This section is required. It is a free-form text value that usually follows the form: `<Major version>.<Minor version>.`

```
Revision=<Software Revision>
```

This section is also required. It is the revision of the software. If the revision of the software is irrelevant, then leave this key empty.

```
Pass=<Software Pass Number>
```

This section is required. It identifies the pass/build number of the software. If the pass/build number information is irrelevant, then leave this key empty.

[DetailFileInformation]

```
<FileName>=<Path Information>, <Major Version>, <Minor Version>, <Major
Revision>, <Minor Revision>, [OS Target]
```

This section is required. To allow SSM to determine if a specific package version is installed on a system, the package must have a significant file that has the correct version in its resource header. For example, this file could be the main .EXE, .DLL, or .SYS file. Any time the version and revision of a package change, the value in the resource header of this significant file must also change.

SSM uses this information to determine whether the version of the software update is newer than the version installed on the client computer. Therefore, the [DetailFileInformation] section needs to describe the file that contains the version information in the resource header.

To determine the version of the package programmatically, the major and minor version and the major and minor revision numbers must be entered, separated by commas. This information is the same information contained in the **VS_VERSION_INFO** field located in the [File Version] section of the significant file. The DWORD value for major and minor revision is located in the [File Version] section and is listed as hex numbers in the form `0x0000,0x0000,0x0000,0x0000`. The version information needs to be entered in this format. For example, a significant file with file version 1.2.3.40 would translate to `0x0001,0x0002,0x0003,0x0028`.

The root of the folder is one of the following values:

Value	Description
<WINDIR>	The main Windows folder on the target system (for example, `C:\Windows` or `C:\WinNT`).
<WINSYSDIR>	The Windows System folder on the target system (for example, `C:\Windows\System32`).
<WINDISK>	The drive letter of the disk containing the Windows folder (for example, C:).
<WINSYSDISK>	The drive letter of the disk containing the Windows System folder (for example, C:).
<PROGRAMFILESDIR>	The Program Files folder on the target system (for example, `C:\Program Files`).
<COMMONFILESDIR>	The Common Files folder on the target system (for example, `C:\Program Files\Common Files`).

Value	Description
<FONTDIR>	The folder on the target system where the Windows fonts folder is located (for example, `C:\Windows\Fonts`).
<WINSYSDIRX86>	The Windows System folder in Windows Vista 64-bit systems (for example, `C:\Windows\SysWOW64`).
<WINSYSDISKX86>	The drive letter of the disk containing the Windows System folder on a 64-bit system (for example, C:).
<DRIVERS>	The Windows drivers directory (for example, `C:\Windows\system32\drivers`)
<DRIVERSX86>	The Windows drivers folder for Windows Vista 64-bit systems (for example, `C:\Windows\syswow64\drivers`)
<PROGRAMFILESDIRX86>	The Program Files folder for 32-bit applications installed on a 64-bit target system (for example, `C:\Program Files(x86)\`).
<COMMONFILESDIRX86>	The Common Files folder for 32-bit applications installed on a 64-bit target system (for example, `C:\Program Files(x86)\Common Files`).

For the Windows 7/Vista ATI Graphics Controller drivers, this section would look like this:

```
[DetailFileInformation]
ATIKMDAG.SYS=<WINSYSDIR>\DRIVERS,0x0008,0x0001,0x0001,0x03cd
```

In this case, the actual driver file version would be 8.1.1.973.

A ROM version is designated with the name of the ROM image CAB file or BIN file and the version in one of the following forms:

● `<ROM>.CAB=ROM,<ROM Family>,<Month>,<Day>,<Century>,<Year>`

● `<ROM>.BIN=ROM,<ROM Family>,<Month>,<Day>,<Century>,<Year>`

📝 **NOTE:** The ROM version values are in decimal format, whereas the version values for applications are in "0x__" hex format.

For a ROM, this section would be:

```
[DetailFileInformation]
686T2.CAB=ROM,686T2,10,02,19,98
```

-or-

```
L84_0103.BIN=ROM,L84,03,13,20,14
```

Additionally, SSM supports the definition of multiple [DetailFileInformation] sets based on the operating system target for a software update. This function makes it easier to create single software packages and CVAs that span multiple operating systems within an environment. The following example checks for two distinct binary files based on the operating system that SSM is running on.

```
[DetailFileInformation]
MyFunc.dll=<WINSYSDIR>, 0x5000, 0x1000, 0x0000, 0x0000, WV32EN
MyFunc.dll=<WINSYSDIRX86>, 0x5000, 0x1000, 0x0000, 0x0135, WV64EN
```

Values for the operating system target are defined in .

[Supported Languages]

```
Languages=<Comma-separated language values as described by the two-letter
code>
```

This section is required. This key contains a comma-separated list of languages that the software supports. Each language is specified by a two-character code; for example, US is US English, IT is Italian. This key lets SSM know on which localized operating systems this software can be installed.

If the deliverable supports all languages, then GLOBAL can be specified.

Language abbreviations for use in custom CVA files		
0x0416	BR	Portuguese - Portuguese (Brazilian)
0x0816	BR	Portuguese - Portuguese (Standard) (default to BR)
0x0804	CH	Chinese (PRC) Simplified
0x1004	CH	Chinese (Singapore)
0x0405	CS	Czech - CSY
0x0406	DK	Danish - Danish
0x0c0c	FC	French - Canadian
0x040b	FI	Finnish - Finnish
0x080c	FR	French - Belgian
0x040c	FR	French - French (Standard)
0x100c	FR	French - Swiss
0x0408	GK	Greek - ELL
0x0c07	GR	German - Austrian
0x0407	GR	German - German (Standard)
0x0807	GR	German - Swiss
0x040e	HU	Hungarian - HUN
0x0410	IT	Italian - Italian (Standard)
0x0810	IT	Italian - Swiss
0x0411	JP	Japanese
0x0412	KR	Korean
0x0812	KR	Korean (Johab)
0x0813	NL	Dutch - Belgian (Flemish)
0x0413	NL	Dutch - Dutch (Standard)
0x0414	NO	Norwegian - Norwegian (Bokmal)
0x0814	NO	Norwegian - Norwegian (Nynorsk)
0x0415	PL	Polish - PLK
0x0419	RU	Russian
0x0819	RU	Russian - Moldavia

Language abbreviations for use in custom CVA files		
0x041D	SE	Swedish – Swedish
0x041B	SL	Slovak
0x2c0a	SP	Spanish – Argentina
0x400a	SP	Spanish – Bolivia
0x340a	SP	Spanish – Chile
0x240a	SP	Spanish – Columbia
0x140a	SP	Spanish – Costa Rica
0x1c0a	SP	Spanish – Dominican Republic
0x300a	SP	Spanish – Ecuador
0x100a	SP	Spanish – Guatemala
0x0c0a	SP	Spanish – Modern sort
0x180a	SP	Spanish – Panama
0x3c0a	SP	Spanish – Paraguay
0x280a	SP	Spanish – Peru
0x380a	SP	Spanish – Uruguay
0x200a	SP	Spanish – Venezuela
0x080a	SP	Spanish – Mexican
0x0c0a	SP	Spanish – Spanish (Modern)
0x040a	SP	Spanish – Spanish (Standard/Traditional)
0x041E	TH	Thai
0x041f	TR	Turkish – TRK
0x0404	TW	Chinese (Taiwan)
0x0409	US	English – American
0x0c09	US	English – Australian
0x0809	US	English – British
0x1009	US	English – Canadian
0x1809	US	English – Ireland
0x040f	US	Icelandic – Icelandic
0x040f	US	English – New Zealand
0x0000	US	Default

[Operating Systems]

```
<OS Code>=<Minimum Service Pack Release>
```

This section lists supported operating systems. Each operating system is specified by its code as follows (each code is a unique version of the operating system):

W732 (Windows 7 All 32-bit Editions)

W764 (Windows 7 All 64-bit Editions)

W732SE (Windows 7 Starter 32-bit Edition)

W732HB (Windows 7 Home Basic 32-bit Edition)

W764HB (Windows 7 Home Basic 64-bit Edition)

W732HP (Windows 7 Home Premium 32-bit Edition)

W764HP (Windows 7 Home Premium 64-bit Edition)

W732UL (Windows 7 Ultimate 32-bit Edition)

W764UL (Windows 7 Ultimate 64-bit Edition)

W732PR (Windows 7 Business 32-bit Edition)

W764PR (Windows 7 Business 64-bit Edition)

W732EN (Windows 7 Enterprise 32-bit Edition)

W764EN (Windows 7 Enterprise 64-bit Edition)

W832 (Windows 8 All 32-bit Editions)

W832EN (Windows 8 Enterprise 32-bit Edition)

W832PR (Windows 8 Professional 32-bit Edition)

W832ML (Windows 8 Multi-Language 32-bit Edition)

W832EM (Windows 8 Single Language 32-bit Edition)

W32CH (Windows 8 Country-Specific (China) 32-bit Edition

W864 (Windows 8 All 64-bit Editions)

W864EN (Windows 8 Enterprise 64-bit Edition)

W864PR (Windows 8 Professional 64-bit Edition)

W64ML (Windows 8 Multi-Language 64-bit Edition)

W864EM (Windows 8 Single Language 64-bit Edition)

W864CH (Windows 8 Country-Specific (China) 64-bit Edition

W8.132 (Windows 8.1 All 32-bit Editions)

W8.132EN (Windows 8.1 Enterprise 32-bit Edition)

W8.132PR (Windows 8.1 Professional 32-bit Edition)

W8.132ML (Windows 8.1 Multi-Language 32-bit Edition)

W8.132EM (Windows 8.1 Single-Language (Emerging Markets) 32-bit Edition

W8.132CH (Windows 8.1 Country-Specific (China) 32-bit Edition

W8.164 (Windows 8.1 All 64-bit Editions)

W8.164EN (Windows 8.1 Enterprise 64-bit Edition)

W8.164PR (Windows 8.1 Professional 64-bit Edition)

W8.164ML (Windows 8.1 Multi-Language 64-bit Edition)

W8.164EM (Windows 8.1 Single-Language (Emerging Markets) 64-bit Edition

W8.164CH (Windows 8.1 Country-Specific (China) 64-bit Edition

MULTIPOINT2010 (Windows MultiPoint Server 2010)

MULTIPOINT2011 (Windows MultiPoint Server 2011)

NONE (OS-independent)

Packages must also specify the minimum version service pack release. For example, a driver might only work on Windows Vista Enterprise Service Pack 1 or higher, in which case the CVA file would contain **WV32EN=SP1**. A value must be present after the equal sign for each operating system. There should be one entry only for each operating system.

Acceptable values for `<Minimum Service Pack Release>` are as follows:

OEM (Original OEM release of an operating system)

SP1 (Service Pack 1)

SP2 (Service Pack 2)

SP3 (Service Pack 3)

SP4 (Service Pack 4)

SP5 (Service Pack 5)

If the software is operating system independent, such as ROMPaqs, there should be only one entry in this section:

NONE

[System Information]

```
SysID{AnyUniqueTag}=<System ID>

SysName{AnyUniqueTag}=<Marketing name>
```

This section is required. The `SysID{AnyUniqueTag}` key (where n is an integer greater than zero assigned to each SysID and SysName pair) specifies a system ID for each computer on which software updates are to be deployed. The system ID can be specified in either system ID hexadecimal format or SMBIOS format. The hexadecimal format is **0xXXXX**. For example, the system ID of the HP Compaq dc7900 Convertible Minitower PC system is **0x3032**. The hexadecimal machine ID is listed in the SSM log file.

The syntax for the SMBIOS format is as follows:

```
SysID{AnyUniqueTag}=<SysMan>,<SysProd>,<SysVer>,<BoardMan>,<BoardProd>,<Bo
ardVer>
```

The fields represent the SMBIOS System (record type 1) and SMBIOS Baseboard (record type 2) manufacturer, product, and version fields. Any blank field is considered a wildcard (*) or "don't care." At least one of the six fields must be non-blank. SMBIOS data is available in the SSM log file.

The SysName key specifies the product marketing name, and the associated models for each computer on which software updates are to be deployed.

[SoftPaq]

```
SoftPaqNumber=
```

The SoftPaq number of the software. For custom packages, delete this line.

[Devices]

```
<Device ID>=<Marketing name of the device>
```

This section lists the Device ID and the marketing names of devices with which the software deliverable is associated, if any. In general, this section is required only for device driver software. PCI IDs can be expressed in two-, three-, or four-part names that follow the convention in the Windows Registry (also displayed in the SSM log file).

SSM searches the list of devices currently present in the system as noted by the SSM log file for the specified <Device ID> substring. If the <Device ID> substring is not found in the list of devices on the system, the deliverable is not a possible update for the system.

[US.Enhancements]

```
Item1=<Description of first enhancement in the SoftPaq>
```

This section is optional. These fields contain text descriptions of the CVA file for internal use.

[Install Execution]

```
SilentInstall="<Relative path and Install executable>" <command line
parameters>
```

This section is required. This field contains the relative path, within your package folder structure, of the installation program as well as the command line parameters (if any) required to perform a silent installation of the software while the operating system is running. Enclose the relative path and file name in quotes if a long file name path is used.

If there are questions of how SSM parsed the command line, use the /DEBUG switch, then look in the log file to see the exact command SSM executes.

Examples:

```
SilentInstall=setup.exe -s
```

```
SilentInstall=disk1\setup.exe -s /parameter:"45x32"
```

```
SilentInstall="Long file name\install program.exe" -s /parameter:"45x32"
```

```
DelayAfterInstall=<delay in seconds up to 10 seconds>
```

Optional.

Specifies the delay needed for the installation program to finish its operations. By default, SSM deletes the installation program and the temporary folder it runs from immediately after the installation program returns control to SSM. This might cause problems if the installation is incomplete. This delay forces SSM to wait for a few seconds before it deletes the installation executable. SSM limits the delay to 10 seconds for any value greater than 10 seconds.

Example:

```
DelayAfterInstall=5
```

Sample CVA file

Here is a sample custom CVA file that updates the ROM on an HP EliteBook 850 G1:

```
[CVA File Information]

CVATimeStamp=20140304T120246

CVASyntaxVersion=2.0A1

[Software Title]

US=HP Notebook System BIOS Update

[US.Software Description]

This package creates files that contain an image of the System BIOS (ROM)
for the supported notebook models with a L71 Family ROM. This package is
used to flash the System ROM on support notebook.

[General]

PN=000000-000

Version=01.08

Revision=

Type=ROMPAQ

Category=BIOS

TargetPartition=Primary

SystemMustBeRebooted=1

VendorName=Inventec

VendorVersion=01.08

[SupportedLanguages]

Languages=GLOBAL

Countries=GBL

[DetailFileInformation]

L71_0108.BIN=ROM,L71,02,19,20,14

[Softpaq]

SoftpaqNumber=sp65524

SupersededSoftpaqNumber=sp65364

SoftPaqMD5=3dd34ca82b89fe603ed76b748760af6c

[Operating Systems]

W732EN=OEM
```

```
W764EN=OEM

W732HB=OEM

W764HB=OEM

W732HP=OEM

W764HP=OEM

W732PR=OEM

W764PR=OEM

W732SE=OEM

W732UL=OEM

W764UL=OEM

W832=OEM

W864=OEM

W8.132=OEM

W8.164=OEM

[System Information]

SysId01=0x198F

SysName01=HP EliteBook 840 G1;HP ZBook 14 Mobile Workstation;HP EliteBook
850 G1

SysId02=0x1991

SysName02=HP EliteBook 820 G1

[US.Enhancements]

- Provides support for Elpida VRAM.

- Updates the PXE code to version 1.5.50.

PCR[00]= b951fddec254a8a657f8c04577c404674a7b9c51

[Install Execution]

Install=

SilentInstall="HPBIOSUPDREC.exe" -s -r

[ReturnCode]

3010:SUCCESS:REBOOT=A restart is required to complete the install

1602:CANCEL:NOREBOOT=The install is cannot complete due to a dependency

273:CANCEL:NOREBOOT=Flash did not update because update is same BIOS
version

282:CANCEL:NOREBOOT=Flash did not update because update is an older BIOS
version
```

```
[Private]

Private_SSMCompliant=1

DPB_Compliant=0

Private_ReleaseType=Recommended

Private_ProductType = Notebooks


[CVAGenerationToolDocumentStamp]

Generated by Mobile Release CVA Tool Version 1.0 using Syntax Version 2.0
A1 on 3/4/2014 12:02:46 PM

Copyright (c) 2014 Hewlett-Packard Development Company, L.P.
```

Errors

SSM is not a CVA verification tool and performs only rudimentary checks on the validity of the data in the CVA file. See Error codes on page 29 for detailed information about error messages related to the CVA file. If an error occurs, SSM excludes that software component from the UID and does not deploy it. SSM continues processing the next CVA file. Use the `/debug` command line option to record potential CVA file errors to the file error_log.txt while building the database:

`SSM /am_bld_db /debug`

See also `%TEMP%\%computername%.ssmtrace` for helpful debug information.

E   Error codes

Locating error messages

If errors occur when running SSM on the client computer, the log file contains a section called [Errors]. If there are no errors, this section does not exist in the log file.

Administrative Mode errors

To operate in Update Mode, the ssm.cab file must be present in the File Store. After the administrator selects the File Store folder, the wizard checks for the existence of the ssm.cab file under the File Store folder. If the ssm.cab file does not exist in the File Store folder, it searches for the ssm.cab file and, if found, displays a list of the ssm.cab files found. Select the CAB file you want to copy to the File Store, then click **Next**.

If the search for the ssm.cab file fails, SSM displays a message that the `ssm.cab` file is required in the File Store folder and identifies the File Store folder as `D:\SSMFILE STORE`. Click **Finish** to terminate and reinstall SSM.

If SSM is executed on an unsupported operating system (Windows 3.x or Windows NT® Workstation 3.5X), SSM displays an error message and terminates.

Use the `/debug` command line option to record potential CVA file errors to the file error_log.txt while building the database:

```
SSM /am_bld_db /debug
```

Update Mode errors

In Update Mode, SSM might display a message if it encounters a problem with the database access objects (DAO). If this happens, the SSM log file indicates that there are no possible updates.

All other errors are placed in the log file and described in the table below.

Error message	Description
SSM is for use on Windows 95/98 and Windows NT 4.0 or greater. – SSM Exit	You have tried to execute SSM on an unsupported operating system (Windows 3. X or Windows NT 3.5 X).
Failed to initialize client information. SSM was unable to determine asset information on the client system. Device detection will fail, as well.	Detection failed. SSM was unable to obtain device information on the client. This error indicates a driver interaction issue.
User does not have correct privileges to update system – SSM Exit.	SSM was unable to elevate its privilege to perform updates. The elevated privilege is achieved through the trusted client portions of the administrative wizard.
SSM support CAB component not found – SSM Exit.	SSM was unable to find the CAB support file and aborted execution. This error does not occur if SSM is installed locally through the `/install` command line switch.
Unable to extract cab file.	The CAB support file is corrupted and cannot be expanded. This error does not occur if SSM is installed locally through the `/install` command line switch.

Error message	Description
Cannot set the setup password with error code <value>.	The value in this error message indicates the reason the setup password could not be set. The lower byte of <value> is the ROM return code. (80=Invalid password because it was too long, 86=Command not supported.)
ROM Flash Error code <value> <symbol_name>	SSM was unable to flash the system ROM. The error code indicates the reason for the failure. This is sometimes the result of an incorrect password. (F6 ERROR_LOCKING_MEMORY).
<Update Package Title> has no installer.	The specified update package has a blank `SilentInstall=` entry in its CVA file.
Unable to open <SoftPaq Filename>	SSM expands SoftPaq files before executing the installer for the update. If that expansion fails, this message is entered in the log file. If this is a custom deliverable, ensure that the `SoftPaq=` line in the CVA file is removed.
Unable to expand ROMPaq <SoftPaq Filename>	SSM is expecting a ROM BIOS SoftPaq, but was unable to expand the files.
<Update Package Title>: Failed to terminate in a timely manner.	SSM waits 15 minutes before terminating an update package's installer. If the installer is terminated, this message is entered into the log file.
<Update Package Title>: Returned exit code <value>	This message is placed in the log when an update issues an exit code that is non-zero. This could represent a failed install, because many updates return non-zero exit codes. The exit code is update specific; see the documentation provided with the update to determine what the exit code means.
<Update Package Title>: Failed to spawn command line (<update package installer command line with arguments>) with error (<value>)	This message indicates that the installer for the update failed to execute properly and might be corrupted.
Can't update the replicated setup settings.	The setup password does not match. The replicated setup/security settings module returns this message when the system attempts to modify configuration settings on a client system using an incorrect password.
F6 ERROR_LOCKING MEMORY	The setup password is not correct.
Evo 600c (686DF) BIOS must be updated to 10/17/2001 before upgrading to newer BIOS. Download SP21459.EXE and SP21459.TXT.	Self-explanatory.
Password is set, but no password provided in SSM.BIN.	An attempt to flash the ROM BIOS failed.
RMSET File not found: <Filename>	The file specified with the RMSET option was not found.
Failed to initialize client information.	The required device driver could not be loaded and system information could not be obtained. This can occur if there is a driver version conflict or SSM does not have administrative rights.

Internal errors

The error codes listed below are internal errors. If other error codes are observed in the error logs, it is probably an error with the software installer or a CVA error.

Error code	Description
SSM	

Error code	Description
0x51	Header not found
0x52	Command not supported
0x53	Image Offset incorrect
0x54	Image Length incorrect
0x55	Battery in calibration mode
0x56	Battery too low
0x57	Flash Update failure
0x58	Earlier version of BIOS not allowed to be flashed
0x80	Invalid Password
0x81	Password not protected
0x82	ROM not protected
0x83	Invalid security tag value
0x84	Invalid security tag length
0x85	Invalid buffer size
0x86	Call not supported
0x87	Security set not allowed
0x88	Security Value remove not allowed
0xF0	Flash failed
0xF1	Invalid IOCTL parameter
0xF2	DeviceIO busy
0xF3	Function not supported
0xF4	Flash in progress
0xF5	Invalid ROM image
0xF6	Error locking memory
0xF7	Error allocating work buffers
0xF8	Lock unlock failed
0xF9	Driver access failed
0xFA	IOCTL failed
0xFB	Unknown ROM size

CVA errors

Use `/DEBUG` in Administrative Mode, then check the file error_log.txt to see these errors.

CVA files are formatted like the INI files found in Windows. SSM uses standard Windows calls to read the data from these files. Syntax errors in a heading or a tag field result in that field being ignored. A syntax error in

the data portion of the tag might result in improper execution of the deliverable or a misspelled title in the log file.

A syntax error in the **Language** field or the **Operating System** fields results in the whole CVA file being rejected.

Error message	CVA section	Description
No key file information.	DetailFileInformation	The entire section was omitted. DetailFileInformation is required for the package to be considered SSM compliant.
Key filename is invalid.	DetailFileInformation	Key file name is not specified or is not valid. At least one key file name must be specified.
Key file path is invalid.	DetailFileInformation	Path for the key file is not provided or is not valid.
Major version information is invalid.	DetailFileInformation	The major file version data was omitted.
Minor version information is invalid.	DetailFileInformation	The minor file version data was omitted.
Major revision information is invalid.	DetailFileInformation	The major file revision data was omitted.
Minor revision information is invalid.	DetailFileInformation	The minor file revision data was omitted.
Failed to commit transaction on record add.		SSM was unable to add the data from the current CVA file to the UID. This is a catastrophic error; you must delete the UID and rebuild it.
Database transaction error on SQL: <Error Text>		SSM was unable to make the appropriate system calls to talk to the UID. This is a catastrophic error; you must delete the UID and rebuild it. This also indicates an operating system-level interaction issue.

F   Troubleshooting

Error log contains errors, but drivers installed properly.

Cause	Solution
Silent installation programs for some drivers return a non-zero return code upon completion. SSM expects a return code of zero.	N/A

Controller is not detected by SSM.

Cause	Solution
SSM only detects PCI or plug-and-play controllers.	N/A

SSM ran, but driver was not updated.

Cause	Solution
Driver was uninstalled before running SSM.	Uninstalling the driver might not have deleted all of the driver files from the hard drive. When SSM tries to determine which version is installed, the remaining files might indicate that the driver is still installed. Look for the file name in the [DetailFileInformation] section of the CVA file for the driver update. Delete the driver file from the hard disk.

Need to uninstall SSM. Delete SSM.exe, ssm.cab, and the File Store.

Cause	Solution
SSM components are only installed on client systems when the `/install` switch is used. The components are stored in a folder named `SSM` on the same drive as the operating system.	To remove the client components, delete the `SSM` folder and all files within it.

Deliverable does not appear as a possible update.

Cause	Solution
CVA file is not SSM compliant.	Rebuild the database with the `/debug` switch: `SSM C:\<File Store> /debug /am_bld_db`, then examine the file `%temp%\%computername%.ssmtrace` for clues. SSM will reject CVA files that do not have required fields.
One or more of the CVA fields do not match the current system.	The list of possible updates is determined by matching the system ID, operating system, language, and device IDs. Run SSM with the `/debug` switch and examine the ssmtrace, CVA, and log files for clues to determine which field does not match.

Nothing was installed.

Cause	Solution
One or more items in the CVA file did not contain a value that matched the system.	Use the extended debug feature to create a log file. Under the Match Type column, the items that do not contain a value that matches the system are marked with **NOT MATCH**. Correct these items in the CVA file.