About this guide

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

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

Table 1-1 Minimum hardware and software

<table>
<thead>
<tr>
<th>File Store server</th>
<th>Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows® XP Professional, Windows Vista®, Windows 7,</td>
<td>Windows 7, Windows 8, or Windows 8.1</td>
</tr>
<tr>
<td>Windows 8, or Windows 8.1</td>
<td></td>
</tr>
<tr>
<td>Internet Explorer 4.01 or later</td>
<td>Internet Explorer 4.01 or later</td>
</tr>
</tbody>
</table>

*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 Command line usage on page 10 for a complete list of command-line options.

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>
<thead>
<tr>
<th>Table 1-2 SSM modules and functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module</strong></td>
</tr>
<tr>
<td>SSM.exe</td>
</tr>
<tr>
<td>ssm.cab</td>
</tr>
<tr>
<td>ssm.bin</td>
</tr>
<tr>
<td>ssmcva.mdb</td>
</tr>
<tr>
<td>SSM log file</td>
</tr>
<tr>
<td>SSM trace file</td>
</tr>
</tbody>
</table>
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 UID performance considerations on page 5 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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/accept or /a</td>
<td>U</td>
<td>Accepts the license agreement. This allows SSM to be used in Update Mode without the ssm.bin file.</td>
</tr>
<tr>
<td>/am_bld_db</td>
<td>A</td>
<td>Silently rebuilds the UID in Administrative Mode.</td>
</tr>
<tr>
<td>/cancel: &lt;Numeric Value&gt;[,modem]</td>
<td>U</td>
<td>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.</td>
</tr>
<tr>
<td>/CurrentSetupPasswordFile:&quot;&lt;filename&gt;&quot;—or—/cpwdfile:&quot;&lt;filename&gt;&quot;</td>
<td>U</td>
<td>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.</td>
</tr>
<tr>
<td>/debug or /d</td>
<td>U, A</td>
<td>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.</td>
</tr>
<tr>
<td>/eDebug, /ExtendedDebug, or /extended_debug</td>
<td>U</td>
<td>Provides greater details when an error is encountered. Extended debug is designed to be run in combination with a single SoftPaq.</td>
</tr>
<tr>
<td>/force or /f</td>
<td>U</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Parameter Modes Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/help</td>
<td>U</td>
<td>Displays an abbreviated list of command line parameters short descriptions of their function. (Available in SSM 3.0.1.1 or later.)</td>
</tr>
<tr>
<td>/install</td>
<td>U</td>
<td>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.</td>
</tr>
<tr>
<td>/log: &lt;path&gt;</td>
<td>U</td>
<td>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.</td>
</tr>
<tr>
<td>/nocvaupdate or /nu</td>
<td>A</td>
<td>Tells SSM not to check the HP FTP site for CVA updates. Used with /am_bld_db.</td>
</tr>
<tr>
<td>/nodialog or /s</td>
<td>U</td>
<td>Suppresses the progress dialog and overrides the /title parameter.</td>
</tr>
<tr>
<td>/nolog</td>
<td>U</td>
<td>Suppresses creation of the log file and overrides the /log parameter.</td>
</tr>
<tr>
<td>/noreboot</td>
<td>U</td>
<td>Suppresses a reboot of the machine by SSM. A reboot can be requested by an update through CVA or be requested Administrative Mode. Overrides the /reboot parameter.</td>
</tr>
<tr>
<td>/reboot or /rb</td>
<td>U</td>
<td>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.</td>
</tr>
<tr>
<td>/report or /r</td>
<td>U</td>
<td>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 execute any update packages. <strong>NOTE:</strong> SSM does not report possible /repset BIOS changes.</td>
</tr>
<tr>
<td>/runonce</td>
<td>U</td>
<td>Causes SSM to clear itself from the RunOnce registry key. command-line switch is used as part of a preinstallation package.</td>
</tr>
<tr>
<td>/sort:ascending or /sort:descending</td>
<td>U</td>
<td>/sort:ascending installs starting with the lowest numbered SoftPaqs first; /sort:descending does the opposite. Custom SoftPaqs are always last.</td>
</tr>
<tr>
<td>/timeout:&lt;Numeric Value&gt;</td>
<td>U</td>
<td>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. <strong>NOTE:</strong> A value of 0 means wait indefinitely on the package and is not suggested.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>/title:“&lt;title&gt;” or /ti:“&lt;title&gt;”</td>
<td>U</td>
<td>Specifies the title for the slider bar dialog (must be specified in quotes).</td>
</tr>
<tr>
<td>/useinstallpath:&lt;path&gt;</td>
<td>U</td>
<td>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”.</td>
</tr>
</tbody>
</table>
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
```
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\l", "@tpm.inf,%std_desc%;Trusted Platform Module 1.2", "SecurityDevices"

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

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

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

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

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

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

Device 13: "ACPI\PNP0501\l", "Communications Port (COM1)", "Ports"

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

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

Device 16: "ACPI\PNP0A03\2&daba3ff&l", "@machine.inf,%*pnp0a03.devicedesc%;PCI bus", "System"

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

...
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=<YYYYMMDDhhmms>
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.*

**[Private]**

Private_SSMCompliant = Value Type: 1 or 0

*This section is required. Indicator of whether or not SoftPaq is compatible with SSM*

MS_Store_app = Value Type: 1 or 0

*This section is required. Indicator of whether or not Softpaq contains a Windows Store App*

**[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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;WINDIR&gt;</td>
<td>The main Windows folder on the target system (for example, C: \Windows or C: \WinNT).</td>
</tr>
<tr>
<td>&lt;WINSYSDIR&gt;</td>
<td>The Windows System folder on the target system (for example, C: \Windows \System32).</td>
</tr>
<tr>
<td>&lt;WINDISK&gt;</td>
<td>The drive letter of the disk containing the Windows folder (for example, C).</td>
</tr>
<tr>
<td>&lt;WINSYSDISK&gt;</td>
<td>The drive letter of the disk containing the Windows System folder (for example, C: \Windows \System32).</td>
</tr>
<tr>
<td>Environment Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>&lt;PROGRAMFILES_DIR&gt;</code></td>
<td>The Program Files folder on the target system (for example, C:\Program Files).</td>
</tr>
<tr>
<td><code>&lt;COMMONFILES_DIR&gt;</code></td>
<td>The Common Files folder on the target system (for example, C:\Program Files\Common Files).</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>&lt;FONTDIR&gt;</code></td>
<td>The folder on the target system where the Windows fonts folder is located (for example, C:\Windows\Fonts).</td>
</tr>
<tr>
<td><code>&lt;WINSYSDIRX86&gt;</code></td>
<td>The Windows System folder in Windows Vista 64-bit systems (for example, C:\Windows\SysWOW64).</td>
</tr>
<tr>
<td><code>&lt;WINSYSDISKX86&gt;</code></td>
<td>The drive letter of the disk containing the Windows System folder on a 64-bit system (for example, C:).</td>
</tr>
<tr>
<td><code>&lt;DRIVERS&gt;</code></td>
<td>The Windows drivers directory (for example, C:\Windows\system32\drivers).</td>
</tr>
<tr>
<td><code>&lt;DRIVERSX86&gt;</code></td>
<td>The Windows drivers folder for Windows Vista 64-bit systems (for example, C:\Windows\syswow64\drivers).</td>
</tr>
<tr>
<td><code>&lt;PROGRAMFILESDIRX86&gt;</code></td>
<td>The Program Files folder for 32-bit applications installed on a 64-bit target system (for example, C:\Program Files(x86)).</td>
</tr>
<tr>
<td><code>&lt;COMMONFILESDIRX86&gt;</code></td>
<td>The Common Files folder for 32-bit applications installed on a 64-bit target system (for example, C:\Program Files(x86)\Common Files).</td>
</tr>
</tbody>
</table>

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 [Operating Systems] on page 22.
[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

<table>
<thead>
<tr>
<th>Language Code</th>
<th>Language Abbreviation</th>
<th>Language Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0416</td>
<td>BR</td>
<td>Portuguese - Portuguese (Brazilian)</td>
</tr>
<tr>
<td>0x0816</td>
<td>BR</td>
<td>Portuguese - Portuguese (Standard) (default to BR)</td>
</tr>
<tr>
<td>0x0804</td>
<td>CH</td>
<td>Chinese (PRC) Simplified</td>
</tr>
<tr>
<td>0x1004</td>
<td>CH</td>
<td>Chinese (Singapore)</td>
</tr>
<tr>
<td>0x0405</td>
<td>CS</td>
<td>Czech - CSY</td>
</tr>
<tr>
<td>0x0406</td>
<td>DK</td>
<td>Danish - Danish</td>
</tr>
<tr>
<td>0x0c0c</td>
<td>FC</td>
<td>French - Canadian</td>
</tr>
<tr>
<td>0x040b</td>
<td>FI</td>
<td>Finnish - Finnish</td>
</tr>
<tr>
<td>0x080c</td>
<td>FR</td>
<td>French - Belgian</td>
</tr>
<tr>
<td>0x040c</td>
<td>FR</td>
<td>French - French (Standard)</td>
</tr>
<tr>
<td>0x100c</td>
<td>FR</td>
<td>French - Swiss</td>
</tr>
<tr>
<td>0x0408</td>
<td>GK</td>
<td>Greek - ELL</td>
</tr>
<tr>
<td>0x0c07</td>
<td>GR</td>
<td>German - Austrian</td>
</tr>
<tr>
<td>0x0407</td>
<td>GR</td>
<td>German - German (Standard)</td>
</tr>
<tr>
<td>0x0807</td>
<td>GR</td>
<td>German - Swiss</td>
</tr>
<tr>
<td>0x040e</td>
<td>HU</td>
<td>Hungarian - HUN</td>
</tr>
<tr>
<td>0x0410</td>
<td>IT</td>
<td>Italian - Italian (Standard)</td>
</tr>
<tr>
<td>0x0810</td>
<td>IT</td>
<td>Italian - Swiss</td>
</tr>
<tr>
<td>0x0411</td>
<td>JP</td>
<td>Japanese</td>
</tr>
<tr>
<td>0x0412</td>
<td>KR</td>
<td>Korean</td>
</tr>
<tr>
<td>0x0812</td>
<td>KR</td>
<td>Korean (Johab)</td>
</tr>
<tr>
<td>0x0813</td>
<td>NL</td>
<td>Dutch - Belgian (Flemish)</td>
</tr>
<tr>
<td>0x0413</td>
<td>NL</td>
<td>Dutch - Dutch (Standard)</td>
</tr>
<tr>
<td>0x0414</td>
<td>NO</td>
<td>Norwegian - Norwegian (Bokmal)</td>
</tr>
<tr>
<td>0x0814</td>
<td>NO</td>
<td>Norwegian - Norwegian (Hynorsk)</td>
</tr>
<tr>
<td>0x0415</td>
<td>PL</td>
<td>Polish - PLK</td>
</tr>
<tr>
<td>0x0419</td>
<td>RU</td>
<td>Russian</td>
</tr>
<tr>
<td>0x0819</td>
<td>RU</td>
<td>Russian - Moldavia</td>
</tr>
<tr>
<td>Language Abbreviations for Use in Custom CVA Files</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OS Code</strong></td>
<td><strong>Language Abbreviation</strong></td>
<td><strong>Language</strong></td>
</tr>
<tr>
<td>0x041D</td>
<td>SE</td>
<td>Swedish - Swedish</td>
</tr>
<tr>
<td>0x0418</td>
<td>SL</td>
<td>Slovak</td>
</tr>
<tr>
<td>0x2c0a</td>
<td>SP</td>
<td>Spanish - Argentina</td>
</tr>
<tr>
<td>0x400a</td>
<td>SP</td>
<td>Spanish - Bolivia</td>
</tr>
<tr>
<td>0x340a</td>
<td>SP</td>
<td>Spanish - Chile</td>
</tr>
<tr>
<td>0x240a</td>
<td>SP</td>
<td>Spanish - Columbia</td>
</tr>
<tr>
<td>0x140a</td>
<td>SP</td>
<td>Spanish - Costa Rica</td>
</tr>
<tr>
<td>0x1c0a</td>
<td>SP</td>
<td>Spanish - Dominican Republic</td>
</tr>
<tr>
<td>0x300a</td>
<td>SP</td>
<td>Spanish - Ecuador</td>
</tr>
<tr>
<td>0x100a</td>
<td>SP</td>
<td>Spanish - Guatemala</td>
</tr>
<tr>
<td>0x0c0a</td>
<td>SP</td>
<td>Spanish - Modern sort</td>
</tr>
<tr>
<td>0x180a</td>
<td>SP</td>
<td>Spanish - Panama</td>
</tr>
<tr>
<td>0x3c0a</td>
<td>SP</td>
<td>Spanish - Paraguay</td>
</tr>
<tr>
<td>0x280a</td>
<td>SP</td>
<td>Spanish - Peru</td>
</tr>
<tr>
<td>0x380a</td>
<td>SP</td>
<td>Spanish - Uruguay</td>
</tr>
<tr>
<td>0x200a</td>
<td>SP</td>
<td>Spanish - Venezuela</td>
</tr>
<tr>
<td>0x080a</td>
<td>SP</td>
<td>Spanish - Mexican</td>
</tr>
<tr>
<td>0x0c0a</td>
<td>SP</td>
<td>Spanish - Spanish (Modern)</td>
</tr>
<tr>
<td>0x040a</td>
<td>SP</td>
<td>Spanish - Spanish (Standard/Traditional)</td>
</tr>
<tr>
<td>0x041e</td>
<td>TH</td>
<td>Thai</td>
</tr>
<tr>
<td>0x041f</td>
<td>TR</td>
<td>Turkish - TRK</td>
</tr>
<tr>
<td>0x0404</td>
<td>TW</td>
<td>Chinese (Taiwan)</td>
</tr>
<tr>
<td>0x0409</td>
<td>US</td>
<td>English - American</td>
</tr>
<tr>
<td>0x0c09</td>
<td>US</td>
<td>English - Australian</td>
</tr>
<tr>
<td>0x0809</td>
<td>US</td>
<td>English - British</td>
</tr>
<tr>
<td>0x1009</td>
<td>US</td>
<td>English - Canadian</td>
</tr>
<tr>
<td>0x1809</td>
<td>US</td>
<td>English - Ireland</td>
</tr>
<tr>
<td>0x040f</td>
<td>US</td>
<td>Icelandic - Icelandic</td>
</tr>
<tr>
<td>0x040f</td>
<td>US</td>
<td>English - New Zealand</td>
</tr>
<tr>
<td>0x0000</td>
<td>US</td>
<td>Default</td>
</tr>
</tbody>
</table>

[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)
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)
W8.132CH (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)
W8.164CH (Windows 8.1 Country-Specific (China) 64-bit Edition)
WT32 (Windows 10 All 32-bit Editions)
WT32_1507 (TH1 – Threshold 1 – Released July 2015 – The original version of Windows 10 32-bit)
WT32_1511 (TH2 – Threshold 2 – Released November 2015 – The November update version of Windows 10 32-bit)
WT32_1607 (RS1 – Redstone 1 – Released July 2016 – The Anniversary Update of Windows 10 32-bit)
WT32_1703 (RS2 – Redstone 2 – Released March 2017 – Windows 10, version 1703 32-bit)
WT32_1709 (RS3 – Redstone 3 – Windows 10, version 1709 32-bit)
WT32_1803 (RS4 – Redstone 4 – Windows 10, version 1803 32-bit)
WT32_1809 (RS5 – Redstone 5 – Windows 10, version 1809 32-bit)
WT32_1903 (19H1 – Windows 10, version 1903 32-bit)
WT64 (Windows 10 All 64-bit Editions)
WT64_1507 (TH1 – Threshold 1 – Released July 2015 – The original version of Windows 10 64-bit)
WT64_1511 (TH2 – Threshold 2 – Released November 2015 – The November update version of Windows 10 64-bit)
WT64_1607 (RS1 – Redstone 1 – Released July 2016 – The Anniversary Update of Windows 10 64-bit)
WT64_1703 (RS2 – Redstone 2 – Released March 2017 – Windows 10, version 1703)
WT64_1709 (RS3 – Redstone 3 – Windows10, version 1709 64-bit)
WT64_1803 (RS4 – Redstone 4 – Windows10, version 1803 64-bit)
WT64_1809 (RS5 – Redstone 5 – Windows 10, version 1809 64-bit)
WT64_1903 (19H1 – Windows 10, version 1903 64-bit)
WTIOT32 (Windows 10 IOT Enterprise 32-bit Edition)
WTIOT64 (Windows 10 IOT Enterprise 64-bit Edition)
WTIOT64_1507 (TH1 – Threshold 1 – Released July 2015 – The original version of Windows 10 IOT 64-bit)
WTIOT64_1607 (RS1 – Redstone 1 – Released July 2016 – Windows 10 IoT Enterprise 2016 LTSC 64-bit)
WTIOT64_1809 (RS5 – Redstone 5 - Windows 10 IoT Enterprise 2019 LTSC 64-bit)
POSREADY732 (Windows Embedded POSReady 7 32-bit)
POSREADY764 (Windows Embedded POSReady 7 64-bit)
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>,<BoardVer>

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

[Store Package Info]

WacomTechnologyCorp.HPPenSettings_7.5.47.0_x64__ss941bf8mfs8a=
WacomTechnologyCorp.HPPenSettings_7.5.47.0_neutral__ss941bf8mfs8a=
WacomTechnologyCorp.HPPenSettings_7.5.47.0_x86__ss941bf8mfs8a=
WacomTechnologyCorp.HPPenSettings_7.5.47.0_neutral__ss941bf8mfs8a=

App package full name is PackageFullName displayed when querying for the app in Windows. The app package full name follows this format:

<Name>_.<Version>_.<Architecture>_.<ResourceId>_.<PublisherId>

When running the PowerShell command Get-AppxPackage, the app properties will be displayed as the example below. Note the content of PackageFullName. The exact value will be input into the key of the app entry in this section.

Name: WacomTechnologyCorp.HPPenSettings
Publisher: CN=E371CB8C-3AF1-4B3A-ACA6-FCB4DF56BD86
Architecture: x64
ResourceId: 
Version: 7.5.47.0
PackageFullName: WacomTechnologyCorp.HPPenSettings_7.5.47.0_x64__ss941bf8mfs8a
InstalledLocation: C:\Program Files\WindowsApps\WacomTechnologyCorp.HPPenSettings_7.5.47.0_x64__ss941bf8mfs8a
IsFramework: False
PackageFamilyName: WacomTechnologyCorp.HPPenSettings_ss941bf8mfs8a
PublisherId: ss941bf8mfs8a
IsResourcePackage: False
IsBundle: False
IsDevelopmentMode: False
IsPartiallyStaged: False
SignatureKind: Store
Status: OK

The app bundle full name can only be retrieved programmatically. Tools can be used to extract the app
package and bundle information from the content of a SoftPaq.
*For tools/solutions consuming the store package information, the information can be extracted into key information: name, version, architecture, resource ID, publisher ID to detect if an app should be recommended-installed to a system. Similar WMI queries like the one can be used to check if the same version or later has been provisioned-installed on the system:

Select * from Win32_InstalledStoreProgram where ProgramId like '
'<Name>_ % <PublisherId>' and Version >= '<Version>'

Select * from Win32_InstalledStoreProgram where ProgramId like
'WacomTechnologyCorp.HPPenSetting_%_ss941bf8mfs8a' and Version >= '7.5.47.0'
Sample CVA file

Here is a sample custom CVA file showing an 840 G5 BIOS.

[CVA File Information]
CVATimeStamp=20190814T102508
CVASyntaxVersion=2.1A6

[Software Title]
US=HP Firmware Pack (Q78)

[US.Software Description]
This package creates files that contain an image of the System BIOS (ROM) for the supported notebook models with a Q78 Family ROM. This package is used to flash the System ROM on support notebook.

[General]
PN=000000-000
Version=01.08.01
Revision=A
Pass=1
Type=ROMPAQ
Category=BIOS
TargetPartition=MFG DIAGS
SystemMustBeRebooted=1
VendorName=HP Inc.
VendorVersion=01.08.01

[SupportedLanguages]
Languages=GLOBAL
Countries=GBL

[ProfessionalInnovations]
HPPI=NO
LearnMore=

[DetailFileInformation]
Q78_010801.bin=ROM,Q78,07,18,20,19

[Softpaq]
SoftpaqNumber=SP98706
SupersededSoftpaqNumber=SP95880
SoftPaqMD5=2ddab4b7882d85690d53a4da5809ee93

[System Information]
SysId01=0x83B2
SysName01=HP ELITEBOOK 840 G5,HP ELITEBOOK 846 G5,HP ELITEBOOK 850 G5,HP ZBOOK 14U G5 MOBILE,HP ZBOOK 15U G5 MOBILE
SysId02=0x83B2
SysName02=HP ELITEBOOK 840 G5 HEALTHCARE EDITION,HP ELITEBOOK 846 G5 HEALTHCARE
EDITION
SysId03=0x83B3
SysName03=HP ELITEBOOK 830 G5,HP ELITEBOOK 836 G5

[Operating Systems]
WT64=OEM

[US.Enhancements]
- Adds an option in the F10 setup to download and install BIOS updates automatically without prompts.
- Adds the Native OS Firmware Update Service feature to the F10 setup interface to enable and disable firmware updates using the Windows Update service.
- Updates the Cypress PD firmware to v7.9.
- Provides separate controls for Intel Management Engine (ME) and Intel Active Management (AMT).
- Updates the Intel silicon reference code to v3.6.7.1.

[Install Execution]
Install="1. Download the file by clicking the Download or Obtain Software button and saving the file to a folder on your hard drive (make a note of the folder where the downloaded file is saved)."
SilentInstall="HpFirmwareUpdRec.exe" -r -b -s

[ReturnCode]
3010:SUCCESS:REBOOT=A restart is required to complete the install.
1602:FAILURE:NOREBOOT=Flash failed due to dependency.
282:CANCEL:NOREBOOT=Flash failed due to update is the same version as the current BIOS.
273:FAILURE:NOREBOOT=Flash failed due to update is older than the current BIOS and -a override flag was not used.
259:FAILURE:NOREBOOT=Returned due to internal failure
3011:CANCEL:NOREBOOT=Rollback successful and previous flash preparation removed. A reboot is required.
290:CANCEL:NOREBOOT=Bitlocker is enabled and utility is running in silent mode.
128:CANCEL:NOREBOOT=Setup password does not match, or unable to read password file.

[Security]
EnterME_PCR00=

FullME_PCR00=143C0A9D61C736975F34B09BCD35E37EDC5C3183DACDAA72C0A786FCBCC2354

[Private]
Private_SSMCompliant=1
DPB_Compliant=0
Private_ReleaseType=Critical
MS_Store_App=0
Private_ProductType = Notebooks
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.

<table>
<thead>
<tr>
<th>Error message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSM is for use on Windows 95/98 and Windows NT 4.0 or greater. - SSM Exit</td>
<td>You have tried to execute SSM on an unsupported operating system (Windows 3.x or Windows NT 3.5 X).</td>
</tr>
<tr>
<td>Failed to initialize client information. SSM was unable to determine asset information on the client system. Device detection will fail, as well.</td>
<td>Detection failed. SSM was unable to obtain device information on the client. This error indicates a driver interaction issue.</td>
</tr>
<tr>
<td>User does not have correct privileges to update system - SSM Exit.</td>
<td>SSM was unable to elevate its privilege to perform updates. The elevated privilege is achieved through the trusted client portions of the administrative wizard.</td>
</tr>
<tr>
<td>SSM support CAB component not found - SSM Exit.</td>
<td>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.</td>
</tr>
<tr>
<td>Unable to extract cab file.</td>
<td>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.</td>
</tr>
<tr>
<td>Error message</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Cannot set the setup password with error code <code>&lt;value&gt;</code>.</td>
<td>The value in this error message indicates the reason the setup password could not be set. The lower byte of <code>&lt;value&gt;</code> is the ROM return code. (80=Invalid password because it was too long, 86=Command not supported.)</td>
</tr>
<tr>
<td>ROM Flash Error code <code>&lt;value&gt;</code> <code>&lt;symbol_name&gt;</code></td>
<td>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).</td>
</tr>
<tr>
<td><code>&lt;Update Package Title&gt;</code> has no installer.</td>
<td>The specified update package has a blank <code>SilentInstall=</code> entry in its CVA file.</td>
</tr>
<tr>
<td>Unable to open <code>&lt;SoftPaq Filename&gt;</code></td>
<td>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 <code>SoftPaq=</code> line in the CVA file is removed.</td>
</tr>
<tr>
<td>Unable to expand ROMPaq <code>&lt;SoftPaq Filename&gt;</code></td>
<td>SSM is expecting a ROM BIOS SoftPaq, but was unable to expand the files.</td>
</tr>
<tr>
<td><code>&lt;Update Package Title&gt;</code>: Failed to terminate in a timely manner.</td>
<td>SSM waits 15 minutes before terminating an update package's installer. If the installer is terminated, this message is entered into the log file.</td>
</tr>
<tr>
<td><code>&lt;Update Package Title&gt;</code>: Returned exit code <code>&lt;value&gt;</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>&lt;Update Package Title&gt;</code>: Failed to spawn command line <code>(update package installer command line with arguments)</code> with error <code>(value)</code></td>
<td>This message indicates that the installer for the update failed to execute properly and might be corrupted.</td>
</tr>
<tr>
<td>Can't update the replicated setup settings.</td>
<td>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.</td>
</tr>
<tr>
<td>F6 ERROR_LOCKING MEMORY</td>
<td>The setup password is not correct.</td>
</tr>
<tr>
<td>Evo 600c (686DF) BIOS must be updated to 10/17/2001 before upgrading to newer BIOS. Download SP21459.EXE and SP21459.TXT.</td>
<td>Self-explanatory.</td>
</tr>
<tr>
<td>Password is set, but no password provided in SSM.BIN.</td>
<td>An attempt to flash the ROM BIOS failed.</td>
</tr>
<tr>
<td>RMSET File not found: <code>&lt;Filename&gt;</code></td>
<td>The file specified with the RMSET option was not found.</td>
</tr>
<tr>
<td>Failed to initialize client information.</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
</table>
## 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.

<table>
<thead>
<tr>
<th>Error message</th>
<th>CVA section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No key file information.</td>
<td>DetailFileInformation</td>
<td>The entire section was omitted. DetailFileInformation is required for the package to be considered SSM compliant.</td>
</tr>
<tr>
<td>Key filename is invalid.</td>
<td>DetailFileInformation</td>
<td>Key file name is not specified or is not valid. At least one key file name must be specified.</td>
</tr>
<tr>
<td>Key file path is invalid.</td>
<td>DetailFileInformation</td>
<td>Path for the key file is not provided or is not valid.</td>
</tr>
<tr>
<td>Major version information is invalid.</td>
<td>DetailFileInformation</td>
<td>The major file version data was omitted.</td>
</tr>
<tr>
<td>Minor version information is invalid.</td>
<td>DetailFileInformation</td>
<td>The minor file version data was omitted.</td>
</tr>
<tr>
<td>Major revision information is invalid.</td>
<td>DetailFileInformation</td>
<td>The major file revision data was omitted.</td>
</tr>
<tr>
<td>Minor revision information is invalid.</td>
<td>DetailFileInformation</td>
<td>The minor file revision data was omitted.</td>
</tr>
<tr>
<td>Failed to commit transaction on record add.</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Database transaction error on SQL: &lt;Error Text&gt;</td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>
F Troubleshooting

Error log contains errors, but drivers installed properly.

<table>
<thead>
<tr>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silent installation programs for some drivers return a non-zero return code upon completion. SSM expects a return code of zero.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Controller is not detected by SSM.

<table>
<thead>
<tr>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSM only detects PCI or plug-and-play controllers.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

SSM ran, but driver was not updated.

<table>
<thead>
<tr>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver was uninstalled before running SSM.</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td>To remove the client components, delete the SSM folder and all files within it.</td>
</tr>
</tbody>
</table>

Deliverable does not appear as a possible update.

<table>
<thead>
<tr>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVA file is not SSM compliant.</td>
<td>Rebuild the database with the /debug switch: SSM C:&lt;File Store&gt; /debug /am_bld_db, then examine the file %temp%%computername%.ssmtrace for clues. SSM will reject CVA files that do not have required fields.</td>
</tr>
<tr>
<td>One or more of the CVA fields do not match the current system.</td>
<td>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.</td>
</tr>
</tbody>
</table>
Nothing was installed.

<table>
<thead>
<tr>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>One or more items in the CVA file did not contain a value that matched the system.</td>
<td>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.</td>
</tr>
</tbody>
</table>