Chapter 5
-
Software Distributor (SD-UX)

INDEX

Introduction

This chapter provides an overview of Software Distributor for HP-UX 11.x (SD-UX) commands and concepts. Our attention is turned especially to patching, which will be deepened in the Patches Chapter.

SD-UX is included with the HP-UX Operating System and by default manages software on the local host only. You can also enable SD-UX to install and manage software simultaneously on multiple remote hosts from a central controller. This chapter does not present a comprehensive view of SD-UX. For in-depth information, consult the SD-UX manuals which are available at http://docs.hp.com:

- Managing HP-UX Software with SD-UX (for 11.0)
- Software Distributor Administration Guide (for 11i)

If you need a good overview for SD-UX of HP-UX 10.x refer to Appendix B of HP-UX 10.x Patch Management Guide (HP-UX 10.x).

Patches for SD-UX

The current Patches for the Software Distributor are:

UX 10.20	PHCO_25316	(or newer)
UX 11.00	PHCO_27672	(or newer)
UX 11.11	PHCO_27671	(or newer)

The Basic SD-UX Object Types

Software Distributor uses a variety of object types. This section gives you a simplified view of the object types. You can find formal definitions in the SD-UX manual or the `sd(4)` man page.

Fileset

A fileset is one or more related files, grouped into a manageable unit. It describes a unique subset of the files that make up a product. A fileset may include scripts that control installation and removal. In general, patches are created and managed at the product level, and patch filesets are delivered only within a patch product. Therefore, you should avoid selecting patches at the fileset level, even though SD-UX permits this kind of selection. Selecting patches by fileset level may cause a fix to be only partially applied.

Product

An HP-UX patch is structured as a single SD-UX product that contains one or more filesets.

HP-UX 11.X patches that require customization include SD-UX control scripts at the product level.

Bundle

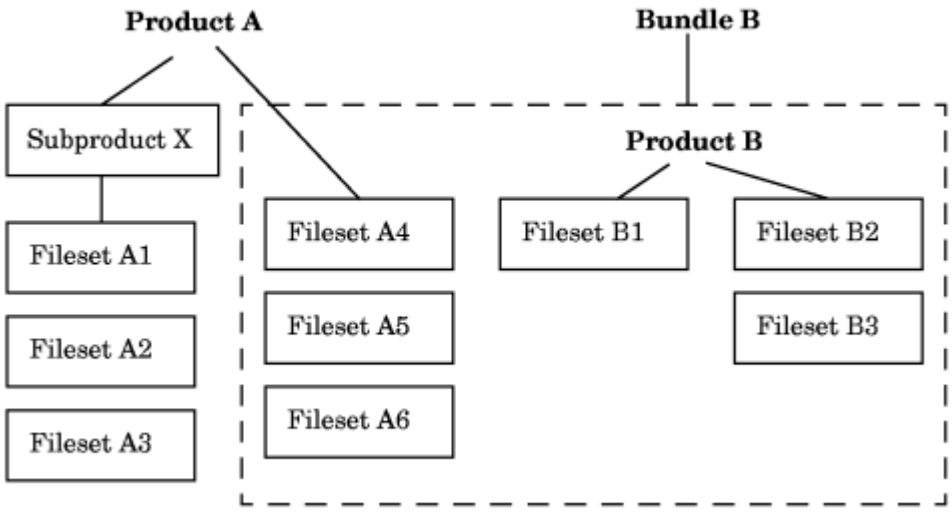
A bundle encapsulates products and filesets into a single software object. Bundles provide a convenient way to group software objects together for easy selection. More than one bundle can contain the same software objects. A bundle can be thought of as a virtual “configuration” of software. HP provides several types of standard patch bundles.

Depot

A depot is a directory that contains software products or bundles that are available for direct or remote installation. You can change the contents of a depot. A depot can also be a distribution media (e.g., CD or tape) or a single, serial file that contains products or bundles.

Software Specifications

When an SD-UX command is applied to a software selection, the selection is comprised of one or more software specifications. A software specification is a unique identifier for an SD-UX software object. A software specification must name either a product or a bundle, and filesets can be specified only within a product. If you explicitly select a bundle, all products within the bundle are also selected. If you select a product, all filesets within that product are also selected.



Example of HP-UX Software Structure

For patch operations, you usually only need to refer to a patch or bundle name.

The software specification takes one of the following formats:

```
product[.fileset][, version]
bundle[.product[.fileset]][, version]
```

where the version has the form:

```
[r= revision][,a= arch][,v= vendor][,c= category]
```

(The version may also have a `l=` locationcomponent that applies only to installed software and refers to software installed to a location other than the default product directory.)

The `software_spec` attribute contains the full software specification for any bundle, product, or patch. You can use the `swlist` command to display this information. The following example shows how `swlist` can create a list of the software specifications for a patch at the fileset level. The software specification for the patch product appears in the output as a comment.

```
# swlist -l fileset -a software_spec PHKL_18543
# PHKL_18543 PHKL_18543,l=/,r=1.0,a=HP-UX_B.11.00_32/64,v=HP
PHKL_18543.C-INC PHKL_18543.C-INC,l=/,r=1.0,a=HP-UX_B.11.00_32/64,v=HP,fa=HP-UX_B.11.00_32/64
PHKL_18543.CORE-KRN PHKL_18543.CORE-KRN,l=/,r=1.0,a=HP-UX_B.11.00_32/64,v=HP,fa=HP-UX_B.11.00_32/64
PHKL_18543.CORE2-KRN PHKL_18543.CORE2-KRN,l=/,r=1.0,a=HP-UX_B.11.00_32/64,v=HP,fa=HP-UX_B.11.00_32
PHKL_18543.KERN2-RUN PHKL_18543.KERN2-RUN,l=/,r=1.0,a=HP-UX_B.11.00_32/64,v=HP,fa=HP-UX_B.11.00_32
PHKL_18543.VXFS-ADV-KRN PHKL_18543.VXFS-ADV-KRN,l=/,r=1.0,a=HP-UX_B.11.00_32/64,v=HP,fa=HP-UX_B.11.00_32
PHKL_18543.VXFS-BASE-KRN PHKL_18543.VXFS-BASE-KRN,l=/,r=1.0,a=HP-UX_B.11.00_32/64,v=HP,fa=HP-UX_B.11.00_32
PHKL_18543.VXFS-PRG PHKL_18543.VXFS-PRG,l=/,r=1.0,a=HP-UX_B.11.00_32/64,v=HP,fa=HP-UX_B.11.00_32/64
```

The IPD – Installed Product Database

An SD “catalog” is a database of information about software. For example, a depot is a “catalog” plus the set of files described by the catalog. The *IPD (Installed Product Database)* is a specific instance of an SD catalog found at `/var/adm/sw/products`. This database (or catalog) does not contain the software itself. It only contains the meta data, i.e. data about SD objects. The kind of data kept in the catalog are all the object attributes (i.e. “revision=1.2”), all the control scripts (preinstall, postremoved, ...), swask response files, permission (swacl) files, etc.

IPD File listing

The IPD is not a single file but a hierarchy of files located in `/var/adm/sw/products`. In the top IPD directory, each bundle and product gets its own directory.

```
# ll /var/adm/sw/products
total 6648
dr-x------  473 root       sys          11264 Apr  9 12:24 ./
drwxr-xr-x  13 bin        bin           1024 Apr  9 16:08 ../
drwxr-xr-x   7 root       sys           1024 Apr 25  2002 Accounting/
drwxr-xr-x   5 root       sys             96 Apr 25  2002 ApacheStrong/
drwxr-xr-x  11 root       sys           1024 Apr 25  2002 Asian-Core/
...
-rw-r--r--   1 root       sys        1960447 Apr  4 10:34 INDEX
...
```

Underneath those directories for all products and bundles one will find the control directories. Control directories have names like “`?files`”. For example “`ifiles`” which is the IPD control directory, “`pfiles`” which are bundle and product control directories and “`dfiles`” which are depot control directories. Control directories contain control information for the object they refer to. Here is the type of information that is saved and what file it is contained in:

File	Control Information
INDEX	Bundle, product, subproduct, fileset and IPD attributes (revision, architecture, is_patch, …). These are the fragment INDEX files from which the master INDEX file gets created.
INFO	File attributes (path, cksum, mode, …)
file	Files referred to by an attribute. For example, "readme README". The file README is saved in the control directory.
control scripts	preinstall, postinstall, configure, …
response file	swask response files are saved in the control directory.
_ACL	root ACL file (in IPD control dir only)
_OWNER	root owner file (in IPD control dir only)

Master Index

The master index is a single file – `/var/adm/sw/products/INDEX` – containing all the attribute-value pairs from each control directory INDEX file. Frequently, if the master INDEX is corrupted, it may be rebuilt by concatenating all the good fragment INDEX's, see Rebuilding the Master INDEX file.

Session Files

Session files let you save your work from a command session. Each invocation of an SD-UX command defines a session. The invocation options, source information, software selections, and target hosts are saved before command execution actually commences. This lets you re-execute the command even if the session ends before proper completion. Each session is saved to the file `$HOME/.sw/sessions/{ command}.last`. This file is overwritten on each invocation.

You can also save session information from interactive or command-line sessions. From an interactive session, you can save session information into a file at any time by selecting the **Save Session** or **Save Session As** option from the **File** menu. From a command-line session, save session information by executing swinstall or swcopy with the `-C session_file` option.

A session file uses the same syntax as the defaults files. You can specify an absolute path for a session file. If you do not specify a directory, the default location for a session file is `$HOME/.sw/sessions/`.

To re-execute a saved session from an interactive session, use the **Recall Session** option from the **File** menu. To re-execute a session from a command-line, specify the session file as the argument for the `-S session_file` option of `swinstall` or `swcopy`.

Note that when you re-execute a session file, the values in the session file take precedence over values in the system defaults file. Likewise, any command line options or parameters that you specify when you invoke `swinstall` or `swcopy` take precedence over the values in the session file.

NOTE: Use of session files is not recommended with `swremove` because the session file could include software selections that you do not want included in the removal operation.

Setting Default Values for Command Options

SD-UX commands have extensive options that alter command behavior. The `/usr/lib/sw/sys.defaults` file is a template that lists and explains each option, default values, all other allowable values, and the resulting system behavior for each. These options are listed as comments that you can copy into the system defaults file (`/var/adm/sw/defaults`) or your personal defaults file (`$HOME/.sw.defaults`).

Values in these option files are specified using this syntax:

```
[command.]option=value
```

These rules govern the way the defaults work:

1. Option values in `/usr/lib/sw/sys.defaults` is only usable as a template for copying to other option files.
2. Option values in `/var/adm/sw/defaults` file affect all users in a system.
3. Option values in your personal `$HOME/.sw.defaults` file affect only you and not the entire system.
4. Option values in a session file affect activities only for that session and revert when that session is completed.
5. Option values changed on the command line affect only that activity.

For system-wide policy setting, use the `/var/adm/sw/defaults` file. Keep in mind, however, that individual users may override these values with their own `$HOME/.sw.defaults` file, session files, or command line changes.

These values can also be overridden by specifying an options file with the `-X option_file` command-line option or with one or more `-x option=value` options directly on the command line. They can also be changed using the GUI Options Editor.

Altering option values and storing them in a defaults file can help when you want the SD-UX command to behave the same way each time the command is invoked. Options in the defaults file are read as part of command initialization. Because the daemon is already running, after changing daemon options, the daemon must be restarted for these options to be recognized. To restart the daemon, type:

```
/usr/sbin/swagentd –r
```

SD-UX Commands

The following section discusses the most commonly used SD-UX commands:

- `swinstall` installs and configures software products.
- `swcopy` copies software products for subsequent installation or distribution.
- `swremove` unconfigures and removes software products.
- `swlist` displays information about software products.
- `swreg` registers or unregisters depots or roots.
- `swmodify` modifies software product information in a target root or depot.
- `swpackage` packages software products into a depot (directory or tape).
- `swverify` verifies available (copied), installed, or configured software products.
- `swconfig` configure, unconfigure and reconfigure software products.
- `cleanup` removes patch rollbacks. Refer to Patches Chapter.
- `show_patches` displays patches (UX 11.X only). Refer to Patches Chapter.

All SD-UX commands run from the command line. `swinstall`, `swcopy`, `swremove`, and `swlist` have an optional GUI mode.

swinstall Command

The `swinstall` command is used to load software from a source depot and onto a target system.

- Because many patches aren't designed for individual installation, the automatic matching options(`autoselect_patches`, `patch_match_target`) should be the preferred method for installing patches.
- `swinstall` has numerous options that you should not use for patching because they lack dependency support. HP recommends that you use only the options discussed below.

Synopsis

```
swinstall [-i] [-p] [-v] [-s source] [-x option=value]...
      [s/w_selections]
```

Command Line Arguments	
`-i`	Use an interactive user interface. If the environment variable DISPLAY is set to a valid X windows display, a graphical user interface is invoked. Otherwise a terminal user interface (TUI) designed for use on ASCII terminals is invoked. The GUI starts by default if you enter `swinstall` without any `s/w_selections`.
`-p`	Previews the install operation without performing the actual installation. Preview mode is *not* enabled by default.
`-v`	Requests verbose mode. This option affects only standard output and not the log files.
`-s source`	Specifies the depot (`source`) containing the software to be installed.

`-x option=value`	Sets the specified command option to the value given, overriding any other values for that option. Patch related command options are specified below.
`s/w_selections`	One or more software specifications.

The following options have the most relevance to patching. Where appropriate, default values are shown. For the full set of available options, consult the `swinstall(1m)` man page or go to the SD-UX manual reference.

Options	
`option=default value`	Menu Path in Interactive Interface
Description	
`autoreboot=false`	None (GUI waits for permission to reboot)
Enables an automatic reboot upon completion of the software installation	
`autoselect_dependencies=true`	**Actions→Mark for Install** (when marking software)
When software is selected for installation with an SD-UX-enforced dependency, that software will automatically be selected for installation if present in the source depot and `autoselect_dependencies` is set to true. While few 11.00 patches exist with dependencies enforced by the SD-UX tools, those that do employ them to enforce critical requirements of content and load order. All 11i patches will contain enforced dependencies except for those that meet strict exception rules. As a result, this option becomes even more important and helpful, eliminating lengthy, manual patch dependency analysis. This option should not be set to false unless directed by an HP Support Engineer.	
`autoselect_patches=true`	**Actions→Manage Patch Selection→Automatically select patches for software to be installed**
When loading a software product, any patches within the same depot for that product will automatically be selected for installation.	
`autoselect_reference_bundles=true`	None
When set to true, any bundle wrappers within the source depot that contain software selected for installation will be automatically selected if the `is_reference` attribute is set to true. Note that this does not mean all of the software listed in the wrapper will be selected, only the bundle wrapper itself.	
`enforce_dependencies=true`	**Options→Change Options→Enforce dependency analysis errors in agent**
Enforces software dependencies. When software is selected for installation with an SD-UX-enforced dependency, if the dependency is not present on the target system and is not selected for installation from the source depot, installation will only proceed if `enforce_dependencies` is set to `false`. While few 11.0 patches exist with dependencies enforced by the SD-UX tools, those that do employ them are enforcing critical requirements of content and load order. Since all 11i patch dependencies will be enforced by SD-UX, it is especially important that this option is taken advantage of in selecting patches and avoiding unwanted configurations. This option should not be set to false unless directed by an HP Support Engineer.	
`enforce_scripts=true`	**Options→Change Options→Enforce script failures**
Each patch may have several installation scripts associated with it. These scripts may issue errors to protect the system from incorrect patch usage. This option should not be used unless directed by an HP Support Engineer.	
`match_target=false`	**Actions→Match What Target Has**
Selects all software within the source depot with an ancestor attribute that matches a fileset currently on the targetsystem. While on HP-UX 10.X systems this option was used to select patches within a depot that applied to the target sys-tem, the `patch_match_target` option is the preferred method for HP-UX 11.X releases to provide this func-tionality.	

`mount_all_filesystems=true`	**Options→Change Options→Mount filesystems in /etc/fstab or /etc/checklist**
By default, `swinstall` requires that all filesystems listed in the systems `/etc/fstab` file are mounted prior to installation. Setting this option to false removes this restriction.	
`patch_filter=software_specification`	**Actions→Manage Patch Selection→Filter**
The patch_filter option can be used to specify a subset of software available to load. This option is not yet recommended for general use in 11.0 as no provision is made for dependencies. However, this option can be used for 11.00 patch selection when combined with user-defined category tags. It is also an especially powerful tool for depot management in 11i since dependencies are dealt with correctly.	
`patch_match_target=false`	**Actions→Manage Patch Selection→Automatically select patches for software installed on the target**
Select all patches within the source depot that modify the existing system software. This is the recommended method to install patches from a managed depot (such as those provided by HP).	
`patch_save_files=true`	**Options→Change Options→Save files replaced by patch for later rollback**
If set to false, patches are loaded directly to the committed state and cannot be rolled back. While a convenient way to control disk usage, this option is not recommended unless alternative recovery mechanisms are available.	
`reinstall=false`	**Options→Change Options→Reinstall filesets even if the same revision exists**
Prevents SD-UX from re-installing (overwriting) an existing revision of a fileset. If set to true, filesets will be re-installed.	
`source_cdrom=/SD-UX_CDROM`	None (default cannot be changed within GUI)
Specify the device file of the CD-ROM to be used as the default.	
`source_tape=/dev/rmt/0m`	None (default cannot be changed within GUI)
Specify the device file of the tape drive to be used as the default.	
`use_alternate_source=false`	**Options→Change Options→All targets to resolve the source locally**
Empowers each target agent to use its own, configured alternate source, instead of the one specified by the user. If false, each target agent uses the same source (the source specified by the user and validated by the command). If true, each target agent uses its own configured value for the source.	
`write_remote_files=false`	None
Prevents installation of files to a target that exists on a remote (NFS) file system. By default, swinstall skips files that would be installed to a remote (NFS) file system (or that are already there). When set to true and superuser has write permission on the remote file system, the remote files are installed.	

Examples

Install from a CD mounted and registered on the system grendel:

```
swinstall -s grendel:/cdrom/QPK1100 \
      -x patch_match_target=true -x autoreboot=true
```

Use the `swinstall` command's preview mode (`-p` option) to get an idea of what to expect for the bundle you want to install. For example:

```
swinstall -p -s grendel:/cdrom/QPK1100 \
      -x patch_match_target=true
      -x autoreboot=true
```

Install of product A1234AA from registered depot `/var/spool/sw` on the system grendel:

```
swinstall –s grendel:/var/spool/sw A1234AA
```

swcopy Command

The swcopy command copies software from one depot to another. This can be particularly useful if software exists in several depots. For example, you can copy all of the contents of individual patches into a single depot from which the group can be loaded in a single session and with a single reboot (if needed).

Note that the swcopy command automatically registers (enables remote access to) any depot that it creates. You do not need to use the swreg command on depots created by swcopy.

Synopsis

```
swcopy [-i] [-p] [-v] [-s source] [-x option=value]
      [s/w_selections] [@ target_selection]
```

Command Line Arguments	
-i	Use an interactive user interface. If the environment variable DISPLAY is set to a valid X windows display, a graphical user interface is invoked. Otherwise a terminal user interface (TUI) designed for use on ASCII terminals is invoked. The GUI starts by default if you enter swcopy without any s/w_selections.
-p	Previews the install operation without performing the actual copy. Preview mode is *not* enabled by default.
-v	Requests verbose mode. This option affects only standard output and not the log files.
-s source	Specifies the depot (source) containing the software to be copied.
-x option=value	Sets the specified command option to the value given, overriding any other values for that option. Patch related command options are specified below.
s/w_selections	One or more software specifications.
target_selections	The absolute path name (directory location) to which you want the s/w_selections to be copied. If you specify a host with the directory, the syntax is host:/directory where the host name can be a name, domain name, or internet address.

The following options have the most relevance to patching. Where appropriate, default values are shown. For the full set of available options, consult the swcopy(1m) man page or refer to the SD-UX manual.

Options	
option=default value	Menu Path in Interactive Interface
Description	
autoselect_dependencies=true	**Actions→Mark for Install** (when marking software)
When software is selected for copying with a registered dependency on other software, other software will automatically be selected to be copied if present in the source depot and autoselect_dependencies is set to true.	
autoselect_reference_bundles=true	None (default cannot be changed within GUI)
When set to true, any bundle wrappers within the source depot that contain software selected for installation will be automatically selected if the is_reference attribute is set to true. Note that this does not mean all of the software listed in the wrapper will be selected, only the bundle wrapper itself.	

`compress_files=false`	**Options→Change Options→Compress files during transfer**
Setting this option to true causes `swcopy` to compress file before transfer to the target depot. This will conserve disk space and can enhance performance on slower networks (50 Kilobytes/second or less), although it may not improve fast networks.	
`enforce_dependencies=true`	**Options→Change Options→Enforce dependency analysis errors in agent**
Enforces software dependencies. When software to be copied has an SD-UX-enforced dependency, if that dependency is not present on the target system and is not marked for copying from the source depot the copy will only proceed if `enforce_dependencies` is set to false. While few 11.00 patches currently exist with dependencies enforced by the SD-UX tools, those that do employ them to enforce critical requirements of content and load order. The majority of 11i patches have SD-UX-enforced dependencies, making this option very useful. This option should not be set to false unless directed by an HP Support Engineer.	
`mount_all_filesystems=true`	**Options→Change Options→Mount filesystems in /etc/fstab or /etc/checklist**
By default, `swcopy` requires that all filesystems listed in the systems `/etc/fstab` file are mounted prior to installation. Setting this option to false removes this restriction.	
`reinstall=false`	**Options→Change Options→ Recopy filesets even if the same revision exists**
Prevents SD-UX from overwriting an existing revision of a fileset. If set to true, filesets will be recopied.	
`source_tape=/dev/rmt/0m`	None (default cannot be changed within GUI)
Specify the device file of the tape drive to be used as the default.	
`uncompress_files=false`	**Options→Change Options→Uncompress files after transfer**
When set to true, files are uncompressed before `swcopy` puts them into the target depot. See also the `compress_files` option.	
`write_remote_files=false`	None
Prevents copying of files to a target that exists on a remote (NFS) file system. By default, `swcopy` skips files that would be copied to an NFS file system (or that are already there). When set to true and superuser has write permission on the remote file system, files are copied to remote systems.	

Examples

With the CD mounted at `/cdrom`, copy the contents of the QPK1100 depot to the local system under the `/var/tmp/MyDepot` directory.

```
swcopy -s /cdrom/QPK1100 \* @ /var/tmp/MyDepot
```

Invoke an interactive session, using the default depot at hostX as the source:

```
swcopy -i -s hostX
```

Copy all patches in current directory to the depot `/hub/patches` (assuming root shell is `/sbin/sh`):

```
for PATCHDEPOT in *.depot
do
swcopy -s $PATCHDEPOT \* @ /hub/patches
done
```

Copy a HP-UX 10.X style depot from the system oldsys to an HP-UX 11.X system.

```
swcopy -s oldsys:/depot -x layout_version=0.8 \* @ /depots/oldsys
```

See also Create a depot on tape (→`swpackage`).

swremove Command

The swremove command deletes software that has been installed on your system. It also removes software from depots.

Note that swremove has several limitations when used for patch operations:

- You cannot use `swremove` to remove committed patches.
- You should not use `swremove` to remove patch information that remains in the IPD after installing a new version of HP-UX, see cleanup(1M) command in the Patches Chapter.
- If you are on an 11.00 system and you use `swremove` to remove a patch, you must make sure you didn't "break" any software dependencies. If the patch was needed to fulfill a documented dependency then patches to satisfy the dependency must be activated via rollback or installation. In HP-UX 11i you cannot remove a patch that is required by another patch. The `swremove` command will fail if the unwanted patch fulfills a dependency.
- Removal of a patch bundle does not automatically return you to the patch state prior to loading that bundle.
- `swremove` may not always be your first and best solution for error recovery. Make sure your other recovery methods are not more appropriate before you use this command.

Synopsis

```
swremove [-i] [-d] [-p] [-v] [-x option=value] [s/w_selections]
      [ @ target]
```

Command Line Arguments	
`-i`	Use an interactive user interface. If the environment variable DISPLAY is set to a valid X windows display, a graphical user interface is invoked. Otherwise a terminal user interface (TUI) designed for use on ASCII terminals is invoked. The GUI starts by default if you enter `swremove` without any `software_selections`.
`-d`	Operate on a depot rather than installed software.
`-p`	Previews the remove operation without performing the actual removal. Preview mode is not enabled by default.
`-v`	Requests verbose mode. This option affects only standard output and not the logfiles. Verbose mode is enabled by default.
`-x option=value`	Sets the specified command option to the value given, overriding any other values for that option. Patch related command options are specified below.
`s/w_selections`	One or more software specifications.
`target`	The depot from which software is to be removed. If not specified, the target is assumed to be the system itself.

The following options have the most relevancy to patching. Where appropriate, default values are shown. For the full set of available options, consult the `swremove(1m)` man page or refer to the SD-UX manual.

Options	
`option=default value`	Menu Path in Interactive Interface
Description	
`autoselect_reference_bundles=true`	None
If true, bundles that have the `is_reference` attribute set to true will be automatically removed when the last of its contents is removed. If false, the bundles will not be automatically removed.	
`enforce_dependencies=true`	**Options→Change Options→Enforce dependency analysis errors in agent**
Enforces software dependencies. When software selected for removal has a registered dependency, if the dependencyis not present on the target system or also selected for removal from the source depot, removal only proceeds if `enforce_dependencies` is set to false. While few 11.00 patches currently exist with dependencies enforced by the SD-UX tools, those that do employ them to enforce critical requirements of content and removal order. However, since all 11i patches enforce depen-dencies, this option is very useful in maintaining patch integrity and system stability. Do not set this option to false unless directed to do so by an HP Support Engineer.	
`enforce_scripts=true`	**Options→Change Options→Enforce script failures**
Each patch may have several removal scripts associated with it. These scripts may issue errors to protect the system from incorrect patch usage. This option should not be used unless directed by an HP Support Engineer.	
`mount_all_filesystems=true`	**Options→Change Options→Mount filesystems in /etc/fstab or /etc/checklist**
By default, SD-UX requires that all filesystems listed in the systems `/etc/fstab` file are mounted prior to removal. Setting this option to false removes this restriction.	
`write_remote_files=false`	None (default cannot be changed within GUI)
Prevents removal of files from a target that exists on a remote (NFS) file system. By default, swremove skips files that would be removed from an NFS file system. When set to true and superuser has write permission on the remote file system, files are removed from remote systems.	

Examples

Remove only the bundle wrapper XSWHWCR1100 from the system, leaving any contents present (Note that the trailing period (`.`) is essential to removing the wrapper only):

```
swremove XSWHWCR1100.
```

Remove all contents of the depot, `/depots/MyDepot`:

```
swremove -d \* @ /depots/MyDepot
```

swlist Command

The `swlist` command provides information on software installed on a system or located in a depot. More specifically:

- See what's installed on a system.
- See what software is in a depot.
- Check attributes of software.

- Browse the patch documentation.
- See what depots are available on remote systems.

Synopsis

```
swlist [-i] [-d] [-v] [-a attribute] [-l level] [-s source]
      [s/w_selections] [ @ target]
```

Command Line Arguments	
`-i`	Invokes a GUI interface that lets you perform interactive software selections. If the environment variable DISPLAY is set to a valid X windows display, a graphical user interface is invoked. Otherwise a terminal user interface (TUI) designed for use on ASCII terminals is invoked. You *must* specify -i to invoke the GUI; it never starts by default.
`-d`	Lists software depots instead of software currently installed on the target system.
`-v`	If no `-a` options are specified, then list all the attributes for an object, one attribute per line. The attributes are listed in the format: `keyword value` If one or more `-a` options are specified, then list the selected attributes in the above format.
`-a attribute`	The named attribute is included in the listing when defined at the specified level. While this option may be specified multiple times, the ordering of the arguments does not control the format of the list.
`-s source`	Specifies the software source to list. This is an alternative way to list a source depot. You can also specify the sources as target depots and list them using the `-d` option.
`s/w_selections`	One or more software specifications.
`target`	The depot to be listed. If not specified, the target is assumed to be the system itself.
`-l level`	Specifies the detail of the `swlist` output. The values used include: • `file` List all files recorded in the IPD. The listing may be limited in scope by the `s/w_selections` specification. Each file is preceded by the product and fileset that is the registered owner of that file. A comment (marked by a leading # character) precedes each block giving the name, revision, and description of the product or fileset to be listed. • `fileset` List all filesets recorded in the IPD (in `product.fileset` format) with the associated revision and description. A comment (marked by a leading # character) precedes each block giving the name, revision, and description of the product. • `product` List all products with revision and description for each. • `bundle` List all bundles with revision and description for each. • `depot` List all registered depots on the target system. • `patch` List all patch filesets using the full software specification, followed by the associated description and current `patch_state`. The listing is sorted by ancestor, and all products and filesets are listed as a comment showing revision and description before any patch fileset that apply to it. • `category` List all category tags currently defined within the target depot. • *default* (no level specified) When no level is specified, `swlist` displays all bundles within the depot followed by any products not contained within a bundle. As is the case with their respective levels, the bundles and products are listed with revision and one-line description.

The following options have the most relevancy to patching. Where appropriate, default values are shown. For the full set of available options, consult the `swlist(1m)` man page or refer to the SD-UX manual.

Options	
`option=default value` Description	Menu Path in Interactive Interface
`show_superseded_patches=false`	
The swlist command lists the contents of the IPD. In 11.00 systems, swlist lists all patches in the IPD, whether or not they are superseded. In 11i, swlist only lists those patches which are active on the system. By setting this option to true the 11.00 swlist functionality can be turned on in an 11i system (all patches in the IPD are listed). However, to get the 11i functionality on an 11.00 system, patch PHCO_22526 must be installed and `swlist.show_superseded_patches=false` must be added to the `/var/adm/sw/defaults` file.	

Examples

List all products with revision and description for each:

```
swlist –l product | more
```

List all filesets which have a state other than configured:

```
swlist –l fileset –a state | grep –v –e '^#' –e configured
```

List all patches in the depot `/var/MyDepot` on the system grendel:

```
swlist -d -l product *,c=patch @ grendel:/var/MyDepot
```

List the filesets modified by installed patch PHSS_8675

```
swlist -a ancestor PHSS_8675
```

List all of the files delivered within patch PHCO_12140 after downloading from the ITRC:

```
swlist -d -l file @ /tmp/PHCO_12140.depot
```

List all patches that have modified the LVM product

```
swlist -l patch LVM
```

Display the documentation for all patches containing critical functionality

```
swlist -a readme -l product *,c=critical
```

List all category tags defined in the depot /var/MyDepot on the system grendel

```
swlist -d -l category @ grendel:/var/MyDepot
```

swreg Command

The `swreg` command registers or unregisters an existing depot. When a depot is registered, it can be accessed from remote systems. Unregistration of a depot can be a convenient way to limit access during development. (Note that unregistered depots are still available locally.)

Synopsis

```
swreg -l depot [-u] [-v] [objects]
```

Command Line Arguments	
`-l depot`	Perform operations on depots. While other levels of SD-UX objects may be modified by `swreg`, they are not within the scope of this tutorial.
`-u`	Causes `swreg` to unregister the specified objects instead of registering them.
`-v`	Requests verbose mode. This option affects only standard output and not the log files.
`objects`	Specifies the path to the object[s] to be registered or unregistered.

For the full set of available options, consult the `swreg(1m)` man page or the SD-UX manual.

Examples

Register the patch depot QPK1100:

```
swreg -l depot /cdrom/QPK1100
```

Disable remote access by unregistering the depot QPK1100 (local access is still enabled):

```
swreg -u -l depot /cdrom/QPK1100
```

swmodify Command

SD-UX commands automatically keep track of software management operations by creating an *Installed Products Database* (IPD) and various catalog files that contain information about the software on the system. Although you cannot edit the IPD or catalog files directly, the swmodify command lets you change the contents of these files via the command line.

WARNING: With the exception of committing patches and creating category tags, the swmodify command is not recommended for general usage. Improper alteration of the information in the IPD could cause unexpected behavior during subsequent patching or system updates and leave your system in an unsupportable state.

NOTE: You may choose the cleanup utility (see Patches Chapter) as a front-end to the `patch_commit` option of `swmodify` that provides an easier interface to commit multiple patches at once.

Synopsis

```
swmodify [-d] [-p] [-v] ]-a attribute[=value]] [-x option=value]
      [s/w_selections] [ @ target]
```

Command Line Arguments	
`-a attribute[=value]`	Add, modify, or delete the value of the given attribute. If the `-u` option is specified, then delete the attribute from the given `software_selections` (or delete the value from the set of values currently defined for the attribute). Otherwise add/modify the attribute for each `software_selection` by setting it to the given value. Multiple `-a` options can be specified. Each attribute modification will be applied to every `software_selection`.
`-d`	Perform modifications on a depot. The given `target` must be a depot.
`-p`	Previews the modify operation without modifying anything. Preview mode is not enabled by default.
`-v`	Requests verbose mode. This option affects only standard output and not the log files.
`-x option=value`	Sets the specified command option to the value given, overriding any other values for that option. Patch related command options are specified below.
`s/w_selections`	One or more software specifications.
`target`	The depot to be modified. If not specified, the target is assumed to be the system itself.

The following options have the most relevancy to patching. Where appropriate, default values are shown. For the full set of available options, consult the `swmodify(1m)` man page or refer to the SD-UX manual.

Options
`option=default value` Description
`patch_commit=false` Commits a patch by removing files saved for patch rollback. The default value is false. When set to true, this option removes the saved files for the patches specified in the software selections for the command. Once you have run this option on a patch, you cannot remove the patch unless you remove the associated base software that the patch modified. See also the cleanup utility described in the Patches Chapter.

Examples

Commit the patch PHKL_1234 and remove its corresponding rollback files:

```
swmodify -x patch_commit=true PHKL_1234
```

Mark all patches in the depot `/depots/newpatches` with a new category tag to indicate that they have been approved:

```
swmodify -a category=approved \* @ /depots/newpatches
```

swpackage Command

While primarily used to create depots from source files, this command allows the transfer of software onto a tape or into a disk depot which can then be used as a software source. Either method can be used to transport the contents of a depot to another system for local access.

The tape can be used in the absence of networking support, and the tape depot could be provided via ftp(1).

Synopsis

```
swpackage [-p] [-v] [-s directory] [-x option=value]
      [s/w_selections] [@ target]
```

Command Line Arguments	
`-p`	Previews the package operation without performing the actual packaging. Preview mode is not enabled by default.
`-v`	Requests verbose mode. This option affects only standard output and not the log files.
`-s directory`	An existing directory depot (which already contains products) to be used as the source.
`-x option=value`	Sets the specified command option to the value given, overriding any other values for that option. Patch related command options are specified below.
`s/w_selections`	One or more software specifications.
`Target`	If you are creating a distribution depot (directory), this operand defines the location of the *directory*. Without this operand, `/var/spool/sw` is used as the default depot directory. If you are creating a distribution tape, this operand names the *device* file on which to write the `tar` archive. The device file must exist so that swpackage can determine if the media is a DDS tape or a disk file. Without this operand, swpackage uses the device file `/dev/swtape`.

The following options have the most relevancy to patching. Where appropriate, default values are shown. For the full set of available options, consult the `swpackage(1m)` man page or refer to the SD-UX manual.

Options
`option=default value` Description
`compress_files=false` Setting this option to true causes `swpackage` to compress files before packaging them. This creates smaller depots.
`layout_version=1.0` Specifies the POSIX layout version to which the SD-UX commands conform when writing distributions. Supported values are 1.0 (default) and 0.8. Refer to the `swpackage(1m)` manpage or the SD-UX manual for more information.
`target_type=directory` Defines the type of distribution to create. The recognized types are `directory` and `tape`.

Examples

Re-package the entire contents of the depot `/var/spool/sw` onto the tape at `/dev/rmt/0m`:

```
swpackage -s /var/spool/sw -x target_type=tape @ /dev/rmt/0m
```

Create a depot on tape

- Which depots are on the server?

```
# swlist -l depot -s upserv01

  /depots/11.11_AppDART56_Mar02
...
```

  In this example, is Glance in a certain depot?

```
# swlist -l bundle -s upserv01:/depots/11.11_AppDART56_Mar02 |\
grep -i glance
B3701AA_TRY    C.03.55.00    Trial HP GlancePlus/UX Pak for s800 11i
...
```

- Copy Glance to a local depot, as it is not possible to copy it directly to tape:

```
# swcopy -s upserv01:/depots/11.11_AppDART56_Mar02 B3701AA_TRY @\
/tmp/glance_depot
```

- Add a patch to the depot:

```
# swcopy -s police:/PATCHDEPOT/PHSS_24864 \* @ /tmp/glance_depot
```

- Check depot contents:

```
# swlist -d -l product @ /tmp/glance_depot
  Glance                 C.03.55.00     HP GlancePlus/UX
  MeasureWare            C.03.55.00     MeasureWare Software/UX
  ...
  PHSS_24864             1.0            PRM C.01.08.2 Cumulative Patch
```

- Insert a DDS tape and copy the depot to the tape:

```
# swpackage -x target_type=tape -s /tmp/glance_depot \* @ /dev/rmt/0m
```

- or to a depot file:

```
# swpackage -x target_type=tape -s /tmp/glance_depot \* @ \
/tmp/glance_depot.depot
```

- Check depot contents:

```
# swlist -d -l product @ /dev/rmt/0m

  Glance                 C.03.55.00     HP GlancePlus/UX
  MeasureWare            C.03.55.00     MeasureWare Software/UX
  ...
  PHSS_24864             1.0            PRM C.01.08.2 Cumulative Patch
```

- Eject the tape:

```
# mt -t /dev/rmt/0m offl
```

swverify Command

The SD-UX `swverify` command verifies available (copied), installed, or configured software products on the specified host. swverify also:

- Determines whether installed or configured software is compatible with the host on which that software is installed.
- Makes sure that all dependencies (prerequisites, corequisites) are being met (for installed software) or can be met (for copied software).
- Executes vendor-specific verification scripts (that is, scripts that testify to the correctness of the product's configuration) if the installed state of the software is configured.
- Reports missing files, checks all file attributes including permissions, file types, size, checksum, mtime, link source and major/minor attributes.

Synopsis

```
swverify [-d|-r] [-v] [-C session_file] [-f software_file]
      [-S session_file] [-t target_file] [-x option=value]
      [-X config_file] [software_selections] [@ target_selections]
```

The swverify command does not feature a GUI. All verify interaction with the system is done on the command line.

Command Line Arguments	
`-d`	Operate on a depot rather than installed software.
`-v`	Turn on verbose output to `stdout` and display all activity to the screen. Lets you see the results of the command as it executes.
`-f software file`	Read a list of software selections from a separate file instead of from the command line.
`-x option=value`	Specify a value to override a default value or a value in an options file.

Examples

To verify an installed fileset `mysoft.myfileset` located on the default depot at `myhosts`, you would type (You could also omit the @ sign and the myhost designation since the software being verified is assumed to be located in the default depot on the local host.):

```
swverify -d mysoft.myfileset @ myhosts
```

To verify the C and Pascal products that are installed on the local host:

```
swverify –v C Pascal
```

Verify the entire contents of a local depot:

```
swverify -d \* @ /var/spool/sw
```

NOTE: Look for errors in the `/var/adm/sw/swagent.log` file.

swconfig Command

The `swconfig` command lets you explicitly configure, unconfigure and reconfigure software products that are installed on a local host by executing the configure script. These scripts are only executed on the host that will actually be running the software.

You can use the swinstall and swremove commands to perform many configuration or unconfiguration tasks. However, the `swconfig` command lets you work independently of these commands. `swconfig` can also be useful when a configuration fails, is deferred, or must be changed. It runs only from the command line interface.

Synopsis

```
swconfig [-p ][-u][-v][-c catalog] [-C session_file]
      [-f software_file] [-S session file] [-t target_file]
      [-x option=value] [-X option_file]
      [software_selections] [@ target_selections]
```

Selected Command Line Arguments	
`-p`	Preview a configuration task from the command line by running it through the Analysis Phase and then exiting.
`-v`	Turn on verbose output to stdout and display all activity to the screen. Lets you see the results of the command as it executes.
`-f software file`	Read a list of software selections from a separate file instead of from the command line.
`-x option=value`	Specify a value to override a default value or a value in an options file.

Examples

To configure `productA`, located in the default depot on the local host:

```
swconfig productA
```

To reconfigure the Omniback product using the default option

```
swconfig -x reconfigure=true Omniback
```

To configure everything, use with caution

```
swconfig \*
```

Troubleshooting

Error Logging

All SD-UX commands (except `swlist` and `swacl`) log summary information about the session, and operation details to a command-specific logfile located (by default) in

/var/adm/sw/<command>.log. For example, if you wanted to examine the logfile for swinstall, you would look in the file /var/adm/sw/swinstall.log.

Of more interest in case of troubleshooting is to examine target agent logfiles for a current session. The location of the agent logfile varies, depending on the type of target:

- /var/adm/sw/swagent.log when operating on a host's primary root.
- /<root_path>/var/adm/sw/swagent.log for an alternate root.
- /<depot_path>/swagent.log for a target or source depot.

The default location of a host's daemon logfile is /var/adm/sw/swagentd.log. This logfile contains information for problems starting agents, particularly for problems where you have access denied to a depot or root.

Error Messages

SD-UX error messages indicate that a problem occurred that will influence the overall outcome of an operation.

For example, if a target in an install session fails the analysis phase due to insufficient disk space, you would find the following error message in the agent log file:

```
ERROR:  The estimated disk space used on filesystem "/" is
14104 Kbyte blocks. This operation will exceed the
minimum free space for this disk. You should free up at
least 2280 Kbyte blocks to avoid installing beyond this
threshold of available user disk space. If you are
running interactive "swinstall", you must return to the
Selection Window and Unmark this target before using
"swremove" to free disk space.
```

Warning Messages

Warning messages let you know that something unexpected and potentially undesirable occurred. A warning does not prevent the SD session from continuing. Warning messages during analysis of an interactive session give you the chance to continue or stop.

For example, if the fileset SD-DATABASE.SD-DATABASE2 is being installed in multiple locations on a target system, you would find the following warning message in the agent log file:

```
WARNING: A version of fileset "SD-DATABASE.SD-DATABASE2,r=9.00.1C"
is already installed in another location (see previous
lines). Installing this version will create multiple
installed versions. This new multiple version will be
installed because the "allow_multiple_versions" option is
set to "true".
```

Notes

Notes are used to notify you of an event that is not erroneous, unexpected or undesirable, but that you should be aware of:

```
NOTE: The fileset "SD-DATABASE.SD-DATABASE1,r=9.00.1C" is
already installed. If you wish to reinstall this
fileset, change the "reinstall" option to "true".
```

swinstall/swremove Fails With a Lock Error or Hangs

Swinstall or swremove fails with the following message:

```
Cannot lock "/" because another command holds a conflicting lock. The
process id of that command is ####.
```

Another SD command is running that prevents the swinstall or swremove command from running. Wait for that command to finish and try again. It may be necessary to kill broken sessions and to restart the `swagentd` as described below (`swagentd -r`).

Sometimes it is necessary to configure some installed software, first:

```
swconfig \*
```

Cannot Contact Target Host's Daemon or Agent

If you see the following error message:

```
ERROR: Could not contact host <hostname>. Make sure the hostname is
correct.
```

It means that the hostname you specified could not be found in the hosts database. Make sure you have typed the hostname correctly and verify hostnames:

```
# hostname
# ping [hostname]
# nslookup [hostname]
# nslookup [IP adress]
```

If the target hostname is not in the hosts database, but you know its network address, you can use it (in standard "dot" notation) in place of the hostname.

If you see this error message:

```
ERROR:Remote Procedure Call to a daemon has failed.
Could not start a management session for <target>.
Make sure the host is accessible from the network,
and that its daemon, swagentd, is running. If the
daemon is running see the daemon logfile
on this target for more information.
```

It means SD-UX could not contact the daemon program on a specific target system. Note that this may occur even if you haven't specified any targets, for example, if the daemon on your local host is not running.

On the target system, type:

```
ps -e | grep swagentd
```

If the daemon does not appear to be running, you can start it by typing (as root on the target system):

```
/usr/sbin/swagentd
```

If you attempt to start a daemon when one is already running, you will see a message about the other daemon; this is harmless. You can also kill and restart a currently running daemon by typing:

```
/usr/sbin/swagentd –r
```

Slow Network Performance

When using `swinstall` or `swcopy` in an environment where network bandwidth is the "bottleneck," the file transfer rate between source and target can become very slow.

The `compress_files=true` (**Options→Change Options→Compress files during transfer**) option compresses files transferred from a source depot to a target. This can reduce network usage by approximately 50%; the exact amount of compression depends on the type of files. Binary files compress less than 50%, text files more.

Sometimes, the network affects the installation or copy of software even if it is done locally, e.g. from the local tape or CD. The `use_alternate_source=true` option (GUI: **Options→Change Options→All targets to resolve the source locally**) can help.

Debugging SD-UX Commands

Many times, if SD-UX is hanging in a particular function or for a variety of other reasons, SD-UX debug variables can be turned on to get more information generated to the `/var/adm/sw/sw*.log` files.

Example:

```
# export SDU_DEBUG=1
# export SDU_DEBUG_TIMESTAMP=hour:minute:second:millisecond
# script -a /tmp/SDdebug.out
```

Perform SD-UX command.

```
# exit
# unset SDU_DEBUG
# unset SDU_DEBUG_TIMESTAMP
```

Analyze `/tmp/SDdebug.out`.

The variable `SDU_DEBUG` can have a value between 1 and 4:

Level 1: Trace entry to major function calls.
Level 2: Add trace of exit from major function calls.
Level 3: Add trace of some internals or variables in major function calls.
Level 4: Add tracing in functions that are called frequently.

Please remember that the `SDU_DEBUG` variables are neither officially supported nor documented, and their behavior may change without notice.

See also:
http://sdweb.fc.hp.com/projects/sd/doc/general/conventions/using_sdu_debug_envvars.html (HP internal)

swinstall in Single User Mode

swinstall needs the Core networking functionality. So it won´t work in single user mode.

1st of all, try to bring up minimum networking functionality after booting into single user mode:

```
# init 1
# /sbin/init.d/net start
# /sbin/init.d/swagentd start
# /sbin/init.d/swconfig start
```

Now, try to swinstall. If it is still impossible, you may try the following unsupported procedure in single user mode:

`dlpi` and `lan0` drivers must be in the kernel. Then backup the following files:

```
# cp /etc/hosts /etc/hosts.org
# mv /etc/resolv.conf /etc/resolv.conf.org (move!)
```

It might be necessary to kill the `named` process in order to switch off DNS.

`vi` the `/etc/hosts` and replace the `localhost` entry by the actual *hostname*:

```
127.0.0.1       hostname        loopback
```

(Re-)start the `swagentd`:

```
# /usr/sbin/swagentd –r
```

Then use swinstall with the `use_alternate_source=true` option:

```
# swinstall –x use_alternate_source=true ...
```

Good luck! Don´t forget to reboot into single user mode and move the `hosts.orig` and `resolv.conf.orig` files back.

Rebuilding the Master INDEX file

Did you ever had one of the following SD-UX messages?

```
Incomplete definition for the product …
The expected depot or root does not exist at "/".
There is currently no installed software on host…
Ignoring duplicate definition …
Ignoring unknown keyword …
```

It may come from the removal or corruption of the master INDEX file.

The following unsupported procedure should work for HP-UX 10.20, 11.0 and 11i:

```
# mv /var/adm/sw/products/INDEX /tmp/INDEX.bak
# echo fileset > /var/adm/sw/products/INDEX
# swmodify -a title
```

Unbelievable, isn't it?

NOTE: If for some reason not only the master INDEX file got lost but the IPD got corrupted anywhere, it might be necessary to restore `/var/adm/sw/products` from backup.

Additional Information

Software Distributor Website:

http://software.hp.com/products/SD_AT_HP/

Software Depots

http://www.software.hp.com/

swdepot.atl.hp.com (15.51.240.89) (HP internal)
gicsrv10.bbn.hp.com (15.140.138.175) (HP internal)
upserv01.grc.hp.com (15.137.20.53) (HP internal)
sdsms.grc.hp.com (15.140.11.156) (HP internal)
hpcsmaag.bbn.hp.com (15.136.112.68) (HP internal)
haweb.cup.hp.com (15.13.172.89) (HP internal)