NAME

install — copy files or make directories with ownership, mode, and preserved context

SYNOPSIS

install [**-vp**] [**-z**|**-Z**|**--context**[=...]] [**-C**|**-s** [**--strip-program**=*strip*]] [**-b**|**--backup**[=...] [**-S** *suffix*]] [**-m** *mode*] [**-o** *owner*] [**-g** *group*] [**-DT**] *file into*

install [**-vp**] [**-z**|**-Z**|**--context**[=...]] [**-C**|**-s** [**--strip-program**=*strip*]] [**-b**|**--backup**[=...] [**-S** *suffix*]] [**-m** *mode*] [**-o** *owner*] [**-g** *group*] *file...* *into*

install [**-vp**] [**-z**|**-Z**|**--context**[=...]] [**-C**|**-s** [**--strip-program**=*strip*]] [**-b**|**--backup**[=...] [**-S** *suffix*]] [**-m** *mode*] [**-o** *owner*] [**-g** *group*] **-t** *into* *file...*

install [**-v**] [**-Z**|**--context**[=...]] [**-m** *mode*] [**-o** *owner*] [**-g** *group*] **-d** *directory...*

DESCRIPTION

Copies *file*s to *into* with **cp**, also setting their mode and, optionally, ownership. With **-d** acts like **mkdir -p** instead.

You may be looking for `apt`(1).

OPTIONS

Forwarded

-v, **--verbose**	Note file creation and removal. Forwarded to **cp** and **mkdir**.
-p, **--preserve-timestamps**	Copy access and modification times from *file*s. Effectiely forwarded to **cp** as **--preserve=timestamps**.
-Z, **--context**, **--context**=...	Use default/explicit SELinux context for new files and directories. Ignored if SELinux unavailable. Forwarded to **cp** and **mkdir**.
-b, **--backup**, **--backup**=...	Back up existing destination files.
-S, **-suffix**=*suffix*	Both forwarded to **cp**.
-T, **--no-target-directory**	Fail if *into* is a directory instead of copying below it.
-t, **--target-directory**=*into*	Set *into* at the start instead of end. Both provide same calling convention as **cp**.

install-specific

-z, **--preserve-context**	Copy SELinux context from *file*. Ignored if SELinux unavailable.
-C, **--compare**	If the destination file is a regular file with the same content (and, with **-z**, context) as *file*, just set its mode and ownership instead of copying.
-s, **--strip**	Strip destination file of symbols after copying. If this step fails, the file is deleted.
--strip-program=*strip*	Run *strip* instead of `STRIP` or **strip** to do so.

-m, **--mode**=*mode*
Set the mode of the destination files or *directory*ies to `chmod`(1)-style *mode*. The default is `a=rx,u+w`. *umask* is not taken into account. The source mode (for the purposes of ± and copying) is **0**! `X` is set if **-d**. Extended ACLs are cleared, except if **-d**.

-o, **--owner**=*owner* and **-g**, **--group**=*group*
Set the ownership of the destination files or *directory*ies to *owner* and/or *group*, which may be a name or a numeric ID. By default, the ownership is unchanged (so current user/current group, unless the target already existed and wasn't recreated (**-d**, **-C**)).

-D (with **-t**)	Create missing parent directories of *into* with mode a=rx,u+w and default ownership. *umask* is not taken into account.
-D (one *file*)	Create missing parent directories up to the parent of *into*, such that *file* will end up at precisely *into*.
-D (otherwise)	Ignored.
-d, **--directory**	Make *directory*ies and their parents instead of copying. Mimicks **mkdir -p**, but honours **-mog**. Parent directories as with **-D**.
-c	Ignored for compatibility with 3BSD.

ENVIRONMENT

STRIP	Default **strip** for **-s**.
VERSION_CONTROL, SIMPLE_BACKUP_SUFFIX	Used by **cp** for **-b**.

EXIT STATUS

1 if a *file* didn't exist, **cp**/**mkdir**/**strip** exited non-zero, a *file* was copied and another would end up in the same path, or the mode, ownership, context, ACLs, or times couldn't be changed.

SEE ALSO

chcon(1), chmod(1), chown(1), cp(1), mkdir(1), strip(1), acl(5)

STANDARDS

Compatible with the GNU system. The **-z** spelling and STRIP are extensions.

Many systems carry *an* **install** with a similar function but wildly differing calling convention. This implementation is just about compatible with 4.4BSD's and its **-smog** if **-c** were always set. Avoid its use in portable programs, or, indeed, at all.

HISTORY

Version 5 AT&T UNIX has a /usr/sys/ld, linking the right objects into a kernel. It's entirely likely that this or earlier systems had some sort of build script for other non-single-file utilities, but they've all been lost or stored out-of-tree.

Version 6 AT&T UNIX has /usr/source/run, taking the form

```
chdir /usr/sys; pwd; time sh run

chdir /usr/source/as; pwd; time sh run
chdir /usr/source/c; pwd; time sh run
same for s1 through s7, &c.
```

with the first building the kernel and the remaining ones building and installing the (group of) utilities that the directory. Sans the more complex builds (like with c), these largely take a similar form to (head of /usr/source/s1/run):

```
cc -s -f -O ac.c
cmp a.out /usr/bin/ac
cp a.out /usr/bin/ac

as ar.s
strip a.out
cmp a.out /bin/ar
cp a.out /bin/ar

cc -s -O banner.c
cmp a.out /usr/bin/banner
cp a.out /usr/bin/banner

as bas.s
ld -s -n a.out -l
cmp a.out /bin/bas
```

```
    cp a.out /bin/bas

    yacc bc.y
    cc -s -O y.tab.c -ly
    cmp a.out /usr/bin/bc
    cp a.out /usr/bin/bc
    rm y.tab.c
```
and end with **rm** `a.out`.

Version 7 AT&T UNIX has `make`(1) and builds/installs the kernel and non-single-file programs with it, Each of `/usr/src/lib*` gets its own `compall` and `mklib` with the expected functionality, but neither of them install the result.

Single-file programs are handled by `/usr/src/cmd/cmake`, taking advantage of the freshly-variabled-up Bourne **sh**, with the general form of, though spelled via `basename`(1):
```
    for s; do
        echo $s:
        case ${s%.[cys]} in
            basename) cc -n -s -O basename.c -o basename ;;
            bc)       yacc bc.y && mv y.tab.c bc.c &&
                      cc -n -s -O bc.c -o bc && rm bc.c ;;
            &c.
        esac
    done
```
Additionally, if **-cmp** is specified, and the compilation succeeded, **cmp** is run against the currently-installed version; if **-cp** — the new binary is installed. `/usr/src/cmd/in` runs **cmake -cp** with some sub-set of the available programs; there doesn't appear to be a rhyme to the list in the distribution. Though, naturally, one could very much just use **cmake -cp** *.

Programmer's Workbench (PWB/UNIX) has `/sys/source/{lex, sccs4}/install`; the former just builds and runs tests and the latter just installs SCCS commands, mode `a=rx,u+w`, to the first argument.

AT&T System III UNIX has a shell `install`(1M) ("install commands") in `/etc/install` with a **SYNOPSIS** of

install [**-c** dira] [**-f** dirb] [**-i**] [**-n** dirc] [**-o**] [**-s**] file [dirx ...]

which, with no flags, copies *file* to the result of (effecitvely) **find** *$dirx*... `/bin /usr/bin /etc /lib /usr/lib` **-name** *${file*`##*/}` **-print -quit**, or complains if no such file exists. **-i** removes the default directories from that list.
With **-c**, copies *file* to `dira`, but only if it doesn't already exist.
With **-f**, copies *file* to *dirb*, not changing mode or ownership if it already exists.
With **-n**, behaves as default but falls back to copying to *dirc* instead of dying.

New files are reset to mode `a=rx,u+w` and **bin:bin** ownership. Existing files retain their mode and ownership; if they have a `+t` mode, it's cleared, run with no arguments, copied, and then set again.

With **-o**, backs up existing destinations to `OLD`*${file*`##*/}` before overriding.
By default this is a very talkative process and **-s** silences everything sans errors.

There's undocumented support for an undocumented `/etc/syslist` file, which, when guessing where to put *file* (i.e. not in **-cf** mode), is first consulted as, effectively **grep -m***1* `"/`*${file*`##*/}\$"`. If that returns a result, that's used instead of **find**ing.

The system is built via `mk`(8) scripts — distributed as `/usr/src/:mk` to rebuild and install all, and `/usr/src/:mkcmd`, `/usr/src/:mklib`, `/usr/src/:mkstand`, &c. to rebuild and install specific components. `/usr/src/:mkcmd` uses a genericised build process per source file type with build parameters extracted from the source, and `install`(1M) as the installation step for all commands, driven as, essentially,

```
/etc/install -n /usr/lbin $prog
/etc/install -n /usr/bin $prog
```

AT&T System V Release 2 UNIX defaults to the current user's user and group ownership, and only uses **bin : bin** if run by root. **-mug** are added, with the expected semantics à la **-mog**, except **-ug** are ignored with a warning for non-root.

None of this is documented; a similar `cpset`(1M) ("install object files in binary directories") command appears with a **SYNOPSIS** of

> **cpset [-o]** object directory **[**mode owner group**]**

"used by sites to track local command movement", and has mostly-identical defaults, sans searching for existing destinations. It also reads `/usr/src/destinations` as a final path override.

AT&T System V Release 4 UNIX `install`(1M), when overriding, now changes the ownership to the current user and group. **-mug** now also has an effect here (but only if explicitly specified), but only after the first ownership change. The reason this is uncelar. **-m** is now also ignored if not root. The program now lives in `/usr/sbin`.

The manual, in place of the default directory list, for some reason has

> If no options or directories (*dirx* ...) are given, **install** will search a set of default directories (`/usr/usr/bin`, `/usr/usr/usr/bin`, `/etc`, `/usr/usr/lib`, and `/usr/usr/usr/lib`, in that order) for a file with the same name as *file*.

The 1BSD and 2BSD tapes ship with an `install` script for the flagship programs and sorted by target directory, respectibely.

3BSD has an undocumented `/usr/bin/install`, which bears recalling here entirely:

```
cmd=/bin/mv
case $1 in
        -s )    /usr/bin/strip $2
                shift
                ;;
        -c )    cmd=cp
                shift
esac

if [ ! ${2-""} ]
then    echo 'install: no destination specified.'
        exit 1
fi

$cmd $1 $2
if [ -d $2 ]
then    file=$2/$1
else    file=$2
fi
chmod 755 $file
chown root $file
```

For a **install** [**-c**|**-s**] *file into* calling convention, present-day mode setting and present-day-adjacent **-s**. This is the source version; on the archived system, the **chown** is no-oped away (making it present-dayer).

4.2BSD finally documents it in `install`(1) ("install binaries") with a **SYNOPSIS** of

> **install** [**-c**] [**-m** mode] [**-o** owner] [**-g** group] [**-s**] binary destination

extending it with

```
strip=""
chmod="/bin/chmod 755"
chown="/etc/chown -f root"
chgrp="/bin/chgrp -f staff"
```

in the header and updating the tail to

```
if [ -d $2 ]
then    file=$2/$1
else    file=$2
fi
/bin/rm -f $file
$cmd $1 $file
if [ $strip ]
then    $strip $file
fi
$chown $file
$chgrp $file
$chmod $file
```

Protection against moving/copying a file into itself is also added, as is against specifying too many arguments.

-c is unchanged (except that more than one flag may be specified), **-s** sets *strip* to `/bin/strip`, **-mog** work in the expected way. **-m** is as present-day (except it doesn't start with a **0** mode); **-og** would be if the default were empty.

4.3BSD adds a **-f** to **chmod**, and forbids multiple **-c**s, because without **-c**, the original is now stripped *before* being moved.

Through making the **-d** branch *file=$2/*`**basename** *$1*`, **install** `s1/cp /bin` is allowed and works to install `s1/cp` as `/bin/cp` instead of failing to copy to `/bin/s1/cp`. None of this is documented.

4.4BSD sees a C implementation with a present-day **-sog** (though they need to be specified by name) and a **SYNOPSIS** of

install [**-cs**] [**-f** *flags*] [**-g** *group*] [**-m** *mode*] [**-o** *owner*] *file1 file2*
install [**-cs**] [**-f** *flags*] [**-g** *group*] [**-m** *mode*] [**-o** *owner*] *file1 ... fileN directory*

(*file2* may be a directory).

The file *flags* mechanism is new in 4.4BSD and works here like you'd expect. Non-regular-file *file*s are rejected, unless they're `/dev/null` (the latter part is explicitly documented for making empty files). The **HISTORY** section says that "The **install** utility appeared in 4.2BSD.", a blatant lie by like four years.

AT&T System V Release 4 UNIX includes a BSD **install** in `/usr/ucb`, based on a C version that pre-dates the first appearance of the C implementation in 4.3BSD–Tahoe by a full year. It boasts a usage string of

```
usage: install [-cs] [-g group] [-m mode] [-o owner] file ...  destination
       install  -d   [-g group] [-m mode] [-o owner] dir
```

with a numeric-only *mode* and name-only **-og**, **-d** similar to present-day but affected by the umask (on all levels), **-c** as present-day (inasmuch as it's forced on and ignored), and **-s** implemented by

```
sprintf(buf, "strip %s", path);
system(buf);
```