

- [SS64](#)
- [Linux](#) >
- [How-to](#) >

-

cp

Copy one or more files to another location.

Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Syntax

```
cp [options]... Source Dest
cp [options]... Source... Directory
```

Key

| | |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a, --archive | same as -dpR |
| -b, --backup | Make backup before removal. If the copy will overwrite a file in the destination, then the original file will be backed up as 'filename~' before being overwritten. |
| -d, --no-dereference | preserve links |
| -f, --force | remove existing destinations, never prompt |
| -i, --interactive | prompt before overwrite |
| -l, --link | link files instead of copying |
| -p, --preserve | preserve file attributes if possible |
| -P, --parents | append source path to DIRECTORY |
| -r | copy recursively, non-directories as files |
| --sparse=WHEN | control creation of sparse files |
| -R, --recursive | copy directories recursively |
| -s, --symbolic-link | make symbolic links instead of copying |
| -S, --suffix=SUFFIX | override the usual backup suffix |
| -u, --update | copy only when the SOURCE file is newer than the destination file or when the destination file is missing |
| -v, --verbose | explain what is being done |
| -V, --version-control=WORD | override the usual version control |
| -x, --one-file-system | stay on this file system |
| --help | display this help and exit |
| --version | output version information and exit. |

Many users find it useful to set an [alias](#) `cp="cp -iv"` in [bashrc](#), so that progress is always displayed and files do not get overwritten without a confirmation.

By default, sparse SOURCE files are detected by a crude heuristic and the corresponding DEST file is made sparse as well.

That is the behavior selected by `--sparse=auto`.

Specify `--sparse=always` to create a sparse DEST file whenever the SOURCE file contains a long enough sequence of zero bytes.

Use `--sparse=never` to inhibit creation of sparse files.

The backup suffix is `~`, unless set with `SIMPLE_BACKUP_SUFFIX`.

The version control can be set with `VERSION_CONTROL`, values are:

- none, off
- never make backups (even if `--backup` is given)

numbered, t
make numbered backups

existing, nil
numbered if numbered backups exist, simple otherwise

simple, never
always make simple backups

As a special case, cp makes a backup of SOURCE when the force and backup options are given and SOURCE and DEST are the same name for an existing, regular file.

As a special case, cp makes a backup of SOURCE when the force and backup options are given and SOURCE and DEST are the same name for an existing, regular file.

Examples

Copy demofile to demofile.bak :

```
$ cp demofile demofile.bak  
or  
$ cp demofile{,.bak}
```

With variables make sure you quote everything:

```
$ cp "$SOURCE" "$DEST"
```

Copy demofile.txt to demofile.bak :

```
$ FILE="demofile.txt"  
$ cp "$FILE" "${FILE%.*}.bak"
```

Copy floppy to home directory:

```
$ cp -f /mnt/floppy/* ~
```

Copy all .jpg files to the CA folder, and for those with "New York" in the filename, replace with "California_"
the "\${f/New York/California_}" is an application of [bash parameter expansion](#)

```
$ mkdir CA  
$ for f in *.jpg; do cp "$f" "CA/${f/New York/California_}"; done
```

"Thank you for sending me a copy of your book; I'll waste no time reading it" ~ Moses Hadas

Related linux commands

[dd](#) - Data Duplicator - convert and copy a file (use for RAW storage.)

[cpio](#) - Copy files to and from archives.

[install](#) - Copy files and set attributes.

[mv](#) - Move files.

[rsync](#) - Remote file copy (Synchronize file trees).

[tar](#) - store or extract files to an archive (allows symbolic links to be copied as links).

Equivalent Windows command: [COPY](#) - Copy one or more files to another location.