# diff and patch

Compare and patch files

## CONCEPTS

diff makes the patch, patch applies it. Always preview with --dry-run first.

**Old + new files** two versions
↓ `diff -u`

**Patch file** + and - lines
↓ `patch -p1`

**Patched tree** new version

**diff exit status**

| Code | Meaning |
|---|---|
| 0 | no differences |
| 1 | files differ |
| 2 | trouble (missing file, bad option) |

In a patch, a/ and b/ prefixes are path components: -p1 strips one, -p0 none.

## COMPARE FILES

**diff** *<a> <b>*
Show differences (normal format)

**diff -u** *<a> <b>*
Unified diff, the patch format

**diff -q** *<a> <b>*
Only say whether files differ

**diff --color=auto -u** *<a> <b>*
Colorize added and removed lines

**diff -y** *<a> <b>*
Two columns side by side

**diff -U** *<n> <a> <b>*
Unified diff with n context lines

**diff** *<(<cmd1>) <(<cmd2>)*
Compare the output of two commands

## COMPARE DIRECTORIES

**diff -rq** *<dirA> <dirB>*
List files that differ or exist on one side

**diff -r** *<dirA> <dirB>*
Show all differences recursively

**diff -ru** *<dirA> <dirB>*
Recursive unified diff

**diff -rN** *<dirA> <dirB>*
Show added and removed files as full diffs

**diff -rq -x '***<glob>***'** *<dirA> <dirB>*
Skip files matching a glob

## CREATE PATCHES

**diff -u** *<old> <new>* **>** *<name>***.patch**
Patch for a single file

**diff -ruN** *<old-dir> <new-dir>* **>** *<name>***.patch**
Patch for a directory tree

**cp -r** *<dir> <dir>***.orig**
Keep a pristine copy before editing

**diff -ruN** *<dir>***.orig** *<dir>* **>** *<name>***.patch**
Diff the edited tree against the copy

Put the old tree first and the new one second, or the patch is reversed.

## APPLY PATCHES

**patch -p1 <** *<name>***.patch**
Apply a tree patch from its top directory

**patch** *<file>* **<** *<name>***.patch**
Apply a patch to one named file

**patch -p1 --dry-run <** *<name>***.patch**
Show what would happen, change nothing

**patch -R -p1 <** *<name>***.patch**
Reverse an applied patch

**patch -p1 -d** *<dir>* **<** *<name>***.patch**
Apply inside another directory

**-p levels for a/src/main.c**

| Option | File patched |
|---|---|
| -p0 | a/src/main.c |
| -p1 | src/main.c |
| -p2 | main.c |

## TROUBLESHOOTING

**find . -name '*.rej'**
Find hunks that did not apply

**cat** *<file>***.rej**
Read a rejected hunk, then apply by hand

patch applies hunks that match and rejects the rest, so a failed run can leave a half-patched tree. Use --dry-run, or a clean checkout.

Wrong -p value gives "can't find file to patch". Compare the paths in the patch header with your current directory.