Difftastic Structural diff tool

START HERE

`difft <old> <new>`
Compare two files

`git log -p --ext-diff`
History with difftastic

`difft <dir1> <dir2>`
Compare two directories

`difft --display inline <a> <b>`
One column output

`git -c diff.external=difft diff`
Structural git diff once

`difft --skip-unchanged <d1> <d2>`
Hide identical files

`git config --global diff.external difft`
Make it the default

`difft --list-languages`
Supported languages

Two inputs files, dirs, stdin → `detect language` → **Parse** tree-sitter syntax tree → `graph search` → **Structural diff** nodes added / removed → `too big or parse error` → **Line diff fallback** or plain text

CONCEPTS

difftastic compares syntax trees, not lines: reformatting and re-wrapping show up as no change, and moved code stays aligned.

If a file cannot be parsed, is over the byte or graph limit, or has no known language, difft falls back to a line-based diff.

difft is a viewer, not a patch tool: its output cannot be applied with patch or git apply.

BASICS

`difft <old> <new>`
Diff two files

`difft <dir1> <dir2>`
Diff two directories recursively

`difft - <file>`
Read one side from stdin

`difft <conflicted-file>`
Diff the two sides of conflict markers
Needs a file containing <<<<<<< markers; added in 0.50.

`difft --help`
List every option

`difft --version`
Print installed version

`difft --list-languages`
Show languages and extensions

`difft --skip-unchanged <d1> <d2>`
Show only files that differ

`difft --sort-paths <d1> <d2>`
Sort directory output by path

`difft --exit-code <a> <b>`
Exit 1 when there are changes
Without it the exit code is 0 even when files differ. 2 means bad arguments or unreadable paths.

`difft --check-only <a> <b>`
Only report whether they differ

DISPLAY

`difft --display side-by-side <a> <b>`
Two columns (default)

DISPLAY (cont.)

`difft --display side-by-side-show-both <a> <b>`
Two columns even for pure adds

`difft --display inline <a> <b>`
Single column, diff style

`difft --display json <a> <b>`
Machine-readable output
Unstable: also needs DFT_UNSTABLE set.

`difft --context <n> <a> <b>`
Lines of context around changes

`difft --width <cols> <a> <b>`
Force the wrapping width

`difft --tab-width <n> <a> <b>`
Spaces per tab

`difft --color always <a> <b>`
Force colour (also never / auto)

`difft --background light <a> <b>`
Colours for a light terminal

`difft --syntax-highlight off <a> <b>`
Plain colours, no highlighting

Use --color always when piping into a pager: less -R keeps the colours.

LANGUAGES AND LIMITS

`difft --override '<glob>:<lang>' <a> <b>`
Force a language by file glob
Repeatable; first match wins. Language 'text' forces plain text.

`difft --override '*.conf:text' <a> <b>`
Treat matching files as text

`difft --override-binary '<glob>' <a> <b>`
Always treat matching files as binary

`difft --ignore-comments <a> <b>`
Ignore comment changes

`difft --strip-cr off <a> <b>`
Keep carriage returns in the diff

LANGUAGES AND LIMITS (cont.)

`difft --byte-limit <n> <a> <b>`
Size above which it uses a line diff
Default 1000000 bytes.

`difft --graph-limit <n> <a> <b>`
Cap work on huge diffs
Default 3000000 graph vertices; raise it if you hit the fallback.

`difft --parse-error-limit <n> <a> <b>`
Parse errors tolerated before line diff

GIT

`git -c diff.external=difft diff`
Use difft for one command

`git config --global diff.external difft`
Use difft for every git diff

`git show --ext-diff <commit>`
Show a commit with difft
show and log -p only use the external diff with --ext-diff.

`git log -p --ext-diff`
History with structural diffs

`git diff --no-ext-diff`
Back to the normal diff

`git diff --ext-diff --staged`
Staged changes with difft

`git difftool -t difftastic`
Run via difftool

`git config --global difftool.prompt false`
No prompt per file

`git config --global diff.tool difftastic`
Make difftastic the difftool

`git config --global pager.difftool true`
Page difftool output

`GIT_EXTERNAL_DIFF=difft git diff`
Select it by environment

diff.external is preferred over difftool: git passes it extra information such as renames and mode changes. Git 2.43.1 and older can crash on mode changes; use difftool there.

CONFIGURATION

`DFT_BACKGROUND=light`
Environment form of --background

`DFT_DISPLAY=inline`
Default display mode

`DFT_CONTEXT=<n>`
Default context lines

`DFT_COLOR=always`
Default colour policy

`DFT_SKIP_UNCHANGED=1`
Always skip identical files

`DFT_OVERRIDE='<glob>:<lang>'`
Language override from the environment

Every option has a DFT_<NAME> environment variable, which is the way to set defaults when a VCS invokes difft for you.

RECIPES

Git aliases

```
[alias]
    dft = -c diff.external=difft diff
    dl  = -c diff.external=difft log -p --ext-diff
    ds  = -c diff.external=difft show --ext-diff
```

Difftool config

```
[diff]
    tool = difftastic
[difftool]
    prompt = false
[difftool "difftastic"]
    cmd = difft "$LOCAL" "$REMOTE"
[pager]
    difftool = true
```

Quick review of two trees

```
difft --skip-unchanged --display inline old/ new/
# exit status for scripts
difft --exit-code --check-only a.py b.py
```

Keep difft colours in a pager

```
GIT_EXTERNAL_DIFF=difft git -c core.pager='less -R' diff --color=always
# or one-off
DFT_COLOR=always git diff | less -R
```