GNU Stow

Symlink farm manager

CONCEPTS

Each package is a directory that mirrors the target tree. Stow symlinks its contents, so ~/dotfiles/vim/.vimrc becomes ~/.vimrc.

Stow dir ~/dotfiles
↓ `stow <pkg>`
Package vim/.vimrc
↓ `symlink`
Target dir ~/.vimrc

The default target is the parent of the stow dir, so keep the repo in ~/dotfiles or pass -t ~.

Stow only removes symlinks it owns and reports a conflict rather than overwriting a real file.

STOW AND UNSTOW

`stow <pkg>...`
Link packages (default action)

`stow -S <pkg>`
Explicit stow action

`stow -D <pkg>`
Delete a package's links

`stow -R <pkg>`
Restow: unlink then link again

`stow -n <pkg>`
Simulate, change nothing

`stow -v <pkg>`
Verbose output (repeat up to -vvvvv)

DIRECTORIES

`stow -d <dir> <pkg>`
Set the stow directory

`stow -t <dir> <pkg>`
Set the target directory

`stow -t ~ <pkg>`
Link into your home directory

`STOW_DIR=<dir> stow <pkg>`
Stow dir from environment

DOTFILES AND CONFLICTS

`stow --dotfiles <pkg>`
Turn dot- prefix into a leading dot

`stow --adopt <pkg>`
Move existing target files into package

`stow --no-folding <pkg>`
Link files, never whole directories

--adopt rewrites the files in your stow dir. Commit first so git diff shows what it took, then git checkout . to keep your repo version.

IGNORING AND CONFIG

`stow --ignore=<regex> <pkg>`
Skip names ending in a Perl regex

`~/.stowrc`
Default options, one per line

`.stow-local-ignore`
Per-package ignore patterns file

TREE FOLDING

Folding behaviour

Situation	Result
`Target dir missing`	One symlink to the whole directory
`Target dir already real`	Links go inside it, file by file
`Second package needs folded dir`	Symlink split into real dir, both linked
`Package removed, one left`	Directory refolded to one symlink
`--no-folding`	Always file-level links

If an app writes new files into a folded directory they land in your repo. Use --no-folding or pre-create the directory.