



MACOS 14 AND LATER • ZSH • HOMEBREW

macOS Handbook

For advanced users

Run the system from the inside. Homebrew and Brewfiles, hidden settings with defaults, launchd, logs, networking, security (SIP, Gatekeeper, Touch ID for sudo), APFS and Time Machine, BSD tool differences and Recovery.

CONTENTS

1. Homebrew in depth
2. Hidden settings with defaults
3. launchd: services and scheduled jobs
4. Logs and diagnostics
5. Network commands
6. Security
7. Users and permissions
8. Disks, APFS and Time Machine
9. Updates and installers
10. Handy macOS-only commands
11. Text processing and pipelines
12. Archives, backups, syncing
13. Customizing zsh
14. Performance and power
15. Recovery and reinstalling

Yellow italic text (like *<package>*) marks the parts you fill in. In Terminal, copy is Cmd+C and paste is Cmd+V. Commands that start with `sudo` change the system; read what they do before running them.

Homebrew in depth

```
% brew bundle dump --file=~/.Brewfile
```

Writes every installed package, cask and App Store app to a Brewfile.

```
% brew bundle --file=~/.Brewfile
```

Installs them all on a new Mac with one command.

```
% brew leaves
```

Packages you installed yourself that nothing else depends on.

```
% brew autoremove
```

Removes dependencies that are no longer needed.

```
% brew pin <package>
```

Pins a package against upgrades. `unpin` releases it.

```
% brew deps --tree <package>
```

Dependency tree. Reverse: `brew uses --installed <package>`.

```
% brew services list
```

Services installed with Homebrew (PostgreSQL, Redis...).

```
% brew services start <package>
```

Starts a service and runs it at login.

```
% brew tap <user>/<repo>
```

Adds a third-party formula repository.

```
% brew doctor
```

Reports problems with the Homebrew installation.

```
% brew install mas && mas upgrade
```

Updates App Store apps from the command line.

Hidden settings with defaults

App settings live in plist files; `defaults` reads and writes them. After a change you need to restart the app in question.

```
% defaults write com.apple.finder  
AppleShowAllFiles -bool true && killall  
Finder
```

Always shows hidden files in Finder.

```
% defaults write com.apple.finder ShowPathbar  
-bool true && killall Finder
```

Path bar in Finder.

```
% defaults write com.apple.dock autohide-  
delay -float 0 && killall Dock
```

An auto-hiding Dock opens without delay.

```
% defaults write com.apple.screencapture  
location ~/Pictures/Screenshots && killall  
SystemUIServer
```

Folder for screenshots (create it first).

```
% defaults write com.apple.screencapture  
disable-shadow -bool true
```

Removes the shadow from window screenshots.

```
% defaults write -g ApplePressAndHoldEnabled  
-bool false
```

Key repeat instead of the accent menu when holding a key (log in again).

```
% defaults write -g KeyRepeat -int 2
```

Faster key repeat (lower is faster).

```
% defaults read com.apple.dock
```

All settings of an app.

```
% defaults delete com.apple.dock autohide-  
delay && killall Dock
```

Resets a setting to its default.

launchd: services and scheduled jobs

On macOS launchd manages services and scheduled jobs. User services are defined by plist files in `~/Library/LaunchAgents/`, system services in `/Library/LaunchDaemons/`.

```
% launchctl list | grep -v com.apple
```

Loaded non-Apple user services.

```
% launchctl print gui/$(id -u)/<Label>
```

Detailed status of a service.

```
% launchctl bootstrap gui/$(id -u)  
~/Library/LaunchAgents/<name>.plist
```

Loads and starts a user service.

```
% launchctl bootout gui/$(id -u)/<Label>
```

Stops and unloads a service.

```
% launchctl kickstart -k gui/$(id -u)/<Label>
```

Restarts a service.

```
% sudo launchctl list
```

System services.

TIP In a plist, the `StartCalendarInterval` key runs a job at a set time, `StartInterval` at set intervals; it's the macOS answer to cron.

Logs and diagnostics

```
% log show --last 30m --predicate  
'messageType == error' --style compact
```

Errors from the last 30 minutes.

```
% log stream --predicate 'process == "  
<process>"'
```

Follows a process's log live.

```
% log show --last 1h --predicate  
'eventMessage contains "<text>"'
```

Searches the log for text.

```
% open -a Console
```

The Console app: logs and crash reports.

```
% ls ~/Library/Logs/DiagnosticReports
```

App crash reports.

```
% sudo sysdiagnose
```

A comprehensive diagnostics archive to send to Apple Support.

Network commands

```
% ipconfig getifaddr en0
```

IP address of the Wi-Fi interface.

```
% networksetup -listallhardwareports
```

Network interfaces and their names.

```
% networksetup -getdnsservers Wi-Fi
```

DNS servers.

```
% sudo networksetup -setdnsservers Wi-Fi  
1.1.1.1 9.9.9.9
```

Changes the DNS servers. Use `Empty` to go back to the default.

```
% sudo dscacheutil -flushcache; sudo killall  
-HUP mDNSResponder
```

Clears the DNS cache.

```
% lsof -iTCP -sTCP:LISTEN -n -P
```

Listening ports and their programs.

```
% networkQuality
```

Built-in internet speed test.

```
% route -n get default
```

The default gateway.

```
% scutil --dns
```

Detailed DNS configuration.

```
% ping -c 4 <address>
```

Reachability test. `traceroute` shows the route.

Security

```
% fdesetup status
```

FileVault status.

```
% csrutil status
```

System Integrity Protection (SIP) status. It can only be turned off from Recovery; not recommended.

```
% spctl --status
```

Gatekeeper status.

```
%  
/usr/libexec/ApplicationFirewall/socketfilterfw --getglobalstate
```

Application firewall status. To turn it on: `sudo ... --setglobalstate on`.

```
% codesign -dv --verbose=4  
/Applications/<App>.app
```

An app's signature and developer.

```
% sudo cp /etc/pam.d/sudo_local.template  
/etc/pam.d/sudo_local && sudo nano  
/etc/pam.d/sudo_local
```

Touch ID for sudo: remove the # in front of the `auth sufficient pam_tid.so` line (macOS 14 and later; survives updates).

```
% tccutil reset All <bundle-id>
```

Resets all privacy permissions of an app (e.g. `com.apple.Terminal`).

TIP Terminal needs permission for some folders (Documents, Desktop, external drives). For batch jobs, add Terminal to System Settings > Privacy & Security > Full Disk Access.

Users and permissions

On macOS sudo rights come from membership in the `admin` group.

```
% id
```

Your user ID and groups.

```
% dscl . list /Users | grep -v '^_'
```

Real user accounts.

```
% sudo dseditgroup -o edit -a <user> -t user  
admin
```

Makes a user an administrator.

```
% chmod +x <file>
```

Makes it executable.

```
% sudo chown -R <user>:staff <path>
```

Changes the owner (users' default group on macOS is `staff`).

```
% ls -le <path>
```

Lists with ACL permissions.

Disks, APFS and Time Machine

```
% diskutil apfs list
```

APFS containers and volumes.

```
% diskutil info /
```

Details of the startup volume.

```
% diskutil verifyVolume /
```

Checks a volume. For repairs, Disk Utility > First Aid.

```
% tmutil listlocalsnapshots /
```

Local Time Machine snapshots.

```
% sudo tmutil deletelocalsnapshots <date>
```

Deletes a specific local snapshot (date from the list above).

```
% tmutil startbackup
```

Starts a Time Machine backup now. `tmutil latestbackup` shows the last one.

```
% du -sh ~/Library/* 2>/dev/null | sort -h | tail
```

What takes the most space in your Library folder.

```
% hdiutil create -srcfolder <folder> archive.dmg
```

Creates a disk image from a folder.

Updates and installers

```
% softwareupdate --list
```

Pending updates.

```
% sudo softwareupdate -ia --restart
```

Installs all updates and restarts if needed.

```
% softwareupdate --list-full-installers
```

Full macOS installers available for download.

```
% softwareupdate --fetch-full-installer --full-installer-version <version>
```

Downloads the full installer to Applications.

```
% sudo /Applications/Install\ macOS\ <Name>.app/Contents/Resources/createinstallmedia --volume /Volumes/<USB>
```

Makes a bootable macOS installer USB (the USB is erased).

```
% softwareupdate --install-rosetta --agree-to-license
```

Installs Rosetta 2 for Intel apps on Apple silicon.

Handy macOS-only commands

```
% sips -Z 1600 *.jpg
```

Resizes images so the longest side is 1600 pixels (overwrites the files).

```
% sips -s format png <image>.heic --out  
<image>.png
```

Converts a HEIC photo to PNG.

```
% textutil -convert docx <file>.rtf
```

Converts document formats (txt, rtf, html, docx).

```
% screencapture -T 5 ~/Desktop/screen.png
```

Takes a screenshot after 5 seconds.

```
% say "<text>"
```

Reads text aloud. To be told when a long job finishes: `long-command; say done`.

```
% mdls <file>
```

A file's Spotlight metadata (size, resolution, download source).

```
% open -a "<App>" --args <parameter>
```

Starts an app with parameters.

Text processing and pipelines

grep, sed and find on macOS are the BSD versions; some options differ from Linux. For the GNU versions: `brew install coreutils gnu-sed grep findutils` (installed as gsed, ggrep, gfind).

```
% grep -rni "<text>" <folder>
```

Recursive, case-insensitive search with line numbers.

```
% sed -i '' 's/<old>/<new>/g' <file>
```

Find and replace in a file. BSD sed needs empty quotes (`''`) after `-i`.

```
% awk '{print $1, $3}' <file>
```

1st and 3rd column.

```
% sort <file> | uniq -c | sort -rn | head
```

Most frequently repeated lines.

```
% find . -name "*.log" -mtime +7 -delete
```

Deletes .log files older than 7 days.

```
% <command> | tee <file>
```

Writes output to both the screen and a file.

```
% diff -u <old> <new>
```

Differences between two files.

```
% curl -s <URL> | jq '.'
```

Pretty-prints JSON (`brew install jq`).

Archives, backups, syncing

```
% tar -czvf <archive>.tar.gz <folder>
```

gzip archive.

```
% tar -xvf <archive>
```

Extracts an archive.

```
% ditto -c -k --keepParent <folder>  
<archive>.zip
```

The same zip as Finder's "Compress"; keeps macOS metadata.

```
% zip -r <archive>.zip <folder> -x  
"*.DS_Store"
```

Leaves .DS_Store files out of a zip headed for Windows.

```
% rsync -avh --progress <source>/ <target>/
```

Syncs a folder; copies only what changed.

```
% rsync -avh --delete --dry-run <source>/  
<target>/
```

Mirror; see what would be deleted first with `-dry-run`.

Customizing zsh

zsh settings are in `~/.zshrc` in your home folder; things to run once at login go in `~/.zprofile`.

```
% nano ~/.zshrc
```

Opens the shell configuration file (creates it if missing).

```
alias update='brew update && brew upgrade &&  
softwareupdate --list'
```

An update check in one word.

```
alias ll='ls -lahG'
```

Colored detailed listing.

```
export PATH="$HOME/bin:$PATH"
```

Adds a folder to PATH.

```
% source ~/.zshrc
```

Loads the changes.

```
% brew install zsh-autosuggestions zsh-  
syntax-highlighting
```

Suggestions from history and syntax highlighting; add the `source` lines printed at the end of the install to `.zshrc`.

Performance and power

```
% pmset -g
```

Power settings.

```
% pmset -g assertions
```

Processes keeping the Mac from sleeping.

```
% sudo pmset -a lowpowermode 1
```

Turns on Low Power Mode (0 turns it off).

```
% system_profiler SPPowerDataType | grep -E  
'Cycle Count|Condition|Maximum Capacity'
```

Battery cycle count and health.

```
% sudo powermetrics --samplers  
cpu_power,gpu_power -i 1000 -n 5
```

Measures CPU and GPU power draw.

```
% sudo purge
```

Flushes the disk cache (before taking measurements).

Recovery and reinstalling

TIP Entering Recovery. Apple silicon: with the Mac off, hold the power button; when “Loading startup options” appears, choose Options. Intel: hold Cmd+R at startup.

TIP Safe Mode. Apple silicon: hold the power button to reach the options, select the startup disk, hold Shift and choose “Continue in Safe Mode”. Intel: hold Shift at startup. For weeding out problem login items and extensions.

Recovery > Disk Utility > First Aid

Repair the container first, then the volumes.

Recovery > Reinstall macOS

Reinstalls the system; personal files and apps stay in place.

Recovery > Restore from Time Machine

Returns the whole system to a backup.

```
% resetpassword
```

Opens the password reset assistant from Terminal in Recovery.

TIP If an Apple silicon Mac won't start at all, Apple Configurator on another Mac can "Revive" it (data stays) or "Restore" it (everything is erased) in DFU mode.

Commands are written for macOS 14 Sonoma and later (Apple silicon and Intel). Menu names follow the English interface. Commands that start with `sudo` change the system; read what they do before running them.