



☰ macOS companion app > **macOS Release**

macOS companion app

macOS Release

This app now ships Sparkle auto-updates. Release builds must be Developer ID-signed, zipped, and published with a signed appcast entry.

Prereqs

Developer ID Application cert installed (example: `Developer ID Application: <Developer Name> (<TEAMID>)`).

Sparkle private key path set in the environment as `SPARKLE_PRIVATE_KEY_FILE` (path to your Sparkle ed25519 private key; public key baked into Info.plist). If it is missing, check `~/.profile` .

Notary credentials (keychain profile or API key) for `xcrun notarytool` if you want Gatekeeper-safe DMG/zip distribution.

We use a Keychain profile named `openclaw-notary` , created from App Store Connect API key env vars in your shell profile:

```
APP_STORE_CONNECT_API_KEY_P8 , APP_STORE_CONNECT_KEY_ID ,  
APP_STORE_CONNECT_ISSUER_ID  
  
echo "$APP_STORE_CONNECT_API_KEY_P8" | sed 's/\\n/\\n/g' >  
/tmp/openclaw-notary.p8  
  
xcrun notarytool store-credentials "openclaw-notary" --key  
/tmp/openclaw-notary.p8 --key-id "$APP_STORE_CONNECT_KEY_ID" --issuer  
"$APP_STORE_CONNECT_ISSUER_ID"
```



```
pnpm deps installed ( pnpm install --config.node-linker=hoisted ).
```

```
Sparkle tools are fetched automatically via SwiftPM at  
apps/macos/.build/artifacts/sparkle/Sparkle/bin/ ( sign_update ,  
generate_appcast , etc.).
```

Build & package

Notes:

`APP_BUILD` maps to `CFBundleVersion` / `sparkle:version` ; keep it numeric + monotonic (no `-beta`), or Sparkle compares it as equal.

Defaults to the current architecture (`$(uname -m)`). For release/universal builds, set `BUILD_ARCHS="arm64 x86_64"` (or `BUILD_ARCHS=all`).

Use `scripts/package-mac-dist.sh` for release artifacts (zip + DMG + notarization). Use `scripts/package-mac-app.sh` for local/dev packaging.



```

# From repo root; set release IDs so Sparkle feed is enabled.
# APP_BUILD must be numeric + monotonic for Sparkle compare.

BUNDLE_ID=bot.molt.mac \
APP_VERSION=2026.2.18 \
APP_BUILD="$(git rev-list --count HEAD)" \
BUILD_CONFIG=release \
SIGN_IDENTITY="Developer ID Application: <Developer Name> (<TEAMID>)" \
scripts/package-mac-app.sh

# Zip for distribution (includes resource forks for Sparkle delta support)
ditto -c -k --sequesterRsrc --keepParent dist/OpenClaw.app dist/OpenClaw-2026.2.18

# Optional: also build a styled DMG for humans (drag to /Applications)
scripts/create-dmg.sh dist/OpenClaw.app dist/OpenClaw-2026.2.18.dmg

# Recommended: build + notarize/staple zip + DMG
# First, create a keychain profile once:
#   xcrun notarytool store-credentials "openclaw-notary" \
#     --apple-id "<apple-id>" --team-id "<team-id>" --password "<app-specific-pass>"
NOTARIZE=1 NOTARYTOOL_PROFILE=openclaw-notary \
BUNDLE_ID=bot.molt.mac \
APP_VERSION=2026.2.18 \
APP_BUILD="$(git rev-list --count HEAD)" \
BUILD_CONFIG=release \
SIGN_IDENTITY="Developer ID Application: <Developer Name> (<TEAMID>)" \
scripts/package-mac-dist.sh

# Optional: ship dSYM alongside the release
ditto -c -k --keepParent apps/macos/.build/release/OpenClaw.app.dSYM dist/OpenClaw

```

Appcast entry

Use the release note generator so Sparkle renders formatted HTML notes:

```
SPARKLE_PRIVATE_KEY_FILE=/path/to/ed25519-private-key scripts/make_appcast
```

Generates HTML release notes from `CHANGELOG.md` (via [scripts/changelog-to-html.sh](#)) and embeds them in the appcast entry. Commit the updated `appcast.xml` alongside the release assets (zip + dSYM) when publishing.

>

Publish & verify

Upload `OpenClaw-2026.2.18.zip` (and `OpenClaw-2026.2.18.dSYM.zip`) to the GitHub release for tag `v2026.2.18`.

Ensure the raw appcast URL matches the baked feed:

`https://raw.githubusercontent.com/openclaw/openclaw/main/appcast.xml`.

Sanity checks:

```
curl -I
https://raw.githubusercontent.com/openclaw/openclaw/main/appcast.xml
returns 200.
```

```
curl -I <enclosure url> returns 200 after assets upload.
```

On a previous public build, run “Check for Updates...” from the About tab and verify Sparkle installs the new build cleanly.

Definition of done: signed app + appcast are published, update flow works from an older installed version, and release assets are attached to the GitHub release.

< macOS Signing

Gateway on macOS >

Powered by [mintlify](#)