



☰ macOS companion app > macOS Dev Setup

macOS companion app

macOS Dev Setup

This guide covers the necessary steps to build and run the OpenClaw macOS application from source.

Prerequisites

Before building the app, ensure you have the following installed:

1. **Xcode 26.2+:** Required for Swift development.
2. **Node.js 22+ & pnpm:** Required for the gateway, CLI, and packaging scripts.

1. Install Dependencies

Install the project-wide dependencies:

```
pnpm install
```

2. Build and Package the App

To build the macOS app and package it into `dist/OpenClaw.app` , run:

```
./scripts/package-mac-app.sh
```

If you don't have an Apple Developer ID certificate, the script will automatically use **ad-hoc signing** (-).

For dev run modes, signing flags, and Team ID troubleshooting, see the macOS app README:

<https://github.com/openclaw/openclaw/blob/main/apps/macos/README.md>

Note: Ad-hoc signed apps may trigger security prompts. If the app crashes immediately with "Abort trap 6", see the [Troubleshooting](#) section.

3. Install the CLI

The macOS app expects a global `openclaw` CLI install to manage background tasks.

To install it (recommended):

1. Open the OpenClaw app.
2. Go to the **General** settings tab.
3. Click **"Install CLI"**.

Alternatively, install it manually:

```
npm install -g openclaw@<version>
```

Troubleshooting

Build Fails: Toolchain or SDK Mismatch

The macOS app build expects the latest macOS SDK and Swift 6.2 toolchain.

System dependencies (required):



Latest macOS version available in Software Update (required by Xcode 26.2 SDKs)

Xcode 26.2 (Swift 6.2 toolchain)

Checks:

```
xcodesbuild -version
xcrun swift --version
```

If versions don't match, update macOS/Xcode and re-run the build.

App Crashes on Permission Grant

If the app crashes when you try to allow **Speech Recognition** or **Microphone** access, it may be due to a corrupted TCC cache or signature mismatch.

Fix:

1. Reset the TCC permissions:

```
tccutil reset All bot.molt.mac.debug
```

2. If that fails, change the `BUNDLE_ID` temporarily in to force a “clean slate” from macOS.

Gateway “Starting...” indefinitely

If the gateway status stays on “Starting...”, check if a zombie process is holding the port:



```
openclaw gateway status  
openclaw gateway stop
```

```
# If you're not using a LaunchAgent (dev mode / manual runs), find the listener:  
lsof -nP -iTCP:18789 -sTCP:LISTEN
```

If a manual run is holding the port, stop that process (Ctrl+C). As a last resort, kill the PID you found above.

< iOS App

Menu Bar >

Powered by [mintlify](#)