Media and devices

Voice Wake

OpenClaw treats **wake words as a single global list** owned by the **Gateway**.

> There are **no per-node custom wake words**.
>
> **Any node/app UI may edit** the list; changes are persisted by the Gateway and broadcast to everyone.
>
> Each device still keeps its own **Voice Wake enabled/disabled** toggle (local UX + permissions differ).

Storage (Gateway host)

Wake words are stored on the gateway machine at:

```
~/.openclaw/settings/voicewake.json
```

Shape:

```
{ "triggers": ["openclaw", "claude", "computer"], "updatedAtMs": 17300
```

Protocol

Methods

```
voicewake.get  →  { triggers: string[] }
```

`voicewake.set` with params `{ triggers: string[] }` → `{ triggers: string[] }`

Notes: ›

- Triggers are normalized (trimmed, empties dropped). Empty lists fall back to defaults.
- Limits are enforced for safety (count/length caps).

Events

`voicewake.changed` payload `{ triggers: string[] }`

Who receives it:

- All WebSocket clients (macOS app, WebChat, etc.)
- All connected nodes (iOS/Android), and also on node connect as an initial “current state” push.

Client behavior

macOS app

- Uses the global list to gate `VoiceWakeRuntime` triggers.
- Editing “Trigger words” in Voice Wake settings calls `voicewake.set` and then relies on the broadcast to keep other clients in sync.

iOS node

- Uses the global list for `VoiceWakeManager` trigger detection.
- Editing Wake Words in Settings calls `voicewake.set` (over the Gateway WS) and also keeps local wake-word detection responsive.

Android node

Exposes a Wake Words editor in Settings.

Calls `voicewake.set` over the Gateway WS so edits sync everywhere.

‹ Talk Mode

Location Command ›