

WhispSkrid — Manuel utilisateur

Installer et utiliser la dictée appui-pour-parler hors ligne

Version v1.0.1

2026-09-18

Table des matières

Chapitre 1 — Installer	2
En deux minutes	2
Prérequis	2
Deux façons d’installer	3
Récupérer un modèle	4
Premier démarrage	4
Une première dictée	4
Désinstaller et mettre à jour	5
Chapitre 2 — Utiliser	6
Pour qui est cet outil	6
Le modèle d’interaction : appui-pour-parler strict	6
Deux voies pour déclencher une dictée	6
Session persistante et client	7
Configuration	7
Gérer les modèles	9
Récapitulatif de la ligne de commande	10
Chapitre 3 — Dépanner	11
Comment le texte arrive dans l’application	11
Le microphone	11
Les outils d’injection et leur choix automatique	11
Raccourcis : X11 et Wayland ne sont pas égaux	12
--diagnose	12
Symptôme → cause probable → action	12
Où chercher un message d’erreur	14
Si le problème persiste	14

Chapitre 1 — Installer

En deux minutes

Pour qui veut dicter tout de suite, sans lire le reste.

1. Installer le paquet natif de votre distribution.

```
# Debian 12 et 13, Ubuntu 22.04 et 24.04 et leurs dérivées
sudo apt install ./whispskrid_1.0.1_all.deb
```

```
# Fedora 42
sudo dnf install ./whispskrid-1.0.1-1.fc42.noarch.rpm
```

```
# openSUSE Leap 15.6
sudo zypper install ./whispskrid-1.0.1-1.leap156.noarch.rpm
```

```
# Arch Linux
sudo pacman -U whispskrid-1.0.1-1-any.pkg.tar.zst
```

2. Récupérer un modèle **Whisper**, si aucun n'est présent (le paquet n'en embarque pas — trop volumineux) :

```
whispskrid --download-model base
```

Le modèle base est téléchargé, vérifié, puis rangé dans `~/.local/share/whispskrid/whisper-models/`. Au tout premier lancement sans aucun modèle, l'outil propose ce même téléchargement de façon interactive.

3. Lancer l'outil.

```
whispskrid
```

Il affiche un bandeau « prêt » et attend. Tenir la touche **Maj droit**, parler, relâcher : le texte transcrit s'écrit dans la fenêtre active.

C'est tout pour commencer. La suite explique comment installer depuis les sources, changer de modèle ou de langue, et piloter l'outil au clavier du bureau (Chapitre 2).

Prérequis

- **Un système Linux** avec une session graphique X11 ou Wayland. Sans serveur d'affichage, il n'y a pas de fenêtre active où écrire le texte transcrit.
- **Python 3** et un environnement virtuel (`python3-venv`).
- **Un outil d'injection de texte** : `ydotool` ou `xdotool` pour la frappe, `wl-clipboard` sous Wayland ou `xclip` sous X11 pour le presse-papiers. Sans moteur de frappe fonctionnel, l'outil ne refuse pas de démarrer : il passe en mode dégradé (le texte transcrit reste affiché dans le terminal, sans injection — Chapitre 3).
- **Un microphone reconnu par le système** comme périphérique d'entrée par défaut (Chapitre 3).
- **De l'espace disque pour le modèle** : de l'ordre de 145 Mo pour le modèle base, davantage pour les modèles plus grands (Chapitre 2). En fonctionnement, une session garde le modèle en mémoire — de l'ordre de 150 à 300 Mo de RAM pour base.

Le paquet natif déclare et installe lui-même toutes ses dépendances, système et Python, quelle que soit la distribution. En mode clone Git, `.venv/bin/pip install -e .` installe les bibliothèques Python; l'outil d'injection (ydotool ou xdotool, plus `wl-clipboard` ou `xclip`) est alors à installer séparément.

Deux façons d'installer

WhispSkrid propose deux canaux, sans que l'un remplace l'autre.

Le paquet natif (recommandé pour un usage courant)

```
# Debian 12 et 13, Ubuntu 22.04 et 24.04 et leurs dérivées
sudo apt install ./whispskrid_1.0.1_all.deb
```

```
# Fedora 42
sudo dnf install ./whispskrid-1.0.1-1.fc42.noarch.rpm
```

```
# openSUSE Leap 15.6
sudo zypper install ./whispskrid-1.0.1-1.leap156.noarch.rpm
```

```
# Arch Linux
sudo pacman -U whispskrid-1.0.1-1-any.pkg.tar.zst
```

Le paquet installe une commande unique, `whispskrid`, dans `/usr/bin/`. Il crée aussi un environnement Python isolé et y installe le moteur de transcription (`faster-whisper`) — cette étape a besoin du réseau au moment de l'installation, une seule fois. Rien n'est requis en tant que superutilisateur après coup.

Chaque paquet a été installé, contrôlé par `whispskrid --diagnose` puis désinstallé proprement dans un conteneur de chacune des versions ci-dessus (Debian 12 et 13, Ubuntu 22.04 et 24.04, Fedora 42, openSUSE Leap 15.6, Arch Linux). Le paquet Debian a en outre été validé en usage réel, dictée comprise, sur plusieurs machines physiques, dont une sous Debian 13; les autres paquets ne l'ont pas encore été en session graphique réelle. Les autres systèmes ne sont pas couverts — pas par principe, simplement le périmètre qui a été testé.

L'installation elle-même ne modifie jamais les fichiers personnels : aucune configuration existante n'est écrasée, aucun répertoire du compte utilisateur n'est créé ni touché à ce stade. Ce n'est qu'au premier lancement, et à la récupération d'un modèle, que des fichiers apparaissent dans le répertoire personnel — jamais avant, jamais sans une action explicite de l'utilisateur.

Le clone Git (pour suivre le développement ou contribuer)

```
git clone https://github.com/RonanDavalan/whispskrid.git
cd whispskrid
python3 -m venv .venv
.venv/bin/pip install -e .
.venv/bin/whispskrid --diagnose
```

Dans ce mode, l'outil se lance par `.venv/bin/whispskrid` plutôt que par la commande `whispskrid` du paquet, et sa configuration vit dans `config/config.yaml`, à côté du code — jamais dans le répertoire personnel. Les deux modes ne se mélangent pas : un clone Git ignore la configuration du paquet, et

récioproquement.

Récupérer un modèle

Aucun des deux canaux n'embarque de modèle de transcription — ils pèsent de l'ordre de 75 Mo (tiny) à plusieurs gigaoctets (large-v3), trop pour être livrés avec l'outil.

Le plus simple, en installation par paquet, est de laisser l'outil s'en charger :

```
whispskrid --download-model base
```

Cette commande récupère le modèle demandé — base par défaut, ou l'un de tiny, small, medium, large-v3 (et leurs variantes .en, spécialisées anglais) — vérifie qu'il se charge, l'installe dans ~/.local/share/whispskrid/whisper-models/, puis s'arrête. Au tout premier lancement sans aucun modèle, la même proposition est faite de façon interactive.

Le modèle et la langue sont deux choix distincts : un modèle Whisper est multilingue, c'est l'option -l (ou default_language dans la configuration, Chapitre 2) qui fixe la langue de reconnaissance. Sans elle, Whisper détecte la langue tout seul.

Si aucun modèle n'est trouvé au lancement, l'outil affiche les emplacements cherchés et la marche à suivre — jamais un échec silencieux.

Premier démarrage

```
whispskrid
```

(en mode clone Git, la même commande est .venv/bin/whispskrid, depuis la racine du dépôt)

Au lancement, l'outil charge la configuration, charge le modèle une fois pour toutes, ouvre le microphone, puis affiche un bandeau « prêt » et attend. Ce processus reste au premier plan : c'est la session persistante. Il garde le modèle en mémoire pour que chaque dictée démarre sans délai de chargement (Chapitre 2). Une seule session peut tourner à la fois.

Pour quitter : Ctrl+C dans le terminal de lancement, ou whispskrid --stop depuis un autre terminal.

Une première dictée

WhispSkrid fonctionne en appui-pour-parler : on tient une touche pendant qu'on parle, on relâche quand on a fini.

1. Donner le focus clavier à l'application cible (éditeur, navigateur, terminal).
2. **Tenir la touche Maj droit.** Un signal sonore court confirme le début de la capture.
3. **Parler** normalement, tant que la touche est tenue.
4. **Relâcher.** L'audio capturé part en transcription ; le texte obtenu est injecté dans la fenêtre active.

Il n'y a pas de phrase de réveil, pas d'état d'attente, pas de minuteur : la capture dure exactement le temps

où la touche est tenue. Chaque relâche produit une injection indépendante.

Sous une session **Wayland**, cette touche n'est captée que pour les fenêtres passant par XWayland ; pour une fenêtre Wayland native, il faut lier une sous-commande de contrôle à un raccourci du bureau (Chapitre 2).

Désinstaller et mettre à jour

Paquet natif. `sudo apt remove whispskrid` (Debian), `sudo dnf remove whispskrid` (Fedora), `sudo zypper remove whispskrid` (openSUSE) ou `sudo pacman -R whispskrid` (Arch) retire la commande et l'environnement Python isolé (`/usr/lib/whispskrid/venv`). Sur Debian, `sudo apt purge whispskrid` efface en plus le modèle de configuration d'usine. Le fichier de configuration personnel (`~/.config/whispskrid/config.yaml`) et les modèles (`~/.local/share/whispskrid/whisper-models/`) ne sont jamais touchés par la désinstallation, sur aucune distribution — les supprimer à la main si on le souhaite.

Clone Git. Supprimer le répertoire cloné : il contient tout (code, environnement virtuel `venv/`, éventuels modèles dans `whisper-models/`). Rien n'est écrit ailleurs dans ce mode.

Mise à jour. Avec le paquet, installer le fichier plus récent avec la même commande que pour l'installation (`apt install`, `dnf install`, `zypper install` ou `pacman -U`) : il remplace la version en place sans toucher à la configuration personnelle. Avec le clone Git, faire `git pull`, puis relancer `.venv/bin/pip install -e .` pour répercuter d'éventuelles nouvelles dépendances.

Chapitre 2 — Utiliser

Pour qui est cet outil

WhispSkrid est un outil de dictée en ligne de commande, hors ligne, construit autour de Whisper. Il vise un usage simple et prévisible : on tient une touche, on parle, on relâche, le texte s'écrit dans la fenêtre active. Rien d'autre — pas de phrase de réveil, pas de commandes vocales, pas d'état d'attente à gérer.

La transcription est servie par `faster-whisper`, une implémentation de Whisper fondée sur `CTranslate2`, optimisée pour tourner sur processeur. Le modèle par défaut est `base`, bon compromis vitesse/précision pour la plupart des machines ; il se change à tout moment (voir plus bas). Toute la reconnaissance est locale : aucun son ni texte ne quitte la machine, à aucun moment.

Le modèle d'interaction : appui-pour-parler strict

Une dictée est un épisode isolé, sans mémoire d'un épisode à l'autre :

1. **Appui** — presser et maintenir la touche d'appui-pour-parler. La capture audio démarre ; un signal sonore court le confirme.
2. **Parole** — l'audio est accumulé tant que la touche est tenue. Aucune coupure automatique, aucun minuteur, aucune détection de fin de phrase.
3. **Relâche** — la capture s'arrête. Le tampon audio complet est transcrit en une seule passe.
4. **Injection** — le texte transcrit, après un post-traitement léger, est injecté dans la fenêtre active (Chapitre 3 pour le mécanisme).

Il n'y a pas de texte « accumulé » entre deux dictées, donc pas de commande pour « copier la session » : chaque relâche produit sa propre injection.

Le garde-fou de durée

Seul arrêt automatique : `capture.max_seconds` (300 secondes par défaut). Si la touche reste tenue au-delà, la capture s'arrête d'elle-même, ce qui a été enregistré est transcrit et injecté normalement, et un avertissement est journalisé. Ce n'est pas un minuteur de confort — c'est une protection contre une touche bloquée ou un tampon audio qui gonfle sans fin.

L'arrêt sur silence n'est pas activé

L'arrêt automatique quand on cesse de parler (VAD) n'est pas encore disponible. Les clés `vad.enabled` (défaut `false`) et `vad.silence_ms` sont déjà présentes dans la configuration ; `vad.enabled: true` émet un avertissement « non encore implémenté » et l'outil poursuit en appui-pour-parler strict.

Deux voies pour déclencher une dictée

La touche locale

Un écouteur de raccourci (`pynput`) observe le clavier. La touche par défaut est **Maj droit** (`shift_r`), sans raccourci global par défaut sous les bureaux courants et facile à tenir. Elle se change par `hotkeys.push_to`

_talk dans la configuration. Sur cette voie, le maintien est natif : la capture dure exactement de l'appui à la relâche.

pynput observe le serveur X. Sous Wayland, il ne capte que les fenêtres passant par XWayland, jamais une fenêtre Wayland native — c'est un confort best-effort. Pour ne jamais démarrer cet écouteur, mettre `hotkeys.pynput_enabled: false` et piloter par la voie ci-dessous.

Les sous-commandes de contrôle

Voie universelle, indépendante du type de session : on lie ces sous-commandes aux raccourcis globaux du bureau. Comme un raccourci de bureau ne transmet pas la notion « touche tenue / touche relâchée », cette voie fonctionne en bascule :

Action	Sous-commande
Démarrer la capture	<code>whispskrid --dictate</code>
Arrêter la capture et injecter	<code>whispskrid --dictate-stop</code>
Démarrer ou arrêter selon l'état	<code>whispskrid --toggle</code>
Jeter la capture en cours sans injecter	<code>whispskrid --cancel</code>
Connaître l'état courant	<code>whispskrid --status</code>
Arrêter la session en cours	<code>whispskrid --stop</code>

Exemple sous KDE Plasma : dans *Paramètres système* → *Raccourcis* → *Raccourcis personnalisés*, créer une entrée « Commande/URL » dont la commande est `whispskrid --toggle`, et lui attribuer une combinaison. Un premier appui démarre la capture, un second l'arrête et injecte.

Les deux voies pilotent le même état interne. Une capture démarrée à la touche peut être annulée par `--cancel`, et inversement.

Session persistante et client

`whispskrid` lancé sans argument de contrôle démarre une **session persistante** au premier plan : elle charge la configuration et le modèle, ouvre le microphone, démarre l'écouteur de touche et la socket de contrôle, puis attend.

`whispskrid` lancé avec un argument de contrôle (`--dictate`, `--status`, `--stop...`) agit en **client** : il se connecte à la session en cours par une socket Unix (dans `$XDG_RUNTIME_DIR`), envoie la commande, affiche la réponse (OK ou ERR) et sort. S'il n'y a pas de session en cours, il le dit et sort en erreur.

Une seule session persistante peut tourner à la fois : si la socket répond déjà, une nouvelle instance refuse de démarrer.

Configuration

Tout le comportement passe par un unique fichier YAML, `config.yaml`.

Où se trouve le fichier

L'emplacement suit un ordre de priorité — le premier qui existe l'emporte, les suivants ne sont jamais consultés :

1. La variable d'environnement WHISPSKRID_CONFIG, si définie — utilisée telle quelle, jamais créée automatiquement.
2. config/config.yaml du dépôt, en mode clone Git.
3. ~/.config/whispskrid/config.yaml, en installation par paquet — créé automatiquement au premier lancement à partir d'un modèle d'usine, avec un message indiquant où le fichier a été créé.
4. Le modèle d'usine lui-même, en lecture seule.

En pratique, quel que soit le paquet natif installé, on modifie ~/.config/whispskrid/config.yaml — fichier personnel, jamais écrasé par une mise à jour du paquet.

Changer de modèle

```
models:
  default: "base"
  device: "auto"
  compute_type: "auto"
```

models.default est un nom court résolu dans le dossier des modèles, ou un chemin absolu. device vaut cpu, cuda ou auto.

compute_type accepte quatre valeurs : int8, int8_float16, float16 et auto. Avec auto, le choix se fait tout seul : int8 sur processeur, float16 sur carte graphique.

--model NOM sur la ligne de commande surcharge models.default pour une session. Un modèle plus grand transcrit plus finement, au prix d'une empreinte mémoire et d'une latence plus élevées.

Forcer la langue

```
default_language: ""
```

Vide, l'interface prend la langue du système et la reconnaissance détecte la langue toute seule. -l fr (ou --lang fr) sur la ligne de commande force la langue de l'interface et de la reconnaissance pour une session, sans toucher au fichier.

La touche d'appui-pour-parler

```
hotkeys:
  pynput_enabled: true
  push_to_talk: ["shift_r"]
  min_hold_ms: 250
  mode: "hold"
```

Une touche de fonction convient aussi (["f4"], à l'abri des raccourcis du bureau). Lister plusieurs touches

en fait une **combinaison** : toutes doivent être tenues ensemble pour engager la dictée, et en mode hold relâcher l'une d'elles arrête et injecte. `min_hold_ms` annule sans transcrire un appui plus bref — garde contre un effleurement qui ouvrirait une capture sur du silence, que Whisper peut halluciner ; sans effet hors mode hold. `mode: "toggle"` change la sémantique : un appui démarre la capture, l'appui suivant l'arrête et injecte, la relâche ne fait plus rien. `mode: "armed"` (écoute de phrases vocales) demande des modèles non livrés : armer échoue proprement tant qu'ils sont absents.

Le garde-fou de durée et le post-traitement

capture:

```
max_seconds: 300
```

post_processing:

```
trim: true
```

```
capitalize_sentence_start: true
```

`trim` rogne les blancs de bord du texte transcrit. `capitalize_sentence_start` force une capitale sur la première lettre injectée. Whisper ponctue et met les majuscules de phrase lui-même : le post-traitement se limite à ces deux réglages ; il n'y a ni table de substitution, ni règle par langue, ni insertion d'espaces fines.

Le collage sous Wayland

clipboard:

```
defer_restore: false
```

Passé à `true`, le presse-papiers n'est restauré qu'en fin de session au lieu de l'être après chaque dictée. Utile avec certaines applications GTK sous Wayland (Chapitre 3).

Gérer les modèles

Les modèles vivent dans un dossier géré :

- installé par paquet : `/usr/share/whispskrid/whisper-models/` (lecture seule), complété par `~/.local/share/whispskrid/whisper-models/` pour les téléchargements de l'utilisateur ;
- depuis les sources : `whisper-models/` à la racine du dépôt.

`whispskrid --download-model [NOM]` télécharge un modèle dans le dossier utilisateur et vérifie qu'il se charge. Ordre de grandeur de l'empreinte disque en quantification `int8` :

tiny	75 Mo
base	145 Mo
small	480 Mo
medium	1,5 Go
large-v3	3 Go

En mémoire vive, au pic d'utilisation : en CPU tiny ~0,4 Go, base ~0,5–0,6 Go, small ~1,3 Go, medium ~3,5–3,7 Go, large-v3 ~6,8 Go ; en GPU (x86) tiny ~0,2 Go, base ~0,3 Go, small ~0,7 Go, medium ~1,7–2 Go, large-v3 ~3,2–3,9 Go de VRAM. Fourchettes mesurées sur cinq machines, quatre langues — valeurs par

modèle. En pratique : `tiny` et `base` suivent le temps réel partout, y compris sur ARM ; `small` décroche sur ARM (2 à 3 fois trop lent) ; `medium` exige un x86 ; `large-v3` exige un x86 avec ~8 Go de RAM et ne sert qu'en GPU (en CPU, même un x86 dépasse le temps réel).

L'outil fonctionne hors ligne une fois le modèle présent : le cache Hugging Face n'est qu'un ultime repli, pas le chemin normal.

Récapitulatif de la ligne de commande

```
whispskrid [-l LANG] [--model NOM]
whispskrid --dictate | --dictate-stop | --toggle | --cancel
whispskrid --status | --stop
whispskrid --diagnose
whispskrid --download-model [NOM]
whispskrid --help | --version
```

Chapitre 3 — Dépanner

Comment le texte arrive dans l'application

WhispSkrid n'a pas d'interface graphique — il écrit le texte transcrit directement dans l'application qui a le focus clavier, quelle qu'elle soit.

Le texte n'est **pas tapé caractère par caractère**. Le mécanisme réel : le texte transcrit est placé dans le presse-papiers, puis un raccourci «coller» (`Ctrl+V`) est simulé. C'est plus fiable et plus rapide qu'une simulation de frappe complète, surtout sur de longs passages.

Le presse-papiers d'origine est restauré juste après le collage — dicter n'écrase donc pas durablement ce qu'on y avait copié. Sous Wayland avec une application GTK, cette restauration peut arriver avant que la fenêtre ait fini de lire le presse-papiers, qui colle alors l'ancien contenu; le réglage `clipboard.defer_restore: true` diffère la restauration à la fin de la session et supprime ce risque.

Le microphone

WhispSkrid capte le son sur le **périphérique d'entrée par défaut du système** (PulseAudio ou PipeWire). Il n'y a pas de réglage de périphérique dans `config.yaml` : pour changer de micro, on définit le bon périphérique comme entrée par défaut — dans les réglages son du bureau, avec `pavucontrol`, ou en ligne de commande (`pactl set-default-source <nom>`, la liste étant donnée par `pactl list sources short` ou `arecord -l`).

`whispSkrid --diagnose` ouvre un flux de test sur ce périphérique et signale si l'ouverture échoue — micro absent, coupé, ou déjà utilisé par une autre application.

Les outils d'injection et leur choix automatique

L'injection demande **deux** outils : un moteur de frappe et un outil de presse-papiers. Le moteur de frappe est choisi une seule fois au démarrage, par sondage direct — jamais retesté avant chaque collage :

- **Premier choix** : `ydotool` (frappe au niveau du pilote via le démon `ydotoold`, codes de touches dépendants de la disposition du clavier).
- **Sinon** : `xdotool` (voie standard, avec détection de la fenêtre).
- **Sinon** : mode dégradé — le texte transcrit reste affiché dans le terminal et la frappe est désactivée pour la session, plutôt qu'un échec silencieux à chaque dictée.

Le presse-papiers suit le type de session : `wl-clipboard` sous Wayland (toujours requis, même avec `ydotool`), `xclip` sous X11. Sans outil de presse-papiers, le texte transcrit n'atteint jamais le presse-papiers : rien ne s'écrit dans l'application et un `Ctrl+V` recolle le dernier contenu copié à la main.

Activer ydotool (solution de repli)

Cette section ne concerne que les systèmes où le moteur retenu est `ydotool`. Quand `xdotool` assure la frappe, rien de ce qui suit n'est nécessaire — en particulier, il est inutile de rejoindre le groupe `input`.

Trois conditions, à vérifier dans l'ordre si le texte ne s'écrit pas :

1. **Le paquet ydotool est installé.** Sur Debian, il vit dans trixie-backports, pas dans la branche stable — activer ce dépôt si nécessaire.
2. **Le service ydotool tourne.** ydotool a besoin de ce démon en arrière-plan.
3. **L'utilisateur appartient au groupe input.** C'est ce groupe qui donne accès à /dev/uinput, sans quoi ydotool échoue silencieusement :

```
sudo usermod -aG input $USER
```

Se déconnecter puis se reconnecter est nécessaire pour que ce changement de groupe prenne effet.

Raccourcis : X11 et Wayland ne sont pas égaux

La touche d'appui-pour-parler intégrée est écoutée par l'outil lui-même via pynput, qui observe le serveur X. Sous Wayland, pynput n'entend que les fenêtres passant par **XWayland** (beaucoup d'applications Electron et anciennes), jamais une fenêtre **Wayland native**. Pour ces dernières, il faut lier une sous-commande de contrôle (`--toggle`, `--dictate`, `--dictate-stop`) à un raccourci de l'environnement de bureau (Chapitre 2).

--diagnose

whispSkrid --diagnose vérifie l'environnement et sort — code de sortie 0 si tout passe, non nul sinon. Chaque point est rendu en ligne ok / !! :

- **Environnement** — type de session (X11 / Wayland), backend d'injection retenu.
- **Outils** — présence de ydotool, ydotool, xdotool, wl-copy / wl-paste ou xclip, paplay; accès en lecture-écriture à /dev/uinput.
- **Backend** — import de faster_whisper et de ctranslate2; device et compute_type effectifs; CUDA disponible ou non (informatif).
- **Modèles** — dossier des modèles résolu; présence du modèle par défaut; chargement d'essai (montée en mémoire puis libération).
- **Entrée audio** — ouverture d'un flux de capture au format attendu.
- **Presse-papiers** — aller-retour lecture / écriture.
- **Socket de contrôle** — chemin, répertoire d'exécution accessible en écriture, session en cours ou non.
- **Configuration** — chemin du config.yaml effectivement chargé.

Symptôme → cause probable → action

Symptôme	Cause probable	Action
Le texte transcrit reste dans le terminal, rien n'est injecté	Aucun moteur de frappe fonctionnel (ni <code>ydotoool</code> ni <code>xdotoool</code>) : mode dégradé	Installer <code>ydotoool</code> ou <code>xdotoool</code> , puis relancer la session
Message « aucun modèle » au démarrage	Aucun modèle trouvé aux emplacements attendus	<code>whispskrid</code> <code>--download-model base</code> — voir Chapitre 1
Rien n'est transcrit, le micro semble muet	Mauvais périphérique d'entrée par défaut, ou micro coupé au niveau système	Vérifier / changer l'entrée par défaut (<code>pavucontrol</code>); <code>whispskrid --diagnose</code>
<code>--diagnose</code> signale l'échec d'ouverture du flux audio	Périphérique occupé par une autre application, ou droits d'accès insuffisants	Fermer l'autre application; vérifier les permissions du périphérique
Rien ne s'écrit dans l'application, aucune erreur affichée	L'outil s'est rabattu sur <code>ydotoool</code> , mais l'utilisateur n'est pas dans le groupe <code>input</code>	Rejoindre le groupe, se reconnecter — voir plus haut
L'application recolle toujours le dernier contenu copié à la main	Aucun outil de presse-papiers installé (<code>wl-clipboard</code> sous Wayland, <code>xclip</code> sous X11)	L'installer
Sous Wayland avec une application GTK, la fenêtre colle par instant l'ancien contenu	La restauration du presse-papiers arrive avant que la fenêtre ait lu la sélection	<code>clipboard.defer_restore: true</code> dans la configuration
La touche Maj droit ne déclenche rien	Fenêtre Wayland native au premier plan — <code>pynput</code> ne voit que les fenêtres X11 et XWayland	Lier <code>--toggle</code> / <code>--dictate</code> à un raccourci du bureau — voir Chapitre 2
Aucun son au début de la capture	<code>paplay</code> absent, ou aucun fichier son configuré	Avertissement bénin — installer <code>pulseaudio-utils</code> si le son est souhaité
La capture s'arrête seule après cinq minutes	Garde-fou <code>capture.max_seconds</code> atteint	Comportement voulu; relever la valeur dans la configuration si besoin — voir Chapitre 2
<code>vad.enabled: true</code> sans effet	La détection de silence n'est pas implémentée dans cette version	Réglage réservé; l'outil poursuit en appui-pour-parler strict — voir Chapitre 2
La transcription est lente	Modèle trop grand pour la machine, ou <code>compute_type</code> inadapté	Passer à <code>base</code> ou <code>small</code> ; laisser <code>compute_type: auto</code> — voir Chapitre 2
La configuration modifiée ne semble jamais prise en compte	Le fichier édité n'est pas celui réellement utilisé	Vérifier l'ordre de résolution — voir Chapitre 2

« Une session est déjà en cours » au lancement	Une session persistante tourne déjà (une seule à la fois)	<code>whispskrid --stop</code> , ou utiliser la session existante
--	---	---

Où chercher un message d'erreur

WhispSkrid affiche ses erreurs directement dans le terminal où la session persistante a été lancée. Relancer depuis un terminal visible est la première chose à faire pour diagnostiquer un problème non listé ici. En cas de doute, `whispskrid --diagnose` vérifie tout et affiche l'état de chaque point.

Si le problème persiste

Le dépôt du projet a une page de suivi publique : <https://github.com/RonanDavalan/whispskrid/issues>