



DOCUMENTATION TECHNIQUE

Sygal CLI

Installation, configuration et utilisation complète de
la ligne de commande Sygal — conversion de
documents en Markdown/JSON, en local.

Version couverte — sygal-cli 0.2.0

Plateformes — macOS (Apple Silicon) ·
Windows x64 · Linux x64

Date — 20 juillet 2026

Auteur — Cyrill Semah

Sommaire

1. Présentation

- Ce qu'est Sygal CLI
- Positionnement (humains et agents IA)
- Confidentialité et fonctionnement local

2. Prérequis

3. Installation

- Installation globale via npm
- Vérification de l'installation
- Mise à jour
- Désinstallation

4. Premier lancement — licence

- Essai gratuit de 14 jours
- Achat et activation d'une licence
- Statut de la licence
- Désactivation (changement de machine)

5. Configuration

- Provider OCR
- Clé API
- Langue, conversion avancée, budget OCR
- Emplacement des fichiers sur le disque

6. Utiliser la conversion

- Syntaxe et options complètes
- Exemples par cas d'usage
- Formats supportés

7. Sortie JSON et usage par un agent IA

- Mode `--json`
- Mode OCR différé + `sygal inject`
- Codes d'erreur

8. Variables d'environnement

9. Dépannage

10. Référence rapide des commandes

1 Présentation

Ce qu'est Sygal CLI, à qui il s'adresse et comment il fonctionne.

Ce qu'est Sygal CLI

Sygal CLI est la version ligne de commande de l'application desktop Sygal. Il convertit des fichiers, des dossiers entiers ou des pages web en **Markdown** et/ou **JSON**, en utilisant exactement le même moteur de conversion que l'application (pipeline local MarkItDown), avec un appel à une API d'OCR pour les images.

Positionnement — humains et agents IA

Le CLI est pensé pour deux publics :

- **Les développeurs**, qui veulent convertir des documents depuis un terminal ou un script, sans ouvrir l'application.
- **Les agents IA** (assistants de type Hermes, agents autonomes, pipelines d'automatisation), avec une sortie `--json` stricte, des codes d'erreur stables et un mode d'OCR différé permettant à l'agent de transcrire lui-même les images sans appel API supplémentaire.

Confidentialité et fonctionnement local

Le pipeline de conversion (MarkItDown) est **100 % local** : aucun document n'est envoyé sur le réseau pour les formats bureautiques, PDF, tableurs, etc. Le seul trafic réseau concerne :

- L'**OCR des images** (envoi de l'image au provider choisi — OpenAI, Mistral, Claude, NVIDIA ou Gemini) ;
- L'**import d'une URL** (récupération de la page demandée, aucun appel de modèle).

À retenir

Sygal CLI est local-first : pas de compte, pas de synchronisation cloud, pas de backend Sygal. Vos documents ne transitent jamais par nos serveurs.

2 Prérequis

Ce qu'il faut avoir installé avant de commencer.

Élément	Exigence
Node.js	<code>>= 20</code> — vérifier avec <code>node --version</code>
npm	Fourni avec Node.js — vérifier avec <code>npm --version</code>
Système	macOS (Apple Silicon), Windows x64, ou Linux x64
Compte OCR (optionnel)	Une clé API OpenAI, Mistral, Claude, NVIDIA ou Gemini — nécessaire uniquement pour convertir des images

Linux — compatibilité glibc

Le binaire Linux nécessite glibc 2.28 ou supérieur (Ubuntu 18.04+, Debian 10+, RHEL 8+, Amazon Linux 2+).

Non compatible avec Alpine/musl

— utiliser une image de base glibc (Debian/Ubuntu) pour un déploiement en conteneur.

3 Installation

Installation globale, vérification, mise à jour et désinstallation.

Installation globale via npm

```
# Installe la commande "sygal" globalement
npm install -g sygal-cli
```

Cette commande installe automatiquement, en fonction de votre plateforme, le moteur de conversion natif correspondant :

Plateforme	Paquet installé automatiquement
MACOS Apple Silicon (arm64)	<code>@sygal/sidecar-darwin-arm64</code>
WINDOWS x64	<code>@sygal/sidecar-win32-x64</code>
LINUX x64	<code>@sygal/sidecar-linux-x64</code>

Vous n'avez rien à choisir : npm sélectionne le bon paquet automatiquement via les métadonnées de plateforme (`os` / `cpu`).

Vérification de l'installation

```
sygal --version
# → 0.2.0

sygal --help
# → liste des commandes disponibles
```

Mise à jour

```
# Vérifie s'il existe une nouvelle version, sans l'installer
sygal update --check

# Vérifie et installe la dernière version (équivalent à un nouveau
# npm install -g sygal-cli)
sygal update
```

Désinstallation

```
npm uninstall -g sygal-cli
```

Les paquets sidecar (`@sygal/sidecar-*`) sont supprimés automatiquement en tant que dépendances optionnelles du CLI.

4 Premier lancement — licence

Le CLI démarre en essai gratuit automatiquement ; aucune étape n'est requise avant la première conversion.

Essai gratuit de 14 jours

La toute première utilisation de `sygal convert` démarre automatiquement un essai gratuit de **14 jours**, sans compte ni carte bancaire. Le compteur commence à courir dès la première conversion.

```
sygal license status
# → ⓘ Essai : 14 jour(s) restant(s)
```

Postes partagés entre l'app et le CLI

Chaque poste de votre licence Sygal peut faire tourner l'application desktop ou le CLI : les deux comptent de la même façon comme un poste actif. Le nombre de postes dépend du pack acheté (1 pour Sygal CLI, 2 pour Solo, 5 pour Team, 10 pour Business, 25 pour Enterprise, 50 pour Ultimate). Pour changer d'appareil, désactivez l'ancien poste avant d'en activer un nouveau.

Achat et activation d'une licence

Une fois l'essai terminé, ou pour activer une licence achetée immédiatement :

```
# Boutique
# → https://sygal.lemonsqueezy.com

# Activation (consomme un poste du pool)
sygal license activate <clé>
```

La clé peut aussi être fournie via la variable d'environnement `SYGAL_LICENSE_KEY` pour une activation automatique (utile en environnement scripté/CI).

Statut de la licence

```
sygal license status
# → ✓ Licence active (masquée : ****-****-1234)
# → ⓘ Essai : N jour(s) restant(s)
# → ✗ Essai terminé – achetez une licence : https://sygal.lemonsqueezy.com
```

Désactivation (changement de machine)

Pour libérer un poste avant de réinstaller sur une autre machine :

```
sygal license deactivate
# → ✓ Licence désactivée – poste libéré
```

5 Configuration

Provider OCR, clé API, langue et réglages par défaut.

Provider OCR

L'OCR (transcription des images) n'est nécessaire que pour convertir des **images** ou des **images embarquées** dans des documents (mode `---advanced`). Cinq providers sont disponibles :

Identifiant	Fournisseur
<code>openai</code>	OpenAI
<code>mistral</code>	Mistral AI
<code>claude</code>	Anthropic Claude
<code>nvidia</code>	NVIDIA (modèle par défaut, gratuit)
<code>gemini</code>	Google Gemini

```
sygal config set-ocr-provider openai
```

Clé API

```
# Lue sur stdin – recommandé, évite que la clé apparaisse
# dans l'historique du shell
sygal config set-api-key openai

# Ou passée directement en argument (déconseillé)
sygal config set-api-key openai sk-...

# Supprimer une clé enregistrée
sygal config clear-api-key openai
```

La clé est **chiffrée sur disque**, liée à la machine courante, jamais affichée en clair ni journalisée.

Langue, conversion avancée, budget OCR

```
# Langue des libellés ajoutés par l'OCR (fr | en | it | de | es)
sygal config set-locale fr

# Active/désactive l'OCR des images embarquées par défaut
sygal config set-advanced on

# Garde-fou de coût : budget maximum d'appels OCR par exécution
sygal config set-max-ocr-calls 50
sygal config set-max-ocr-calls off # illimité

# Voir la configuration actuelle (jamais les valeurs des clés)
sygal config show
```

Emplacement des fichiers sur le disque

Fichier	Emplacement	Contenu
Configuration	<code>~/.sygal/config.json</code>	Provider, langue, clés API chiffrées, réglages par défaut
Licence	<code>~/.sygal/license.enc</code>	État de licence chiffré (essai / active)

Les deux fichiers sont créés automatiquement au premier usage (permissions restreintes, `chmod 600`).

6 Utiliser la conversion

Syntaxe complète, toutes les options, exemples par cas d'usage.

Syntaxe

```
sygal convert <chemins...> [options]
```

`<chemins...>` accepte un ou plusieurs fichiers, dossiers (parcoursus récursivement) ou URLs `http(s)`, dans n'importe quelle combinaison.

Options complètes

Option	Effet
<code>-o, --output <dir></code>	Dossier de sortie. Par défaut : dossier source de chaque fichier.
<code>-f, --format <fmt></code>	<code>md</code> (défaut) · <code>json</code> · <code>both</code>
<code>--ocr <mode></code>	<code>auto</code> (défaut, appelle le provider configuré) · <code>defer</code> (aucun appel API, images extraites pour transcription externe)
<code>--advanced</code>	Transcrit aussi les images embarquées (docx, pptx, pdf, xlsx, html, eml)
<code>--pdf-ocr <mode></code>	Pour un PDF scanné : <code>all</code> (toutes les pages) · <code>first</code> (première page) · <code>page</code> (une page précise, avec <code>--page</code>)
<code>--page <n></code>	Numéro de page pour <code>--pdf-ocr page</code>
<code>--max-ocr-calls <n></code>	Budget d'appels OCR pour cette exécution uniquement (remplace le réglage global)
<code>--translate <locale></code>	Traduit le Markdown final (<code>fr</code> · <code>en</code> · <code>it</code> · <code>de</code> · <code>es</code>) — incompatible avec <code>--ocr defer</code>
<code>--json</code>	Sortie machine JSON sur stdout, aucun habillage (voir section 7)
<code>--batch</code>	Accepté pour compatibilité — les dossiers sont développés automatiquement, cette option n'a plus d'effet propre

Exemples par cas d'usage

Convertir un seul fichier

```
sygal convert rapport.pdf
# → rapport.pdf.md créé à côté du fichier source
```

Convertir un dossier entier, Markdown + JSON, vers un dossier de sortie dédié

```
sygal convert dossier/ --format both --output ./sorties
```

Convertir une page web

```
sygal convert https://example.com/article  
# → article.md, nommé d'après le titre de la page
```

PDF scanné – OCR page par page

```
sygal convert scan.pdf --pdf-ocr all
```

Document avec images embarquées à transcrire

```
sygal convert rapport.docx --advanced
```

Convertir puis traduire le résultat

```
sygal convert note.txt --translate en
```

Plusieurs sources en une seule commande

```
sygal convert facture.pdf notes/ https://example.com/doc --format both
```

Formats supportés

Routage automatique par type de fichier — aucun choix de pipeline à faire manuellement. Les images passent par l'OCR configuré, tout le reste par Markdown (100 % local).

Catégorie	Extensions
Documents bureautiques	pdf · docx · xlsx · pptx · xls
Web / texte / données	html · csv · txt · md · json · xml
Autres documents	epub · msg · eml · ipynb · zip
Images (via OCR)	png · jpg · gif · webp · tiff · heic
Web (URL)	Pages http(s) , flux RSS/Atom

7 Sortie JSON et usage par un agent IA

Sortie machine stricte, codes d'erreur stables, extraction sans appel API.

Mode `--json`

Ajouter `--json` à n'importe quelle commande `convert` bascule la sortie en JSON strict sur stdout (aucun texte décoratif), avec un code de sortie `1` en cas d'échec :

```
sygal convert dossier/ --json
```

```
{
  "version": "0.2.0",
  "summary": { "done": 1, "failed": 0, "ocrCalls": 0 },
  "results": [
    {
      "input": "/chemin/rapport.pdf",
      "name": "rapport.pdf",
      "pipeline": "markdown",
      "status": "done",
      "outputs": ["/chemin/rapport.pdf.md"],
      "sizeBytes": 4213,
      "durationMs": 87,
      "mdTokens": 612,
      "jsonTokens": 780
    }
  ]
}
```

Mode OCR différé + `sygal inject`

Pour laisser un agent IA transcrire lui-même les images (sans appel API OCR facturé par Sygal), utiliser `--ocr defer` :

```
sygal convert doc.docx --advanced --ocr defer --json
```

Résultat :

- Les images sont extraites dans `doc.docx.images/` ;
- Le Markdown contient des marqueurs `![sygal:ocr:N](...)` à la position exacte de chaque image ;
- Le JSON liste les transcriptions en attente (`deferred` : index, chemin du fichier image, page).

Une fois l'agent en possession des transcriptions, il les réintègre au document avec :

```
sygal inject doc.docx.md --transcripts transcriptions.json
```

Format attendu de `transcriptions.json` :

```
[
  { "index": 1, "text": "Description de l'image...", "provider": "agent" }
]
```

Chaque transcription est injectée à la position exacte de son marqueur d'origine.

Codes d'erreur

Stables entre versions, présents dans le champ `error.code` de la sortie `--json` :

Code	Signification
<code>unsupported_format</code>	Extension de fichier non reconnue
<code>api_key_missing</code>	Aucune clé API configurée pour le provider OCR actif
<code>invalid_api_key</code>	Clé API rejetée par le provider
<code>rate_limited</code>	Limite de débit atteinte côté provider OCR
<code>network_error</code>	Échec réseau (OCR ou import URL)
<code>scanned_pdf_detected</code>	PDF scanné détecté, nécessite <code>--pdf-ocr</code>
<code>ocr_call_limit_exceeded</code>	Budget <code>--max-ocr-calls</code> atteint
<code>license_expired</code>	Essai terminé, aucune licence active
<code>sidecar_down</code>	Moteur de conversion injoignable
<code>sidecar_not_found</code>	Paquet sidecar de la plateforme introuvable — réinstaller
<code>timeout</code>	Dépassement du délai de conversion

8 Variables d'environnement

Pour un usage scripté ou en environnement CI, toute la configuration peut être fournie sans fichier.

Variable	Rôle
<code>SYGAL_OCR_PROVIDER</code>	Provider OCR actif (prioritaire sur le fichier de config)
<code>SYGAL_OCR_API_KEY</code>	Clé API du provider actif (prioritaire sur le fichier de config)
<code>SYGAL_LICENSE_KEY</code>	Clé de licence — active automatiquement au premier lancement
<code>SYGAL_SIDE CAR_PATH</code>	Chemin explicite vers un binaire sidecar (contourne la détection automatique)
<code>SYGAL_CONFIG_PATH</code>	Emplacement alternatif du fichier de configuration

9 Dépannage

Problèmes fréquents et solutions.

« **sidecar_not_found** » ou « **Moteur de conversion introuvable** »

Réinstaller proprement :

```
npm uninstall -g sygal-cli
npm install -g sygal-cli
```

Si le problème persiste, vérifier que la plateforme est supportée (section 2) et, en dernier recours, définir `SYGAL_SIDECAR_PATH` vers un binaire valide.

« **api_key_missing** » sur une conversion d'image

Configurer une clé pour le provider actif :

```
sygal config set-ocr-provider openai
sygal config set-api-key openai
```

« **license_expired** »

L'essai de 14 jours est terminé. Activer une licence :

```
sygal license activate <clé>
# achat : https://sygal.lemonsqueezy.com
```

« **scanned_pdf_detected** »

Le PDF ne contient pas de texte extractible (images scannées). Relancer avec l'option OCR appropriée :

```
sygal convert scan.pdf --pdf-ocr all
```

Changement de machine

Désactiver la licence sur l'ancienne machine avant d'activer sur la nouvelle (le nombre de postes de votre licence est limité) :

```
# Sur l'ancienne machine
sygal license deactivate

# Sur la nouvelle machine
sygal license activate <clé>
```

10 Référence rapide des commandes

Toutes les commandes de Sygal CLI, en un coup d'œil.

Commande	Rôle
<code>sygal convert <chemins...></code>	Convertit un ou plusieurs fichiers/dossiers/URLs
<code>sygal inject <markdown> --transcripts <file></code>	Réintègre des transcriptions OCR différées
<code>sygal update [--check]</code>	Vérifie/installe la dernière version
<code>sygal license activate <clé></code>	Active une licence
<code>sygal license status</code>	Affiche l'état de la licence
<code>sygal license deactivate</code>	Libère le poste courant
<code>sygal config set-ocr-provider <id></code>	Choisit le provider OCR
<code>sygal config set-api-key <provider> [clé]</code>	Enregistre une clé API (chiffrée)
<code>sygal config clear-api-key <provider></code>	Supprime une clé API
<code>sygal config set-locale <locale></code>	Langue des libellés OCR
<code>sygal config set-advanced <on off></code>	OCR des images embarquées par défaut
<code>sygal config set-max-ocr-calls <n off></code>	Budget d'appels OCR par exécution
<code>sygal config show</code>	Affiche la configuration actuelle
<code>sygal --version</code>	Affiche la version installée
<code>sygal --help</code> / <code>sygal <cmd> --help</code>	Aide intégrée

Sygal CLI — documentation générée le 20 juillet 2026 pour la version 0.2.0.
Boutique de licence : <https://sygal.lemonsqueezy.com> — Site : <https://sygal.app>