



TECHNICAL DOCUMENTATION

Sygal CLI

Installation, configuration and full usage of the Sygal command line, converting documents to Markdown/JSON, locally.

Version covered — sygal-cli 0.2.0

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

Date — July 20, 2026

Author — Cyrill Semah

Table of contents

1. Overview

What Sygal CLI is

Who it's for (humans and AI agents)

Privacy and local-first operation

2. Requirements

3. Installation

Global install via npm

Verifying the install

Updating

Uninstalling

4. First run — licence

14-day free trial

Purchasing and activating a licence

Licence status

Deactivation (changing machine)

5. Configuration

OCR provider

API key

Locale, advanced conversion, OCR budget

File locations on disk

6. Using conversion

Full syntax and options

Examples by use case

Supported formats

7. JSON output and usage by an AI agent

`--json` mode

Deferred OCR mode + `sygal inject`

Error codes

8. Environment variables

9. Troubleshooting

10. Quick command reference

1 Overview

What Sygal CLI is, who it's for, and how it works.

What Sygal CLI is

Sygal CLI is the command-line version of the Sygal desktop app. It converts files, entire folders, or web pages to **Markdown** and/or **JSON**, using exactly the same conversion engine as the app (local MarkItDown pipeline), with a call to an OCR API for images.

Who it's for — humans and AI agents

The CLI is built for two audiences:

- **Developers**, who want to convert documents from a terminal or a script, without opening the app.
- **AI agents** (assistants like Hermes, autonomous agents, automation pipelines), with a strict `--json` output, stable error codes, and a deferred OCR mode letting the agent transcribe images itself, with no extra API call.

Privacy and local-first operation

The conversion pipeline (MarkItDown) is **100% local**: no document is ever sent over the network for office formats, PDFs, spreadsheets, etc. The only network traffic involves:

- **Image OCR** (sending the image to the provider you chose — OpenAI, Mistral, Claude, NVIDIA or Gemini);
- **URL import** (fetching the requested page, no model call).

Keep in mind

Sygal CLI is local-first: no account, no cloud sync, no Sygal backend. Your documents never pass through our servers.

2 Requirements

What you need installed before you start.

Item	Requirement
Node.js	<code>>= 20</code> — check with <code>node --version</code>
npm	Bundled with Node.js — check with <code>npm --version</code>
System	macOS (Apple Silicon), Windows x64, or Linux x64
OCR account (optional)	An OpenAI, Mistral, Claude, NVIDIA or Gemini API key — only needed to convert images

Linux — glibc compatibility

The Linux binary requires glibc 2.28 or later (Ubuntu 18.04+, Debian 10+, RHEL 8+, Amazon Linux 2+).

Not compatible with Alpine/musl

— use a glibc-based image (Debian/Ubuntu) for container deployments.

3 Installation

Global install, verification, updating, and uninstalling.

Global install via npm

```
# Installs the "sygal" command globally
npm install -g sygal-cli
```

This command automatically installs the matching native conversion engine for your platform:

Platform	Package installed automatically
MACOS — Apple Silicon (arm64)	@sygal/sidecar-darwin-arm64
WINDOWS — x64	@sygal/sidecar-win32-x64
LINUX — x64	@sygal/sidecar-linux-x64

You don't have to choose anything: npm selects the right package automatically via platform metadata (`os` / `cpu`).

Verifying the install

```
sygal --version
# → 0.2.0

sygal --help
# → list of available commands
```

Updating

```
# Checks for a new version, without installing it
sygal update --check

# Checks and installs the latest version (equivalent to a fresh
# npm install -g sygal-cli)
sygal update
```

Uninstalling

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

Sidecar packages ([@sygal/sidecar-*](#)) are automatically removed as optional dependencies of the CLI.

4 First run — licence

The CLI starts a free trial automatically; no step is required before your first conversion.

14-day free trial

The very first use of `sygal convert` automatically starts a free **14-day** trial, with no account or card. The countdown starts on your first conversion.

```
sygal license status
# → Trial: 14 day(s) left
```

Seats shared between the app and the CLI

Each seat on your Sygal licence can run the desktop app or the CLI: both count the same way as one active seat. The number of seats depends on the pack you bought (1 for Sygal CLI, 2 for Solo, 5 for Team, 10 for Business, 25 for Enterprise, 50 for Ultimate). To switch devices, deactivate the old seat before activating a new one.

Purchasing and activating a licence

Once the trial ends, or to activate a licence you bought right away:

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

# Activation (uses one seat)
sygal license activate <key>
```

The key can also be supplied via the `SYGAL_LICENSE_KEY` environment variable for automatic activation (useful in scripted/CI environments).

Licence status

```
sygal license status
# → ✓ Licence active (masked: ****-****-1234)
# → ⌚ Trial: N day(s) left
# → ✗ Trial ended — buy a licence: https://sygal.lemonsqueezy.com
```

Deactivation (changing machine)

To free up a seat before reinstalling on another machine:

```
sygal license deactivate
# → ✓ Licence deactivated — seat freed
```

5 Configuration

OCR provider, API key, locale, and default settings.

OCR provider

OCR (image transcription) is only needed to convert **images** or **images embedded** in documents (`--advanced` mode). Five providers are available:

Identifier	Provider
<code>openai</code>	OpenAI
<code>mistral</code>	Mistral AI
<code>claude</code>	Anthropic Claude
<code>nvidia</code>	NVIDIA (default model, free)
<code>gemini</code>	Google Gemini

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

API key

```
# Read from stdin – recommended, keeps the key out of
# your shell history
sygal config set-api-key openai

# Or passed directly as an argument (not recommended)
sygal config set-api-key openai sk-...

# Remove a stored key
sygal config clear-api-key openai
```

The key is **encrypted on disk**, bound to the current machine, never shown in clear or logged.

Locale, advanced conversion, OCR budget

```
# Language of labels added by OCR (fr | en | it | de | es)
sygal config set-locale en

# Enable/disable OCR of embedded images by default
sygal config set-advanced on

# Cost guard rail: max OCR calls per run
sygal config set-max-ocr-calls 50
sygal config set-max-ocr-calls off # unlimited

# View current configuration (never key values)
sygal config show
```

File locations on disk

File	Location	Content
Configuration	<code>~/.sygal/config.json</code>	Provider, locale, encrypted API keys, default settings
Licence	<code>~/.sygal/license.enc</code>	Encrypted licence state (trial / active)

Both files are created automatically on first use (restricted permissions, `chmod 600`).

6 Using conversion

Full syntax, all options, examples by use case.

Syntax

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

`<paths...>` accepts one or more files, folders (walked recursively), or `http(s)` URLs, in any combination.

Full options

Option	Effect
<code>-o, --output <dir></code>	Output folder. Default: each file's source folder.
<code>-f, --format <fmt></code>	<code>md</code> (default) · <code>json</code> · <code>both</code>
<code>--ocr <mode></code>	<code>auto</code> (default, calls the configured provider) · <code>defer</code> (no API call, images extracted for external transcription)
<code>--advanced</code>	Also transcribes images embedded in docx, pptx, pdf, xlsx, html, eml
<code>--pdf-ocr <mode></code>	For a scanned PDF: <code>all</code> (every page) · <code>first</code> (first page) · <code>page</code> (one specific page, with <code>--page</code>)
<code>--page <n></code>	Page number for <code>--pdf-ocr page</code>
<code>--max-ocr-calls <n></code>	OCR call budget for this run only (overrides the global setting)
<code>--translate <locale></code>	Translates the final Markdown (<code>fr</code> · <code>en</code> · <code>it</code> · <code>de</code> · <code>es</code>) — incompatible with <code>--ocr defer</code>
<code>--json</code>	Strict machine JSON on stdout, no decoration (see section 7)
<code>--batch</code>	Accepted for compatibility — folders are expanded automatically, this option has no effect of its own anymore

Examples by use case

Convert a single file

```
sygal convert report.pdf
# → report.pdf.md created next to the source file
```

Convert an entire folder, Markdown + JSON, to a dedicated output folder

```
sygal convert folder/ --format both --output ./output
```

Convert a web page

```
sygal convert https://example.com/article  
# → article.md, named after the page title
```

Scanned PDF – OCR page by page

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

Document with embedded images to transcribe

```
sygal convert report.docx --advanced
```

Convert then translate the result

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

Multiple sources in a single command

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

Supported formats

Automatic routing by file type — no pipeline to choose manually. Images go through the configured OCR, everything else through Markdown (100% local).

Category	Extensions
Office documents	<code>pdf</code> · <code>docx</code> · <code>xlsx</code> · <code>pptx</code> · <code>xls</code>
Web / text / data	<code>html</code> · <code>csv</code> · <code>txt</code> · <code>md</code> · <code>json</code> · <code>xml</code>
Other documents	<code>epub</code> · <code>msg</code> · <code>eml</code> · <code>ipynb</code> · <code>zip</code>
Images (via OCR)	<code>png</code> · <code>jpg</code> · <code>gif</code> · <code>webp</code> · <code>tiff</code> · <code>heic</code>
Web (URL)	Pages <code>http(s)</code> , RSS/Atom feeds

7 JSON output and usage by an AI agent

Strict machine output, stable error codes, extraction with no API call.

Mode `--json`

Adding `--json` to any `convert` command switches the output to strict JSON on stdout (no decorative text), with exit code `1` on failure:

```
sygal convert folder/ --json
```

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

Deferred OCR mode + `sygal inject`

To let an AI agent transcribe images itself (with no OCR API call billed by Sygal), use `--ocr defer`:

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

Result:

- Images are extracted into `doc.docx.images/` ;
- The Markdown contains `![sygal:ocr:N](...)` markers at each image's exact position;
- The JSON lists the pending transcriptions (`deferred` : index, image path, page).

Once the agent has the transcriptions, it feeds them back into the document with:

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

Expected format of `transcriptions.json` :

```
[
  { "index": 1, "text": "Image description...", "provider": "agent" }
]
```

Each transcription is injected at the exact position of its original marker.

Error codes

Stable across versions, present in the `error.code` field of `--json` output:

Code	Meaning
<code>unsupported_format</code>	Unrecognized file extension
<code>api_key_missing</code>	No API key configured for the active OCR provider
<code>invalid_api_key</code>	Key rejected by the provider
<code>rate_limited</code>	Rate limit reached on the OCR provider side
<code>network_error</code>	Network failure (OCR or URL import)
<code>scanned_pdf_detected</code>	Scanned PDF detected, needs <code>--pdf-ocr</code>
<code>ocr_call_limit_exceeded</code>	<code>--max-ocr-calls</code> budget reached
<code>license_expired</code>	Trial ended, no active licence
<code>sidecar_down</code>	Conversion engine unreachable
<code>sidecar_not_found</code>	Platform sidecar package not found — reinstall
<code>timeout</code>	Conversion time limit exceeded

For scripted use or CI environments, all configuration can be supplied with no file.

Variable	Role
<code>SYGAL_OCR_PROVIDER</code>	Active OCR provider (takes priority over the config file)
<code>SYGAL_OCR_API_KEY</code>	API key for the active provider (takes priority over the config file)
<code>SYGAL_LICENSE_KEY</code>	Licence key — activates automatically on first run
<code>SYGAL_SIDE CAR_PATH</code>	Explicit path to a sidecar binary (bypasses auto-detection)
<code>SYGAL_CONFIG_PATH</code>	Alternate location for the configuration file

9 Troubleshooting

Common issues and fixes.

"sidecar_not_found" or "Conversion engine not found"

Reinstall cleanly:

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

If the problem persists, check that your platform is supported (section 2) and, as a last resort, set `SYGAL_SIDECAR_PATH` to a valid binary.

"api_key_missing" on an image conversion

Configure a key for the active provider:

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

"license_expired"

The 14-day trial has ended. Activate a licence:

```
sygal license activate <key>  
# buy: https://sygal.lemonsqueezy.com
```

"scanned_pdf_detected"

The PDF has no extractable text (scanned images). Re-run with the right OCR option:

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

Changing machines

Deactivate the licence on the old machine before activating on the new one (your licence's seat count is limited):

```
# On the old machine  
sygal license deactivate  
  
# On the new machine  
sygal license activate <key>
```

10 Quick command reference

Every Sygal CLI command, at a glance.

Command	Role
<code>sygal convert <paths...></code>	Converts one or more files/folders/URLs
<code>sygal inject <markdown> --transcripts <file></code>	Re-integrates deferred OCR transcriptions
<code>sygal update [--check]</code>	Checks/installs the latest version
<code>sygal license activate <key></code>	Activates a licence
<code>sygal license status</code>	Shows licence state
<code>sygal license deactivate</code>	Frees the current seat
<code>sygal config set-ocr-provider <id></code>	Chooses the OCR provider
<code>sygal config set-api-key <provider> [key]</code>	Stores an API key (encrypted)
<code>sygal config clear-api-key <provider></code>	Removes an API key
<code>sygal config set-locale <locale></code>	Language of OCR labels
<code>sygal config set-advanced <on off></code>	OCR of embedded images by default
<code>sygal config set-max-ocr-calls <n off></code>	OCR call budget per run
<code>sygal config show</code>	Shows current configuration
<code>sygal --version</code>	Shows installed version
<code>sygal --help</code> / <code>sygal <cmd> --help</code>	Built-in help

Sygal CLI — documentation generated on July 20, 2026 for version 0.2.0.

Licence store: <https://sygal.lemonsqueezy.com> — Site: <https://sygal.app>